Research Progress Report

Botao Zhu

July 10, 2019

1 Reading and Research Activities

Considering energy efficient and density clustering protocol based on kernel soft k-Means for wireless sensor network

1.1 Problem statement

At present, most of the research on wireless sensor networks are based on the assumption that the nodes are evenly distributed. Different application scenarios may result in non-uniform distribution of nodes. [1] [2] proposed W-LEACH algorithm to handle non-uniform network, which chooses a number of sensors in each cluster to send data to their CHs based on a fixed percentage. The target is to reduce the number of sending sensors for extending the lifetime of network. However, it may result in the loss of important information. Hence, it is significant to study non-uniform distributed wireless networks. The research can be divided into two aspects: clustering algorithm and cluster head selection.

K-means clustering algorithm is used by [3] firstly in LEACH of wireless sensor networks. However, there are some drawbacks. Firstly, K-Means algorithm chooses the initial centroid randomly, which cannot ensure that clustering is optimal. If the initial centroid is far away from the cluster's true centroid, it will cause the result that the number of iterations required to optimize the centroid takes longer and an incorrect clustering may be obtained.

Secondly, the node with the highest residual energy in the cluster is elected as the cluster head(CH). It can lead to a potential problem:unbalanced energy load and excessive energy expenditure. For example, node A in Fig.1 has higher residual energy than other nodes in the cluster, which means node A can be elected as the CH with the highest probability. However, this way enables the other nodes in the same cluster to send data in the opposite direction to the base station, causing higher energy consumption.

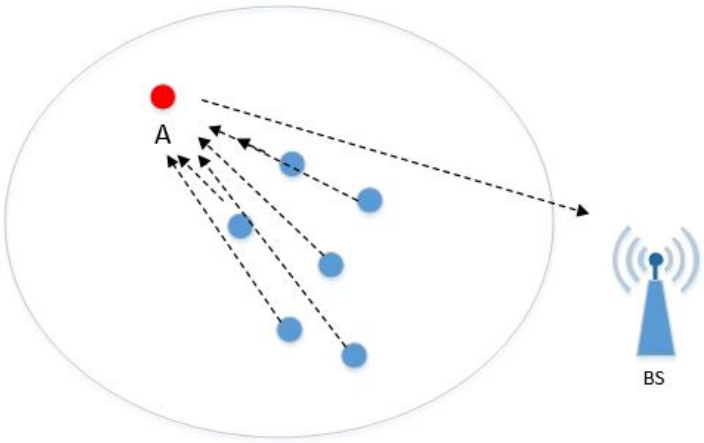


Figure 1: A example of CH selection

Thirdly, it does not consider that some of sensor nodes in a cluster can be reallocated to another cluster. As shown in Fig.2, if we use hard classification algorithm, K-Means, node n belongs to cluster B. However, if we consider this question from the perspective of probability, node n joins the cluster B with a certain probability, 55%. Then, it will have a 45% probability to join the cluster A. Because the density of cluster B is greater than that of cluster A, node n can be allocated to cluster A in order to reduce the stress of CH of cluster B.

Fourthly, because each cluster only selects one cluster head, cluster head's energy will be depleted quickly for aggregating, compressing and transmitting information when there are too many sensor nodes in the cluster. [4] proposed a method to prolong the lifetime of each round by balancing the energy consumption of the nodes. The nodes are divided into three types in each cluster: CH, vice CH and member nodes. The vice CH plays the role of CH when the CH dies before the completion of current round, which can diminish the frequency of re-clustering and extend network lifetime. [5] also comes up with a similar concept. After stable clusters have been achieved, CH node assigns its two nearest nodes as CHs, which can ensure load balancing. However, neither of them considers the problem of cluster size, that is non-uniform distributed wireless networks. In Fig.2, when CHs of cluster B runs out of energy, CHs of cluster A still have a lot of energy because it contains fewer member nodes, which can result in unbalanced load.

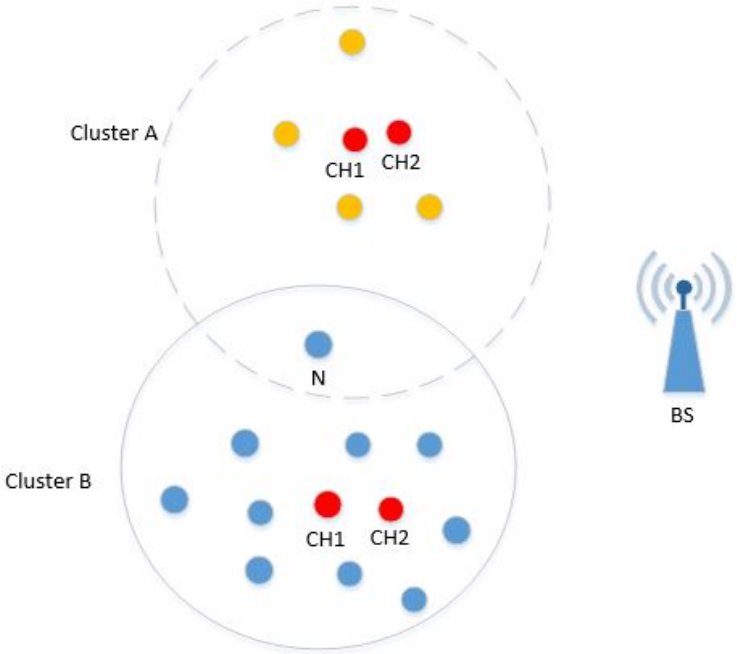


Figure 2: A example of non-uniform wireless network

As mentioned above, our goal is to improve these four problems. 1) optimizing the selection of the initial centroid of clustering algorithm by using kernel density estimation. 2) adjusting cluster size dynamically to balance node load by soft K-Means. 3) determining the location of CHs according to the energy density by kernel density estimation. 4) selecting the number of CH according to the size of clusters.

1.2 Soft K-Means

[6] K-Means is the simplest clustering algorithm in unsupervised learning, which partitions the data set in to k clusters using some distance measurement methods, like Euclidean distance. It is a hard clustering method, that is to say the membership degree of one node has only two values 0 and 1 for a specific class. However, in some cases, there are a few data points for which it is not quite so obvious to which cluster they belong. Soft K-Means clustering decides to which degree each data point belongs to , the assignments to clusters will be probabilistic. Generally speaking, soft K-Means clustering can be seen as the problem of finding k cluster centroids with the aim of minimizing the error function. Given a set of data points $X = \{x_1, x_2, ..., x_n\}$, the error function is

$$E(\mu_1, \mu_2, ..., \mu_k) = \sum_{i=1}^{k} E(\mu_i) = \sum_{i=1}^{k} \sum_{j=1}^{n} z_{ij} ||x_j - \mu_i||^2 \tag{1}$$

where μ_i is the centroid of each cluster, z_{ij} is the indicator variable.
For the traditional K-Means clustering

$$z_{ij} = \begin{cases} 1, \text{if } x_j \in C_i \\ 0, \text{otherwise} \end{cases} \tag{2}$$

where C_i represents cluster i, this equation means whether x_i belongs to cluster C_i. However, z_{ij} is not a integer value for soft K-Means clustering

$$z_{ij} = \frac{e^{-\beta||x_j - \mu_i||^2}}{\sum_{l=1}^{k} e^{-\beta||x_j - \mu_l||^2}} \tag{3}$$

where β is the stiffness parameter and greater than 0. From equation (3), we can get $z_{ij} \in [0, 1]$ and $\sum_i z_{ij} = 1$.
How to update the centroid μ_i of each cluster until convergence? For a specific class C_i

$$E(\mu_i) = \sum_{j=1}^{n} z_{ij} ||x_j - \mu_i||^2 \tag{4}$$

the target is to minimize the error function, so the problem can be turned into the following optimization problem

