Online Unsupervised Event Detection in Wireless Sensor Networks

Majid Bahrepour, Nirvana Meratnia , Paul J. M. Havinga

Pervasive Systems Group, University of Twente
P.O.Box 217, 7500 AE Enschede, the Netherlands
{m.bahrepour, n.meratnia, p.j.m.havinga}@utwente.nl

Abstract— **Event detection applications of wireless sensor networks (WSNs) highly rely on accurate and timely detection of out of ordinary situations. Majority of the existing event detection techniques designed for WSNs have focused on detection of events with known patterns requiring a priori knowledge about events being detected. In this paper, however, we propose an online unsupervised event detection technique for detection of unknown events. Traditional unsupervised learning techniques cannot directly be applied in WSNs due to their high computational and memory complexities. To this end, by considering specific resource limitations of the WSNs we modify the standard K-means algorithm in this paper and explore its applicability for online and fast event detection in WSNs. For performance evaluation, we investigate event detection accuracy, false alarm, similarity calculation (using the Rand Index), computational and memory complexity of the proposed approach on two real datasets.**

I. INTRODUCTION

Events have different meaning in different applications and research domains. Generally speaking, event detection in wireless sensor networks (WSNs) is the process of identifying those sensor readings that do not conform to normal behavior and model of data and indicate occurrence of an out of ordinary situation.

Interest in event detection in wireless sensor networks is increasing and many applications such as safety and security [1], health and well-being [2], hazardous detection [3], activity monitoring [4], and vehicle tracking [5] depend more and more on fast and accurate detection of out of ordinary situations. From the data processing point of view, there are generally two main approaches to tackle the problem of event detection in WSNs, i.e., (i) to transmit raw sensor data to a base station for centralized event detection, and (ii) to enable event detection locally at each sensor node and report the detection result to a base station or sink. One of the drawbacks of the former approaches is rapid energy exhaustion of sensor nodes due to continuous data transmissions, which in turn will shorten the lifetime of the WSN. Additionally, since these approaches often rely on availability of large amount of data for accurate event detection, event detection and notification process usually suffer from delays, which in turn make these approaches not suitable for real-time applications. Approaches of the latter case, on the other hand, offer the benefit of fast and energy-efficient event detection due to local detection of events while may suffer from low detection accuracy and lack of generality as the event detection is based on data of individual sensor nodes with their limited perception [6].

The event detection itself can be performed using simple comparisons against some threshold values or using sophisticated machine learning techniques. While comparing against threshold values has the advantage of having low computation and communication complexity, it is limited in a sense that it fails in detecting complex and multivariate events. There are a number of machine learning and artificial intelligent based event detection techniques in literature. However, they are mostly supervised techniques and perform well in detecting events with known patterns and semantics.

By targeting events with unknown patterns and signatures, in this paper we focus on unsupervised learning techniques and propose a modified version of K-means algorithm for fast in-network unsupervised event detection in WSNs. The reason to modify the standard K-means algorithm is to cope with the problem of batch offline processing, which the standard K-means algorithm is known for. The reason we explore use of unsupervised learning techniques is their ability to detect patterns that may not have been previously seen [7].

The rest of this paper is organized as follows: Section II surveys the related work. An introduction to standard K-means algorithm is presented in Section III. Our unsupervised event detection approach is explained in Section IV. Analysis of datasets we use in our performance evaluation is presented in Section V, while Section VI reports the empirical results. Finally Section VII summarizes the research and draws some conclusions.

II. RELATED WORK

Classical research on event detection can be categorized into research on (i) pattern matching for known events and (ii) pattern recognition for unknown events. Supervised learning techniques are often used for finding events that are similar to predefined event signatures, while unsupervised learning techniques are used for finding hidden patterns.

While many studies target supervised learning in WSN, to the best of the authors' knowledge, there are only a few studies, which target unsupervised learning in WSNs and even then they have not considered event detection as their core application. For instance, authors of [8] propose a combination of clustering technique and support vector machine (SVM) to detect intrusions in WSNs. As proposed in [9], self-organizing map (SOM) and genetic algorithm (GA)

can be used for finding trust values in WSNs. Zhang et al. propose an unsupervised centered quarter-sphere support vector machine for outlier detection in WSNs [10]. Aggregation Tree is also proposed for unsupervised outlier detection in WSNs [11]. Adaptive Resonance Theory (ART) and Fuzzy-ART are proposed in [12] to classify environmental data into clusters in an unsupervised fashion.

