


Intelligent energy optimization for advanced IoT analytics edge computing on wireless sensor networks

International Journal of Distributed
Sensor Networks
2020, Vol. 16(7)
© The Author(s) 2020
DOI: 10.1177/1550147720908772
journals.sagepub.com/home/dsn


Israel Edem Agbehadji¹ , Samuel Ofori Frimpong¹, Richard C Millham¹,
Simon James Fong^{2,3} and Jason J Jung⁴

Abstract

The current dispensation of big data analytics requires innovative ways of data capturing and transmission. One of the innovative approaches is the use of a sensor device. However, the challenge with a sensor network is how to balance the energy load of wireless sensor networks, which can be achieved by selecting sensor nodes with an adequate amount of energy from a cluster. The clustering technique is one of the approaches to solve this challenge because it optimizes energy in order to increase the lifetime of the sensor network. In this article, a novel bio-inspired clustering algorithm was proposed for a heterogeneous energy environment. The proposed algorithm (referred to as DEEC-KSA) was integrated with a distributed energy-efficient clustering algorithm to ensure efficient energy optimization and was evaluated through simulation and compared with benchmarked clustering algorithms. During the simulation, the dynamic nature of the proposed DEEC-KSA was observed using different parameters, which were expressed in percentages as 0.1%, 4.5%, 11.3%, and 34% while the percentage of the parameter for comparative algorithms was 10%. The simulation result showed that the performance of DEEC-KSA is efficient among the comparative clustering algorithms for energy optimization in terms of stability period, network lifetime, and network throughput. In addition, the proposed DEEC-KSA has the optimal time (in seconds) to send a higher number of packets to the base station successfully. The advantage of the proposed bio-inspired technique is that it utilizes random encircling and half-life period to quickly adapt to different rounds of iteration and jumps out of any local optimum that might not lead to an ideal cluster formation and better network performance.

Keywords

Edge computing, wireless sensor network, clustering algorithm, Kestrel-based search algorithm, heterogeneous energy environment, Internet of things

Date received: 19 July 2019; accepted: 5 February 2020

Handling Editor: Ashish Luhach

Introduction

The Internet of things (IoT) era has enhanced data sharing among connected objects and people.¹ IoT is an ecosystem that consists of technologies such as radio frequency identification (RFID), sensors, and smart devices, which are connected together to form a network for data transmission and reception.² Basically, a network facilitates communication among people and “things” and encourages intelligent collaboration. The

¹ICT and Society Research Group, Department of Information Technology, Durban University of Technology, Durban, South Africa

²ICT and Society Research Group, Department of Computer and Information Science, University of Macau, Taipa, Macau SAR

³Durban University of Technology, Durban, South Africa

⁴Chung-Ang University, Seoul, South Korea

Corresponding author:

Richard C Millham, ICT and Society Research Group, Department of Information Technology, Durban University of Technology, Ritson Campus, Durban 4001, South Africa.
Email: richardm1@dut.ac.za



intelligence relates to how quickly decisions are made on a network to increase the performance and lifetime of the network. Network lifetime is the probability of a network to continuously be available irrespective of the network load.³ Collecting network intelligence on energy is crucial in edge computing particularly when many devices with different energy requirements can be connected at any time, and as the network scales-up, its lifetime should be maintained.

Generally, an energy management system manages demand and supply of energy to its users. On the demand side, balancing energy load connected to energy management systems often consider user's behavior and preferences. By definition, a load is an electrical appliance that can be controlled and measured; in addition, it can be discrete (i.e. on or off) or variable (i.e. having series of consumptions levels).⁴ Mostly, the load is balanced to ensure efficient resource utilization and to help optimize energy needs in order to avoid overload or under-load.⁵ Thus, efficient energy management of connected devices can be achieved through load balance.

The energy management model enables data collection, preprocessing, and analysis for efficient service delivery.⁶ There are several tools and methods which could be deployed to manage and analyze energy. Generally, energy-related systems that are developed utilizing the cloud computing environment are designed to send data to a centralized location (i.e. cloud) for processing and further analysis. In using cloud computing, the end-user can monitor and personalize their energy management needs according to their priority and comfort.⁷ Existing energy management systems include Schneider Electric StruxureWare,⁸ Honeywell Attune Advisory Services,⁹ eSight,¹⁰ and Predictive Energy Optimization.¹¹

Mostly, sensor devices for energy management systems are limited in terms of computing capability, memory, and battery power, and it is a challenge to replenish the battery of sensors.¹² In these regards, methods to preserve energy levels are significant on wireless sensor networks (WSNs). The main contribution of this article is to investigate the problem of energy consumption in the context of IoT devices with different energy requirements. These IoT devices perform activities such as processing data, transmitting packets, reading sensor values, and actuating a device. In performing these activities, the device tends to lose some amount of energy which can limit network performance. In general, different communication devices have different numbers of packets sent and received, as well as the length of the packet. In view of this, we focus on the clustering algorithm for energy optimization of sensor nodes on the edge of a WSN. The objective of this article is to propose a model for energy optimization on a WSN.

In this article, the proposed model for energy optimization is based on the intelligent behavior and characteristics of a bird. The clustering technique is an energy-efficient approach for WSN, in which the cluster head (CH) selection is optimized by utilizing the proposed algorithm. The proposed algorithm (Kestrel-based search algorithm (KSA)) takes inspiration from the hunting strategy of Kestrel birds, which is applied to heterogeneous WSN to optimize energy and increase network performance.

The rest of this article is organized as follows: second section details "Related work/related terminologies". Third section is about "Proposed model." Fourth section is about "Proposed algorithm." Fifth section details "Parameter setting for simulation of the network." Sixth section presents "Simulation results." Seventh section presents conclusions and future work.

Related works/related terminologies

Classical energy models have been used as a guide for the design of low-level energy consumption devices, which are used to transmit and receive data.^{13,14} The energy models are also referred to as routing protocols, which include Energy-Aware Clustering Algorithm,¹⁵ Hierarchical Energy-Efficient Routing Protocol,¹⁶ and Energy-Efficient Hierarchical Clustering.¹⁷ In this section, we will review related works on energy optimization for WSNs.

Internet of Agent model for energy management

Conceptually, the Internet of Agent (IoA) model is based on communication and collaboration among connected agents. These agents negotiate among themselves to find an optimal way of working together in order to form a connected environment of agents. Subsequently, the optimal result is communicated to all other agents to make their own decision. Specifically, each agent is responsible to make its own decision, which might lead to a unanimous decision. For instance, there are "agents" to facilitate communication with other agents in which "agents" can be delegated to facilitate collaboration among agents, while another "agents" can be delegated to learn from a different environment and in order to make a unanimous decision. Based on IoA framework, devices connected, such as home electrical appliances, negotiate to agree on an optimal way to exchange information. These agents can create multi-agent systems for communication and collaboration and provide an optimal way to effectively optimize energy in real time.⁴ Thus, consensual negotiation ensures dynamic scheduling through load shifting to reduce energy consumption.

Load balance

Load balance is a way to control energy usage and the rate of usage. Shivapur et al.¹⁸ indicated that load

balancing does not seek equal distribution of network load but on how to balance the load on a single node based on the status of the network. The method to achieve load balance is clustering. In clustering, sensor nodes are connected to form a cluster,¹⁹ where each cluster has a CH that connects another CH to form a network of clusters. CHs might have different initial energy requirements or the same initial energy requirement. Mostly, when CHs have the same energy requirement, it is referred to as a homogeneous network; otherwise, it is referred to as a heterogeneous network.

Benchmarked clustering algorithms for the homogeneous networks include low-energy adaptive clustering hierarchy (LEACH),²⁰ power-efficient gathering in sensor information systems (PEGASIS), and hybrid energy-efficient distributed clustering (HEED).²¹ However, Qing et al.²² indicate that the complication in network operation and configuration of connected devices are some of the challenges to devise an energy-efficient clustering algorithm for the heterogeneous network.²³ In spite of these challenges, some benchmarked clustering algorithms for heterogeneous networks have been proposed; these include stable election protocol (SEP) for two-level heterogeneous networks and for multi-level heterogeneous networks, distributed energy-efficient clustering (DEEC).²³

There are different versions of LEACH-like algorithms for clustering in both homogeneous and heterogeneous sensor network schemes.²² The DEEC model is based on a two-level heterogeneous network in which the sensor nodes are assumed to have a normal and advanced battery level. E_DEEC uses the idea of three levels of heterogeneous sensor networks. This is different from the DEEC model in the sense that the EDEEC model is based on a three-level heterogeneous WSN where sensor nodes are thought to have normal, advanced, and super-battery levels. DDEEC model utilizes the same network structure with other energy models like EDEEC.²⁴

On one hand, in comparison to benchmarked algorithms for homogeneous networks, the LEACH algorithm can select a cluster and periodically rotate cluster position, it maintains cluster hierarchy, and it could perform well in a homogeneous network. However, performance degrades on a heterogeneous network²³ because there is no inter-cluster communication on the network for each CH to send data directly to a sink/BS.²⁵ PEGASIS applies hierarchical routing which is challenged by high power usage, while HEED selects CHs by using stochastic technique.²²

On the other hand, in heterogeneous network, low energy devices often have different energy makeup, which should be considered when designing an energy-efficient model for heterogeneous network.

Comparatively, Uplap and Sharma²⁶ indicate that the drawback of a homogeneous network is that all

nodes on the network can act as a CH, which can lead to uniform energy drainage. Contrarily, a heterogeneous network increases the lifetime of the WSNs and often has lower hardware cost.