$$e = \text{argmin} \sum_{j=1}^{n} z_{ij} ||x_j - \mu_i||^2 \tag{5}$$

which is a convex function and has a unique minimum value. The equation (4) can be written as

$$E(\mu_i) = \sum_{j=1}^{n} z_{ij} (x_j - \mu_i) \top (x_j - \mu_i) \tag{6}$$

$$= \sum_{j=1}^{n} z_{ij} (x_j \top x_j - 2 x_j \top \mu_i + \mu_i \top \mu_i) \tag{7}$$

deriving $E(\mu_i)$ with respect to μ_i and denoting to zero, we can have

$$\mu_i = \frac{\sum_{j=1}^{n} z_{ij} x_j}{\sum_{j=1}^{n} z_{ij}} \tag{8}$$

Each cluster updates the centroid according to equation (3) and (8) until the probabilities of the clusters in which the data points are located remain unchanged or the maximum number of iterations are reached.

1.3 Kernel density estimation

The first law of geography states that all things are interrelated. The closer they are, the stronger they are. Kernel density estimation is based on this law, its value decreases gradually with the increase of the central radiation distance. From two-dimensional or three-dimensional surface of density, we can intuitively acquire the features of point sets.

[7] [8]Kernel density is one of the non-parametric estimation methods used to estimate the unknown density function. Each point is covered with a smooth surface. The surface value at the location of the point is the highest and decreases with the increase of distance until the value is zero when the distance equals the search radius, which looks like the density map of Gauss distribution function. Then, the values of kernel density estimation of all points are superimposed and a set of points is transformed into a surface showing continuous density changes.
Supposing $X = \{x_1, x_2, \ldots, x_n\}$ is independent distributed random variable, density function is $f(x)$, $x \in R$

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{9}$$

where $\hat{f}(x)$ is the kernel density estimation of $f(x)$, K is the kernel function, h is the bandwidth, n is the number of points in the bandwidth range.

Kernel function is usually required to satisfy the following conditions:

$$K(-\mu) = K(\mu) \tag{10}$$

$$\mathrm{Sup}|K(\mu)| < \infty \tag{11}$$

$\hat{f}(x)$ is related not only to the data, but also to the kernel function and the window width parameter h. Here are the usual kernel functions:
(1) Gaussian kernel

$$K(\mu) = \frac{1}{\sqrt{2\pi}} e^{\frac{\mu^2}{2}} \tag{12}$$

(2) Boxcar kernel

$$K(\mu) = \begin{cases} 1, |\mu| \leq 0.5 \\ 0, |\mu| > 0.5 \end{cases} \tag{13}$$

(3) Epanechnikow kernel

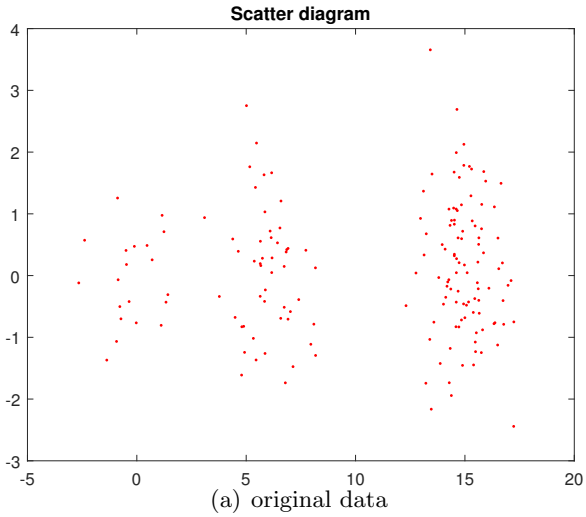
$$K(\mu) = \begin{cases} \frac{3(1-u^2)}{4}, |\mu| \leq 1 \\ 0, |\mu| > 1 \end{cases} \tag{14}$$

We generate three sets of random numbers by Gauss function and implement kernel density estimation for them. Fig.3(b) and Fig.3(c) show the density distribution graph of two-dimensional and three-dimensional respectively. The discrete point set is transformed into a smooth density map, which shows its spatial distribution. The higher the density value, the higher the aggregation degree of the point is.

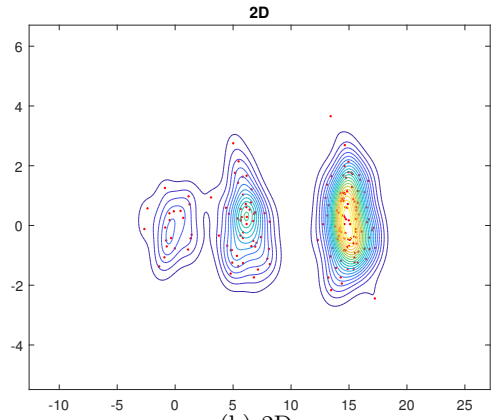
1.4 Network Model and Energy Model

1.4.1 Network Model

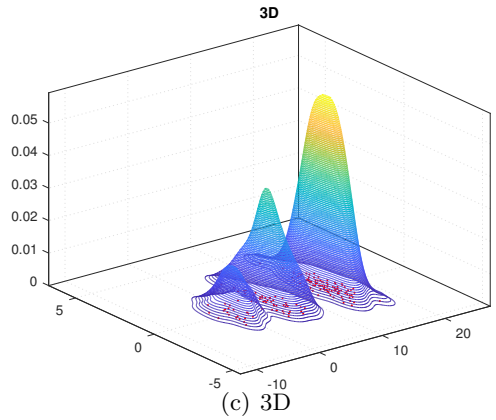
We have following assumptions regarding the network model [9] [10]:
(a) The BS is a high energy node and is located far away from the extremities of the sensor network.
(b) All sensor nodes having the same computational and transmission capabilities. In other words, all nodes are capable of acting as CH nodes.
(c) The sensor nodes can vary the power with which they transmit signals according to the received signal strength indication of a particular node.
(d) The sensor nodes scans its environment at a fixed rate and will contain data to be sent to the



(a) original data



(b) 2D



(c) 3D

Figure 3: A example of kernel density estimation

BS at all intervals.
(e) The sensor nodes and BS are all static after deployment.
(f) The sensor nodes in general have location information such as a GPS support.
(g) The CHs perform data compression to reduce the amount of bits transmitted to the BS.
(h) The BS indicates all nodes to reinitiate clustering when all the CH nodes in the network have insufficient energy.

1.4.2 Energy model

Because the main energy consumption of the protocol is only for receiving and sending data, the first order radio model as the energy model is in line with the needs [11]. As shown in Fig. 4, the energy consumption of nodes comes from the sum of the energy consumption of signal transmitting, signal amplifying and receiving. The longer transmission distance each node has, the more the signal

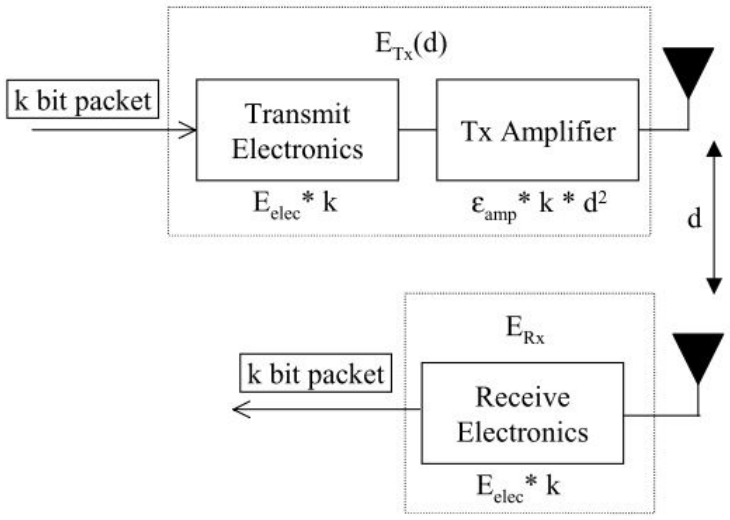


Figure 4: First order ratio model

intensity will lose. Assuming the transmission distance is d, whenever the data of K bit is transmitted or received, the energy consumption formula of the transmitting is as follows

$$E_{send} = E_{TX\_elec}(k) + E_{TX\_amp}(k,d) = k \times E_{static} + k \times \sigma \times d^{\beta} \tag{15}$$

the energy consumption formula of the receiving

$$E_{receive} = E_{RX\_elec}(k) = k \times E_{static} \tag{16}$$

E_{static} is the energy consumption of receiving and transmitting. Amplification multiples of signal amplifier is expressed as σ, β is the route loss index. We need to choose the corresponding energy consumption model according to the length of transmission path for calculating the energy of data transmission. In the first-order radio energy consumption model, there are two kinds of distance d: self-space model and multi-path fading model. Free space model will be used at close range, and multi-path fading model will be used at long range.

$$\begin{cases} \sigma_{fs} \times k \times d^2 & d \le d_0 \\ \sigma_{amp} \times k \times d^4 & d > d_0 \end{cases} \tag{17}$$

σ_{fs} and σ_{amp} represent the energy consumption parameters of amplifier of free space model and multi-path fading model respectively.

1.5 Proposed Algorithm

The protocol is divided into many rounds. Each round contains setup phase and steady phase. In this research, we only focus on the setup phase and improve some existing methods. During the setup

phase, the BS station gathers the location and residual energy of each node in the whole wireless network. Also, it generates k initial centers by kernel density estimation as the input of clustering algorithm, soft K-Means, which can avoid to get the results of local optimum by traditional K-Means clustering method. And then, classification algorithm is implemented. Each node calculates the distance between itself and center points and chooses to join the nearest cluster with a certain probability. At the same time, if the nodes on the edge of two clusters have similar distance to these two cluster centers, they will preferentially join the cluster with smaller density. After clustering, the final cluster centers are selected according to the energy density by kernel density estimation. The detailed discussion is given below.

1.5.1 Selection of Initial Cluster Heads

Kernel density estimation is used to assign the initial center of the soft K-Means. Since the soft K-Means randomly choose the initial centroid, it is not guaranteed that clustering by the soft K-Means is optimal. The algorithm is described below.
Step 1: Kernel density estimation is applied to sample data to obtain a image with continuous density.
Step 2: The image is divided into low density area and high density area.
Step 3: Focus statistical tool is used to obtain max density point sets.

Algorithm 1: Selection of initial cluster heads

Input: A set of n data items $X = \{x_1, x_2, \ldots, x_n\}$
Output: Max density point sets
1 Kernel density estimation for X;
2 Selecting high density and low density areas;
3 Obtaining max density point sets as the input of clustering algorithm

1.5.2 Cluster Formation

Compared with the K-Means algorithm, our proposed algorithm will take one more step after the classification convergence, assigning nodes at the edge of the clusters to join the approximate cluster. We discuss this problem in two scenarios.
Scenario 1: The node N is at the edge of two clusters, shown in Fig. 5. The distance from node N to the center of cluster B is a little bigger than that from node N to the center of cluster A, which means it has a higher probability to join cluster A. However, if we consider the density of cluster, it is a different story. Cluster A has 5 member nodes and cluster B has 10 member nodes. Assuming that the time interval for each node sending messages to CH are consistent, CH of cluster B will deal with more information from member nodes. So, the better way is that the node N chooses to join cluster A, which can reduce the energy consumption of CH of cluster B and balance energy load. We define that when the difference between the probability of node N joining two different clusters is less than 10%, node N will join the cluster with low density. Otherwise, it will join the cluster with high probability.
Scenario 2: The node N is at the edge of three or more clusters. 6. We can get a set in descending order, $P = \{P_1, P_2, \ldots P_k\}$, represents the probability of node N belonging to different clusters. Clusters with low probability will not be considered. Hence, we take a subset of P, $P_{sub} = \{P_1, P_2, \ldots, P_{k/2}\}$, composed by numbers greater than median of P.

$$D_{1i} = P_1 - P_i, i = 2, \ldots, k/2 \tag{18}$$

where D_{1i} is the difference between P_1, the highest probability, and other probability.
If $D_{1i} < 10\%$, node N will join the cluster with smaller density between 1 and i. Otherwise, it will

join cluster 1. As shown in Fig. 6, node N has similar probability to different centers, which finally joins cluster C because cluster C has lower density than cluster A and B.

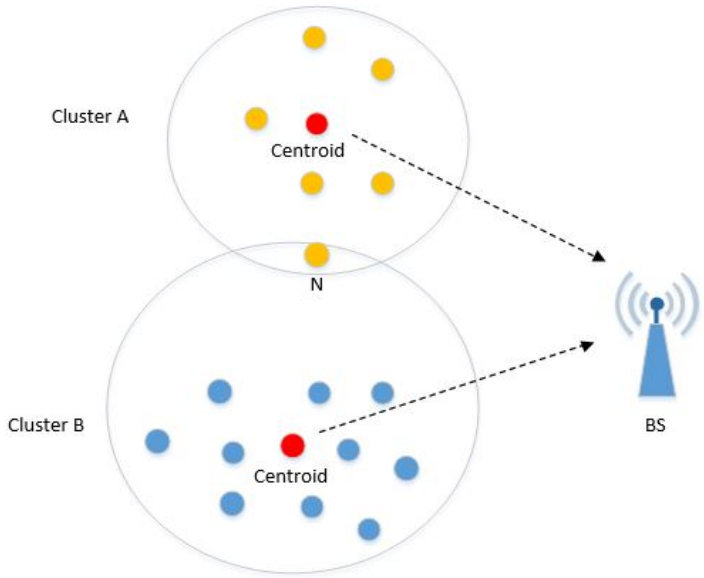


Figure 5: Example of two clusters

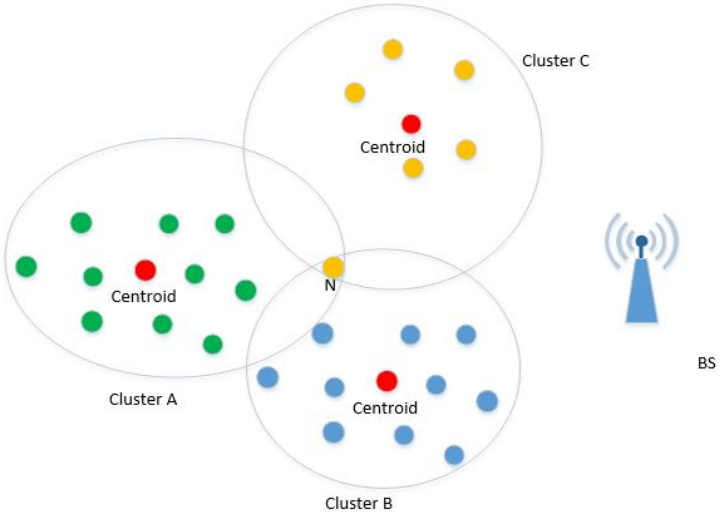


Figure 6: Example of three clusters

Algorithm 2: Cluster formation

Input: Initial centers from **Algorithm** 1, the number of clusters, set of n data items $\{x_1, x_2, \ldots, x_n\}$

Output: A set of k clusters

1 **repeat**
2 Assigning each point x_i to cluster centroids with certain probabilities according to the distance;
3 Re-calculating the cluster centroids by equation 8;
4 **until** *Cluster centroids are not changed any more*;
5 Assigning nodes at the edge of cluster to appropriate cluster according to **Scenario 1** and **2**;
6 Clustering end

1.5.3 Selection of Final Cluster Heads

The most common way to select cluster heads is to choose the nodes with the largest residual energy, which may cause excessive energy consumption. In this step, we still use kernel density estimation to select cluster heads and add a parameter σ in $\hat{f}(x)$, which represents the residual energy of node. Equation (19) can ensure that CHs are selected in the area with the highest energy density.

$$\hat{f}(x) = \frac{\sigma}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{19}$$

At the same time, the number of cluster head nodes is not fixed in each cluster, which is determined by the nodes in the cluster. Here we give a simple rule. We count the nodes of each cluster and sort in descending order of size. The first 30% clusters takes three head nodes, the middle 30% clusters takes two head nodes and the last 30% clusters takes one head node. This method can avoid the communication overhead caused by frequent clustering and take into account the different size of clusters.

Algorithm 3: Selection of final cluster heads

Input: A set of k clusters

Output: A set of k Cluster heads

1 Calculating the size of all k clusters and sort them in descending order;
2 Ensuring the number of cluster head of each cluster, $H = \{H_1, H_2, \ldots H_k\}$;
3 Calculating the average energy of each cluster;
4 Selecting nodes whose energy is greater than the average energy to get a new set X_{new};
5 Applying kernel density estimation to X_{new} to obtain the max energy density areas of each cluster;
6 Selecting final cluster heads according to H from the max density area nearest to the center of each cluster

After CHs for the current round are selected, the BS notifies all nodes to join the cluster to which they belong. CHs then broadcast TDMA schedules for the member nodes to transmit data in different time slots to avoid data collision.

2 Objectives for the Next 2 Weeks

Simulate the algorithm proposed above.

3 Advisor's Comments

Bibliography

[1] H. M. Abdulsalam and L. K. Kamel, "W-leach: Weighted low energy adaptive clustering hierarchy aggregation algorithm for data streams in wireless sensor networks," Dec 2010, pp. 1–8.

[2] H. M. Abdulsalam and B. A. Ali, "W-leach based dynamic adaptive data aggregation algorithm for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 9, no. 9, p. 289527, 2013.

[3] P. Sasikumar and S. Khara, "2012 fourth international conference on computational intelligence and communication networks," Nov 2012, pp. 140–144.

[4] A. S. D. Sasikala and N. Sangameswaran, "Improving the energy efficiency of leach protocol using vch in wireless sensor network," *International Journal of Engineering Development and Research*, vol. 3, no. 2, pp. 918–924, 2015.

[5] S. Periyasamy, S. Khara, and S. Thangavelu, "Balanced cluster head selection based on modified k-means in a distributed wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 12, no. 3, p. 5040475, 2016.

[6] C. Bauckhage, "Lecture notes on data science: Soft k-means clustering," 2015.

[7] C. B, "Lecture notes on machine learning: Kernel k-means clustering," 2019.

[8] T. Buch-larsen, J. P. Nielsen, M. Guillén, and C. Bolancé, "Kernel density estimation for heavy-tailed distributions using the champernowne transformation," *Statistics*, vol. 39, no. 6, pp. 503–516, 2005.

[9] N. Srikanth and M. Ganga Prasad, "Efficient clustering protocol using fuzzy k-means and mid-point algorithm for lifetime improvement in wsns," *International Journal of Intelligent Engineering and Systems*, vol. 11, pp. 61–71, 08 2018.

[10] A. Ray and D. De, "Energy efficient clustering protocol based on k-means (eecpk-means)-midpoint algorithm for enhanced network lifetime in wireless sensor network," *IET Wireless Sensor Systems*, vol. 6, no. 6, pp. 181–191, 2016.

[11] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, Oct 2002.