Supervised learning techniques have been widely used for event detection in WSNs, examples include map-based [13], probabilistic-based [14], K-nearest neighborhood-based (K-NN), maximum likelihood-based, support vector machines-based (SVM) [15], naïve Bayes-based [16] [17] [18], feed forward neural networks-based [16], distributed fuzzy engine-based [19], voting graph neuron-based [20], and distributed decision trees-based [21] techniques.

Figure 1 presents taxonomy of event detection techniques designed for WSNs.

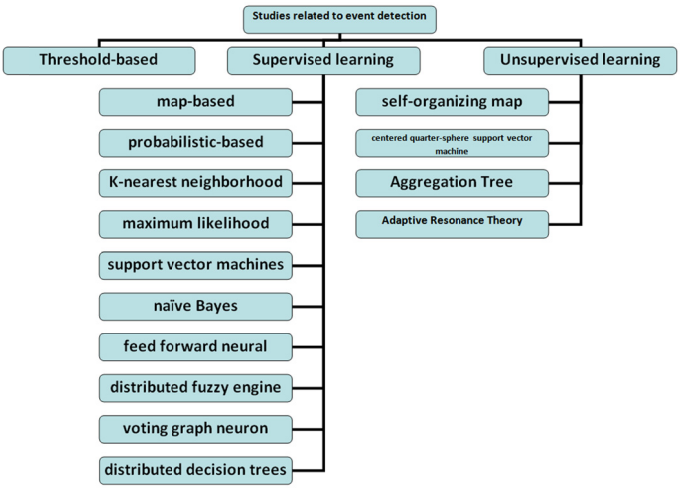


Figure 1: Taxonomy of event detection techniques designed for WSNs

III. Standard K-Means Algorithm

The K-means algorithm is one of the basic unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows an uncomplicated way of classification of a given dataset through a certain number of clusters (k clusters). The main idea of the standard K-means algorithm is to define k centroids, one for each cluster and then partition *n* observations into *k* clusters in such a way that each observation belongs to the cluster with the smallest mean values.

The term "K-means" was firstly used by James MacQueen in 1967 [22]. The K-means algorithm performs the following steps:

1. Placing *K* points into the space representing the center of objects being clustered. These points represent initial group means.
2. Assigning each object to the group that has the smallest mean by using Euclidean distance $d = \sqrt{\sum_i^n (x_i - y_i)^2}$.
3. Recalculating the positions of the K–means (center of clusters) using Equation (1) after all objects are assigned to clusters.

$$new\ Center_l = \frac{1}{n}\sum_{i}^{n} x_i \qquad \text{Equation (1)}$$

4. Repeating Steps 2 and 3 for the rest of data.

The convergence of the algorithm is discussed in [23]

IV. A Fast Unsupervised Event Detection Approach

Major drawbacks of the standard K-means algorithm for being applied in WSNs include its batch and offline clustering characteristics as well as high computational and memory complexity. Therefore in what follows we explain how to modify the algorithm to make it simpler and less computationally exhaustive.

A. Modified K-means Algorithm

Our aim of modifying the standard K-means algorithm is to lower down its time, memory, and computational complexity. This in turn will lead to a simple classifier to be used for fast event detection on resource limited wireless sensor nodes. To do so, we follow the following steps:

1. The first *k* data are assigned to the first *k* event classes. By doing this the first K-means (first *k* cluster centers) are created.
2. A tracking table is made to hold track of the previously seen data in a compact form. The tracking table contains *k* columns, each representing one of the *k* clusters. In each column the center of the clusters as well as the number of populations within the respective clusters are stored. Figure 2 illustrates how a tracking table looks like.
3. The Manhattan distance between the $k+1^{th}$ data instance (the next coming data) and *k* means (center of *k* clusters) is calculated based on Equation 2 and the $k+1^{th}$ data is assigned to the closest cluster.

$$d_j = \sum_{i}^{m} |x_i - y_i| \qquad \text{Equation (2)}$$

Where, j=(1,2,..*k*), d_j is distance to j^{th} cluster center, m is number of features (dimensions), x_i is the i^{th} dimension of the data, and y_i is the center of cluster *j* in i^{th} dimension.