One of the major technical issues (like low throughput and less reliability) of routing algorithm is how to increase the lifetime of network without increasing energy consumption. This is because, in routing operation, some nodes may be over requested to relay information from other neighbors, causing faster energy dissipation that could generate coverage holes in its sensing area. Hence, it is desirable to have node balancing their energy loads in order to maximize the network lifetime.²⁷

Bio-inspired techniques

Bio-inspired techniques have also played a role in clustering. In a sense that it helps to find an optimal way to build a cluster by taking into consideration the energy makeup of devices and the distance among clusters with their respective nodes. In view of this, bio-inspired techniques are applied to clustering models for an efficient and dynamic energy load balance. Examples of bio-inspired techniques are genetic algorithms (GA), particle swarm optimization (PSO),²⁸ ant colony optimization (ACO), wolf search algorithm (WSA), and bat algorithm.

A GA is an evolutionary approach that is based on the survival of the fittest. This survival depends on the mechanism of “natural selection” (Darwin, 1868 as cited by Agbehadji et al.²⁹) where species considered as weak and cannot adapt to the conditions of the habitat are eliminated while species considered as strong and can adapt to the habitat survive. Thus, natural selection is based on the notion that strong species have a greater chance to pass their genes to future generations, while weaker species are eliminated by natural selection. Sometimes, there are random changes that occur in genes due to changes within the external environments of species, which will cause new future species that are produced to inherit different genetic characteristics. At the stage of producing new species, individuals are selected, at random, from the current population within the habitat to be parents and use them to produce the children for the next generation, thus successive generations are able to adapt to the habitat in respect of time.

PSO is a bio-inspired method based on swarm behavior such as fish and bird schools in nature.³⁰ The swarm behavior is expressed in terms of how particles adapt and make a decision depending on a change of position within a space based on the position of other neighboring particles. The advantage of swarm behavior is that as individual particle makes a decision, it leads to emergent behavior. This emergent behavior is based on local interaction among particles in order to

determine a potential solution with the highest momentum to find an optimal solution.

The ACO is bio-inspired by the foraging behavior of real ants in their search for the shortest paths to food sources.³¹ When a source of food is found, ants deposit pheromone to mark their path for other ants to traverse. Pheromone is an odorous substance that is used as a medium of indirect communication between ants. The quantity of pheromone depends on the distance, quantity, and quality of food source. However, the pheromone substance which decays or evaporates with time prevents ants from converging prematurely; thereby ants can explore other sources of pheromone substances within its habitat. In a situation where an ant is lost, it moves at random in search of a laid pheromone, and ants likely will follow the path that reinforces the pheromone trails. Thus, ants make probabilistic decisions on updating their pheromone trail and local heuristic information to explore larger search areas.

WSA is a bio-inspired heuristic optimization algorithm that is based on wolf preying behavior.³² The behavior of wolves includes the ability to hunt independently by remembering their own trait (meaning wolves have memory), ability to only merge with its peer when the peer is in a better position (meaning there is trust among wolves to never prey on each other), only attracts to prey within its visual range, ability to escape randomly upon appearance of a hunter, and the use of scent marks as a way of demarcating its territory and communicating with each other wolf. The multiple behaviors expressed by wolves enable it to randomly adapt to its environment when hunting. Thus, a better position replaces the old position. Each wolf instinctively flocks together in a pack, which indicates a collective behavior and organizes individual searches of an individual wolf. Therefore, the swarming behavior of WSA is delegated to each individual wolf, and this behavior could form multiple leaders swarming from multiple directions toward a point of convergences (that is the best solution) rather than as a single flock searching for an optimum in one direction at a time. The multiple behaviors of wolves can be used in defining multiple search criteria toward convergence into a global solution. The behavior of a wolf is implemented as an iterative search process that starts with the setting of an initial parameter, random initialization of population, evaluation and updating a current population using a fitness test, and continuing on with creating new generations/iterations until the stopping criteria is met. A variant of WSA is the WSA with Minus Step Previous (WSA-MP). The WSA-MP allows wolves to remember a previous best position and avoid the old positions taken which do not produce the best solution. The wolf behavior has been applied in several optimization problems to find the best optimal solution.

Bat algorithm is a bio-inspired method based on the behavior of micro-bats in their natural environment.³³ The unique behavior that characterizes bats is their echolocation mechanism. This mechanism helps bats orient and find prey within their environment. The search strategy of a bat is controlled by the pulse rate and loudness of their echolocation mechanism. While the pulse rate changes to improve the better position that was previously found, the loudness indicates to each other bat that the best position is accepted/found. The bat algorithm search process starts with random initialization of the population, evaluation of the new population using a fitness function, and finding the best population. Unlike the wolf algorithm that uses the attractiveness of prey to govern its search, the bat algorithm uses the pulse rate and loudness to control the search for an optimal solution. A variant of bat algorithm is sampling, improved bat algorithm (SIBA). The bat behavior has been applied in several optimization problems to find the best optimal solution.

In order to create a dynamic energy load balance, bio-inspired techniques explore and exploit different search areas to find an optimal way to form a cluster. The advantage of a bio-inspired technique is the ability to jump out of any local optimum that might not lead to an ideal cluster formation. In respect to this advantage, several energy clustering models are integrated with bio-inspired techniques for clustering so as to solve network energy load balancing problems. Example of such integration includes Energy-aware Clustering for WSNs using the PSO Algorithm;³⁴ Cluster-based WSN Routing using the ABC Algorithm;³⁵ Whale Optimization for clustering and energy optimization for WSN;³⁶ LEACH—Centralized with Simulated Annealing³⁵ and LEACH combine with GA in which LEACH optimizes energy and cluster sensor devices on a networks, whereas GA finds an optimal probability of a node selected to be a CH, thereby minimizing the total energy consumption of a network.³⁷ When dynamic energy load balance techniques are created, it provides an opportunity to control energy consumption needs. Bui and Jung³⁸ indicates that load balance models are synchronization-based negotiation models, which negotiates energy demand for IoT devices connected on the edge of wireless networks. Moreover, with negotiation models, the operation of each device is scheduled in real time so as to identify the load and status of devices either active, finish, wait, on, or off.³⁸

Chan and Han¹² proposed a mechanism to predict the energy consumption by a sensor node, which then constructs an energy map of a sensor network. In using this approach, sensor nodes do not need to transmit energy information periodically. Instead, it requests energy information and the model's parameter. This approach works well when the sensor's energy dissipation rate is relatively stable. However, the performance

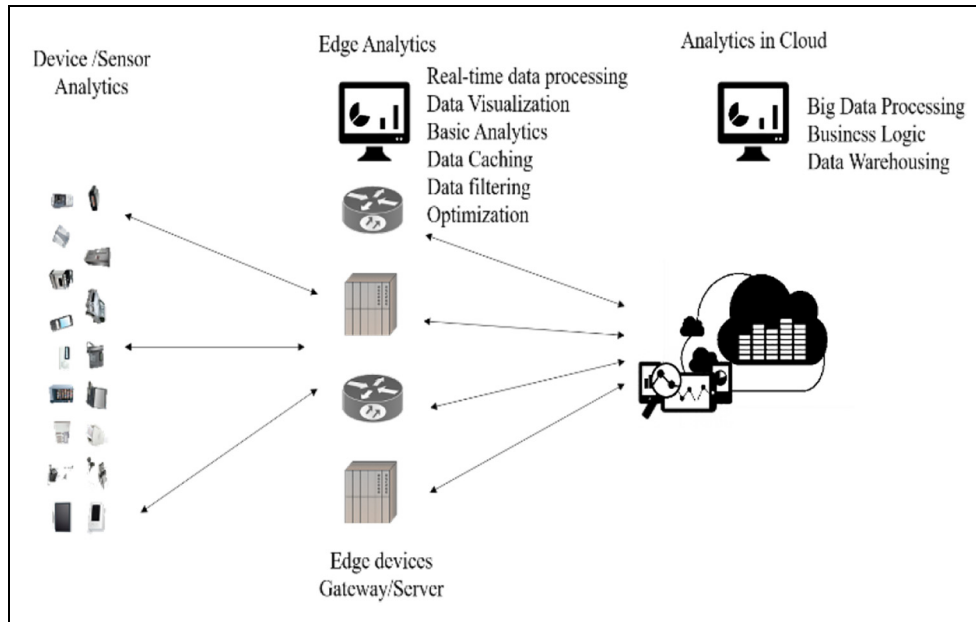


Figure 1. Edge analytics architecture.

Source: Adapted from Bui and Jung.³⁸

decreases along with an increase in the events' randomness within the network.

Bio-inspired distributed energy-efficient clustering (B-DEEC) algorithm based on the artificial bee colony (ABC) algorithm has also been proposed.²⁷ The waggle dance for multiple interactions and information sharing was utilized to optimize the CH selection for heterogeneous WSNs. In the B-DEEC algorithm, probabilistic CH selection was optimized by performing neighborhood search and nominating a node as CH with maximum energy. The B-DEEC increases both network lifetime and throughput of the network.

Edge computing

Edge computing performs real-time analytics to find the best way to send data to the cloud environment, instead of performing analytics directly on the cloud. Varghese et al.³⁹ indicate that when data analytics is done on the cloud, due to bottleneck in data transmission and reception, it creates an increased energy demand. The increase in energy demand can be minimized by incorporating energy management strategies so that analytical tasks are performed on the edge of networks via gateways or base stations (BS) that are closer to the source of data. Basically, the BS is a location for users to access data.

