Improved Soft-k-Means Clustering Algorithm for Balancing Energy Consumption in Wireless Sensor Networks

Botao Zhu, Ebrahim Bedeer, *Member, IEEE,* Ha H. Nguyen, *Senior Member, IEEE,* Robert Barton, and Jerome Henry

***Abstract*—Energy load balancing is an essential issue in designing wireless sensor networks (WSNs). Clustering techniques are utilized as energy-efficient methods to balance the network energy and prolong its lifetime. In this paper, we propose an improved soft-k-means (IS-k-means) clustering algorithm to balance the energy consumption of nodes in WSNs. First, we use the idea of "clustering by fast search and find of density peaks" (CFSFDP) and kernel density estimation (KDE) to improve the selection of the initial cluster centers of the soft k-means clustering algorithm. Then, we utilize the flexibility of the soft-k-means and reassign member nodes considering their membership probabilities at the boundary of clusters to balance the number of nodes per cluster. Furthermore, the concept of multi-cluster heads is employed to balance the energy consumption within clusters. Extensive simulation results under different network scenarios demonstrate that for small-scale WSNs with single-hop transmission, the proposed algorithm can postpone the first node death, the half of nodes death, and the last node death on average when compared to various clustering algorithms from the literature.**

***Index Terms*—Clustering by fast search and find of density peaks (CFSFDP), energy load balancing, kernel density estimation (KDE), multi-cluster heads, soft k-means, wireless sensor networks (WSNs).**

I. Introduction

THE general concept of Internet-of-Things (IoT) is to facilitate the network connection of billions of devices to collect and exchange information to provide various services [1], [2]. Wireless sensor networks (WSNs) are among important parts of an IoT system because they can be used to gather and send data [3]. WSNs are like the eyes and ears of the IoT and they build the bridge between the real and the digital worlds. WSNs typically consist of a large number of low-cost sensor nodes with restricted battery supplies. Sensor nodes are deployed in various application scenarios to monitor and collect physical conditions of the surrounding environment such as temperature, humidity, pressure, position, vibration, and sound, to name a few [4]. The collected data is then sent to the base station (BS) for further analysis and processing.

B. Zhu, E. Bedeer, and H. H. Nguyen are with the Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, Canada S7N 5A9. Emails: {botao.zhu, e.bedeer, ha.nguyen}@usask.ca

R. Barton and J. Henry are with Cisco Systems Inc. Emails: {robbarto, jerhenry}@cisco.com.

This work was supported by NSERC/Cisco Industrial Research Chair in Low-Power Wireless Access for Sensor Networks.

Reducing energy consumption is a key challenge in WSNs, as sensor nodes can be placed in hard-to-reach areas and/or their batteries may not be rechargeable [5]. Clustering in energy-limited WSNs has been widely investigated to reduce the energy consumption [6]. Clustering-based algorithms group sensor nodes into distinct clusters, where each sensor node belongs to one cluster only. All member nodes sense their surrounding environment and send the results to the cluster heads (CHs). Then, CHs collect and process the data and send information to the BS [7]. Each node consumes a certain amount of energy when it collects, processes, and sends data, and a node is defined to be dead when it runs out of energy [8]. Hence, it is crucial to develop efficient clustering algorithms to balance the energy consumption among sensor nodes in WSNs.

Different clustering techniques have been proposed to design energy-efficient WSNs and increase their lifetime. The authors in [9] proposed a CH election method, which rotates the CH positions among the nodes with higher energy in different communication rounds. In particular, the method considers the initial energy, residual energy, and an optimal number of CHs to decide the next group of CHs among the nodes in the network. Then, member nodes join different CHs according to the distances between them and CHs to form clusters. A joint clustering and routing algorithm is proposed in [10] to improve the energy efficiency of large-scale WSNs. This algorithm employs a back-off timer and gradient routing to execute the CH selection and multi-hop routing simultaneously. The authors in [11] presented a node-density-based clustering and mobile elements algorithm (NDCMC) for collecting data in WSNs. In NDCMC, the nodes surrounded by more deployed nodes are selected as CHs in order to improve the efficiency of intra-cluster routing. The authors presented a fixed parameter tractable (FPT) approximation algorithm with an approximation factor of 1.2 based on the parameterized complexity theory in [12] in order to solve load balanced clustering problem (LBCP) in WSNs. The FPT-approximation algorithm determines which gateway each sensor node must be assigned to, which can lead to more balanced load and energy consumption among the gateways. On the other hand, a routing tree for the inter-cluster communication is proposed, which can distribute the overhead of the routing among almost of all of the nodes. In [13], the authors further proposed an FPT-approximation algorithm with an approximation factor of 1.1, which is more precise than previous approximation factors

reported for LBCP. The FPT-approximation algorithm is used to assign sensor nodes to gateways such that the maximum load of the gateways is minimized. Then, an energy-aware routing algorithm is employed to find the optimal routing tree between gateways and the sink with the aim of balancing the energy consumption of the nodes. The same authors also considered another FPT-approximation algorithm with an approximation factor of 1.1 in [14]. In order to make the FPT-approximation algorithm to be practical in large-scale WSNs, a virtual grid infrastructure with several equal-size cells is used where the FPT-approximation algorithm runs in each cell independently. In [15], a distributed multi-objective based clustering algorithm is presented to assign sensor nodes to appropriate CHs. Then, an energy-efficient routing algorithm is proposed to balance the relay load among the CHs. In [16], the authors implemented a distributed clustering algorithm by considering a trade-off between the energy efficiency and coverage requirement. This algorithm can form unequal-size clusters to balance the load of the CHs. The same authors in [17] proposed a distributed fuzzy logic-based unequal clustering approach and routing algorithm (DFCR) to solve the hot spot problem, which is caused by the fact that some CHs deplete their energy much faster as compared to other CHs. The DFCR algorithm designs an unequal clustering mechanism by reducing the cluster size nearest to the BS.

The authors in [18] proposed a modified k-means clustering algorithm that considers two factors, namely, (i) distances among CHs and their member nodes, and (ii) the remaining energy of nodes, to reduce the overall energy consumption and extend the network lifespan. In [19], the authors proposed a hybrid clustering algorithm based on the k-means clustering algorithm and LEACH [20], where balanced clusters are generated by k-means and CHs are selected by LEACH. This hybrid algorithm outperforms LEACH in terms of the energy consumption. However, due to the frequent re-clustering, the energy consumption of the nodes may increase in the phase of cluster formation and CH selection. An energy efficient clustering protocol based on k-means (EECPK-means) is proposed in [21] with the aim of balancing the load of CHs in WSNs. The midpoint method is used to improve the initial selection of centroids in the k-means algorithm in order to generate balanced clusters. In [22], the authors proposed a method based on fuzzy c-means clustering and particle swarm optimization (FCM-PSO) to reduce the total energy consumption of the network and reduce the number of network disconnects. The FCM-PSO algorithm considers the energy consumption and constraints of communication in the calculations of the CHs and nodes' membership probability. The energy-efficient k-means LEACH (KM-LEACH) algorithm is proposed in [23] to create symmetric clusters and reduce the average intra-cluster communication distance, which can save nodes' energy and improve the network lifetime. To address the problem of how to control the failure of a CH in each cluster, the k-medoids clustering algorithm and vice CH scheme (VLEACH) are used together with LEACH in [24]. Vice CH will become a new CH in case the CH of a given cluster dies, which helps to prolong the lifetime of WSNs by balancing the nodes' energy consumption. The authors in [25] used the k-means and Gaussian elimination algorithms to reduce energy consumption of WSNs and extend their lifetime. An innovative classification algorithm based on "clustering by fast search and finding of density peaks" (CFSFDP) [26] algorithm for balancing energy is proposed in [27]. The authors extend the original CFSFDP algorithm to take into account residual energy (in addition to local density and distance) to select CHs, and accordingly cluster nodes based on the selected CHs.

Against the above background, in this paper, an improved soft-k-means (IS-k-means) clustering algorithm is proposed with the aim of balancing the energy consumption of all nodes in WSNs and extending the network lifetime. The proposed IS-k-means can be widely used in industrial control, smart home, smart agriculture, environment perception, health monitoring, etc., because it can extend the life of sensor nodes in these application scenarios. The novelty of the proposed algorithm can be summarized as follows.

1) Compared with existing clustering algorithms that select the initial cluster centers randomly, we choose the initial centroids of the IS-k-means clustering algorithm by using the idea of density from CFSFDP and kernel density estimation (KDE) [28] to achieve a better clustering result. The nodes with high local density and relative large node distances are chosen as the initial centroids.

2) After the proposed algorithm converges, we reassign member nodes that are located at the boundary of two or more clusters to balance the number of nodes per cluster according to the flexibility of the soft-k-means.

3) Since the clustering process needs to be repeated continually, the communication cost during the clustering phase is increased. We use multi-cluster heads (multi-CHs) scheme to balance traffic load of CHs of different clusters and reduce the frequency of clustering.

The rest of this paper is organized as follows. The necessary background for our research is discussed in Section II. Section III describes the proposed IS-k-means algorithm. In Section IV, we compare the performance of the proposed IS-k-means with other algorithms. Finally, Section V concludes the paper.

II. Preliminaries

A. Soft k-Means

The soft k-means [29] is a kind of fuzzy clustering algorithm where clusters are represented by their respective centers. Since traditional k-means clustering techniques are hard clustering algorithms, which may fail to separate overlapping clusters or properly cluster noisy data [30], the soft k-means algorithm can be applied to address these cases. With the soft k-means algorithm, each node may belong to one or more clusters with different degrees of membership [31]. Nodes located at the boundaries of clusters are not forced to fully belong to a given cluster, but rather they can be members of many clusters with membership degrees or probabilities between 0 and 1 [32]. Nodes at the edge of a cluster may have lower membership probabilities than nodes close to the center of a cluster. This flexibility of the soft k-means clustering is in sharp contrast with the k-means clustering, where a node belongs to only a single cluster.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3031272, IEEE Internet of Things Journal

For a set of nodes' locations $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ in WSNs, the goal of the soft k-means is to partition the n nodes into k sets $\boldsymbol{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k\}$ with small intra-cluster distances and large inter-cluster distances. Thus, we define the following cost function:

$$J(\boldsymbol{X}; \boldsymbol{Z}, \boldsymbol{M}) = \sum_{v=1}^{k} \sum_{j=1}^{n} z_{vj} ||\boldsymbol{x}_j - \boldsymbol{\mu}_v||^2, \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{\mu}_v; v = 1, \ldots, k)$ is the matrix of cluster centers, and $\boldsymbol{Z}(z_{vj}; v = 1, \ldots, k; j = 1, \ldots, n)$ is the membership probability matrix of $\boldsymbol{X}$. z_{vj} is the membership value of the jth node to the vth cluster and is defined as [29]

$$z_{vj} = \frac{e^{-\beta||\boldsymbol{x}_j - \boldsymbol{\mu}_v||^2}}{\sum_{l=1}^{k} e^{-\beta||\boldsymbol{x}_j - \boldsymbol{\mu}_l||^2}}, \tag{2}$$

where β is the stiffness parameter that impacts the membership probability of each node. The best clustering solution is obtained by minimizing J, which differs from the conventional k-means since weighted squared errors are used in the cost function instead of squared errors [29]. The result of the soft k-means algorithm will depend on the choice of β. We will discuss the choice of β when presenting simulation results.

In order to minimize the objective function in (1), z_{vj} must satisfy the following three constraints [29].

1) Each node is assigned a membership probability between 0 and 1 for belonging to a cluster:

$$z_{vj} \in [0, 1], \ v = 1, \ldots, k, \ j = 1, \ldots, n. \tag{3}$$

2) The sum of the membership probabilities for one node over all clusters is equal to 1:

$$\sum_{v=1}^{k} z_{vj} = 1, \ j = 1, \ldots, n. \tag{4}$$

3) There will be at least one node with some non-zero membership probability for belonging to each cluster

$$\sum_{j=1}^{n} z_{vj} > 0, \ v = 1, \ldots, k. \tag{5}$$

By minimizing the objective function, we can calculate the cluster centers as [29]

$$\boldsymbol{\mu}_v = \frac{\sum_{j=1}^{n} z_{vj} \boldsymbol{x}_j}{\sum_{j=1}^{n} z_{vj}}. \tag{6}$$

The operations of the soft k-means algorithm can be summarized as follows: the algorithm calculates the membership probabilities and the cluster centers according to (2) and (6) in each round, respectively. If the changes of the membership probabilities $\boldsymbol{Z}$ or the cluster centers $\boldsymbol{M}$ are below given thresholds, the clustering process ends. Otherwise, the algorithm recalculates the new membership probabilities $\boldsymbol{Z}$ and the new cluster centers $\boldsymbol{M}$. If the algorithm does not converge after a given number of iterations, it will reinitiate by choosing new initial cluster centers. Fig. 1 shows an example of the clustering result of 100 nodes by the soft k-means algorithm.

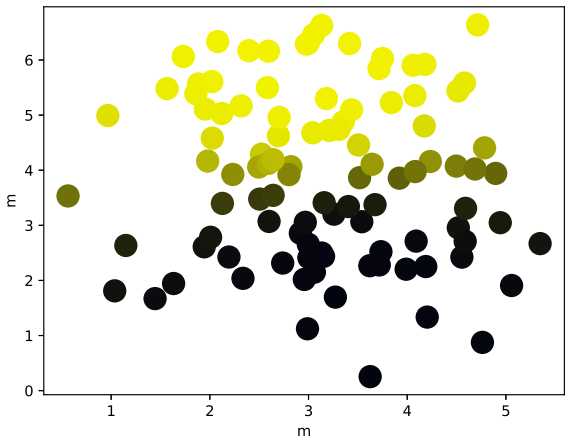


Fig. 1. Example of soft k-means clustering.

B. Kernel Density Estimation

Non-parametric estimators are flexible for modeling probability density function (PDF) of data points. They have no fixed functional form and depend on data points to reach an estimate when compared to parametric estimators [33]. Non-parametric estimators can be classified into histogram-based and kernel-based estimation. A histogram-based estimator needs large data sets to guarantee convergence, and it cannot produce smooth continuous estimation curve [34]. KDE finds the distribution characteristics from data points without attaching any assumptions to data. It can ensure a smooth PDF approximation for given data points [28]. In KDE, the kernel function is centered at each data point, and it has the peak value at the data point location while decreasing in intensity with the distance from this location [28].

Using KDE, the PDF of the nodes' locations $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \in \mathbb{R}^d$ is represented by a weighted sum of the kernel functions [35]

$$\hat{f}_h(\boldsymbol{x}_i) = \frac{1}{nh^d} \sum_{t=1}^{n} \mathcal{K}\left(\frac{\boldsymbol{x}_t - \boldsymbol{x}_i}{h}\right), \tag{7}$$

where h is the smoothing parameter called the bandwidth and it controls the size of the neighborhood around $\boldsymbol{x}_i, i \in 1, \ldots, n$. $\mathcal{K}(\cdot)$ is called the kernel function, which is defined in a d-dimensional space. The kernel function controls the weight given to $\boldsymbol{X}$ at each point $\boldsymbol{x}_i$ based on their proximity. To yield meaningful estimates, a kernel function should satisfy the following conditions [28].

1) Normalization:

$$\int_{\mathbb{R}^d} \mathcal{K}(\boldsymbol{u}) d\boldsymbol{u} = 1. \tag{8}$$

2) Symmetry:

$$\mathcal{K}(-\boldsymbol{u}) = \mathcal{K}(\boldsymbol{u}). \tag{9}$$

3) Non-negative and real-valued integrable:

$$\mathcal{K}(\boldsymbol{u}) > 0. \tag{10}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3031272, IEEE Internet of Things Journal

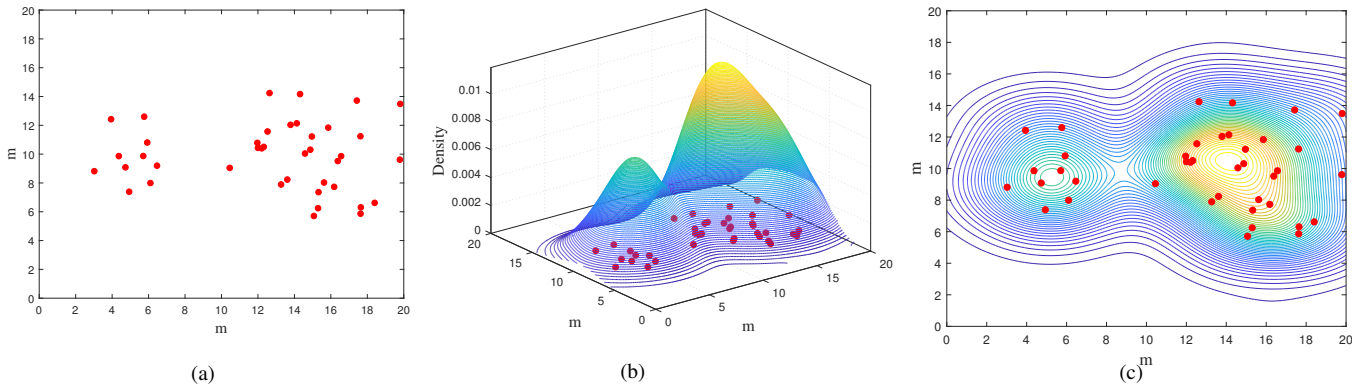


Fig. 2. An example of KDE. (a) Nodes distribution. (b) 3-dimensional density contour of nodes in (a). (c) 2-dimensional density contour of nodes in (a).

A multivariate kernel function can be seen as a product of symmetric univariate kernel functions [36]

$$\mathcal{K}(\boldsymbol{u}) = \prod_{j=1}^{d} \phi(u_j), \tag{11}$$

where u_j is the jth component of the d-dimensional vector $\boldsymbol{u}$, and $\phi(\cdot)$ is a univariate kernel function. In our proposed algorithm, we use the Gaussian kernel function due to its well-known properties [37], which is defined as follows:

$$\phi(u_j) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u_j^2}{2}\right). \tag{12}$$

Fig. 2 is an example of KDE for a set of data. The set of discrete points is transformed into a smooth density map, as shown in Fig. 2 (b), which displays its spatial distribution. The higher the PDF value in a location is, the higher the density is.

C. "Clustering by Fast Search and Find of Density Peaks" Algorithm

CFSFDP is a new clustering algorithm proposed by Rodriguez and Laio [26]. It is based on the assumptions that cluster centers are surrounded by lower local density neighbors and they are at a relatively large distance from any nodes with a higher local density. This method needs to calculate two quantities for each node i: local density ρ_i and distance δ_i. The cluster centers are the nodes with higher local density and larger distance. For a set of nodes' locations $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, and nodes' label set $\boldsymbol{I} = \{1, \ldots, n\}$, the local density of a node $\boldsymbol{x}_i$ is defined as

$$\rho_i = \sum_{i \neq j} \chi(d_{ij} - d_{\mathrm{c}}), \tag{13}$$

where

$$\chi(\alpha) = \begin{cases} 1, & \alpha < 0, \\ 0, & \alpha \geq 0, \end{cases} \tag{14}$$

d_{ij} is the distance between nodes $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and d_{c} is the cutoff distance. The choice of d_{c} should yield an average number of neighbors around 1 to 2% of the total number of nodes. In essence, ρ_i can be seen as the number of nodes that are neighbor to node $\boldsymbol{x}_i$ in the range of d_{c}.

Two cases need to be considered in calculating a node's distance. If node i has the highest density, then its distance δ_i is the maximum value of distances from node i to all other nodes in $\boldsymbol{I}$. Otherwise, the distance of node i is defined as the distance between node i and its nearest neighbor having a higher density [38]. Specifically, the distance δ_i is expressed as

$$\delta_i = \begin{cases} \max(d_{ij})_{j \in \boldsymbol{I}}, & \text{if } \rho_i \text{ is maximum}, \\ \min(d_{ij})_{j \in \boldsymbol{I}^{(i)}}, & \text{otherwise}, \end{cases} \tag{15}$$

$$\boldsymbol{I}^{(i)} = \{t \in \boldsymbol{I} : \rho_t > \rho_i\}, \tag{16}$$

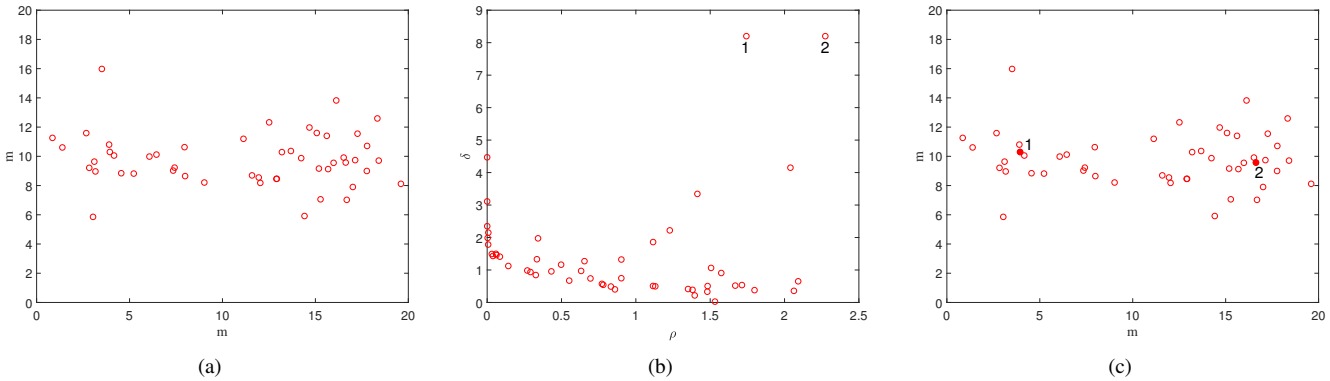
where $\boldsymbol{I}^{(i)}$ is the nodes' label set with node densities greater than ρ_i. After these two quantities are calculated, the cluster centers are selected from nodes with high values of both ρ_i and δ_i. Then, the CFSFDP algorithm assigns other remaining points to the nearest cluster center to form clusters. Specifically, if ρ_i is large and δ_i is small for node i, it means node i is close to the cluster center but not the center. On the other hand, if node i has small ρ_i and large δ_i, it implies that the node is away from the cluster center [39].

Fig. 3 (b) shows the plot of δ_i as a function of ρ_i for each node in Fig. 3 (a). This representation is called the decision graph. According to the decision graph, we can have two nodes with higher values of both density ρ and distance δ. Hence, they can be chosen as cluster centers, as shown in Fig. 3 (c).

III. Proposed IS-k-Means Algorithm

The proposed IS-k-means algorithm involves two phases: (i) set-up phase, and (ii) steady phases. During the set-up phase, each node broadcasts a HELLO message including its ID and location within the range of its coverage so that each node can acquire information of its neighbor nodes. Next, each node sends its information to the BS by the geographic multi-hop routing algorithm [11] because it already knows the positions of its neighbor nodes. The BS runs the proposed IS-k-means algorithm according to the information received from all nodes. The proposed algorithm uses CFSFDP and KDE algorithms to optimize the selection of initial cluster centers of the soft k-means clustering method. Then, the soft

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3031272, IEEE Internet of Things Journal



(a) (b) (c)

Fig. 3. CFSFDP in two dimensions. (a) Nodes distribution. (b) Decision graph for nodes in (a): X-coordinate is local density ρ, and Y-coordinate is δ. (c) Two center nodes are determined.

k-means is used to form clusters and node reassigning scheme is employed to balance the numbers of nodes in different clusters. In order to balance the energy overhead of CHs, the multi-CHs scheme is utilized. After formulation of clusters and selection of CHs are completed, the BS broadcasts the results to all nodes by the restricted flooding method [11]. Thus, each node can identify its role, e.g., CH or member node, and choose to join a corresponding CH if it is a member node. The steady phase is composed of many communication rounds. In each round r, member nodes collect and transmit data to CHs in their allotted time slots, and CHs aggregate the data and send it to the BS. When the energy of a CH is less then a threshold, it will broadcast a SWITCH message to activate the next candidate CH in the same cluster as the new CH and inform member nodes to send data to this new CH. If all CHs in a certain cluster are enabled sequentially, the last working CH will send a RESTART message to the BS to trigger re-clustering. The flowchart of the proposed IS-k-means algorithm is shown in Fig. 4.

A. Energy Model

The first-order radio model [20] is used to calculate the energy consumption of the network. The transmitter's energy consumption involves the transmitter circuitry and the power amplifier, while the energy consumption of the receiver accounts for the receiver circuitry. The free space and the multipath fading models are used in the transmitter power amplifier. If the distance between the transmitter and the receiver is less than a threshold, the power amplifier uses the free space model; otherwise, the multipath model is used [40]. The energy consumption of the transmitter and the receiver for transmitting an l-bit message can be calculated as follows [20]

$$E_{\mathrm{T}} = \begin{cases} lE_{\mathrm{elec}} + l\varepsilon_{\mathrm{fs}}d^2, & d \le d_0, \\ lE_{\mathrm{elec}} + l\varepsilon_{\mathrm{mp}}d^4, & d > d_0, \end{cases} \tag{17}$$

$$E_{\mathrm{R}} = lE_{\mathrm{elec}}, \tag{18}$$

$$d_0 = \sqrt{\frac{\varepsilon_{\mathrm{fs}}}{\varepsilon_{\mathrm{mp}}}}, \tag{19}$$

where E_{T} is the dissipated energy in the transmitter and E_{R} is the dissipated energy in the receiver. E_{elec} is the dissipated energy per bit in both the transmitter circuitry and the receiver circuitry. d is the transmission distance between the transmitter and the receiver. d_0 is the distance threshold. $\varepsilon_{\mathrm{fs}}$ and $\varepsilon_{\mathrm{mp}}$ represent the radio amplifier energy parameter of the free space and multipath fading models [11], respectively.

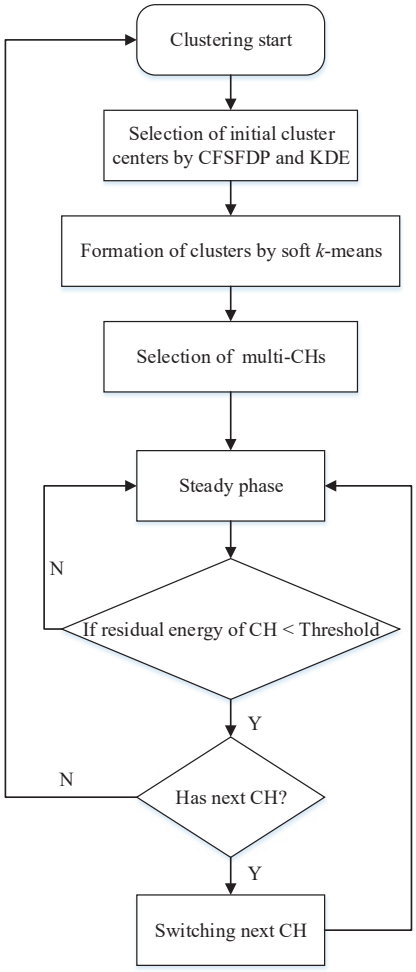


Fig. 4. Flowchart of the proposed algorithm.

Because there are many rounds within the steady phase, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3031272, IEEE Internet of Things Journal

energy consumption of a CH in round r can be calculated as

$$E_{\text{CH}}(r) = gcE_{\text{T}} + g(clE_{\text{DA}} + E_{\text{R}}), \tag{20}$$

where E_{DA} represents the dissipated energy of data aggregation and c is the data aggregation ratio. The first term of the right hand side of (20) is the energy consumption of a CH for sending aggregated data to the BS and the second term is the energy consumption of receiving and aggregating data of g member nodes. The energy consumption of a member node sending data to its CH in round r is

$$E_{\text{nonCH}}(r) = E_{\text{T}}. \tag{21}$$

Hence, the residual energy of node i in round r can be computed by

$$E_i(r) = \begin{cases} E_i(r-1) - E_{\text{CH}}(r), & i \in \text{CHs}, \\ E_i(r-1) - E_{\text{nonCH}}(r), & i \notin \text{CHs}, \end{cases} \tag{22}$$

where $E_i(r-1)$ is the residual energy of node i in the $r-1$ round.

B. Selection of Initial Cluster Centers

We use CFSFDP and KDE algorithms to determine the initial cluster centers as the input to the soft k-means clustering algorithm to produce a better clustering result. Because cluster centers are surrounded by neighbors with lower local density and they are at a relatively large distance from any points with a higher local density, they are selected by the maximum distance δ and relatively high local density ρ, which is illustrated in Fig. 3. First, we calculate the density of each node and find the nodes' set $\boldsymbol{X}'$ with relatively high density $\boldsymbol{\rho}'$. Then, the distances δ among nodes in $\boldsymbol{X}'$ are computed. In order to choose cluster centers, we only choose nodes with relatively high density, and then we multiply their density ρ_i and distance δ_i together as

$$\gamma_i = \rho_i \times \delta_i, \ i \in \{1, \ldots, m\}, \tag{23}$$

where m is the number of nodes with relatively high density. Since each initial cluster center node should have a high γ value, we choose nodes with relatively large γ value as the initial cluster centers. In addition, the value of k is equal to the number of the initial cluster centers. Algorithm 1 describes the detailed steps.

C. Cluster Formation

Some k-means-based algorithms form clusters according to the distances between normal nodes and CHs, such as distributed k-means clustering algorithm [42] and improved k-means cluster-based routing [41]. These k-means-based algorithms can easily lead to a large gap in the number of nodes in different clusters in WSNs and may cause unbalanced energy consumption of CHs. Hence, compared with these k-means-based clustering algorithms, our proposed IS-k-means algorithm uses the soft k-means clustering algorithm to address this problem. Each node can be a member of more than one clusters at the same time according to membership probabilities in the soft k-means. However, member nodes need to join only one cluster with the highest membership probability at a time. Some boundary nodes may have similar probabilities to join multiple clusters. After the convergence of our proposed IS-k-means algorithm, we may reassign nodes to different clusters to balance the number of nodes per cluster. For example, node a is at the edge of two clusters and it has a higher probability to join cluster A, as illustrated in Fig. 5. Before reassigning node a, cluster A already has 10 member nodes and cluster B has 5 member nodes. Since all member nodes send messages to their CH, CH of cluster A will deal with more information from its member nodes. In order to balance the energy consumption of CHs, it is better to reassign node a to cluster B. Reassigning node a from cluster A to cluster B may increase slightly the energy consumption of transmitting messages between node a and its CH because the transmission distance d is increased. However, this slight increase of the transmission energy consumption is negligible as compared to the total energy consumption in CH. If the difference of the probabilities of a node belonging to two clusters is less than a certain threshold, it will join the cluster with low density. If a node is at the boundary of three or more clusters, the proposed algorithm only choose the first two

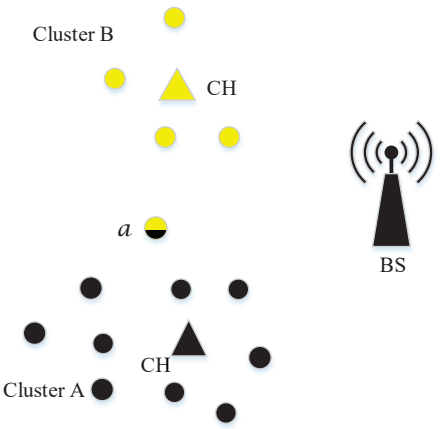


Fig. 5. A node at the boundary of two clusters.

Algorithm 1: Selection of initial cluster centers

Input: $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$
Output: Initial cluster centers: $\boldsymbol{M}$

```
for i = 1 : n do
    calculate ρ_i
end for
ρ = {ρ_1, ..., ρ_n}
choose nodes with local maximum density
    X' = {x_1, ..., x_m} and get their density set
    ρ' = {ρ_1, ..., ρ_m}, m < n
for i = 1 : m do
    calculate δ_i
end for
Δ = {δ_1, ..., δ_m}
calculate γ_i by (23) to determine the initial cluster
    centers
return M = {μ_1, ..., μ_k}
```

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3031272, IEEE Internet of Things Journal

Algorithm 2: Cluster formation

Input: $\boldsymbol{M} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k\}$, $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, the maximum number of iterations $r_{\max}$
Output: k clusters

for r = 1 : $r_{\max}$ **do**
  for v = 1 : k **do**
    for j = 1 : n **do**
      $z' = 0$
      for l = 1 : k **do**
        $z' = z' + e^{-\beta||\boldsymbol{x}_j - \boldsymbol{\mu}_l||^2}$
      end for
      $z_{vj} = \frac{e^{-\beta||\boldsymbol{x}_j - \boldsymbol{\mu}_v||^2}}{z'}$
    end for
  end for
  $\boldsymbol{Z}_r = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ z_{k1} & \cdots & \cdots & z_{kn} \end{bmatrix}$
  for $v = 1 : k$ **do**
    $\boldsymbol{\mu}_v = \frac{\sum_{j=1}^{n} z_{vj} \boldsymbol{x}_j}{\sum_{j=1}^{n} z_{vj}}$
  end for
end for
final membership probabilities $\boldsymbol{Z}_{r_{\max}}$
for j = 1 : n **do**
  assign node j to cluster with the highest probability according to $\boldsymbol{Z}_{r_{\max}}$
end for
k clusters $\boldsymbol{C} = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k\}$
for j = 1 : n **do**
  reassign node j located on the border to different cluster
end for
return k new clusters $\boldsymbol{C}' = \{\boldsymbol{c}'_1, \ldots, \boldsymbol{c}'_k\}$

maximum probabilities and follow the same rule. Algorithm 2 outlines the cluster formation algorithm.

D. Selection of Multi-CHs

Normally, the numbers of nodes in different clusters are different in WSNs. If only one CH is selected in each cluster, CH will consume too much energy to deal with the information from its member nodes in a high density cluster, which will cause its death too early. Hence, our proposed IS-k-means algorithm designs a scheme of multi-CHs. The number of CHs is not fixed in each cluster, and it is determined by the number of nodes per cluster. The larger the number of nodes in a cluster is, the higher the number of CHs will be. The remaining energy of nodes and distances between nodes and their cluster centers are considered in choosing CHs. Nodes close to their cluster center and having higher residual energy than the average energy of the cluster can become CHs. We define a matrix $\mathbf{CHs} = \{\mathbf{CH}_1, \ldots, \mathbf{CH}_k\}$, which is composed of all CHs of k clusters, and $\mathbf{CH}_v, 1 \leq v \leq k$, represents the set of CHs of cluster v. The total remaining energy of cluster

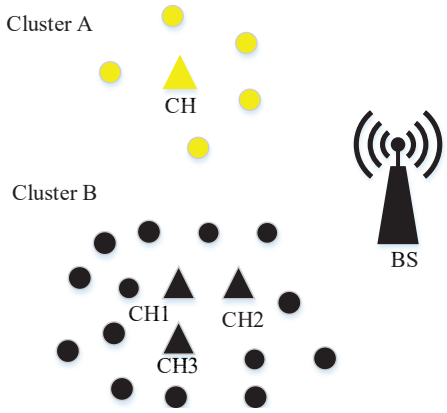


Fig. 6. Multi-CHs scheme.

Algorithm 3: Selection of multi-CHs

Input: $\boldsymbol{C}' = \{\boldsymbol{c}'_1, \ldots, \boldsymbol{c}'_k\}$
Output: **CHs**

for $v = 1 : k$ **do**
  calculate the size of cluster $\boldsymbol{c}'_v$: S_v
  calculate average energy of cluster E_{ave_v}
  $p = \frac{S_v}{\text{constant}}$, the number of CHs of cluster v
  for i = 1 : S_v **do**
    Iterate $\boldsymbol{x}_i$ from near the center of cluster
    if $E_i > E_{\text{ave}_v}$ and $p > 0$ **then**
      $p = p - 1$
      $\mathbf{CH}_v(p) = \boldsymbol{x}_i$
    end if
  end for
end for
return $\mathbf{CHs} = \{\mathbf{CH}_1, \ldots, \mathbf{CH}_k\}$, CHs of k clusters

$v \in \{1, \ldots, k\}$ can be computed as

$$E_v = \sum_{i=1}^{S_v} E_i(r), \tag{24}$$

where S_v is the size of cluster v, $E_i(r)$ is the residual energy of node i in current round r, which can be obtained from (22).

The average energy of cluster v is calculated as

$$E_{\text{ave}_v} = \frac{E_v}{S_v}. \tag{25}$$

As an example, since the number of nodes in cluster B is around 3 times that of cluster A in Fig. 6, cluster B will have three CHs if only one CH is selected in cluster A. After the number of CHs is determined, the nodes which have larger remaining energy and close to the cluster center are selected as CHs. This multi-CHs scheme can balance the energy consumption of CHs per cluster in WSNs, and is summarized in Algorithm 3.

After the set of CHs and clusters are determined, the first node in each $\mathbf{CH}_v$ is selected as the current CH in that cluster and the BS notifies all member nodes to join the cluster to which they belong. CHs broadcast time division multiple access schedules to their member nodes for transmitting data in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3031272, IEEE Internet of Things Journal

Algorithm 4: Switching to a next CH

Input: CHs of k clusters, $\mathbf{CHs} = \{\mathbf{CH}_1, \ldots, \mathbf{CH}_k\}$
Output: Next CH

```
Current round
for v = 1 : k do
    T = (residual energy of CH_v(p) in current round) / (residual energy of CH_v(p) in last round)
    if T < Threshold then
        if CH_v has CH_v(p+1) then
            switch to CH_v(p+1)
        else
            re-clustering
        end if
    end if
end for
```

different time slots to avoid data collision. Then, the network enters the steady phase and begins to exchange data between normal nodes and their CHs.

E. Switching to a Next CH

For balancing the energy consumption of CHs, if the energy consumption ratio of a current CH of any cluster is below a threshold value, the next candidate CH in that cluster is enabled. Until all CHs in a given cluster are executed, the algorithm starts re-clustering. The specific steps are described in Algorithm 4.

F. Complexity Analysis

The run-time complexity of the proposed IS-k-means algorithm mainly involves three phases. In the phase of selecting initial cluster centers, IS-k-means needs $o(n^2)$ operations [44] to execute CFSFDP and KDE to calculate the nodes' densities and distances where n is the number of nodes. Then, the algorithm requires $o(nk^2 r_{\max})$ operations [45] to execute the soft k-means and $o(2n)$ operations to assign nodes to form final clusters. Because the selection of initial cluster centers has been optimized, the algorithm converges quickly and the value of $r_{\max}$ is very small. In the third phase, the algorithm needs $o(n)$ operations to select CHs. Thus, the overall time complexity of the proposed IS-k-means algorithm is $o(n^2 + nk^2 r_{\max} + 3n)$ operations. Obviously, the time complexity of the IS-k-means depends mainly on the execution time of the first phase. The time complexity of the soft k-means algorithm is $o(nk^2 r_{\max})$ operations [45], which is lower than that of our proposed IS-k-means algorithm. However, the higher complexity of the proposed IS-k-means algorithm can be well justified by its ability to better balance the energy consumption of nodes. As for the memory requirement, the proposed algorithm needs $o(n)$ memory units to store nodes first. Then, it costs $o(n)$ memory units [46] to store ρ and δ in the phase of selecting initial cluster centers. Then, $o(nk)$ memory units are required to store membership probabilities in the phase of cluster formation. Hence, the total storage requirement of the proposed algorithm is $o(2n + nk)$ memory units.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Area	100 m × 100 m, 200 m × 200 m
BS coordinates	(50 m, 150 m), (100 m, 200 m)
Initial energy	0.2 J, 1 J
Packet length	4000 bits
Control length	100 bits
E_{T}	50 nJ/bit
E_{R}	50 nJ/bit
$\varepsilon_{\mathrm{fs}}$	10 pJ/bit/m^2
$\varepsilon_{\mathrm{mp}}$	0.0013 pJ/bit/m^4
E_{DA}	5 nJ/bit
d_0	88 m
Number of sensor nodes	28, 100
Maximum communication range	250 m [43]

IV. Experiment Results and Analysis

A. Simulation Settings

To evaluate the performance of the proposed algorithm, we consider two different scenarios. In Scenario 1, the network size is $100\,\mathrm{m} \times 100\,\mathrm{m}$ and the BS is located at $(50\,\mathrm{m}, 150\,\mathrm{m})$. Scenario 2 has the size of $200\,\mathrm{m} \times 200\,\mathrm{m}$ with the BS at location $(100\,\mathrm{m}, 200\,\mathrm{m})$. The main simulation parameters are selected as in [7] and listed in Table I. The experiments are implemented using MATLAB R2017b.

B. Nodes Reassigning of Improved Soft k-Means Analysis

In this subsection, we will show the advantage of the node reassigning scheme incorporated in the proposed IS-k-means algorithm to balance the energy consumption of CHs. A total of 28 sensor nodes are randomly distributed in scenario 1. First, we use the k-means clustering method to classify these nodes and obtain two clusters, as shown in Fig. 7 (a). It is found that cluster 1 contains 20 nodes, which is quite larger than the number of nodes in cluster 2. As a result, CH of cluster 1 will be exhausted much earlier than that of cluster 2. Fig. 7 (b) shows the clustering result of the soft k-means algorithm. In Section III, we define β as the stiffness parameter, which represents the tightness of a node belonging to a cluster. Setting $\beta = 0.2$, we can find that the nodes at the edge of two clusters having similar membership probabilities belonging to these two clusters, such as node 1, node 2, node 3, node 4, and node 5, as shown in Table II. Furthermore, if the value of β changes, the probabilities also will change. When $\beta = 1$, all five nodes belong to the clusters with higher probabilities when compared to the case where $\beta = 0.2$. In our proposed algorithm, we set $\beta = 0.2$ in the following simulations. According to the rule of node reassigning, node 2, node 3, node 4, and node 5 are reassigned to cluster 2 from cluster 1 as shown in Fig. 7 (c), which balances the energy overhead of CHs in these two clusters. The residual energy of CHs, computed by (22), in each round could be used to check the advantage of this scheme. Fig. 8 (a) and Fig. 8 (b) compare the residual energy of CHs among k-means, soft k-means,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3031272, IEEE Internet of Things Journal

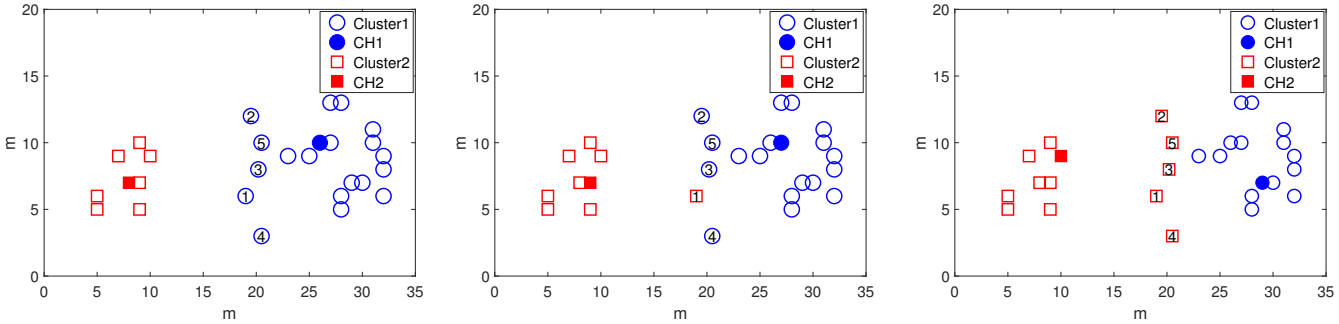


Fig. 7. Comparison of different clustering results, $\beta = 0.2$. (a) k-means clustering result. (b) Soft k-means clustering result. (c) IS-k-means clustering result.

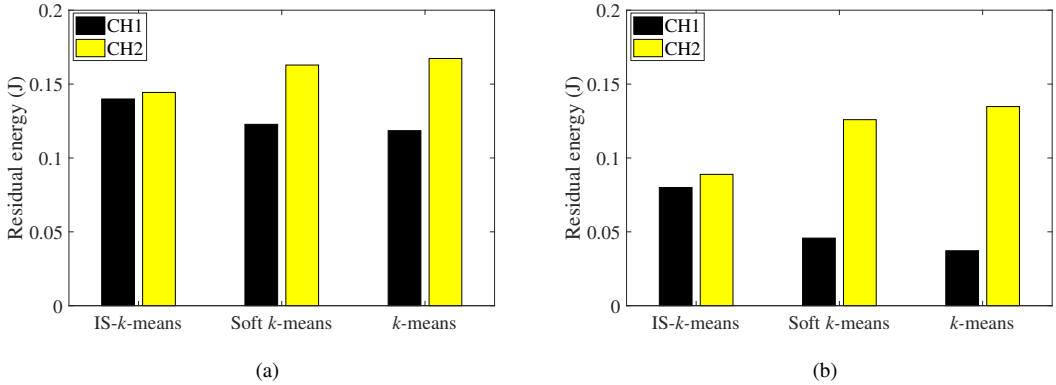


Fig. 8. Comparison of residual energy of CHs. (a) Residual energy of CHs after 5 rounds. (b) Residual energy of CHs after 10 rounds.

TABLE II
PROBABILITIES COMPARISON

Probability		Node 1	Node 2	Node 3	Node 4	Node 5
$\beta = 0.2$	Cluster 1	0.4852	0.5537	0.5684	0.6125	0.6120
	Cluster 2	0.5148	0.4463	0.4316	0.3875	0.3880
$\beta = 1$	Cluster 1	0.0438	0.9787	0.9860	0.9919	0.9992
	Cluster 2	0.9562	0.0213	0.014	0.0081	0.0008

and IS-k-means after 5 rounds and 10 rounds, respectively. The IS-k-means algorithm achieves an equilibrium of energy consumption in both CHs when compared to the k-means and the soft k-means algorithms.

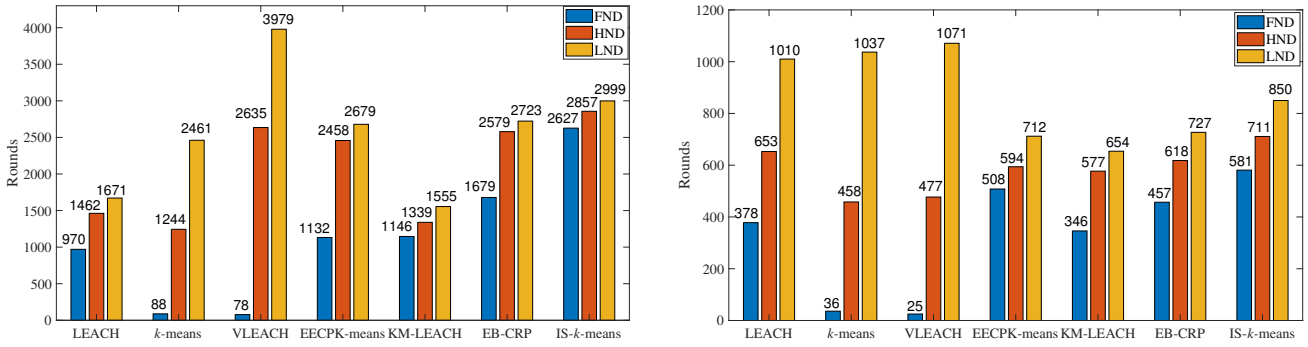
C. Network Lifetime

To test the performance of the proposed IS-k-means algorithm, we compare it with KM-LEACH [23], VLEACH [24], LEACH [20], k-means [47], EECPK-means [21], and EB-CRP [13] with the same parameters shown in Table I. Here, we state two things about the implementation of the EB-CRP algorithm in our experiment. First, the original EB-CRP algorithm does not need to select the CHs because the authors consider a certain number of gateways with enough energy to act as CHs in WSN. However, our implemented EB-CRP algorithm needs to select CHs randomly from all sensor nodes because the network considered in our simulation contains only sensor nodes with the same initial energy and functionality. In order to have a fair comparison, we set the number of CHs in EB-CRP to be the same as that in our proposed algorithm. Thus, the location of CHs may be different in each steady-state phase because all nodes have the same chance to be CHs. Second, the steady-state phase of the original EB-CRP algorithm is composed of pre-specified 75 rounds. This is quite reasonable because the authors set the initial energy of CHs to be 10 J, which can maintain a high number of communication rounds. However, considering the limited energy of CHs in our simulation, each steady-state phase is composed of 20 rounds in our implemented EB-CRP algorithm, which can achieve the best results for the EB-CRP algorithm.

We assume there are 100 sensor nodes that are randomly distributed in both scenario 1 and scenario 2. The obtained results are the averages of 20 independent experiments. The authors in [20] found the optimum number of clusters to be between 3 and 5 for 100-node network in LEACH. Thus, in scenario 1, we set 4 as the initial number of clusters in LEACH. For the other six algorithms, we use CFSFDP and KDE to determine the number of clusters in order to ensure the same number of clusters for each algorithm. The initial number of clusters found from the CFSFDP and KDE algorithms is 4 in scenario 1. In scenario 2, all algorithms are set with the same number of clusters 6 that is determined by CFSFDP and KDE. We assume that the death of 85% nodes means all nodes are dead.

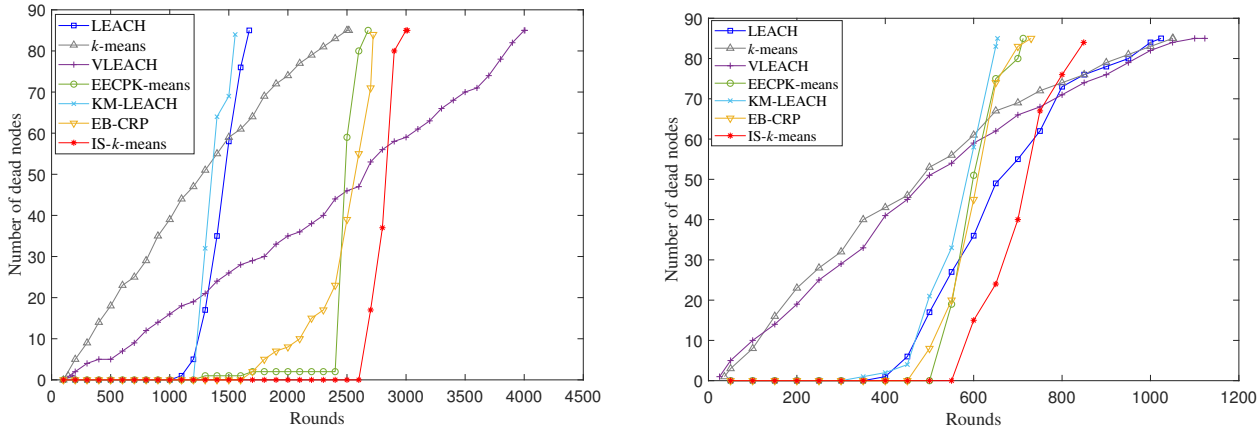
Fig. 9 shows the first node death (FND), half of nodes

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3031272, IEEE Internet of Things Journal



(a) (b)

Fig. 9. Comparison of FND, HND, and LND. (a) Scenario 1. (b) Scenario 2.



(a) (b)

Fig. 10. Comparison of network lifetime of LEACH, k-means, VLEACH, EECPK-means, KM-LEACH, EB-CRP, and IS-k-means. (a) Scenario 1. (b) Scenario 2.

death (HND), and the last node death (LND) for these seven algorithms when the number of nodes is 100. If an algorithm can balance energy well, the first node death will be very late. In Fig. 9 (a), the average number of rounds of FND in k-means is 88, which is much earlier than 970 in LEACH and 2627 in IS-k-means, and the average LND happens later when compared to LEACH and the IS-k-means algorithms. Thus, it is obvious that the energy consumption of k-means is unbalanced. Although VLEACH uses the vice CH scheme in each cluster to extend the network lifetime, it exhibits a poor performance in balancing energy consumption because its FND is 78 and LND is 3979, as shown in Fig. 9 (a). EECPK-means improves the selection of initial cluster centers of the k-means algorithm by using the midpoint algorithm. It outperforms LEACH and KM-LEACH in both balancing energy consumption and extending network lifetime. For the EB-CRP algorithm, its FND is about 1.7 times that of LEACH, 19 times that of k-means, 21 times that of VLEACH, 1.5 times that of EECPK-means, and 1.5 times that of KM-LEACH, which demonstrates that the EB-CRP algorithm can postpone the death of the first node when compared with the other five algorithms. In addition, the HND of EB-CRP is 2579, which is larger than 1462 in LEACH, 1244 in k-means, 2458 in EECPK-means, and 1339 in KM-LEACH. This result means that the EB-CRP algorithm can delay the death of the first 50% of nodes as compared to LEACH, k-means, EECPK-means, KM-LEACH. Thus, the EB-CRP shows a good performance in balancing the energy consumption of the nodes and increasing the network lifetime. In view of Fig. 9 (a), our proposed IS-k-means algorithm can effectively postpone the FND, HND and LND. The average FND of IS-k-means is 2627, which is around 2.7 times that of LEACH, 30 times that of k-means, 34 times that of VLEACH, 2.3 times that of KM-LEACH, 2.4 times that of EECPK-means, and 1.5 times that of EB-CRP. Instead of using a fixed number of communication rounds during each steady-phase, like in the EB-CRP algorithm, the communication rounds in our proposed IS-k-means algorithm are determined by the residual energy of CHs. If the residual energy of any CH is below the threshold, the algorithm will stop the current steady-phase and trigger re-clustering, which can avoid CHs to die earlier than EB-CRP. Thus, the IS-k-means algorithm can keep all nodes in the network alive in most rounds. The average HND of the IS-k-means is also around 2 times among LEACH, k-means, and KM-LEACH.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3031272, IEEE Internet of Things Journal

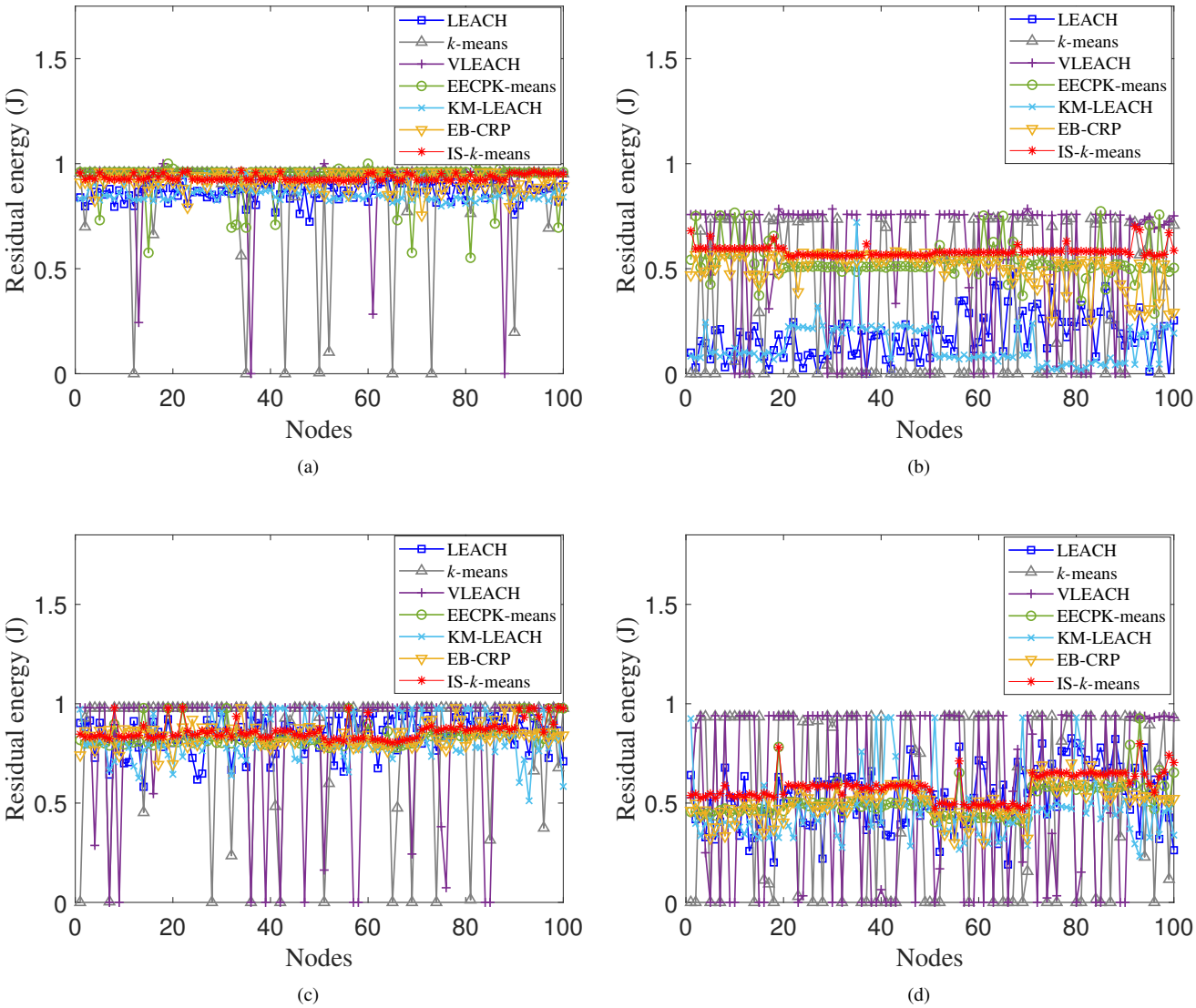


Fig. 11. Comparison of residual energy curve. (a) Residual energy after 400 rounds in scenario 1. (b) Residual energy after 1000 rounds in scenario 1. (c) Residual energy after 100 rounds in scenario 2. (d) Residual energy after 300 rounds in scenario 2.

In Fig. 9 (b), the average FND, HND, and LND of all algorithms are decreased. This is because extending the network size will increase the communication distance of the nodes which leads to an increase in the energy consumption. The VLEACH and k-means algorithms still show a very poor outcome in balancing the energy consumption, a consequence of having small FND and large LND. However, EECPK-means and EB-CRP have relatively large values of the FND and HND, which means most nodes in these two algorithms live longer when compared with k-means, VLEACH, and KM-LEACH. In addition, it is evident that our proposed IS-k-means algorithm has the best results in postponing the FND, HND, and LND as compared with the other six algorithms.

Fig. 10 shows the network lifetime comparison of our proposed IS-k-means algorithm and the other six algorithms. As can be seen from Fig. 10 (a), the network lifetime curves of KM-LEACH, LEACH, EECPK-means, and the proposed IS-k-means algorithms are approximately vertical. This means that, in these algorithms, the majority of nodes die approximately after the same number of rounds. Furthermore, one can see that the proposed IS-k-means algorithm outperforms KM-LEACH, LEACH, and EECPK-means algorithms in terms of the energy consumption equilibrium. The results in Fig. 10 (a) also show that VLEACH has a longer network lifetime than our proposed IS-k-means algorithm. This is reasonable since the objective of VLEACH is to extend the network lifetime, whereas our proposed IS-k-means algorithm aims to balance the energy consumption in the network. As a result, some nodes die very early and others die very late in VLEACH, which likely results in the inability to collect sensing data from certain areas where some nodes are dead. In Fig. 10 (b), although none of the algorithms shows a nearly vertical curve, like in Fig. 10 (a), our proposed algorithm still outperforms the other six algorithms in balancing the energy consumption and prolonging the network lifetime.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3031272, IEEE Internet of Things Journal

TABLE III
COMPARISON OF ENERGY VARIANCE OF DIFFERENT ROUNDS

		LEACH	k-means	VLEACH	EECPK-means	KM-LEACH	EB-CRP	IS-k-means
Scenario 1	200 rounds	0.0018	0.0374	0.0271	0.0127	0.0013	0.0019	0.0002
	400 rounds	0.0039	0.0768	0.0496	0.0264	0.0025	0.0024	0.0004
	600 rounds	0.0074	0.1161	0.0775	0.0396	0.0085	0.0026	0.0004
	800 rounds	0.0108	0.1163	0.0756	0.0432	0.0098	0.0038	0.0005
	1000 rounds	0.0168	0.1148	0.0882	0.0498	0.0094	0.0044	0.0008
	1200 rounds	0.0172	0.1016	0.0904	0.0515	0.0055	0.0067	0.0007
	1400 rounds	0.0095	0.0988	0.0901	0.0496	0.0024	0.0072	0.0009
Scenario 2	100 rounds	0.0081	0.0983	0.1070	0.0031	0.0112	0.0028	0.0022
	200 rounds	0.0131	0.1642	0.1645	0.0058	0.0186	0.0052	0.0045
	300 rounds	0.0248	0.1921	0.1899	0.0073	0.0258	0.0073	0.0046
	400 rounds	0.0375	0.1806	0.1963	0.0105	0.0392	0.0094	0.0080
	500 rounds	0.0388	0.1713	0.1821	0.0135	0.0312	0.0122	0.0110
	600 rounds	0.0357	0.1491	0.1647	0.0167	0.0181	0.0168	0.0141

D. Energy Variance

Fig. 11 compares the average residual energy of all 100 nodes in WSNs among the seven algorithms after different rounds in two scenarios. It is found that the residual energy curve of all nodes in the IS-k-means algorithm is smoother than that of the other six algorithms. This result demonstrates that the IS-k-means algorithm is good at balancing the energy consumption of all nodes in WSNs. For the purpose of estimating performance of the proposed algorithm, we introduce a new parameter called energy variance (EV), which is expressed as

$$\mathrm{EV} = \frac{\sum_{i=1}^{n} \left(E_i\left(r\right) - \overline{E}\right)^2}{n}, \tag{26}$$

where $\overline{E}$ is the average energy of all nodes. Table III clearly reveals that EB-CRP has relatively smaller variances than LEACH, k-means, VLEACH, KM-LEACH, and EECPK-means in different rounds. In addition, our proposed IS-k-means algorithm achieves the smallest variances among seven algorithms, which demonstrates that the IS-k-means can keep the residual energy of 100 nodes to be the most uniform in WSNs.

It is worthy to mention that the EB-CRP algorithm shows better performance in extending the network lifetime for WSNs with large network sizes [13], while the proposed algorithm has good performance in balancing the energy consumption and extending the network lifetime for smaller network sizes. We briefly summarize the reasons why the proposed algorithm performs better than the other six algorithms for WSNs of smaller sizes. First, optimizing the initial cluster centers of the soft k-means algorithm and reassigning nodes can better balance the number of nodes in different clusters to form good clustering results. Second, our algorithm selects nodes with more residual energy as the CHs, which can prevent the CHs from dying too early and support a high number of communication rounds. Third, the multi-CHs scheme of the proposed IS-k-means can reduce the communication energy consumption in the set-up phase caused by re-clustering because it reduces the number of re-clustering. Thus, all sensors can save energy to maintain more communication rounds in the steady phase, which extends the network lifetime. However, for the EB-CRP algorithm, it only chooses one CH in each cluster, which may cause all nodes to re-cluster frequently because CHs may quickly exhaust their energy. Fourth, instead of using a fixed number of communication rounds during each steady-phase, like in EB-CRP, the communication rounds in our proposed IS-k-means algorithm are determined by the residual energy of CHs. If the residual energy of any CH is below the threshold, the algorithm will stop the current steady-phase and trigger re-clustering, which can avoid CHs to die earlier than the EB-CRP algorithm.

V. CONCLUSIONS

In this paper, we proposed an energy balanced IS-k-means algorithm based on the soft k-means for WSNs. The proposed algorithm improves the selection of initial cluster centers by using CFSFDP and KDE algorithms. In order to balance the number of nodes per cluster, the proposed algorithm reassigns nodes at the edge of different clusters to a low-density cluster according to the nodes' membership probabilities. Furthermore, multi-CHs scheme was used in the selection of final CHs, which can effectively balance the traffic load of CHs, reduces the number of re-clustering and saves communication cost in the set-up phase. In order to show the advantages of the IS-k-means in balancing energy consumption, we compared it with LEACH, k-means, VLEACH, EECPK-means, KM-LEACH, and EB-CRP. In scenario 1, simulation results demonstrated that the proposed IS-k-means algorithm postponed the FND by 2.7 times, 34 times, 2.3 times, 2.4 times, 30 times, and 1.5 times when compared to LEACH, VLEACH, KM-LEACH, EECPK-means, k-means, and EB-CRP on average, respectively. The HND of the IS-k-means algorithm also was delayed by 2 times when compared to LEACH, k-means, and KM-LEACH. In addition, the IS-k-means algorithm achieved an excellent result in postponing the FND and HND in scenario 2 as compared with other mentioned algorithms. The IS-k-means algorithm also extended network lifetime in both scenarios as compared to

KM-LEACH, EECPK-means, and EB-CRP. Furthermore, the proposed algorithm also yields smoother average remaining energy curves of all nodes in different rounds and smaller average energy variances. Hence, the proposed IS-k-means algorithm is promising in balancing energy consumption in WSNs. In a future work, we plan to design an energy-efficient multi-hop routing algorithm to extend the IS-k-means algorithm to large-scale WSNs.

REFERENCES

[1] L. Chettri and R. Bera, "A comprehensive survey on internet of things (IoT) toward 5G wireless systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16–32, Jan. 2020.

[2] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the internet of things (IoT) forensics: challenges, approaches, and open issues," *IEEE Commun. Surv. Tutor.*, vol. 22, no. 2, pp. 1191–1221, Jan. 2020.

[3] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. N. Hindia, "An overview of internet of things (IoT) and data analytics in agriculture: benefits and challenges," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3758–3773, Oct. 2018.

[4] T. M. Behera, S. K. Mohapatra, U. C. Samal, M. S. Khan, M. Daneshmand, and A. H. Gandomi, "I-SEP: an improved routing protocol for heterogeneous WSN for IoT-based environmental monitoring," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 710–717, Jan. 2020.

[5] F. Deng, X. Yue, X. Fan, S. Guan, Y. Xu, and J. Chen, "Multisource energy harvesting system for a wireless sensor network node in the field environment," *IEEE Internet Things J.*, vol. 6, pp. 918–927, Feb. 2019.

[6] L. Xu, R. Collier, and G. M. P. O'Hare, "A survey of clustering techniques in WSNs and consideration of the challenges of applying such to 5G IoT scenarios," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1229–1249, Oct. 2017.

[7] H. Xie, Z. Yan, Z. Yao, and M. Atiquzzaman, "Data collection for security measurement in wireless sensor networks: a survey," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2205–2224, Apr. 2019.

[8] J. Lee and T. Kao, "An improved three-layer low-energy adaptive clustering hierarchy for wireless sensor networks," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 951–958, Dec. 2016.

[9] T. M. Behera, S. K. Mohapatra, U. C. Samal, M. S. Khan, M. Daneshmand, and A. H. Gandomi, "Residual energy-based cluster-head selection in WSNs for IoT application," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5132–5139, Jun. 2019.

[10] Z. Xu, L. Chen, C. Chen and X. Guan, "Joint clustering and routing design for reliable and efficient data collection in large-scale wireless sensor networks," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 520–532, Aug. 2016.

[11] R. Zhang, J. Pan, D. Xie and F. Wang, "NDCMC: a hybrid data collection approach for large-scale WSNs using mobile element and hierarchical clustering," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 533–543, Aug. 2016.

[12] R. Yarinezhad and S. N. Hashemi, "A routing algorithm for wireless sensor networks based on clustering and an FPT-approximation algorithm," *J. Syst. Softw.*, vol. 155, pp. 145–161, Sep. 2019.

[13] R. Yarinezhad and S. N. Hashemi, "Increasing the lifetime of sensor networks by a data dissemination model based on a new approximation algorithm," *Ad Hoc. Netw.*, vol. 100, Jan. 2020.

[14] R. Yarinezhad and S. N. Hashemi, "Solving the load balanced clustering and routing problems in WSNs with an FPT-approximation algorithm and a grid structure," *Pervasive Mob. Comput.*, vol. 58, Jun. 2019.

[15] N. Mazumdar and H. Om, "DUCR: distributed unequal cluster-based routing algorithm for heterogeneous wireless sensor networks," *Int. J. Commun. Syst.*, vol. 30, no. 18, Jul. 2017.

[16] N. Mazumdar and H. Om, "Distributed fuzzy logic based energy-aware and coverage preserving unequal clustering algorithm for wireless sensor networks," *Int. J. Commun. Syst.*, vol. 30, no. 13, Sep. 2017.

[17] N. Mazumdar and H. Om, "Distributed fuzzy approach to unequal clustering and routing algorithm for wireless sensor networks," *Int. J. Commun. Syst.*, vol. 31, no. 12, May 2018.

[18] S. Randhawa and S. Jain, "Performance analysis of LEACH with machine learning algorithms in wireless sensor networks," *Int. J. Comput. Applica.*, vol. 147, no. 2, pp. 7–12, Aug. 2016.

[19] A. Mahboub, M. Arioua *et al.*, "Energy-efficient hybrid k-means algorithm for clustered wireless sensor networks," *Int. J. Elec. Comput. Engin.*, vol. 7, no. 4, pp. 2054–2060, Aug. 2017.

[20] W. R. Heinzelman *et al.*, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wirel. Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[21] 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 Wirel. Sens. Syst.*, vol. 6, no. 6, pp. 181–191, Dec. 2016.

[22] N. T. Tam, D. T. Hai, L. H. Son, and L. T. Vinh, "Improving lifetime and network connections of 3D wireless sensor networks based on fuzzy clustering and particle swarm optimization," *Wireless Netw.*, vol. 24, no. 5, pp. 1477–1490, Nov. 2016.

[23] M. Bidaki, R. Ghaemi, and S.R.K Tabbakh, "Towards energy efficient k-means based clustering scheme for wireless sensor networks," *Int. J. Grid Distrib. Comput.*, vol. 9, no. 7, pp. 265–276, Jul. 2016.

[24] A. S. D. Sasikala and N. Sangameswaran, "Improving the energy efficiency of LEACH protocol using VCH in wireless sensor network," *Int. J. Eng. Dev. Res.*, vol. 3, no. 2, pp. 918–924, 2015.

[25] E. Rabiaa, B. Noura, and C. Adnene, "Improvements in LEACH based on k-means and Gauss algorithms," *Procedia Comput. Sci.*, vol. 73, pp. 460–467, Dec. 2015.

[26] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492-1496, Jun. 2014.

[27] Y. Zhang, M. Liu, and Q. Liu, "An energy-balanced clustering protocol based on an improved CFSFDP algorithm for wireless sensor networks," *Sensors*, vol. 18, no. 3, p. 1–18, Mar. 2018.

[28] A. Ihsani and T. H. Farncombe, "A kernel density estimator-based maximum a posteriori image reconstruction method for dynamic emission tomography imaging," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2233–2248, May 2016.

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

[30] P. Shen and C. Li, "Distributed Information Theoretic Clustering," *IEEE Trans. Signal Process.*, vol. 62, no. 13, pp. 3442–3453, Jul. 2014.

[31] R. Sharma, V. Vashisht, and U. Singh, "EEFCM-DE: energy-efficient clustering based on fuzzy C means and differential evolution algorithm in WSNs," *IET Commun.*, vol. 13, no. 8, pp. 996–1007, May 2019.

[32] H. Yang, Q. Yao, A. Yu, Y. Lee, and J. Zhang, "Resource assignment based on dynamic fuzzy clustering in elastic optical networks with multi-core fibers," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3457–3469, May 2019.

[33] A. Majdara and S. Nooshabadi, "Nonparametric density estimation using copula transform, bayesian sequential partitioning, and diffusion-based kernel estimator," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 821–826, Apr. 2020.

[34] Y. Li, Y. Zhang, M. Yu, and X. Li, "Drawing and studying on histogram," *Cluster Comput.*, vol. 22, pp. 3999–4006, Mar. 2019.

[35] A. Qahtan, S. Wang, and X. Zhang, "KDE-track: an efficient dynamic density estimator for data streams," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 3, pp. 642–655, Mar. 2017.

[36] V. Savic, E. G. Larsson, J. Ferrer-Coll, and P. Stenumgaard, "Kernel methods for accurate UWB-based ranging with reduced complexity," *IEEE Trans. Wirel. Commun.*, vol. 15, no. 3, pp. 1783–1793, Mar. 2016.

[37] B. Qin and F. Xiao, "A non-parametric method to determine basic probability assignment based on kernel density estimation," *IEEE Access*, vol. 6, pp. 73509–73519, Nov. 2018.

[38] R. Mehmood *et al.*, "Clustering by fast search and find of density peaks via heat diffusion," *Neurocomputing*, vol. 208, pp. 210–217, Oct. 2016.

[39] J. Zhong, P. Tse, and Y. Wei, "An intelligent and improved density and distance-based clustering approach for industrial survey data classification," *Exp. Syst. Appl.*, vol. 68, pp. 21–28, Feb. 2017.

[40] W. Heinzelman, "Application-specific protocol architectures for wireless networks," Ph.D. thesis, Mass. Inst. Technol., Cambridge, 2000.

[41] M. Lehsaini and M. B. Benmahdi, "An improved k-means cluster-based routing scheme for wireless sensor networks," in *Proc. IEEE Int. Symp. Program. Syst. (ISPS)*, Apr. 2018, pp. 1–6.

[42] J. Qin, W. Fu, H. Gao, and W. X. Zheng, "Distributed k-means algorithm and fuzzy c-means algorithm for sensor networks based on multiagent consensus theory," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 772–783, Mar. 2017.

[43] C. Lipi, S. Paawan, B. Govind, and K. Adesh, "Wireless sensor network based smart grid communications: cyber attacks, intrusion detection system and topology control," *Electronics*, vol. 6, pp. 1–22, Jan. 2017.

[44] W. Zhang and J. Li, "Extended fast search clustering algorithm: widely density clusters, no density peaks," *arXiv:1505.05610*, pp. 1–17, May 2015.

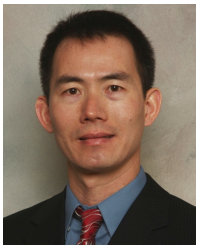
[45] T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall, and M. Palaniswami, "Fuzzy c-means algorithms for very large data," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 6, pp. 1130–1146, Dec. 2012.
[46] M. Wang, F. Min, Y. Wu, and Z. Zhang, "Active learning through density clustering," *Expert Syst. Appl.*, vol. 85, pp. 305–317, Nov. 2017.
[47] P. Sasikumar and S. Khara, "K-means clustering in wireless sensor networks," in *Proc. IEEE Fourth Int. Conf. of Computational Intelligence and Communication Networks (CICN)*, Nov. 2012, pp. 140–144.

Botao Zhu is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, Canada. As the first inventor, he has more than a dozen authorized patents related to his research in China. His current research interests include Wireless Sensor Networks, Machine Learning, and unmanned aerial vehicles (UAVs).

Ebrahim Bedeer received the B.Sc. (Hons.) and M.Sc. degrees from Tanta University, Tanta, Egypt and the Ph.D. degree from Memorial University, St. Johns, NL, Canada, all in Electrical Engineering.

He joined the Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, Canada, in 2019 as an Assistant Professor. Before that, he was an Assistant Professor (Lecturer in the UK) at Ulster University, United Kingdom, and postdoctoral fellow at Carleton University, Ottawa, ON, Canada and the University of British Columbia, Kelowna, BC, Canada. His current research interests include applications of optimization techniques in signal processing and wireless communications, spectral efficient communication systems, and internet-of-things (IoT).

Dr. Bedeer is an Associate Editor of the IEEE COMMUNICATIONS LETTERS. He has served on the Technical Program Committees of numerous major international communication conferences, such as IEEE GLOBECOM, IEEE ICC, and IEEE VTC. Dr. Bedeer received numerous awards including the Exemplary Reviewer of IEEE COMMUNICATIONS LETTERS and IEEE WIRELESS COMMUNICATIONS LETTERS.



Ha H. Nguyen received the B.Eng. degree from Hanoi University of Technology (HUT), Hanoi, Vietnam, in 1995, the M.Eng. degree from the Asian Institute of Technology (AIT), Bangkok, Thailand, in 1997, and the Ph.D. degree from the University of Manitoba, Winnipeg, MB, Canada, in 2001, all in electrical engineering. He joined the Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, SK, Canada, in 2001, and became a full Professor in 2007. He currently holds the position of *NSERC/Cisco Industrial Research Chair in Low-Power Wireless Access for Sensor Networks*. His research interests fall into broad areas of Communication Theory, Wireless Communications, and Statistical Signal Processing. Dr. Nguyen was an Associate Editor for the *IEEE Transactions on Wireless Communications* and *IEEE Wireless Communications Letters* during 2007-2011 and 2011-2016, respectively. He currently serves as an Associate Editor for the *IEEE Transactions on Vehicular Technology*. He served as a Technical Program Chair for numerous IEEE events. He is a coauthor, with Ed Shwedyk, of the textbook "A First Course in Digital Communications" (published by Cambridge University Press). Dr. Nguyen is a Fellow of the Engineering Institute of Canada (EIC) and a Registered Member of the Association of Professional Engineers and Geoscientists of Saskatchewan (APEGS).

Robert Barton is a Distinguished Engineer working in Cisco's Digital Transformation and Innovation group, where he has the role of Chief Architect for Cisco Canada and Cisco's global IoT sales organization. Rob has worked in the IT industry for over 23 years, the last 20 of which have been at Cisco. Rob Graduated from the University of British Columbia with a degree in Engineering Physics. Rob is a published author, with titles on the subjects of Network QoS, Wireless, IoT, Machine Learning and Data Analytics. Rob has also contributed to many academic papers, and leads Cisco's university research partnership program. Rob also holds many patents in the areas of wireless communications, segment routing, and Machine Learning. Rob's current areas of work include wireless communications, Industrial Networking, IoT, and AI/ML in networking systems.

Jerome Henry is a principal engineer in the office of the Wireless CTO at Cisco systems. He has more than 20 years of experience in the wireless industry, focusing on optimization of short and long range performances. An IEEE participant, Jerome was elevated to the rank of Senior Member in 2013. Jerome is a published author, with titles on Wireless technologies, location, 802.11 security, IoT, QoS and Machine Learning, and has also contributed to numerous academic papers on the same topics. Jerome's current area of work include wireless communications and location with a particular focus on accuracy improvement through machine learning.