4. The tracking table will be updated so that the population of the assigned cluster is increased by one. Consequently, the center of the clusters will be updated based on Equation 3.

$$y_j = \frac{(P_j \times OldCenter_j) + x_i}{P_j + 1} \qquad \text{Equation (3)}$$

Where, j=(1,2,..*k*), y_j is j^{th} cluster center, *m* is number of features (dimensions), x_i is the i^{th} dimension of the data (i=1,2,..,*m*), and P_j is population of j^{th} cluster.

5. Steps 3 and 4 will be repeated for all incoming data instances.

	Clusters			
	Cluster *1*	**Cluster *2***	**...**	**Cluster *K***
Centers	$<c_{1,1}, c_{1,2}, .., c_{1,m}>$	$<c_{2,1}, c_{2,2}, .., c_{2,m}>$	...	$<c_{k,1}, c_{k,2}, .., c_{k,m}>$
Population	P_1	P_2	...	P_k

Figure 2: Tracking table which holds a track of previously seen data in a compact form

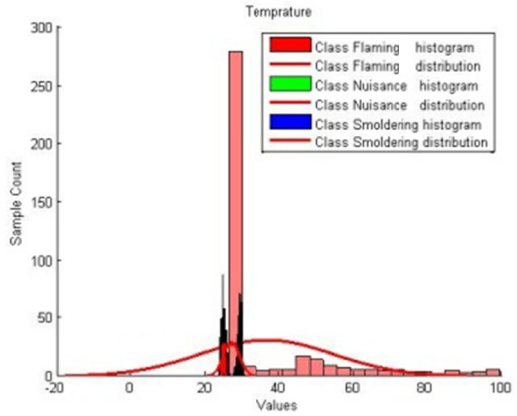
The proposed algorithm takes similar steps as the standard K-means algorithm and its convergence is proofed as the standard K-means [23].

V. Data Analysis

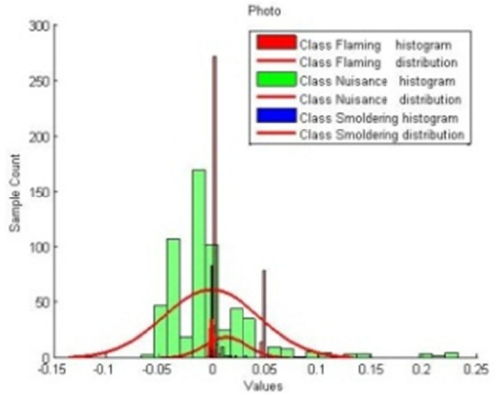
To do a cross validation of our proposed approach, we will use two datasets. The first dataset is a real residential fire dataset collected by the National Institute of Standards and Technology (NIST) [3]. The second dataset used is a real activity dataset of a healthy person performing standing, siting, and walking activities collected by Medisch Spectrum Twente (MST) [4].

The fire dataset contains 1400 data instances, 4 features (temperature, ionization, photoelectric and CO), and 3 classes (flaming fires, smoldering fires, and nuisance). Figure 3 illustrates the role of each feature in generation of different classes (events).

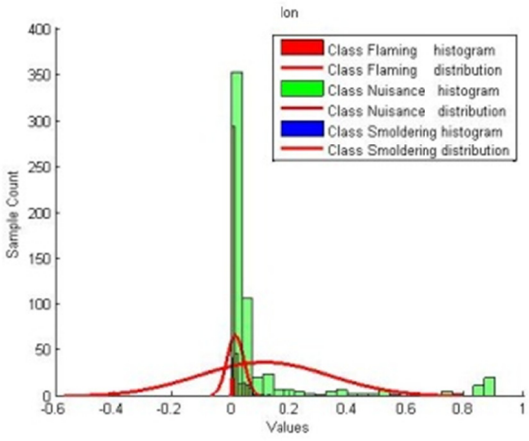
The activity dataset contains 8330 data instances and 3 classes namely walking, sitting, and standing still. It also contains four features ('Z' vector of gyroscope installed on the right foot, 'Y' vector of accelerometer installed on trunk, 'Z' vector of accelerometer installed on trunk, and 'X' vector of accelerometer installed on left foot). Figure 4 shows the role of each feature in generation of different classes (events). As it can be seen from Figure 4, the activity data shows high degree of overlaps between features in the generated classes, which makes classifications of data difficult.



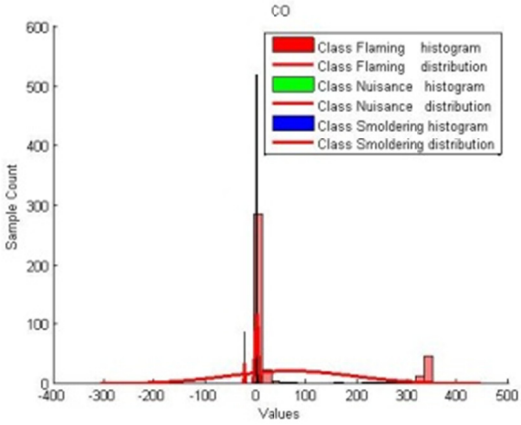
(a)



(b)

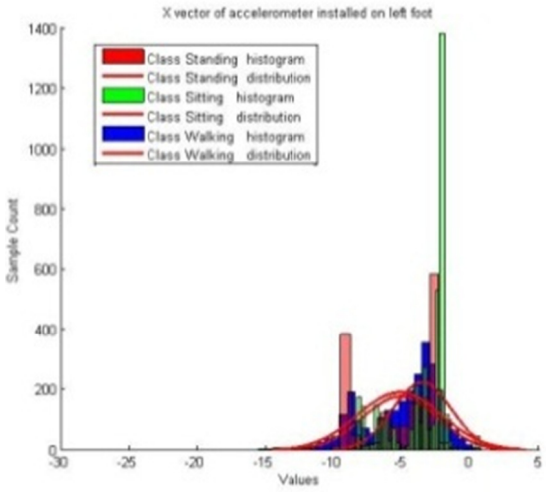


(c)

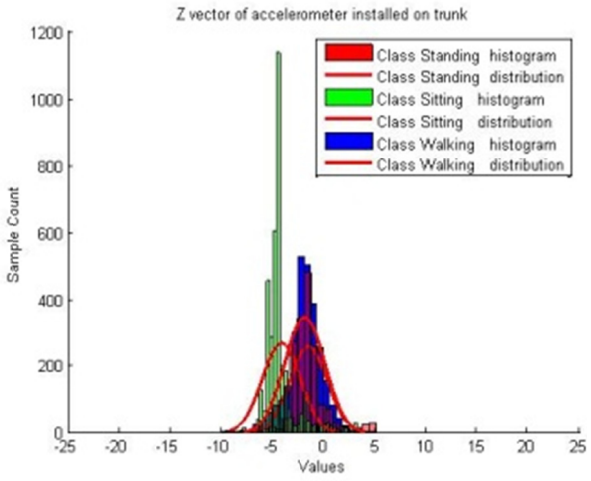


(d)

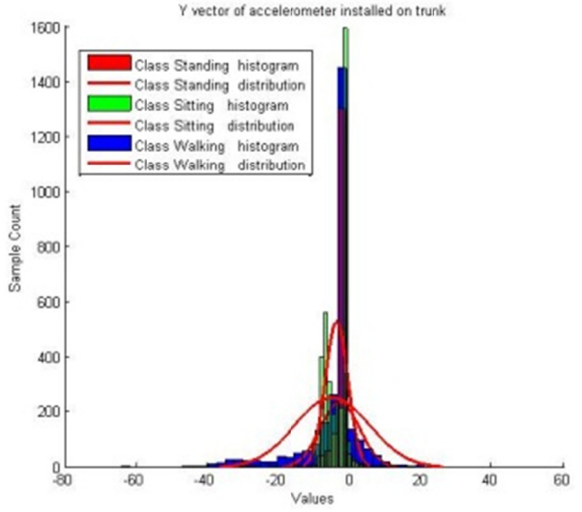
Figure 3: Data distribution of fire dataset for four features: (a) temperature sensor, (b) photoelectric (photo), (c) ionization (ION), and (d) CO features.



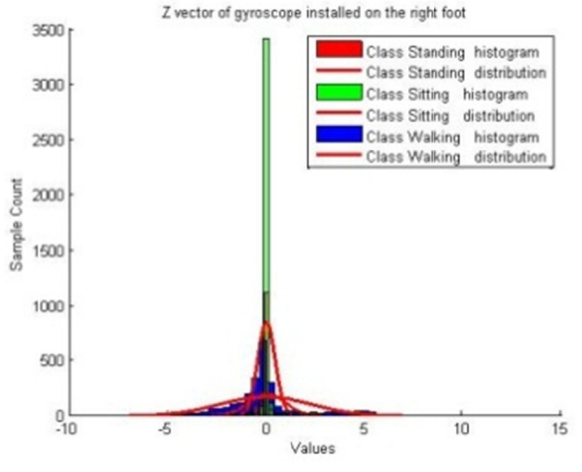
(a)



(b)



(c)



(d)

Figure 4: Data distribution of activity data set for four features: (a) 'X' vector of accelerometer installed on left foot, (b) 'Z' vector of accelerometer installed on trunk, (c) 'Y' vector of accelerometer installed on trunk, (d) 'Z' vector of gyroscope installed on the right foot

VI. Empirical Results

To evaluate our fast unsupervised event detection algorithm, the fire and activity datasets described in Section V are used in a simulation environment developed in Matlab®.

We compare our algorithm with the standard K-means algorithm described in Section III. Additionally, we investigate the effect of the distance function used in the algorithm and compare performance of the algorithm using Euclidean distance, Manhattan distance (Equation 2), and Malahanobis distance (Equation 5).

$$d_j = \sqrt{(\mu - x)^T S^{-1} (\mu - x)} \qquad \text{Equation (5)}$$

Where, d_j (j=1,2,…,k) is Malahanobis distance between x to μ, μ is $1 \times m$ matrix holding mean values of a population for distance calculation and m is the number of features, S is covariance matrix and x is the data in m dimensional space.

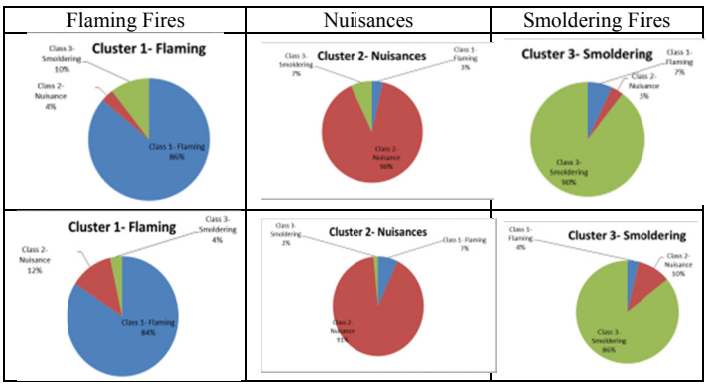
For the performance evaluation, we consider five metrics, i.e., (i) event detection accuracy, (ii) false alarm, (iii) similarity calculation (using the Rand Index), (iv) computational complexity, and (v) memory complexity. Table I and II present our simulation results in terms of detection accuracy and false alarm for the two datasets.

TABLE I: PERFORMANCE EVALUATION IN TERMS OF DETECTION ACCURACY AND FALSE ALARM FOR THE FIRE DATASET

Technique	Detection Accuracy	False Alarm
Modified K-means using Manhattan distance	87,16%	12,84%
Modified K-means using Euclidean distance	87,85%	12,15%
Modified K-means using Malahanobis distance	72,71%	27,29%
Standard K-means	88,41%	11,59%

TABLE II: PERFORMANCE EVALUATION IN TERMS OF DETECTION ACCURACY AND FALSE ALARM FOR THE ACTIVITY DATASET

Technique	Detection Accuracy	False Alarm
Modified K-means using Manhattan distance	48,65%	51,35%
Modified K-means using Euclidean distance	50,31%	49,69%
Modified K-means using Malahanobis distance	45,42%	54,58%
Standard K-means	51,32%	48,68%



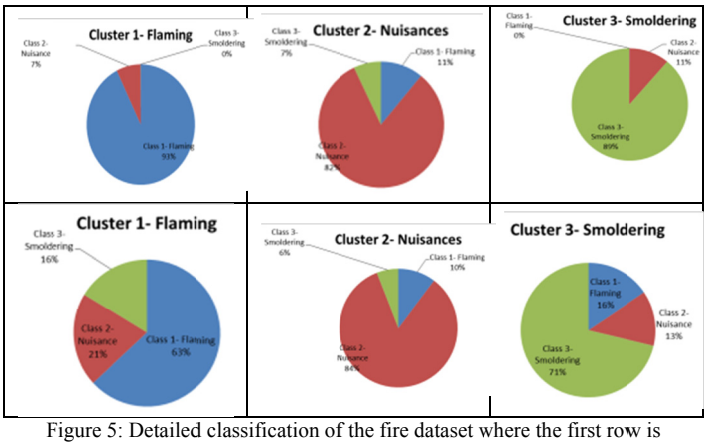


Figure 5: Detailed classification of the fire dataset where the first row is Standard K-means, the second row is Modified K-means using Manhattan distance, the third row is Modified K-means using Euclidean distance, and the forth row is Modified K-means using Malahanobis distance

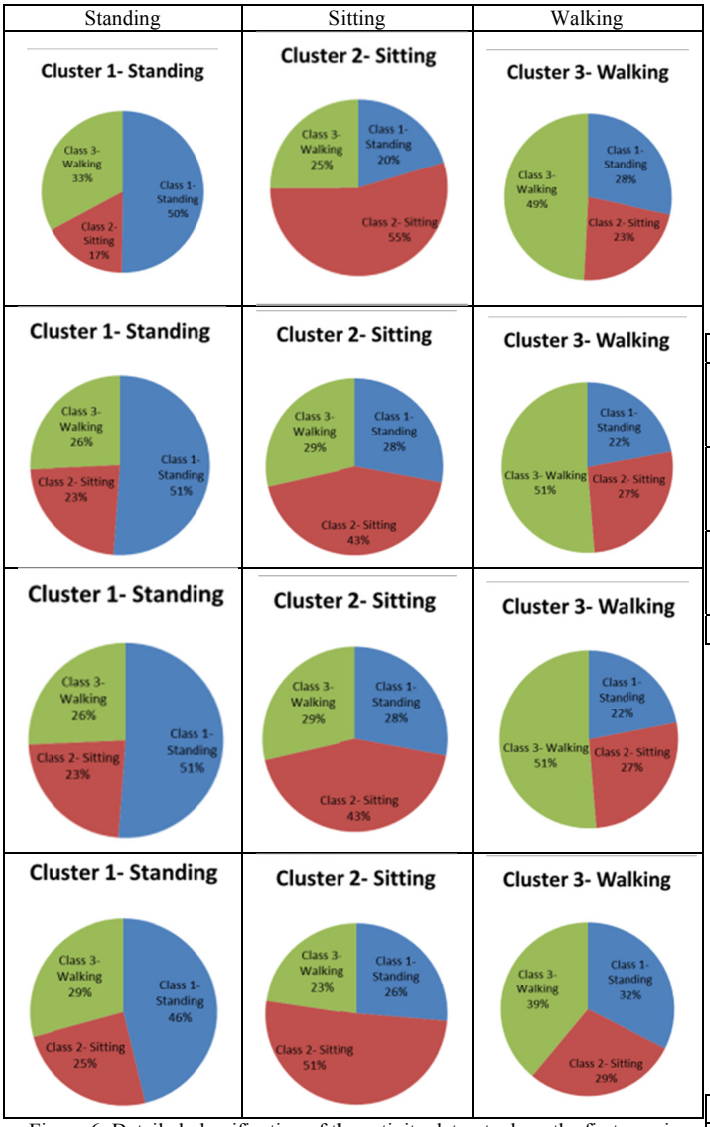


Figure 6: Detailed classification of the activity dataset where the first row is Standard K-means, the second row is Modified K-means using Manhattan distance, the third row is Modified K-means using Euclidean distance, and the forth row is Modified K-means using Malahanobis distance

As it can be seen from Table I and II in the first instance all algorithms have almost the same detection accuracy. However, standard K-means performs slightly better than the rest. The modified K-means algorithm using Malahanobis is the least accurate one.

One can noticed that the overlaps of each class shown in Figures 5 and 6 indicate the false alarm rate of the classification process.

Figures 5 and 6 present the event detection accuracy of the K-means and our modified K-means algorithms more clearly by showing percentages of the correctly classified data (detection rate) as well as wrongly classified data(false alarm).

Similarity calculation shows how similar the clustering results are with the actual values. This similarity can be calculated using the Rand Index presented by Equation 6. Table III reports the Rand Index values for the standard K-means and our modified K-means algorithms techniques.

$$RI = \frac{TP + TN}{TP + TN + FP + FN} \qquad \text{Equation (6)}$$

Where, RI is Rand Index ($0 \leq RI \leq 1$, 0 means no similarity and 1 shows 100% similarity), TP is true positive, TN is true negative, FP is false positive and FN is false negative.

As Table III shows, similarity between clustering results and actual values for the standard K-means and the proposed approach using Manhattan and Euclidean distances is almost the same.

TABLE III: THE RAND INDEX FOR STANDARD AND MODIFIED K-MEANS EVENT DETECTION TECHNIQUES

Techniques	Fire dataset	Activity dataset
Modified K-means using Manhattan distance	91,44%	65,77%
Modified K-means using Euclidean distance	91,90%	65,77%
Modified K-means using Malahanobis distance	81,81%	63,62%
Standard K-means	92,27%	67,54%

Another metric to consider is the computational complexity. The most time consuming part of a clustering algorithm is calculating the distance. We ran 100 time distance calculation in an HP laptop with Intel Dual Core 2,5 GHz CPU with 4 GB RAM. The average running time of distance calculation for each algorithm is reported in Table IV. It can be seen that calculation of the Manhattan distance is the fastest and calculation of the Malahanobis distance is the slowest. In fact, calculation of the Manhattan distance is 1.44 times faster than calculation of the Euclidean distance and 95.44 faster than calculation of Malahanobis distance.

TABLE IV: AVERAGE RUNNING TIME OF THE DISTANCE FUNCTIONS

Distance Measurement	Average Running Time
Manhattan distance	2.6033e-006 sec.
Euclidean distance	3.7490e-006 sec.
Malahanobis distance	2.4845e-004 sec.

The last metric to consider is the memory complexity to get an idea about how much memory is needed for executing the algorithm. Table V reports the memory complexity of our

modified K-means algorithm versus the standard K-means algorithm in terms of Big O notation.

TABLE V: MEMORY COMPLEXITY OF THE MODIFIED K-MEANS AND THE STANDARD K-MEANS ALGORITHMS

Standard K-means	$f(n) = O[(n+k)m]$
Proposed algorithm	$f(n) = O(m \times k)$
Where , *n* is total data, *m* is dimension of data (number of features) and *k* is number of clusters (or classes)	

It can be seen from Table V that for the standard K-means algorithm we need to store *k* means (*k* cluster centers) and the entire data instances in an *m* dimensional table inside the memory. For our modified K-means algorithm, we only need to keep the tracking table in memory which is a table of *k* columns and *m* rows, where *k* is number of classes and *m* is number of features. Memory complexity of the propose algorithm is independent of number of inputs and therefore occupies less space in memory.

VII. CONCLUSION AND FUTURE WORK

In this paper, we propose a modified version of the K-means algorithm for fast unsupervised event detection in WSNs. The advantage of our unsupervised event detection technique is its ability to detect unknown events in absence of a priori knowledge about event semantic and signature. It is also simple in terms of memory and computation complexity.

To evaluate the proposed approach, two real datasets are used and the algorithms are compared in terms of detection accuracy, false alarm rate, computational complexity and memory complexity. The results of these evaluations show that compared with the standard K-mean algorithm, our modified K-means algorithm using Manhattan distance is the faster and also requires lesser memory. The detection accuracy of both algorithms, however, remains (almost) the same.

In our future work, we will explore the possibility and usefulness of making the modified K-means algorithm distributed. This will be done with the idea of making the algorithm more accurate and achieving a better load balancing in the WSNs.

ACKNOWLEDGMENT

Authors would like to thank Dr. Mannes Poel and Dr. Behrooz Safarinejadian for their valuable comments and suggestions for this study. This work is supported by the EU Seventh Framework Programme, IS-ACTIVE and the SENSEI project.

REFERENCES

[1] T. He, *et al.*, "VigilNet: An integrated sensor network system for energy-efficient surveillance," *ACM Transactions on Sensor Networks (TOSN),* vol. 2, pp. 1-38, 2006.

[2] E. Shih, *et al.*, "Sensor Selection for Energy-Efficient Ambulatory Medical Monitoring," presented at the MobiSys '09 2009.

[3] M. Bahrepour, *et al.*, "Fast and Accurate Residential Fire Detection Using Wireless Sensor Networks," *Environmental Engineering and Management Journal,* vol. 9, pp. 215-221, 2010.

[4] M. Bahrepour, *et al.*, "Sensor Fusion-based Activity Recognition for Parkinson Patients," in *Sensor Fusion - Foundation and Applications*, ed: InTech, 2011, pp. 171-190.

[5] M. Duarte and Y. Hu, "Vehicle Classification in Distributed Sensor Networks," *Journal of Parallel and Distributed Computing,* 2004.

[6] G. Wittenburg, *et al.*, "A System for Distributed Event Detection in Wireless Sensor Networks," presented at the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, 2010.

[7] B. A. Wooley. (1999). *Scaling Clustering for the Data Mining Step in Knowledge Discovery*. Available: http://www.cs.utexas.edu/users/csed/doc_consortium/DC99/wooley-abstract.html

[8] W. S. Hortos, "Unsupervised algorithms for intrusion detection and identification in wireless ad hoc sensor networks," presented at the Intelligent Sensing, Situation Management, Impact Assessment, and Cyber-Sensing, 2009.

[9] Z. Banković, "Unsupervised intrusion detection for wireless sensor networks based on artificial intelligence techniques," PhD, Universidad Politécnica de Madrid, Madrid, 2011.

[10] Y. Zhang, *et al.*, "An online outlier detection technique for wireless sensor networks using unsupervised quarter-sphere support vector machine," presented at the Intelligent Sensors, Sensor Networks and Information Processing, Sydney, Australia, 2008.

[11] K. Zhang, *et al.*, "Unsupervised Outlier Detection in Sensor Networks Using Aggregation Tree," in *Advanced Data Mining and Applications*. vol. 4632, R. Alhajj, *et al.*, Eds., ed: Springer Berlin / Heidelberg, 2007, pp. 158-169.

[12] A. Kulakov and D. Davcev, "Tracking of unusual events in wireless sensor networks based on artificial neural-networks algorithms," in *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, 2005, pp. 534-539 Vol. 2.

[13] A. Khelil, *et al.*, "MWM: A Map-based World Model for Wireless Sensor Networks," presented at the Autonomics, Turin, Italy 2008.

[14] J. Yin, *et al.*, "Spatio-temporal event detection using dynamic conditional random fields," in *International Joint Conference On Artificial Intelligence*, Pasadena, California, USA, 2009.

[15] D. Li, *et al.*, "Detection, classification, and tracking of targets," *Signal Processing Magazine, IEEE,* vol. 19, pp. 17-29, 2002.

[16] M. Bahrepour, *et al.*, "Use of AI Techniques for Residential Fire Detection in Wireless Sensor Networks," presented at the AIAI 2009, Greece, 2009.

[17] B. Krishnamachari and S. Iyengar, "Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," *IEEE Transactions on Computers* vol. 53, pp. 241-250, 2004.

[18] M. Bahrepour, *et al.*, "Sensor Fusion-based Event Detection in Wireless Sensor Networks," presented at the SensorFusion'09, Toronto, Canada, 2009.

[19] M. Marin-Perianu and P. J. M. Havinga, Eds., *D-FLER - A Distributed Fuzzy Logic Engine for Rule-Based Wireless Sensor Networks*. Springer Verlag, Germany, 2007, p.^pp. Pages.

[20] M. Baqer and A. I. Khan, "Event Detection in Wireless Sensor Networks Using a Decentralised Pattern Matching Algorithm," *White Paper,* 2008.

[21] M. Bahrepour, *et al.*, "Distributed Event Detection in Wireless Sensor Networks for Disaster Management," presented at the International Conference on Intelligent Networking and Collaborative Systems, INCoS 2010, Thessaloniki, Greece, 2010.

[22] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations," in *5th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, 1967, pp. 281–297.

[23] P. E. H. Richard O. Duda, David G. Stork, *Pattern Classification (2nd Edition)*: Wiley-Interscience, 2000.