Generally, the advantage of edge computing is the efficient distributed computing such that when a sensor node dies or goes off quickly, because of workload in collecting, aggregating, and sending data to a BS, a different sensor node is selected as CH.²⁵

The edge analytics framework is one of the proposed frameworks for real-time data collection in WSNs. Figure 1 depicts the edge analytics architecture for edge devices (e.g. sensor-enabled) to perform analytics in real time or in nearby locations (e.g. ceilings).

Figure 1 shows the edge analytics architecture which consists of three analytics layers, namely device/sensor layer, edge analytics layer, and analytics in the cloud layer. First, the edge analytics layer is the layer that supports Fog computing.

Basically, Fog computing is a computing framework that liaises between the device/sensor layer and analytics in the cloud layer. Conceptually, Fog computing is an extension of cloud computing, to ensure real-time or near real-time data processing, optimization, and so on. Mostly, in Fog computing, data analytics is delegated to edge devices/gateway rather than being delegated to a central cloud server. In this regard, it reduces data transfer to the cloud computing environment and minimizes data analytics latency. Second, analytics in the cloud layer supports further data processing and storage particularly through the use of the Internet. Third, the device/sensor layer houses all data collection devices such as smart appliances, low powered devices that are connected to a single location mostly referred to as sink node or BS. It is significant to note that fog computing devices may not necessarily be at the edge of a network, but rather it may reside close to the edge of the network. Contrarily, edge devices reside on the edge of networks; therefore, it is often the first point of contact in IoT analytics. In essence, fog computing and edge computing are both close to the IoT end-devices, but

the edge computing devices are often closer.⁴⁰ In many related works, fog computing and edge computing have been used interchangeably.

Generally, the rise in the utilization of sensor devices and heterogeneity of sensor devices result in finding ways to improve the lifetime of WSNs. Furthermore, when different sensor devices are randomly placed in different locations to transmit data from one point to another, some amount of energy is dissipated.³⁶ Bio-inspired algorithms are important in edge computing as they provide an optimal way to adjust energy needs in real time without being stuck in the local optimum.

WSNs

WSN consists of sensor nodes equipped with the capability to sense, compute, and communicate with another sensor node. Routing algorithms are required when nodes are unable to send data to BSs.²³ The operation of WSN is such that, sensor nodes are grouped into clusters, where individual sensor node transmits data to CH in either single hops or multiple hops and then forwards the data to a BS or sink. The BS then runs a clustering algorithm and notifies all sensor nodes on a network of the best gateway for each sensor node to use.⁴¹ In WSNs, data are transmitted in real time and clustering algorithms continuously update their network status. Each iteration to update the network is referred to as a round.⁴² Sensor devices have different parts that consume energy, namely the microcontroller processing, radio transmission and receiving, transient energy, sensor sensing, and sensor logging and actuation. In addition, the location of sensor nodes in a cluster, which leads to having different transmit distance to a CH, also consumes energy. The energy spent by neighboring sensor nodes in transmitting and receiving (that is communication energy) data plays an important role in WSNs (see Figure 2). In WSN, energy consumption in data transmission often reduces and this affects the overall performance of a network, stability of network, and efficiency of information transmission.⁴³ In this regard, the energy-efficient clustering algorithm in WSN is significant. The description of WSN is depicted in Figure 2.¹⁸

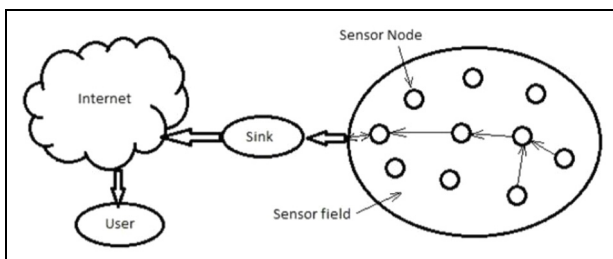


Figure 2. Structure of a wireless sensor network.

The sink node shown (see Figure 2) represents the edge device gateway of the edge analytics architecture (see Figure 1). The communication among sensor nodes is achieved by using sensor radio, and when distance increases the energy dissipation for the communication leads to an increase in the cost.⁴² In respect of optimality, an optimal cluster for a number of sensor nodes is based on distance and energy dissipation per round. In this regard, an optimal number of clusters is defined as the number of clusters that reduces energy dissipation.⁴²

In general, the following are excerpts from the related works: heterogeneous network might increase the lifetime of WSN; it is desirable to have node balancing their energy loads in order to maximize the network lifetime; clustering algorithms continuously update their network status; the location of sensor nodes in a cluster leads to different transmit distance to a CH which also consumes energy; bio-inspired algorithms are important because they provide an optimal way to adjust energy needs in real time without being stuck in the local optimum.

As technology advances, newly developed sensor-enabled devices should be able to interact with existing network infrastructure. Therefore, it is important to develop new algorithms to support device interaction without having to utilize more energy in forming clusters and transmitting data. In this article, a novel intelligent energy-efficient model based on the behavior and characteristics of a bird, called Kestrel, is proposed for clustering on WSN.

Proposed model

This article proposes a novel clustering scheme exploiting the KSA to optimize energy in heterogeneous environments. The approach to optimizing energy is based on the behavior and characteristics of birds/animals mostly referred to as bio-inspired approach. A bio-inspired approach is applied because of its distributed cooperative nature that enables the processing of data from heterogeneous environments. In a distributed environment, bio-inspired search strategies can apply randomization and efficient local and global search to achieve a new optimal solution.⁴⁴ The bio-inspired approach can help form basic rules that aim at some level of intelligent by providing an optimal way to adjust energy needs.

Our proposed model focuses on receiving sensors' location, their energy level (that is when devices in a cluster do not have the same initial energy)²³ and a number of sensors. The proposed approach uses the idea of having a BS (that is, end-user), which computes average energy levels such that sensors with energy below specified average energy are not eligible to be

selected as a CH. A cluster member is then identified as a CH based on the position, energy consumption, and data transfer with each cluster member. When CH is selected, it acts as a local controller to coordinate the data transmission among each cluster member and avoids a collision that might occur in sending data from an active device (e.g. appliance). The BS is located at a given coordinate, whereas the number and locations of sensors are optimized.⁴⁵ Thus, only sensors with an energy level above the average energy are eligible to be candidates for a CH position. In order to assign sensors to a cluster, the BS considers energy level and the distance between a sensor and selected CHs. The CH plays a significant role in setting up Time Division Multiple Access (TDMA) schedule which helps to avoid collision among sensors and only allows devices to be turned on when it wants to send data, thus reducing the energy consumption.³⁵ The application of TDMA mechanism for energy saving builds a time slot for each device by dividing the time into the various size of a fixed number of slots, each slot unit lasts at some time period.

Proposed clustering algorithm based on KSA

The KSA is based on characteristics, namely random encircling, trail evaporation based on half-life period, position, and velocity of Kestrels.⁴⁴ Although the algorithm has been applied to different problem domains such as missing value estimation,⁴⁴ association rule mining,⁴⁶ and feature selection in classification.^{29,47} In this study, we applied the KSA to clustering in the case of heterogeneous energy requirements. The aim of clustering using the Kestrel is to optimize energy consumption by balancing nodes on the edge computing network. Basically, Kestrel achieves this optimization by changing position, velocity, and trail evaporation and ensures randomness.

The KSA starts by initialing a set of random Kestrels at the set-up phase of first round/iteration to determine the energy requirements of heterogeneous devices and find an optimal parameter. Devices are said to be heterogeneous because each has a different energy requirement.

The position of KSA is expressed as follows

$$x_{i+1}^k = x_i^k + \beta_o e^{-\gamma r^2} (x_j - x_i) + f_i^k \quad (1)$$

where x_{i+1}^k is the current best position of a Kestrel that represents a candidate solution; x_i^k is the previous position of Kestrel based on random encircling formulation;⁴⁴ $\beta_o e^{-\gamma r^2}$ is the attractiveness which indicates the light reflected from a trail, where the variable β_o represents initial attractiveness, r represents distance measurements expressed using Minkowski distance,⁴⁴ γ represents a variation of light intensity between [0, 1]; x_j represents a Kestrel with a better position; and f_i^k is

the frequency of bobbing as expressed by Agbehadji et al.⁴⁴ The random encircling formulation is expressed by

$$\vec{x}(t+1) = \vec{x}_p(t) - \vec{A} * \vec{D} \quad (2)$$

Thus

$$\vec{A} = 2 * \vec{z} * r_2 - \vec{z} \quad (3)$$

$$\vec{D} = |\vec{C} * \vec{x}_p(t) - \vec{x}(t)| \quad (4)$$

$$\vec{C} = 2 * r_1 \quad (5)$$

where \vec{A} is the coefficient vector, \vec{D} is the encircling value obtained, $\vec{x}_p(t)$ is the position vector of the prey, and $\vec{x}(t+1)$ represents the previous position of Kestrels. Where \vec{C} is the coefficient vector, $\vec{x}(t)$ indicates the position vector of a Kestrel, and r_1 and r_2 are the random numbers generated between 0 and 1, and as Kestrels shift the center of encircling, it maximizes the chances of locating its prey hence the constant value of 2. \vec{z} represents a parameter to control the active mode with \vec{z}_{hi} as the parameter for flight mode and \vec{z}_{low} as the parameter for perched mode, which linearly decreases from 2 (high active mode value) to 0 (low active mode value), respectively, during the iteration process. This is expressed as

$$\vec{z} = \vec{z}_{hi} - (\vec{z}_{hi} - \vec{z}_{low}) \frac{itr}{Max_itr} \quad (6)$$

where itr is the current iteration and Max_itr is the total number of iterations that are performed during the search. Other Kestrels that are involved in the search update their position according to the best position of the leading Kestrel.

Finally, the velocity of Kestrel is updated by

$$v_{t+1}^k = v_t^k + x_t^k \quad (7)$$

where v_{t+1}^k is the current best velocity, v_t^k represents the initial velocity, while x_t^k represents the current best position of a Kestrel.

Trail evaporation. In meta-heuristic algorithms, ants use trails both to trace the path to a food source and to prevent themselves from getting stuck in a single food source. Thus, ants, using these trails, can search many food sources in a search space. As ants continue to search, trails are drawn and pheromones are deposited on a trail. This pheromone helps ants to communicate with each other about the location of food sources. Therefore, other ants continuously follow this path and also deposit substances for the trail to remain fresh. Similar to ants, Kestrels use trails in search of food sources. However, these trails are rather deposited by

prey which provides an indication to Kestrels on the availability of food sources. The assumption is that the substances deposited by prey are similar to pheromone deposited on ants' pheromone trail. In addition, when the source of food depletes, Kestrels no longer follow this path that leads to the location of prey. Consequently, the pheromone trail begins to diminish with time at an exponential rate causing trails to become old.^{44,47} This diminishment denotes the unstable nature of the trail substances which can be theoretically stated as if there are N unstable nodes (that is, different energy requirements), then the rate at which the "substance" decays with time t is expressed by

$$\frac{dN}{dt} = -\gamma N \quad (8)$$

Thus, the decay rate (γ) with time (t) is simplified as

$$\gamma_t = \gamma_o e^{-\varphi t} \quad (9)$$

where γ_o represents initial value and t is the time of decay. The decay constant φ which shows how long it takes for a "substance" to decay is re-expressed as

$$\varphi = \frac{\ln 0.5}{-t_{\frac{1}{2}}} \quad (10)$$

where φ is the decay constant and $t_{\frac{1}{2}}$ is the half-life period. If the value of decay constant is greater than 1, then the trail is considered as new else the trail is considered as old, which is expressed by

$$\text{if } \varphi \rightarrow \begin{cases} \varphi > 1, \text{ trail is new} \\ 0, \text{ otherwise} \end{cases} \quad (11)$$

where φ is the decay constant.

In WSNs, the heterogeneity of devices on the edge of networks makes it impossible for all nodes to go off at the same time. Similarly, it could be said that each node has its own half-life. In this regard, the decay rate

of nodes on the network can be determined by applying the half-life formulation. Moreover, as nodes are heterogeneous, a degree of randomness is introduced which is accounted for by the decay process.

Heterogeneous network model for energy optimization

The energy model finds energy dissipated when sensor node transfers or receives data on the network. In this article, the DEEC model is adopted and integrated with the behavior and characteristics of Kestrel formulation. In the transfer of data, some amount of energy is dissipated, and to optimize this energy, the radio energy dissipation model³⁷ shown (see Figure 3) is applied.

In this model, the energy required by the transmit amplifier $E_{TX}(l, d)$ to transmit an l -bit message over a distance d between a transmitter $E_{TX}(l, d)$ and receiver $E_{RX}(l)$ is expressed by

$$E_{TX}(l, d) = \begin{cases} l * E_{elec} + l * \epsilon_{fs} * d^2 & \text{if } d \leq d_o \\ l * E_{elec} + l * \epsilon_{mp} * d^4 & \text{if } d \geq d_o \end{cases} \quad (12)$$

where d_o is the threshold transmission distance between the transmitter and receiver which is expressed as $d_o = \sqrt{\epsilon_{fs}/\epsilon_{mp}}$. E_{elec} is the energy consumption in electronics for sending or receiving a bit, and ϵ_{fs} and ϵ_{mp} represent amplifier parameters for free state and two ray models, respectively. d^2 and d^4 refers to short- and long-distance transmissions, respectively. When an l -bit packet is received, the energy required by the receiver $E_{RX}(l)$ is expressed by

$$E_{RX}(l) = l * E_{elec} \quad (13)$$

In this section, we described the energy model for a multi-level heterogeneous network. In the multi-level heterogeneous model, the energy of sensor nodes is randomly distributed in size of $M \times M$ meters. In this context, the total initial energy for all sensor node is expressed by

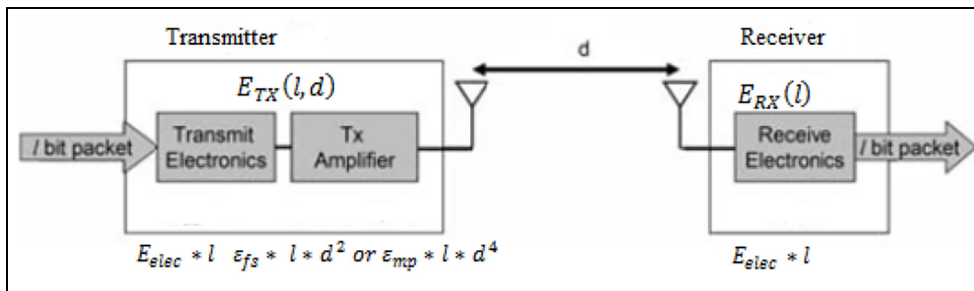


Figure 3. First-order radio model.

$$E_{total} = \sum_{i=1}^n E_o(1 + a_i) = E_o \left(\left(n + \sum_{i=1}^n a_i \right) \right) \quad (14)$$

where E_o represents the initial energy of a node; in this context, a node has initial energy $E_o(1 + a_i)$ which is a_i times more than the lower/initial bound E_o .

When a cluster is formed, each CH dissipates energy in receiving a signal from nodes and then aggregates the signals and transmits the aggregate signal to the BS which is far from nodes. Thus, a cluster-head should have enough energy to reach a BS. The energy of the cluster is expressed as

$$E_{CH} = l * \frac{n}{k} * (E_{elec} + E_{DA}) + l * \varepsilon_{mp} * d_{toBS}^4 \quad (15)$$

where E_{CH} represents the energy of a CH, n is the number of sensor nodes, k is the number of clusters, E_{elec} is the transmitter electronics, E_{DA} is the energy for aggregating data, l is the data packet, and ε_{mp} transmits amplifier in long-distance d_{toBS}^4 to a BS. Similarly, the energy dissipated by non-CHs E_{n-CH} is expressed by

$$E_{n-CH} = l * E_{elec} + l * \varepsilon_{fs} * d_{toCH}^2 \quad (16)$$

where l is the data packet, E_{elec} is the transmitter electronics, and ε_{fs} represents transmitter amplifier for the free state in short-distance d_{toCH}^2 to CH. The distances, both short and long distances, are expressed by²²

$$d_{toBS}^4 = \frac{M}{\sqrt{2\pi k}}, d_{toCH}^2 = 0.765 \frac{M}{2} \quad (17)$$

where M is the size of sensor field, k is the number of clusters, d_{toCH}^2 is a short distance to CH, and d_{toBS}^4 is the long distance to the BS. Thus, total energy E_{total} dissipated by a cluster is expressed by

$$E_{Cluster} = E_{CH} \left(\frac{n}{k} - 1 \right) E_{n-CH} \approx E_{CH} + \frac{n}{k} E_{n-CH} \quad (18)$$

$$E_{total} = k * E_{Cluster} \quad (19)$$

where $E_{Cluster}$ is the energy dissipated by a cluster among cluster members, E_{CH} represents the energy of CH, n is the number of sensor nodes, k is the number of clusters, and E_{n-CH} is the energy dissipated by non-cluster. In addition, each non-CH sends l -bits message to the CH in a round; therefore, total energy dissipated in the network during a round is expressed by

$$E_{round} = l * (2 * n * E_{elec} + n * E_{DA} + k * \varepsilon_{mp} * d_{toBS}^4 + n * \varepsilon_{fs} * d_{toCH}^2) \quad (20)$$

where E_{round} represents energy at a round, l is the data packet, n is the number of sensor nodes, E_{elec} is the transmitter electronics, E_{DA} is the energy for aggregating data, ε_{mp} is the transmitter amplifier in long-distance d_{toBS}^4 to a BS, ε_{fs} is the transmitter amplifier for the free

state in short-distance d_{toCH}^2 to CH, and k is the number of clusters. The BS and CH are always located at a distance, and due to randomization, the distance to send a data packet is always computed in each round/iteration. In view of this, the threshold transmission distance d_o and d are compared to find the distance to send packets to BS and CH.

In respect of clusters, the optimal number of clusters k_{opt} which replaces k is expressed as

$$k_{opt} = \frac{\sqrt{n}}{\sqrt{2\pi}} * \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}} * \frac{M}{d_{toCH}^2} \quad (21)$$

where n is the number of sensor nodes, ε_{mp} is the transmitter amplifier, ε_{fs} represents transmitter amplifier for the free state in short-distance d_{toCH}^2 to CH, and M is the size of sensor fields.

It is important to select an optimal CH; therefore, a probability threshold $T(s_i)$ is applied to determine the optimal CH in a round. If the probability is less than a threshold $T(s_i)$ value, the node is selected as a CH for that round. $T(s_i)$ is expressed by

$$T(s_i) = \begin{cases} \frac{p_i}{1-p_i \left(r \text{mod} \frac{1}{p_i} \right)} & \text{if } s_i \in G \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

where p_i is the user set probability for a CH, r represents the current round, and G is the set of nodes that have not been selected as CHs in the previous $1/p_i$ rounds. Thus

$$p_i = \frac{p_{opt} n(1+a) E_i(r)}{\left(n + \sum_{i=1}^n a_i \right) \bar{E}(r)} \quad (23)$$

where p_{opt} represents the reference value of the average probability of p_i , n is the number of nodes, $E_i(r)$ is the residual energy, and $\bar{E}(r)$ is the estimated energy that serves as standard reference energy for each node. This reference energy indicates that each node has its own energy in each round to keep the network alive, and this introduces some heterogeneity on the network. Moreover, in a heterogeneous network, it is important to ensure that there is enough energy for data transmission. In view of this, both initial energy and residual energy level of nodes are used to select cluster-heads at each round. Since nodes have different energy requirements, the network identifies the best node base on the average energy $\bar{E}(r)$ at a round of the network which is computed by

$$\bar{E}(r) = \frac{1}{n} * E_{total} \left(1 - \frac{r}{R} \right) \quad (24)$$

where R is the total network lifetime. The assumption for considering network lifetime is that should all the

nodes die simultaneously, R is the total of rounds from the time the network begins to time the node dies.²² Furthermore, network lifetime can be categorized into two periods such as stable and unstable periods. Whereas, the stable period refers to the period from the beginning of a transmission to the period the first node dies and the unstable period refers to a period from the death of the first node till the death of the last node on the network.²⁸ Therefore, the energy consumed by the network in each round is denoted by E_{round} , and therefore, R is expressed by

$$R = \frac{E_{total}}{E_{round}} \quad (25)$$

where E_{total} is the total energy.

Objective function

In order to select a node with an adequate amount of energy in a cluster, the objective function $ofunc$ is expressed as

$$ofunc = \alpha * f_1 + (1 - \alpha) * f_2 \quad (26)$$

$$f_1 = \max_{i \in [1, K]} \left(\sum_{s_i \in C_k} \frac{d(sensor_i, CH_k)}{|C_k|} \right) \quad (27)$$

$$f_2 = \frac{\sum_{i=1}^N E(sensor_i)}{E(CH_k)} \quad (28)$$

where α is a user-defined parameter between 0 and 1, N is the number of sensors, $|C_k|$ is the number of sensors that belong to a cluster C_k , f_1 is the maximum average distance of sensors and their CHs, and f_2 is the ratio of the total initial energy of all sensors with the total current energy of CH in a round, where i be the amount of energy consumed from i to N . If the value of the objective function is less, the node becomes a cluster-head for the current round.

In respect of the DEEC, the fitness function is expressed by equation $T(s_i)$, similar to Ari.³⁵ It is ideal to have the same fitness function for each proposed and comparative algorithm. However, in this study, divergent fitness functions were applied.

Proposed algorithm

The algorithm to implement the proposed solution is as follows:

1. Set the model parameters.
2. Initialize energy for all sensor nodes using equation (14)

3. Initialize population of n Kestrels using equation (2) and evaluate objective function using equation (26)
4. Start iteration (loop until termination criteria is met)
 - Compute Half-life of trail using equation (9)
 - Compute position for each Kestrel using equation (1)
 - Calculate the energy required by the transmit amplifier $E_{TX}(l, d)$ using equation (12) and compute the energy required by the receiver $E_{RX}(l)$ using equation (13)
 - Compute energy for the sensor node in next round using equation (14)
 - Compute probability threshold $T(s_i)$ to find the optimal CH in a round using equation (22)
 - Evaluate objective function using equation (26)
 - If $ofunc_i < ofunc_j$ then
 - Move Kestrel i toward j
 - End if
 - Update position of Kestrel
 - Find the optimal energy
5. End loop
6. Display results of optimal energy

The flowchart of the proposed algorithm is shown in Figure 4:

Parameter setting for simulation of network

The parameters for KSA are $zmin = 0.2$ (that is, parameter for perched mode) and $zmax = 0.8$ (that is, parameter for flight mode).⁴⁴ Following the network parameter settings by Jadhav and Shankar,³⁶ the network parameter for the energy model is set. Initially, the network load is 100 nodes, the transmitter electronics is set as 5 nJ/bit, the initial energy requirement is between 0.5 and 0.8 J, data aggregation is 5 nJ/bit/message, transmitter amplifier in long distance is 0.0013 pJ/bit/m⁴, while the transmitter amplifier in short distance is 10 pJ/bit/m², data packets size is 4000 and the network coverage in terms of size is 100 m × 100 m. $Popt$ represents the optimum probability which is set to 0.1. The network parameter settings are summarized in Table 1.

The underlying assumption for the proposed model is that all the nodes can communicate with the BS directly. The BS, which has a continuous energy supply, sends a request to all nodes in the sensor network, requesting them to collect residual energy. We assume that the sensor network consists of N sensor nodes of size m by m . The positions of the sensor nodes are

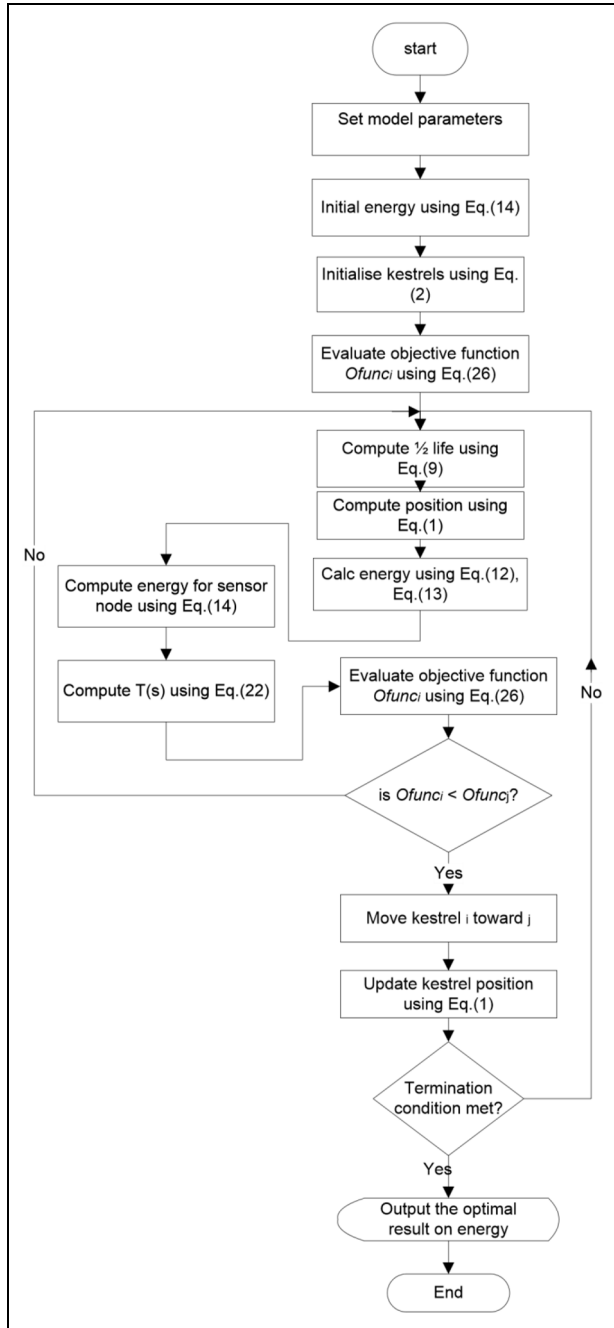


Figure 4. Flowchart on proposed algorithm.

generated randomly. Each node has a different energy requirement. The sensor network consumes energy according to the proposed DEEC-KSA model.

The proposed DEEC-KSA model consists of the following process: determination of clusters using random encircling, evaluation of the fitness of each encircled position, and determination of energy consumption. There are different optimal parameter for KSA, whereas the comparative algorithms used optimal parameter (p_{opt}) defined in Table 1. The network performance is evaluated based on when the first node and

Table 1. Parameter settings.

Parameter	Values
Number of nodes	100
Transmitter electronics E_{elec}	5 nJ/bit
Initial energy E_o	0.5–0.7 J
Data aggregation E_{DA}	5 nJ/bit/message
Transmitter amplifier ϵ_{mp} if $d \geq d_o$	0.0013 pJ/bit/m ⁴
Transmitter amplifier ϵ_{fs} if $d \leq d_o$	10 pJ/bit/m ²
Data packet size l bit	4000
Network size	100 m × 100 m
p_{opt}	0.1

KSA: Kestrel-based search algorithm.

the last node die, network lifetime, and network throughput in terms of packets sent to BS.

The first node death (FND) is the number of rounds in the network until the first node has depleted its energy and died.⁴⁸ In WSNs, performance tends to decline with the nodes' death. Normally, the network is in a stable period before the first node dies. The death of the first node indicates that the network is in an unstable state; hence, the performance of the network starts to decline.⁴³ However, the last node dead is the number of rounds in the network until all nodes in the network had depleted their energy and died. Therefore, network stability is categorized into stable and unstable periods. The stable period is a period from the beginning of a transmission to the period the first node dies, and an unstable period is a period from the death of the first node till the death of last node on the network. Network lifetime is the number of alive nodes on the network. Network throughput is defined as the number of data packets successfully received at BS. In other words, it is expressed as the number of packets sent to BS minus the number of packets dropped.^{49,50}

Simulation results

In this section, we present the simulation result of the proposed DEEC-KSA algorithm and compare it with existing clustering algorithms, namely DEEC, developed distributed energy-efficient clustering (DDEEC), an extended version of distributed energy-efficient clustering (E_DEEC) with normal, advance, and super node classifications.

A clustering algorithm is evaluated in terms of stability of network on FND and the number of rounds in the network until all nodes deplete their energy and died (last node death (LND)), network lifetime, and network throughput in terms of packets sent to BS.

Comparison of FND

In WSNs, the network is in a stable period before the first node dies. When the first node dies, then network

Table 2. Alive nodes during network lifetime for 100 nodes using 0.5 J.

Algorithms	Popt	FND	Tenth node dead (tenth_dead)	LND	Packets to BS	Time (s)
DEEC-KSA	0.001	2004	2093	3909	291,792	22.685
DEEC	0.1	937	1211	3902	66,876	46.708
DDEEC	0.1	1083	1340	3096	53,164	59.243
E_DEEC	0.1	768	1088	0	216,284	33.349

FND: first node death; BS: base stations; DEEC: distributed energy-efficient clustering; DDEEC: developed distributed energy-efficient clustering; E_DEEC: extended version of distributed energy-efficient clustering; LND: last node death; KSA: Kestrel-based search algorithm.

Table 3. Alive nodes during network lifetime for 100 nodes (initial energy of 0.6 J).

Algorithms	Popt	FND	Tenth node dead (tenth_dead)	LND	Packets to BS	Time (s)
DEEC-KSA	0.0455	2291	2534	4470	336740	18.613
DEEC	0.1	1311	1480	4118	79411	53.425
DDEEC	0.1	1264	1505	3813	62795	67.435
E_DEEC	0.1	923	1143	0	238121	38.071

FND: first node death; BS: base stations; DEEC: distributed energy-efficient clustering; DDEEC: developed distributed energy-efficient clustering; E_DEEC: extended version of distributed energy-efficient clustering; KSA: Kestrel-based search algorithm.

Table 4. Alive nodes during network lifetime for 100 nodes (initial energy of 0.7 J).

Algorithms	Popt	FND	Tenth node dead (tenth_dead)	LND	Packets to BS	Time (s)
DEEC-KSA	0.11361	2591	2774	0	380,894	19.260
DEEC	0.1	1468	1598	4577	96,109	55.514
DDEEC	0.1	1257	1636	4060	102,285	76.603
E_DEEC	0.1	1141	1459	0	281,118	38.921

FND: first node death; BS: base stations; DEEC: distributed energy-efficient clustering; DDEEC: developed distributed energy-efficient clustering; E_DEEC: extended version of distributed energy-efficient clustering; KSA: Kestrel-based search algorithm.

performance tends to decline, which results in an unstable period. The results are presented in Tables 2–5 for heterogeneous initial energy values between 0.5 and 0.8 J. Based on the result presented in Table 2, it shows the performance of FND for the proposed DEEC-KSA, DEEC, DDEEC, and E_DEEC as 2004, 937, 1083, and 768, respectively. Similarly, it is observed that the FND of DEEC, DDEEC, and E_DEEC are completely different from the proposed DEEC-KSA as observed in Tables 2–5.

In order to observe the performance of the proposed DEEC-KSA in selecting an optimal cluster-head, we chose to use different *Popt* parameters that are randomly generated, whereas the comparative algorithms have the same *Popt* parameter. In terms of percentage of a parameter, the proposed DEEC-KSA varies from 0.1%, 4.5%, 11.3%, and 34% while the percentage of the parameter of comparative algorithms maintained at 10%.

In the context of LND, the proposed DEEC-KSA has the highest (3909), DEEC is second (3902), DDEEC is third (3096), and E_DEEC is fourth. The

proposed DEEC-KSA has an advantage in delaying the round of the FND after several rounds of iteration thereby extending the death of the first node. It also retains a higher number of active nodes in the final round of iteration in Tables 2 and 3. However, in Tables 4 and 5, the number of round on the network is zero, meaning their energy is depleted and have died in respect of the proposed DEEC-KSA. Similarly, among Tables 2–5 for E_DEEC, the number of round on the network is zero, meaning their energy is depleted. Again, in Table 5, it is observed that DEEC has one round on the network. It is possible that as nodes with different energy tend to send a higher number of packets to BS, it depletes the energy within different time(s).

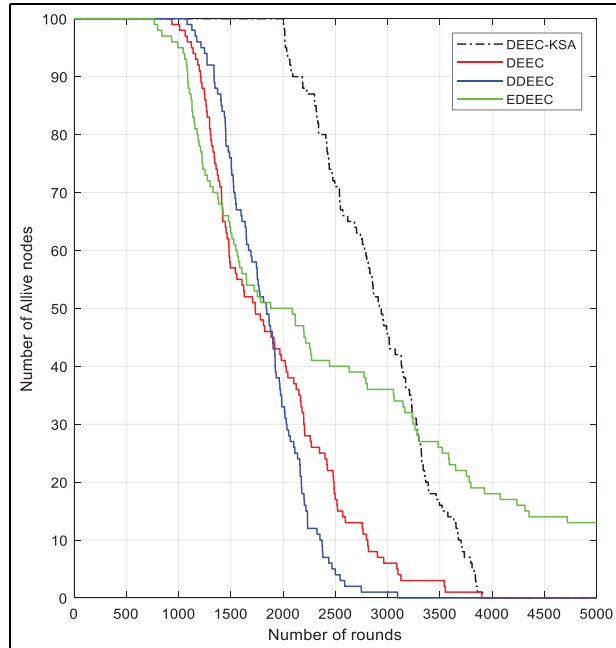
Comparison of network lifetime performance

Figure 5 shows the graphical display of alive nodes in each round of iteration for 0.5 J. Further simulation is performed using 100 nodes with heterogeneous initial energy between 0.6 and 0.8 J, and simulation result is presented in Figures 6 to 8. It is observed in Figure 5

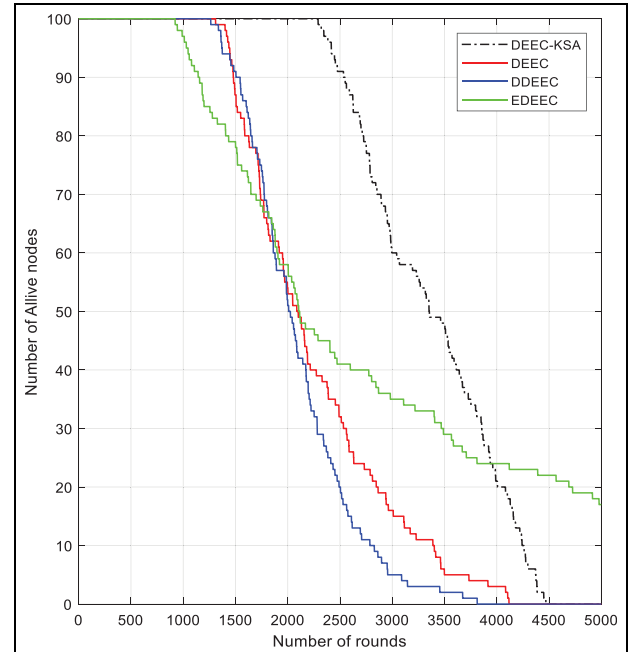
Table 5. Alive nodes during network lifetime for 100 nodes (initial energy of 0.8 J).

Algorithms	Popt	FND	Tenth node dead (tenth_dead)	LND	Packets to BS	Time (s)
DEEC-KSA	0.33625	2770	3178	0	372,660	19.306
DEEC	0.1	1628	1787	1	118,733	70.378
DDEEC	0.1	1682	2093	0	121,266	85.099
E_DEEC	0.1	1144	1594	0	303,156	37.256

FND: first node death; BS: base stations; DEEC: distributed energy-efficient clustering; DDEEC: developed distributed energy-efficient clustering; E_DEEC: extended version of distributed energy-efficient clustering; KSA: Kestrel-based search algorithm.

**Figure 5.** Alive nodes per round for 100 nodes with 0.5 J.

that DEEC-KSA has 100 nodes in the 2000th round, while EDEEC has 100 nodes close to the 750th round, DEEC has 100 nodes in the 1000th round, and DDEEC has 100 nodes in the 1000th round. Although all the clustering algorithms have 100 nodes in different rounds, the proposed DEEC-KSA performs better by retaining 100 nodes up to 2000th round. Nodes in the comparative algorithms reduce close to 1000th round of iteration before nodes in DEEC-KSA. In comparison to network lifetime performance, it is observed that DEEC-KSA achieved better performance as compared with the three other algorithms. This could be attributed to the ability to adequately perform random encircling and trail evaporation at each round. DDEEC network lifetime performance is second, DEEC is third, while E_DEEC is fourth. This indicates that the proposed DEEC-KSA ensures more round of network communication in all cases of simulation (see Figures 6 and 7) as compared with other algorithms that have different energy requirements.

**Figure 6.** Alive nodes per round for 100 nodes (initial energy of 0.6 J).

Comparison of the network throughput

In WSN, network throughput is fundamental to evaluate the efficiency of algorithms. It refers to a number of data packets in the network successfully sent at BS. As cluster member node sends information in the form of packets to CH, and the CH fuses the information it sensed and finally sends to BS as packets. During this period, if the energy of the CH is insufficient to receive, fuse, or transmit the packets, all the information of the cluster in the round is not transmitted to the BS, resulting in a decrease in network throughput. Simulation is performed using 100 nodes with heterogeneous initial energy between 0.5 and 0.8 J, and the simulation result is presented in Figures 9 to 12. Based on Figure 9, the number of packets sent successfully to BS using DEEC-KSA, E_DEEC, DEEC, and DDEEC is, respectively, 2.8×10^5 , 2.2×10^5 , 0.7×10^5 , and 0.52×10^5 , with respect to the number of rounds. The result shows

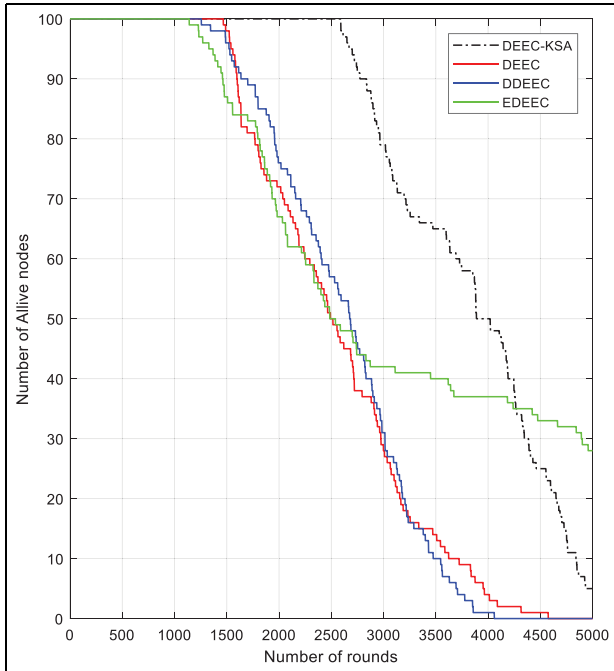


Figure 7. Alive nodes per round for 100 nodes (initial energy of 0.7 J).

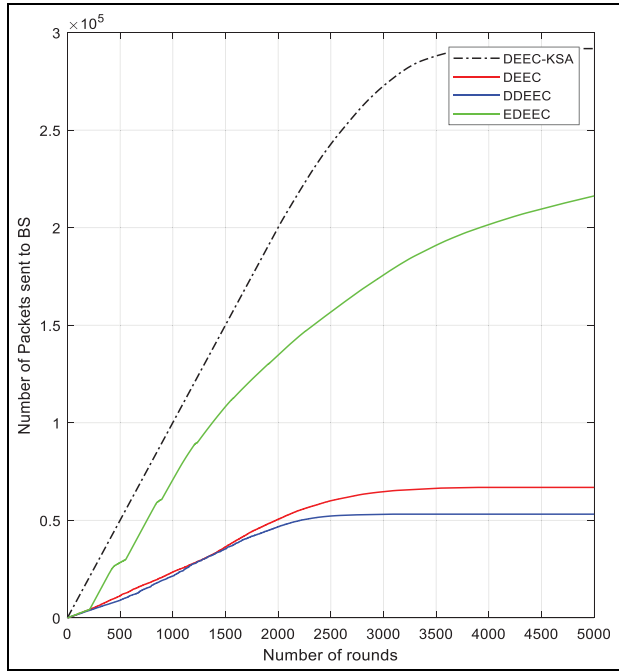


Figure 9. Packets sent to BS per round with 0.5 J.

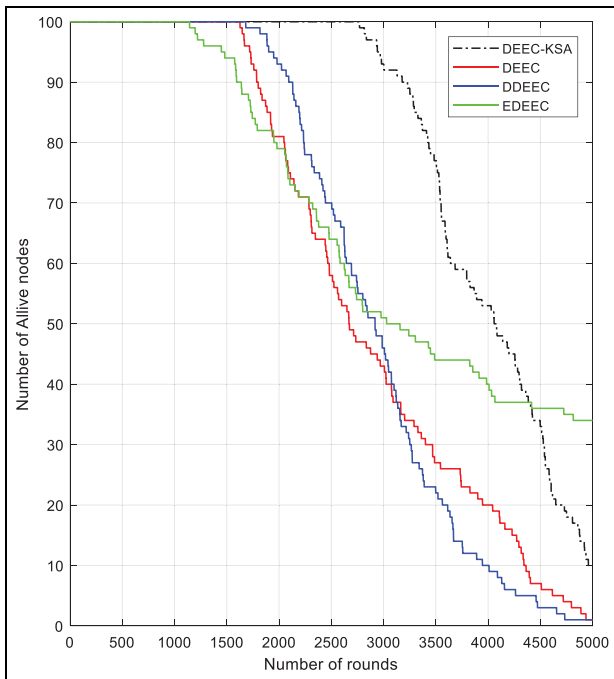


Figure 8. Alive nodes per round for 100 nodes (initial energy of 0.8 J).

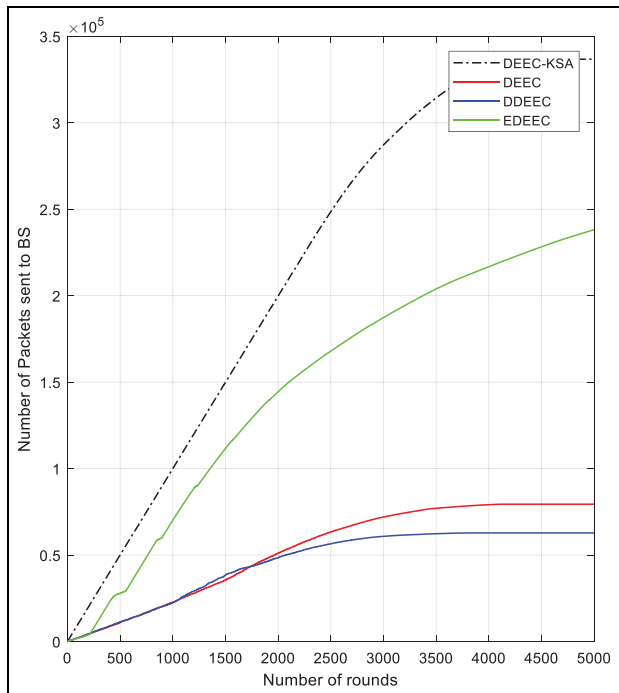


Figure 10. Packet sent to BS on 100 nodes (initial energy of 0.6 J).

that DEEC-KSA has the highest network throughput in all cases of simulation results (Figures 10 to 12).

In respect of time, the simulation results in Tables 2–5 indicate that the proposed DEEC-KSA has the least

time to send successful packets to BS, E_DEEC is second, DEEC is third, and DDEEC is fourth in terms of time to send successful packets. This suggests that the proposed DEEC-KSA is the efficient clustering

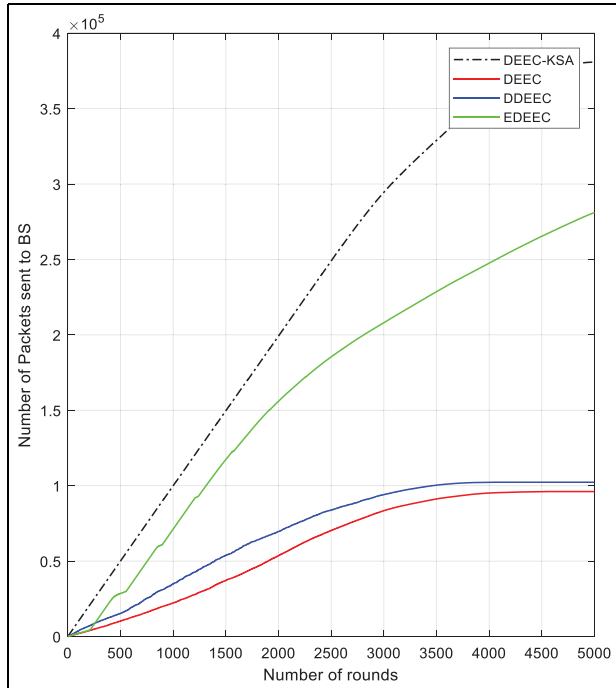


Figure 11. Packet sent to BS on 100 nodes (initial energy of 0.7 J).

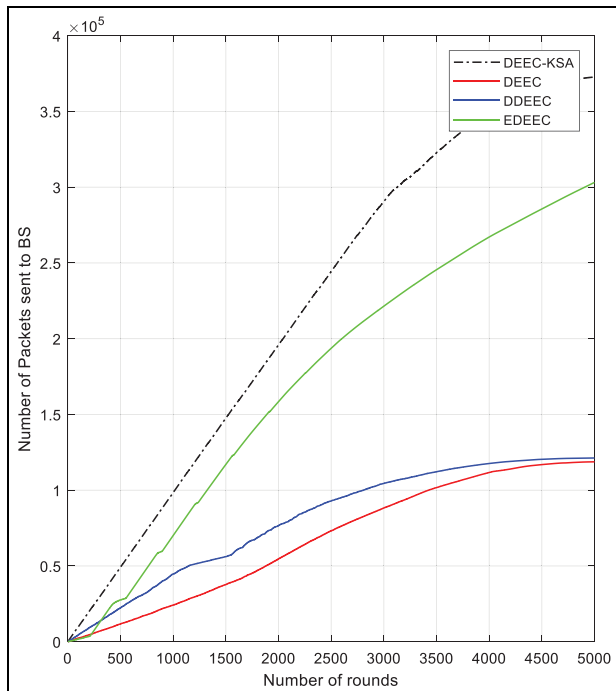


Figure 12. Packet sent to BS on 100 nodes (initial energy of 0.8 J).

algorithm since it spent less time to send a higher number of packets to BS. It is ideal that efficient clustering algorithms spend less time to send a higher number of

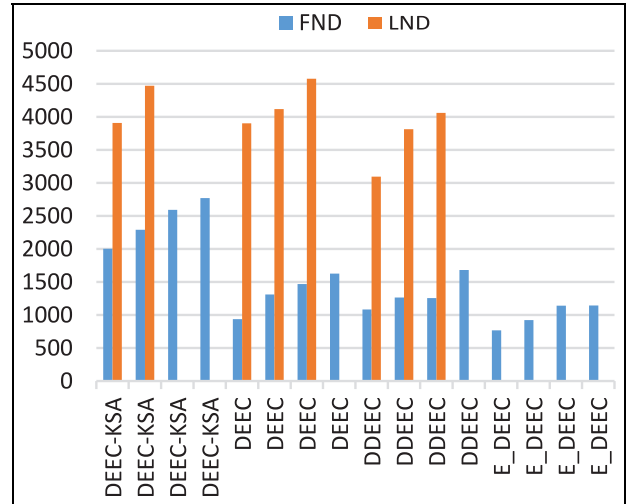


Figure 13. Network lifetime showing FND and LND based on 100 nodes.

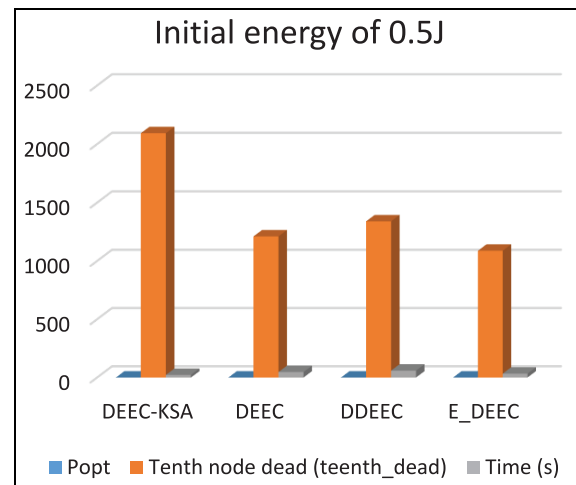


Figure 14. Network lifetime based on 100 nodes with initial 0.5 J.

packets thereby reducing the energy consumption in WSNs.

Based on the simulation result obtained and subsequently presented in Tables and Figures, once a node runs out of its energy, it is considered to be dead and it can no longer transmit or receive any data. Thus, simulation ends when all the nodes in the network run out of their energy. High energy efficiency means low energy consumption and a long stability period. From the simulation results, it is evident that network lifetime (in round) increases in DEEC-KSA with different initial energy between 0.5 and 0.8 J as shown in Figure 13.

Figures 13 to 17 show the Tenth node dead (teenth_dead) and *Popt* and time(s) to send successful packets. It is evident that the proposed DEEC-KSA is the

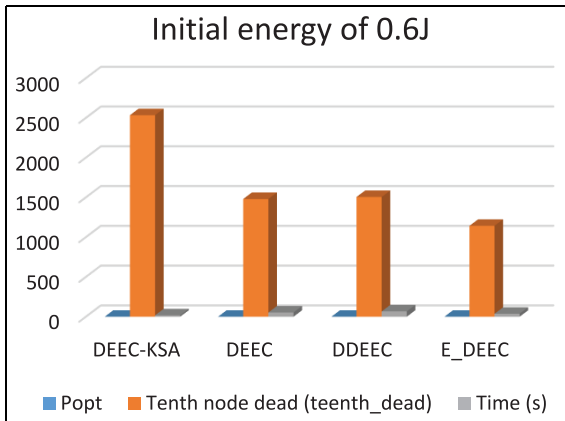


Figure 15. Network lifetime based on 100 nodes with initial 0.6 J.

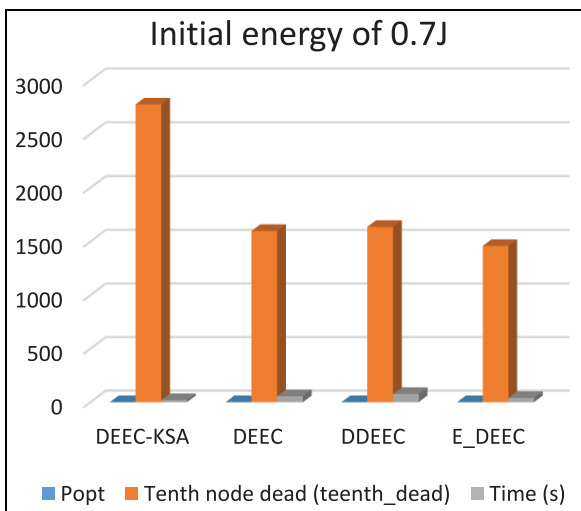


Figure 16. Network lifetime based on 100 nodes with initial 0.7 J.

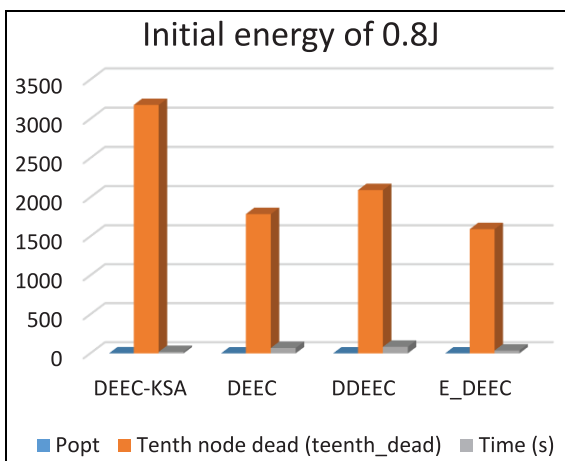


Figure 17. Network lifetime based on 100 nodes with initial 0.8 J.

efficient clustering algorithm since it spent less time to send a higher number of packets to BS, which suggests a reduction in energy.

Conclusion and future work

This article presented a bio-inspired approach called DEEC-KSA for optimizing energy in WSNs. The proposed approach considered the heterogeneity of energy requirements of sensor nodes. The simulation result showed that the proposed DEEC-KSA performed optimally in comparison with the existing benchmarked clustering algorithms for heterogeneous networks. In addition, the proposed DEEC-KSA has the highest network throughput spending limited time, and it has the best network stability than the comparative algorithms considered in this article. It is observed that the proposed DEEC-KSA is efficient in terms of stability period, network lifetime performance, and network throughput in terms of packets successfully sent to BS relative. In addition, the proposed DEEC-KSA has an optimal time to send packets successfully to BS. The base on the proposed DEEC-KSA intelligently optimizes energy; it can, therefore, be concluded that our proposed DEEC-KSA provides an energy-efficient clustering algorithm that ensures a longer stability period for WSNs. In the future, the proposed algorithm will be applied to a large number of nodes with different energy requirements to evaluate its efficiency with other nature-inspired algorithms. In addition, since it is possible for nodes with higher energy to be drained resulting in no active node on the network, future works should also focus on how to overcome this challenge.


Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The authors are thankful for the research supported grant by both the National Research Foundation of South Africa with grant number 117799 and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP; NRF-2018K1A3A1A09078981).

ORCID iD

Israel Edem Agbehadji  <https://orcid.org/0000-0001-5718-4494>

References

1. Siow E, Tiropanis T and Hall W. Analytics for the internet of things: a survey. *ACM Comp Surv* 2018; 51(4): 74.

2. Ristl A. *The internet of things: IoT analytics from the edge to core to cloud*. Hopkinton, MA: Dell EMC, 2017, p.45.
3. Manichavelu D and Vaidyanathan RU. Particle swarm optimization (PSO)-based node and link lifetime prediction algorithm for route recovery in MANET. *J Wirel Commun Netw* 2014; 2014: 107.
4. Gomes L, Sousa F and Vale Z. An agent-based IoT system for intelligent energy monitoring in buildings. In: *IEEE 87th vehicular technology conference*, Porto, 3–6 June 2018, pp.1–5. New York: IEEE.
5. Xu X, Fu S, Cai Q, et al. Dynamic resource allocation for load balancing in fog environment. *Wirel Commun Mob Comput* 2018; 2018: 6421607.
6. Zhou K, Fu C and Yang S. Big data driven smart energy management: from big data to big insights. *Renew Sustain Energy Rev* 2016; 56(2016): 215–225.
7. Fotopoulou E, Zafeiropoulos A, Terroso-Sáenz F, et al. Providing personalized energy management and awareness services for energy efficiency in smart buildings. *Sensors* 2017; 17(9): 2054.
8. Expert EPM. <http://www.schneider-electric.com/b2b/en/solutions/enterprise-solutions/enterprise-software-suites/index.jsp>
9. Buildingiq 2020, <https://buildingiq.com/products/predictive-control/>
10. eSight. <https://www.esightenergy.com/about/>
11. Optimization™ PE. 2019, <https://buildingiq.com/products/predictive-controlold/buildingiq-platform/>
12. Chan E and Han S. Energy efficient residual energy monitoring in wireless sensor networks. *Int J Distrib Sensor Netw* 2009; 5: 748–770.
13. Heinzelman WR, Chandrakasan A and Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks. In: *IEEE Hawaii international conference on systems sciences*, Maui, HI, 7 January 2000. New York: IEEE.
14. Torres MGC. *Energy consumption in wireless sensor networks using GSP*. Medellin, Colombia: Universidad Pontificia Bolivariana, 1995, p.95.
15. Heinzelman WB, Chandrakasan A and Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans Wirel Commun* 2002; 1(4): 660–670.
16. Nesrine K and Jemaa MB. HEERP: hierarchical energy efficient routing protocol for wireless sensor networks. In: *International conference on communications and information technology (ICCIT)*, Hammamet, