SENOCLU, Energy Efficient Approach for Unsupervised Node Clustering in Sensor Networks

Adriola Faqolli

Mathematics, Statistics & Applied Informatics Department,

Faculty of Economy, University of Tirana, Albania

adriola_faqolli@hotmail.com

***Abstract* – Acquisition and analysis of data from sensor networks, where nodes operate in unsupervised way, has become a ubiquitous issue. The biggest challenge in this process is related to limited energy, computational and memory capacity of sensor nodes. Therefore, the main goal of our work is to devise and evaluate the contribution of an energy efficient algorithm for data acquisition in sensor networks.**

The proposed SENOCLU algorithm considers specific requirements of sensor network application like energy efficiency, state change detection, load balancing, high-dimensions of the sensed data etc. By applying these techniques, this algorithm contributes in filling the gap between distributed clustering and high-dimensional clustering algorithms that are available in the literature. This work evaluates the contribution of this algorithm in comparison to other competing state-of-the-art techniques.

The experiments show that by applying SENOCLU algorithm better life times of sensor networks are achieved and longer monitoring of different phenomena is provided.

***Index Terms* - Sensor Networks, Energy Efficiency, Self-Organization, Change Detection, Load Balancing.**

I. INTRODUCTION

Nowadays devices with sensing ability are useful for monitoring different phenomena and making processes of our life easier. These devices produce a large amount of data across wide sensor networks. Therefore, gathering and analyzing data of sensor network has become an important issue. This process is challenging because sensor nodes that are part of these networks, have limited energy, computational and memory capacity. Limitation in energy especially affects sensor network lifetime.

To face these challenges and to make this process more efficient, it is proposed a new algorithm that introduces an energy efficient way for data acquisition in sensor networks.

The scope of this work is evaluating the contribution of this algorithm in energy efficient node clustering.

The proposed SENOCLU algorithm suggests a self-organization of sensor nodes into clusters. This is an important feature that makes this algorithm useful for most of sensor network applications, where nodes operate in an unsupervised way. According to research over Mica2 [1] and Telos [2] sensor nodes, usually 90-95% of total energy consumption in a node goes for communication process. Thus, in this algorithm only a subset of nodes in the network has direct communication with data sink. In this way, more energy is saved in communication and longer sensor network lifetime is provided.

This algorithm takes into consideration the distributed nature of data in sensor networks and the high dimensionality of it. By adding specific requirements of sensor network applications like energy efficiency, state change detection, load balancing etc., this algorithm contributes in filling the gap between distributed clustering and high-dimensional clustering algorithms that are available in the literature. These techniques are adapted to sensor network applications.

Evaluation shows that this algorithm reduces energy consumption in sensor nodes and provides an accurate way for collecting data from sensor networks.

II. SENOCLU ALGORITHM

In this algorithm, nodes maintain a cache of their previous measurements for each attribute. Based on the previous data, each node is able to detect the outlier values. Not all the nodes communicate directly with data sink. One representative per each cluster forwards the statistics of the sensed data to data sink. Nodes do not send data continuously to their representatives. They send data about the event that they monitor, only when they detect a state change in this event. Representatives calculate the statistics for each attribute, for all the values that they receive from their cluster nodes. They report these statistics to data sink only when they detect a state change in the attribute values of their clusters. When any representative realizes that its energy capacity falls under a certain threshold, it delegates its representative authority to another suitable node on its cluster.

If the measurements sensed by a node are not anymore similar to the measurements of other cluster nodes, this node leaves its current cluster and tries to find for itself any other suitable neighbor representative. If it finds, it joins to that cluster, otherwise it remains alone in its cluster. Nodes that are alone in their clusters periodically send heartbeats to their neighbor representatives and ask them to join their clusters. If any representative itself measures very different values from its cluster nodes, all the nodes of its cluster perform locally once more clustering process.

A. Initialization of Data History

In this algorithm, nodes are organized in one cluster based on their geographical proximity and similarities in the attribute measurements that they sense. Each node estimates this similarity based on the measurements that it receives from its neighbor nodes. To provide an accurate estimate, the measurements that every node broadcasts to its neighbors should not be outliers. Moreover, in order to improve the energy efficiency in sensor networks, we try to reduce communication cost as much as possible in this algorithm. In each cluster, nodes do not have a constant communication with their representatives. Each node sends data to representative only when it detects a state change in the event that it is monitoring. In order to detect a state change, each node compares for each attribute the recent data value that is senses to the previous one. If the difference of these data values is greater than a certain threshold, the node decides to send the data to the representative; otherwise, there is no need of sending these data.

The check for accuracy of data is needed before clustering process starts. In this way nodes make sure that the attribute measurements that they broadcast to their neighbors are accurate ones. Therefore, the first step of this algorithm is initializing of data history. Each node senses the first *k* measurements for each attribute and stores them in the cache of data history.

B. Detection of Geographical Neighbors

In order to improve energy efficiency in sensor networks, long-distance communication should be reduced as much as possible. Therefore, the first condition for a group of nodes to be part of the same cluster is their geographical proximity. In the second step of this algorithm every node detects its geographical neighbors *GN* by running spatial similarity queries with radius ε. Before this process starts, every node *n* broadcasts within a radius ε its *ID* and its spatial attributes < $ID_n; x_n, y_n, z_n$ > In this way, every node becomes aware of geographical coordinates of its neighbors.

Spatial similarity query $sim_{spatial}$ with radius ε, executed by node $n \in SN$ (sensor nodes) returns the set of nodes n_i such that: $sim_{spatial}\,(n,n_i) = \{ n_i \in SN \mid d(n, n_i)) \leq \varepsilon \}$

The similarity distance *d* for spatial attributes is Euclidean distance.

C. Setting Relevant Attributes

In SENOCLU algorithm subspace clustering is applied. This means that sensor nodes are clustered together according to similarities of a subset of non-spatial attributes. In this step, the representatives of each cluster decide which attributes are relevant for the possible cluster represented by each of them.

Initially each node senses the next measurement for each attribute. Based on its data history it analyses whether these measurements are correct ones or outlier values. The measurements that are confirmed to be correct values are broadcasted to all the geographical neighbors. The ones that are suspected to be outliers are not broadcasted. Instead of them, sensor node broadcasts to its neighbors the corresponding previous measurements that it has already stored in its cache.

Each node after receiving this data from its neighbors, calculates for each attribute the mean value and standard deviation of the measurements of all its neighbors, including the measurements of itself. Based on these values it will set the relevant attributes for the possible cluster of nodes represented by it. A non-spatial attribute a_m, where $1\leq m \leq k$, is called a relevant attribute for possible cluster with representative n, iff $a_n m \in [-2\sigma_m + \mu_m,\ 2\sigma_m + \mu_m]$, where $a_n m$ is the current measurement for attribute *m* of representative *n* ; σ_m and μ_m are respectively the standard deviation and mean value of measurements for non-spatial attribute *m*.

D. Detection of Candidate Cluster Member

In this step, each node finds out which of the nodes among its geographical neighbors are similar to it according to the measurements of its relevant attributes. These nodes will be the candidate members of its cluster. Let *RA* be the set of relevant attributes for possible cluster *C* with representative *n.* A geographical neighbor *q* of node *n* can be a candidate cluster member for *n* if the following condition is satisfied for all the relevant attributes of node *n*.

$$\forall\, a_r \in RA:\quad a_q r \in [-2\sigma_r + \mu_r,\ 2\sigma_r + \mu_r]$$

$a_q r$ is the current value of neighbor node *q* for each relevant attribute a_r of possible representative *n*.

Node *n* repeats the same procedure for every node that belongs to its *GN* set. Each node $n_i \in SN$ performs the same procedure as well. At the end of this step, each node n_i of the sensor network has detected all its candidate cluster members CCM_i for the possible cluster represented by itself.

E. Estimation of Representation Quality for Each Node

In this step of algorithm, each node analyzes how good it is in representing the rest of its cluster nodes in the network. To estimate its representation quality, each node refers to its *CCM* set and its own residual energy *RE*. Based on the density of nodes that belong to its *CCM* set and on the residual energy in the moment of calculation, each node assigns to itself a representation quality parameter *RepQ.*

Higher this density is, less energy is spent in communication between the candidate representative and its cluster nodes. Higher the residual energy of the candidate representative is, longer and more efficiently it can perform its representative task. Therefore, a mixture of these two features in RepQ parameter makes a good criterion in selecting representatives in SENOCLU algorithm. Out of the nodes of one cluster, the node with the highest RepQ value will be selected as representative. This representative will represent a high density of nodes and will have sufficient residual energy to perform its tasks for a long period.

F. Selection of Local Representatives

Each node n broadcasts its *RepQ* value to every node n_i that belongs to its *CCM.* In this way, every node will be aware about the *RepQ* value of all nodes that pretend to represent it in the network. Every node *n ϵ SN* stores the list of candidate local representatives *CLR*, together with the *RepQ* values received by them. It includes also itself in this list. It ranks this list in decreasing order based on the *RepQ* value of each node. Then, it decides about its representative based on the following rules:

- If its own *RepQ* value is greater than all the *RepQ* values of other nodes that belong to its *CLR*, node *n* claims itself representative.
- If the greatest *RepQ* value is the same for more than one node of *CLR*, the closer node is selected as representative.
- If none of the above conditions is true, node n selects as its local representative node n_i *ϵ CLR* that has the greatest *RepQ* value.

G. Load Balancing Among Representatives

To avoid overloading of one representative comparing to other representatives in the network, we set a threshold *MaxNds* for the maximum number of nodes that can be represented by a representative. In other words, for the maximum number of nodes that is included in one cluster.

Based on this idea each local representative checks whether the condition CluSize ≤ *MaxNds* holds or not.

If this condition holds, no change is done in the current cluster of the respective local representative. Otherwise, the representative excludes from its cluster the most distant cluster members. In this way, it is provided a uniform utilization of energy resources in sensor network. This process is illustrated in Figure 1.

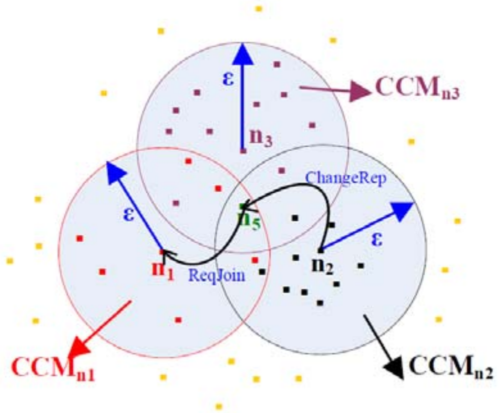

Fig. 1 Load Balancing Among Representatives

H. Check for Outlier Detection

To enhance the accuracy of the attribute measurements that are reported to the data sink and to provide a stable cluster structure, we address outlier detection in our approach. Otherwise, clustering structure would be based on wrong values of attributes and the similarity among nodes would not be detected correctly. The coming measurements would contradict this wrong decision and in this way re-clustering process would take place. Under this scenario energy consumption would increase. To prevent this situation, each sensor node detects the deviating measurements that it senses and excludes them from the data that is reported to the respective local representative. These measurements are ignored and they are not stored even in the corresponding cache of data history.

I. Communication

This algorithm initiates communication process only when a state change is detected. Nodes communicate with their representatives only when they detect a state change in the attribute measurements of the event they are monitoring. The same technique is applied in communication between representatives and data sink as well. Representatives send data to data sink only if they detect a state change in the statistics of the measurements collected from all the nodes of their clusters.

Nodes communicate with their representatives not only for reporting the measurements that they have sensed, but also for reporting their residual energy. We have assumed that nodes in our algorithm are energy aware. When a node notices that its residual energy has decreased too much and its value is under a very low threshold, it sends to its local representative the current value of its residual energy. After this information, local representative will be able to realize that in the next iterations the missing measurements from this node will be due to its shortage of energy.

J. Maintenance of Clusters

The main goal of a sensor network is to continuously monitor the physical phenomena in the environment. To achieve this goal continuous data acquisition from sensor network is provided. Therefore, the self-organizing and distributed clusters of sensor nodes are being maintained during all the application. This process is maintained by sensor nodes themselves in an unsupervised way.

K. Optimization in Case of Single Node Cluster

Due to the differences in non-spatial attribute values, any of the nodes may be alone in its cluster. Single node clusters are not a good option, because in this case the single node of the cluster continuously communicates in long distance with the data sink, consuming in this way a lot of energy. To avoid this scenario, in this algorithm each node that is alone in its cluster sends periodically heartbeats to its neighbor representatives and asks them to join their clusters.

Each neighbor representative after receiving the *'join request'* from any node checks whether the attribute measurements of this node fulfill the criteria of its own cluster. If these criteria are fulfilled it sends back an *'approve acknowledgment'*. If these criteria are not fulfilled then it sends back a *'reject acknowledgment'*.

This node waits for *'approve acknowledgment'* from other neighbor representatives. If it receives more than one *'approve acknowledgment'* it selects as its new representative the nearest neighbor representatives among those nodes that have approved it. Unless it receives any *"approve acknowledgment",* it continues being alone in its cluster.

L. Delegation of Representative Authority

Nodes that are selected as representatives perform more communication and computation tasks than other nodes in the sensor network. In order to save the energy of representatives, to provide a uniform utilization of energy sources in the network and to enhance the sensor network lifetime, when a representative notices that its energy capacity is under a very low threshold, it delegates its representative authority to another node of its cluster.

Each local representative is aware of its residual energy. In the moment when it notices that its energy capacity is decreased under a certain threshold, local representative sends a message to its cluster nodes and asks them to send back their residual energy value. After this, representative checks whether the difference of its own residual energy with the residual energy of any other cluster node is greater than a certain threshold or not. If there are more than one cluster nodes that fulfill this condition, representative decides to delegate its authority to the one with the highest residual energy value among them. If there is only one node that fulfills this condition, representative directly delegates its authority to this node. If this condition is not fulfilled by any of the cluster nodes, the current representative continues being the representative of its cluster and performs the same check later again.

III. COMPARISON WITH RELATED WORK

For evaluating the contribution of SENOCLU algorithm on energy efficient sensor nodes clustering, the features of this algorithm are compared to related works on traditional clustering algorithms, distributed clustering algorithms, energy efficient clustering techniques and techniques for selecting node representatives.

A. Traditional Clustering Algorithms

There is extensive literature on clustering techniques in data mining and machine learning communities. In general clustering algorithms can be classified into many categories as: partitioning algorithms, density based algorithms, hierarchical algorithm etc.

In *Partitioning Clustering* algorithms (*k-Means* [3], *k-Medoids* [4]) the number of clusters *k* is an input parameter. An inappropriate choice of *k* may yield poor results in clustering sensor nodes. Due to the change of attribute values sensed by sensor nodes, it is necessary to rearrange the organization of the clusters in certain instants of time. A fixed structure of cluster organization would not be the optimal representation for sensor nodes and would increase the communication cost.

Hierarchical Clustering (*BIRCH* [5]) creates a hierarchical decomposition of the given set of objects. The fact that once the step of merging or splitting is done, it can never be undone is a drawback of these techniques. The constant change in topology of sensor networks makes hierarchical techniques even more inconvenient for sensor network applications.

Density Based Clustering (*DBSCAN* [6]) algorithms do not require the number of clusters in advance, as an input parameter. They can detect clusters with arbitrary shape or size. They are robust to outliers. They tackle the complex and important problem of distributing clustering. They operate with one single scan over the input data. All these advantages motivate our choice of density based clustering technique as the core of clustering technique of the proposed algorithm.

B. Distributed Clustering Algorithms

In most of the sensor network applications, sensor nodes are not located in one site. They are distributed in the environment to monitor different phenomena. Therefore, the only clustering algorithms useful for sensor network applications are distributed algorithms. Recently a distributed algorithm called *ELink* [7] has been proposed to perform spatial clustering in sensor networks. The goal of this algorithm is to partition the network into clusters of nodes that have observed similar phenomena. This clustering is called σ-clustering, and the data dissimilarity between any two nodes inside a cluster is at most σ. ELink applies Auto-Regression (AR) to model the time series of individual nodes. This model is based on communication graph. Initially a set of nodes in the graph are nominated as root nodes. Clustering starts with root nodes and expands to include other nodes, if the Euclidean distances between their model coefficients are less than a predefined threshold. When an element cannot expand anymore, it passes the root feature to another element in the cluster to do it. Clustering terminates when none of the nodes in the cluster can expand. Although ELink outperforms some other algorithms in the literature, its performance is limited because in this algorithm nodes are not self-organized in clusters. So, this algorithm cannot be applicable in those cases when nodes operate in unsupervised way. It does not pay attention to provide a uniform utilization of resources in the network. Therefore, it does not provide a continuous data acquisition from all the locations that are monitored. Moreover, each cluster is coarsely represented by the features of the cluster root rather than the statistics of the whole cluster.

DGClust [8] is a Distributed Grid Clustering system for sensor data streams. Each sensor node receives data from a given source and produces a univariate data stream. This data stream is potentially infinite, that is why each sensor's data is processed locally and is incrementally discretized into a univariate adaptive grid. Each new data point reflects the current state of the data stream at the local site by triggering a cell in this grid. Whenever a local site changes its state, that is, the triggered cell changes, the new state is communicated to a central site. This approach introduces the idea of communicating the state to the central server only if it has changed. This principle of communication between nodes in this approach is the same as communication principle in SENOCLU algorithm. However a grid based representation of sensor nodes distribution is not an optimal one. In real application we have neither a uniform distribution of sensor nodes, nor a constant similarity among the data that the nodes sense. In this way the nodes that are part of one grid for one time interval, may not be part of it for all the application. Maintaining of this grid structure is difficult under circumstances of sensor network applications.

DSIC [9] is a Distributed Single-pass Incremental Clustering technique to cluster the time series obtained at sensor nodes. The underlying infrastructure of this algorithm is hierarchical organization of sensor network. In this algorithm sensor nodes are self-organized into a set of physical clusters based on available energy resources. Each physical cluster consists of a cluster head (CH) and several cluster members *(CMs).* Each cluster head collects data from the members and performs most of computational tasks in its cluster. All the cluster heads form a multi-hop routing tree back to the gateway. This algorithm gives a good contribution in reducing data acquisition and transmission cost in sensor networks. But its hierarchical underlying infrastructure is difficult to be maintained especially along a continuous clustering application. With passing of time the sensor readings change and according to them the positions of respective nodes in the hierarchy must change as well. In applications with frequent state change, this technique needs a frequent reorganization of all nodes in the hierarchy. Moreover, this infrastructure contains a multi-hop routing tree back to the gateway. According to the new readings, nodes have to reconsider their decision in choosing the successive cluster head to transmit the compressed data. All these restrictions make infrastructure maintenance process in this algorithm too costly in terms of energy.

SDBDC [10] is a Scalable Density Based Distributed Clustering algorithm. This approach, differently from previous density based clustering techniques, does not produce a fixed number of representatives. It allows the user to find an individual trade-off between cluster quality and runtime. Local representatives represent dense areas of nodes in a local site and they tend to be in the middle of their clusters. Although this algorithm is not aimed for clustering in sensor networks, the idea of quality driven determination of local representatives that this algorithm introduces, is useful for self-organizing clustering in sensor networks. The principle of this idea is also used in representative selection process in SENOCLU algorithm. Since this algorithm is not meant for clustering in sensor network applications, it does not take into consideration the restrictions and challenges of sensor network domain. Nevertheless, the idea that this algorithm introduces about self-organizing objects in cluster, is useful for sensor network applications, especially in those scenarios where the nodes have to operate in an unsupervised way.

C. *Energy Efficient Clustering Techniques*

In sensor network applications it is very important to provide a continuous data collection over the monitored phenomena. This continuity should be in terms of area location, size and frequency of data. If the nodes follow the straightforward solution by sending every measurement continuously to the central server, their energy will be consumed quickly. To give a solution of this problem, a group of researchers from Duke University proposed a *Data-Driven Processing Technique in Sensor Networks* [11]. The goal of this technique is to provide continuous data without continuous reporting, but with checks against the actual data. This is a common goal between this approach and our algorithm. To achieve this goal, this approach introduces temporal and spatio-temporal suppression schemes, which use the in-network monitoring to reduce the communication rate to the data sink. According to these schemes, each sensor node (updater) based on individual temporal correlation checks the recent received data against the previous data. Only when it detects a state change, it forwards the change of the data toward some successive nodes, called observers. Based on these schemes, data is routed over chain architecture from updater to observers and vice versa. In this way only the nodes that are most near to the data sink send the aggregate change of data to it. Since only these nodes perform long-distance communication, their energy will be consumed much earlier comparing to the rest of nodes. Their task will be performed by the set of other nodes that are next most near to the data sink. Soon, these nodes will face the same problem of running out of energy. This would disturb the continuity of data collection from the location near to data sink. While in SENOCLU approach, when a representative detects that its energy capacity has fallen under a certain threshold, it delegates its representative authority to another suitable node in the network. This helps in a better usage of resources of all the nodes, provides a stable cluster structure by avoiding frequent reclustering and prolongs significantly the sensor network lifetime.

Snapshot Queries: Towards Data-Centric Sensor Networks [12] is another approach that introduces a platform for energy efficient data collection in sensor networks. By selecting a small set of representative nodes, this approach provides a quick answer to user queries and reduces substantially the energy consumption in the network. Process of selecting representatives in this approach is similar to the one we perform in the proposed algorithm. Each node after comparing the representation quality of itself with the one of each candidate cluster member, it decides whether it will be represented by another node or it will be itself a representative. After a node decides which of its neighbors it can represent, it broadcasts its list of candidate cluster members to all its neighbors. Each node selects as its representative that neighbor that can represent it and that additionally has the longest list of candidate cluster members. The process of broadcasting the list of candidate cluster members is a factor that increases the energy cost of representatives selecting process in this approach. While in SENOCLU algorithm, based on the residual energy and on the density of neighbor nodes that one sensor can represent, it assigns to itself a parameter for its representation quality. It broadcasts only this parameter to its neighbors. This makes the energy cost of representatives selecting process in SENOCLU algorithm cheaper than in this approach. This model is very expensive in terms of energy, as all nodes needs to exchange all historical readings among each other. In SENOCLU algorithm, each sensor node maintains a small cache of past measurements of itself for each attribute. Based on distribution of its previous measurements it analyzes each new attribute value that it senses. After making sure that this value is an accurate one and is not an outlier, it broadcasts it to its neighbors.

Moreover, in SENOCLU algorithm subspace clustering is applied, while Kotidis's approach applies full space clustering. Since clustering is based on similarity of all attributes in full space clustering, there is high probability that due to a state change in any of the attributes, a node is not anymore similar to other nodes of current cluster. In case of subspace clustering the number of attributes is smaller, so this probability is lower. Therefore, subspace clustering provides a more stable cluster structure than full space clustering.

According to all this comparative analysis between two algorithms, we can conclude that the process of selecting representatives is more energy efficient and provides a more stable cluster structure in SENOCLU algorithm compared to Kotidis's approach. Nevertheless, since these two algorithms share similarities in principle of clustering, Kotidis's algorithm is used in our experiments.

ECLUN [13] is a Self-Organizing, Energy Aware algorithm for Clustering Nodes in sensor networks. This algorithm has inherited most of the crucial features of SENOCLU algorithm such as: reducing of communication burden by allowing communication only when state change is detected; applying subspace clustering in order to assure stability of cluster structure and more energy efficiency in clustering process; uniform utilization of energy resources in sensor network by applying delegation of local representative authority and load balancing among representatives etc. Nevertheless, SENOCLU algorithm is the output of a very extensive research work, while ECLUN is a partial extract of this work. SENOCLU algorithm is compared in details with many other related works and its relevant advantages and contribution are identified. Compared to ECLUN algorithm, in SENOCLU algorithm it is applied a more consolidated and efficient outlier detection algorithm. By detecting and removing outliers, not only the accuracy of transmitted data is increased, but also the energy that would be spent for sending this unnecessary data to representative is saved. Moreover, reporting of outliers to representative impacts negatively the stability of cluster structure. A reported outlier would show to representative a fake state change in the measurements of the respective sensor node. This state change would be noticed only in this node. Apparently this node would not appear to be anymore similar to other cluster nodes. Therefore, representative would ask this node to re-cluster. In this way, this node unfairly would spend some more energy in additional communication and computation to find a suitable new cluster for itself. Outlier detection prevents all these scenarios and improves the energy efficiency as well as accuracy of data in sensor networks.

In addition, in our work for SENOCLU algorithm a large set of experiments have been performed. In these experiments, the features of algorithm are switched on and off and each feature's specific impact on algorithm efficiency and accuracy is evaluated. In these experiments it is analyzed also the scalability of SENOCLU algorithm against different portions of number of outliers in dataset. As a conclusion, due to outlier detection feature, SENOCLU algorithm is scalable to number of outliers. It follows the same flow of performance even in cases when number of outliers is increased.

D. Techniques of Selecting Representatives

Selecting representatives is an important process of SENOCLU algorithm. This process is repetitive during all the runtime of algorithm, as soon as reclustering process takes place. Therefore, higher quality of this process, better performance of algorithm can be provided.

SERENE [14] is a framework for SElecting REpresentatives in a sensor NEtwork. It uses clustering techniques to select the subset of nodes that can best represent the rest of sensors in the network. In order to reduce communication, rather than directly querying all network nodes, only the representative sensors are queried. To select an appropriate set of representative sensors, SERENE performs the analysis of historical readings of sensor nodes, in order to find out the correlations both in space and time dimensions among sensors and sensor readings. Sensors may be physically correlated. Sensor readings may be correlated in time. Physically correlated sensors with correlated readings are assigned to the same cluster. Then each cluster performs further analysis in order to select the sensors with the highest representation quality.

Similar to SENOCLU algorithm, this technique uses density-based clustering algorithm, DBSCAN [6]. Nevertheless, different from the proposed algorithm, in SERENE approach the first stage of clustering process is analysis of historical data for detecting correlations among nodes and sensor readings. Due to restrictions of energy, computational and memory capacity in sensor nodes, this analysis can not be performed by the nodes themselves. Continuous storing of historical data for all nodes that are spatially correlated, in order to analyze correlation of their readings, requires more memory capacity than a sensor node possesses. Processing of all the analyses over measurements of sensors to find out correlations needs high computation resources as well. This is followed by high energy consumption in nodes, due to frequent communication and data exchange with more than one node in their clusters. Due to all these restrictions, in this approach sensor nodes cannot be self-organized into clusters. As a result, this technique is suitable only for those scenarios where nodes operate in supervised way.

Another difficult part of this technique is related with maintenance of SERENE platform. With passing of time, the readings of sensor nodes change, consequently the same set of sensors may not be any more correlated with each other, or a new correlation may immense among some other nodes. This change requires a reorganization of nodes in clusters. Reclustering process is followed by additional communication among nodes for updating historical data. This will increase the communication burden and the size of transmitted data will be significantly high. All the above mentioned reasons make this approach expensive in terms of energy and not easy to maintain in cases of continuous clustering applications.

IV. CONTRIBUTION OF SENOCLU ALGORITHM

By applying clustering technique, SENOCLU algorithm reduces ***communication burden*** in sensor network. Due to important feature of ***state change detection***, the continuous communication of nodes is avoided and the communication frequency is reduced significantly without having impact in the accuracy of data. In this way, energy consumption in sensor network is reduced. ***Self-organization*** of nodes into clusters makes this algorithm useful for most of sensor network applications where nodes operate in unsupervised way. ***Delegation of representative authority*** in case of energy shortage and ***load balancing among the representatives*** in the cluster, are two important features of this algorithm that contribute in a uniform utilization of energy resources in sensor network. This provides continuity in monitoring of phenomena from all the locations where the nodes are deployed. Managing of multi-dimensional data in sensor network is performed by ***subspace clustering*** technique. This feature of algorithm contributes in detecting clusters in the network and providing stable cluster structure. Another feature that helps in stability of cluster structure and accuracy of data as well is outlier detection.

The presence of these efficient techniques as important components of SENOCLU algorithm is the best proof for efficiency of this algorithm in data acquisition in sensor networks. Figure 2 illustrates the contribution of SENOCLU algorithm.

V. EXPERIMENTAL RESULTS

The experiments that we perform for evaluating SENOCLU algorithm focus on two main directions:

- Testing and evaluating the impact of each feature of algorithm in the overall performance of it
- Analyzing the efficiency of this algorithm with respect to competitive algorithm, Snapshot Queries [12].

Two datasets were used for the experimental evaluation: a processed and partial version of Dataset of Intel Berkeley Research Lab [15] and generated synthetic data set.

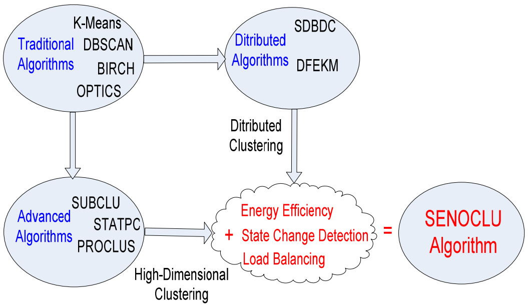

Fig. 2 Contribution of SENOCLU Algorithm

A. Data Sets

Real data set

In these experiments we have not used the full version of dataset of Intel Berkeley Research Lab. We have processed it by removing the readings with missing values and the nodes that contain a small number of readings. After processing, the dataset that is used for our experiments contains 51 nodes. The coordinates of nodes are preserved the same. The initial energy of each sensor node is 295 J. The dataset that we are using contains 15730 readings for each node. Each reading contains measurement for 4 attributes respectively: temperature, humidity, light and voltage. Readings are stored every 31 sec. This means that these attribute values are collected by sensor nodes for 5 days, 15 hours, 27 min and 10 sec period of time.

Synthetic data set

The synthetic dataset that we have generated for our experiments consists of 49 sensor nodes that are uniformly distributed in one grid with 7x7 dimensions. One node is located in each vertex of 1x1 dimensions square unit of the grid. Each sensor node is characterized by its node id, its spatial attributes consisting of its coordinates in the grid and its non-spatial attributes consisting of six other attributes. Dataset consists of 15000 readings for each node, where each reading contains values for 6 attributes. It is assumed that each reading is stored every 31 sec. The initial energy for each sensor node is 280 J.

B. Experiments for the Features of SENOCLU

This set of experiments aims to test and evaluate the impact of each feature of SENOLCU algorithm in the overall performance of it. In some of these experiments, a single feature or a combination of certain features of the algorithm are disabled and then the performance of algorithm is compared to the performance of original version of algorithm, where all the features are enabled. In some other experiments, all the important features are disabled and then the performance of the original version of algorithm is compared to the most naive version of it. These experiments are performed either over the processed version of dataset from Intel Berkeley Research Lab, or over the synthetic dataset that we have generated. In both cases the experiments run until the moment when the last readings in the respective dataset are processed by sensor nodes. In case when all the sensor nodes run out of energy before the entire readings end, the experiments stop running as soon as the last node in the network runs out of energy, although there may be still unprocessed readings in the dataset. All these experiments help us to evaluate better SENOCLU algorithm and to understand the specific contribution of each feature of algorithm in the overall performance of it. The results of some of these experiments are displayed in the figures below.

In the experiment displayed in Figure 3, we have disabled the following features: state change detection for decision of sending data from local representative to data sink, state change detection for decision of sending data from node to local representative, outlier detection, subspace clustering, delegation of representative authority in case of energy shortage and load balancing among representatives. The definition of cluster, the set of thresholds and the criteria for reclustering are the same in both cases of experiment.

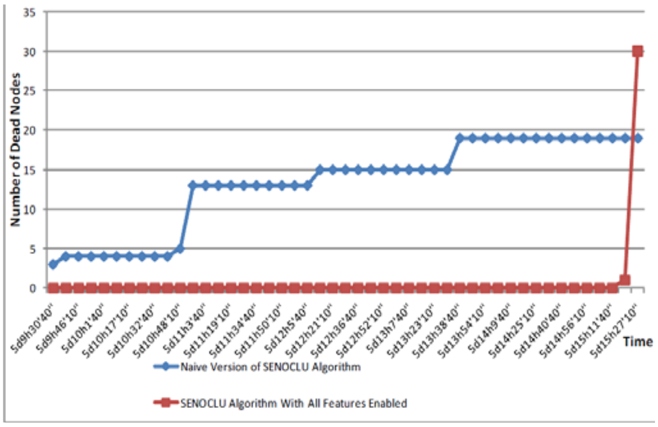

Fig. 3 Performance of SENOCLU algorithm in case when the most important features of it are disabled with respect to the case when all the features are enabled

In the following experiments a single feature or a combination of certain features of algorithm are disabled.

As it is displayed in the figures below, the results of these experiments show that each of the features that SENOCLU algorithm contains has a valuable contribution in overall performance of this algorithm. They contribute in reducing the energy consumption in sensor nodes and provide a longer sensor network lifetime.

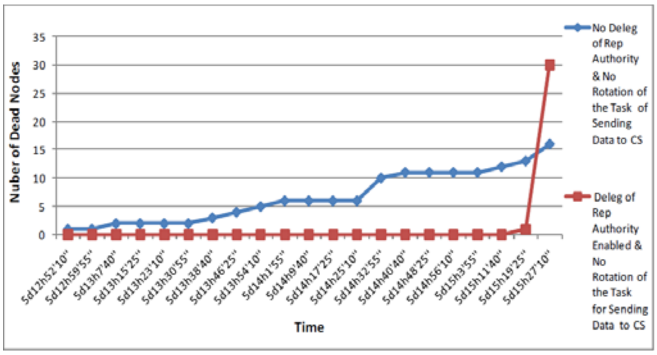

Fig. 4 Performance of SENOCLU algorithm in case when the feature of rotating the task of sending data to central server (CS) is disabled and the feature of delegating representative authority is initially disabled and then enabled

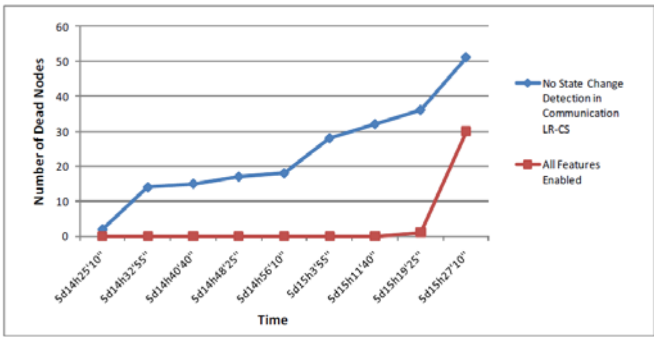

Fig. 5 Performance of SENOCLU algorithm in case when we disable the feature of sending data from local representative (LR) to central server (CS) only when a state change is detected, compared to the case when all the features of algorithm are enabled

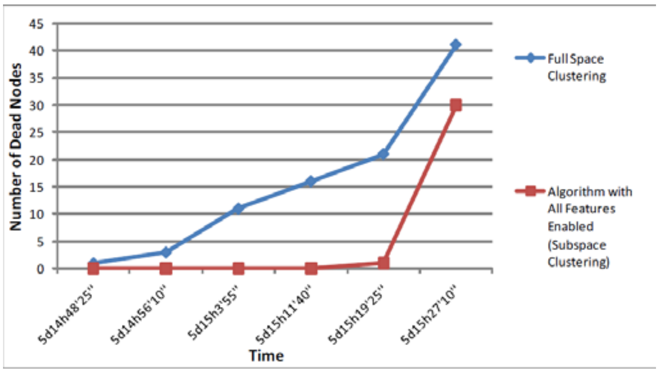

Fig. 6 Performance of SENOCLU algorithm in case when we apply in it subspace clustering compared to the case when we apply full space clustering

B. Comparison with Snapshot Queries

Performance of SENOCLU algorithm is compared experimentally to the performance of Snapshot Queries approach [12]. In the first experiment (Figure 8), we compare the energy cost for the process of selecting representatives in both approaches. In the second experiment (Figure 9), we compare the overall performance of our algorithm to the performance of both versions of Kotidis's approach.

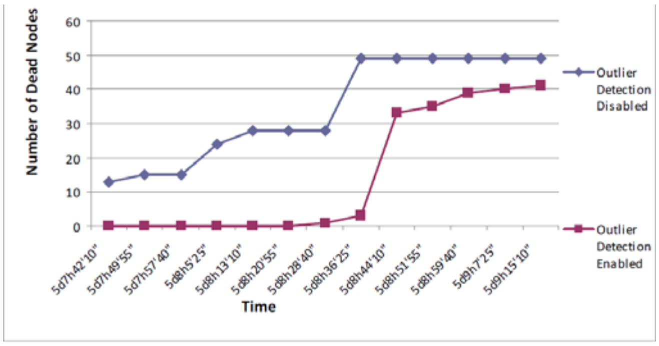

Fig. 7 Performance of SENOCLU algorithm in case when we initially disable and then enable outlier detection feature

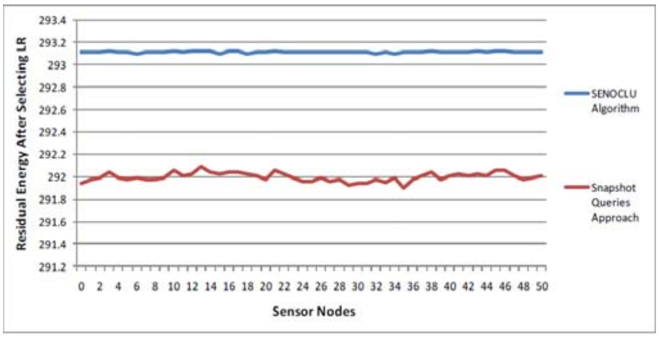

Fig. 8 Energy cost of the process of selecting representatives in SENOCLU algorithm with respect to Snapshot Queries approach

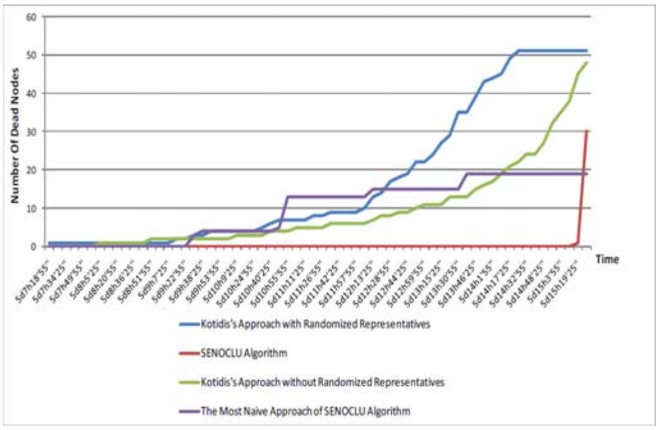

Fig. 9 Performance of SENOCLU algorithm with respect to Kotidis's approach

The results of these experiments show that the process of selecting representatives is more energy efficient and provides a more stable cluster structure in SENOCLU algorithm compared to Kotidis's approach. Moreover, the overall performance of SENOCLU is better than any version of Kotidis's algorithm. In Kotidis's algorithm with randomization in selection of representatives, nodes start to run out of energy almost 8 hours earlier than in SENOCLU algorithm. While in Kotidis's algorithm without randomization in selection of representatives, nodes start to run out of energy 7 hours and 15 min earlier than in SENOCLU algorithm. In both cases, the number of dead nodes at the end of experiment is greater than in SENOCLU algorithm. This better performance of SENOCLU algorithm is due to important set of features that it includes.

V. CONCLUSIONS AND FUTURE WORK

In this work we propose and evaluate SENOCLU algorithm, an energy efficient approach for unsupervised node clustering in sensor networks. This algorithm takes into consideration the distributed nature of data in sensor networks and the high dimensionality of it. By adding specific requirements of sensor network applications like energy efficiency, state change detection, load balancing and outlier detection , this algorithm contributes in filling the gap between distributed clustering and high-dimensional clustering algorithms that are applied in sensor networks. The experimental results show that due to the set of features that SENOCULU algorithm contains, it is highly efficient in unsupervised node clustering. It reduces energy consumption in sensor nodes and provides an accurate way for collecting data from sensor networks.

In the future, our algorithm might be extended to deal with missing values in the measurements of sensor nodes. It might be improved to be applicable in the applications of mobile sensor nodes as well. It might also treat the case when the nodes fail due to some hardware or environmental problems rather than shortage of energy that we have included in our algorithm. Additional techniques may be applied to improve the energy efficiency in data acquisition.

VI. REFERENCES

[1] V. Shnayder, M. Hempstead, B. Chen, G. Allen, and M. Welsh, "Simulating the Power Consumption of LargeScale Sensor Network Applications", *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, SenSys 2004, Baltimore, MD, USA.

[2] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling Ultra-Low Power Wireless Research", *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks*, IPSN 2005, Los Angeles, California, USA.

[3] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations", *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, 1: 281-297, 1967.

[4] L. Kaufman and P.J.Rousseeuw," Clustering by Means of Medoids in Statistical Data Analysis Based on the L1 Norm", Y. Dodge, Ed., pp. 405-416. North Holland/Elsevier, Amsterdam, 1987.

[5] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases", *SIGMOD '96* 6/96 Montreal Canada.

[6] M. Ester, H. Kriegel, J. Sander, X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Data Bases with Noise", *2-nd International Conference on Knowledge Discovery and Data Mining, KDD-96*.

[7] A. Meka and A. K. Singh, "Distributed Special Clustering in Sensor Networks", *EDBT 2006*, LNCS 3896, pp 980-1000, 2006

[8] P. P. Rodrigues, J. Gama and L. Lopes, "Clustering Distributed

Sensor Data Streams", *ECML PKDD 2008*, LNAI. Springer-Verlag 2008.

[9] J.Yin, M. M. Gaber, "Clustering Distributed Time Series in Sensor Networks", in Proceedings of the Eighth IEEE Conference on Data Mining, ICDM, 2008.

[10] E. Januzaj, H. Kriegel, M. Pfeifle, "Scalable Density-Based Distributed Clustering", *Proc. 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Pisa, Italy, 2004.

[11] A. Silberstein, R. Braynard, G. Filpus, G. Puggioni, A. Gelfand, K. Munagala, and J. Yang, "Data-Driven Processing in Sensor Networks", *3rd Biennal Conference on Innovative Data Systems Research (CIDR)*, Asilomar, California, USA, 2007.

[12] Y. Kotidis, "Snapshot Queries: Towards Data-Centric Sensor Networks", *Proceeding of the 21st International Conference on Data Engineering,* ICDE 2005.

[13] M. Hassani, E. Müller, P. Spaus, A. Faqolli, Th. Palpanas, Th. Seidl, "Self-Organizing Energy Aware Clustering of Nodes in Sensor Networks Using Relevant Attributes", *SensorKDD'10*, July 25, 2010, Washington, DC, USA.

[14] E. Baralis, T. Cerquitelli, "Selecting Representatives in a Sensor Network", *SEBD 2006*: 351-360.

[15] Dataset of Intel Berkeley Research Lab, "http://db.csail.mit.edu/ labdata/labdata.html", in [online], 2004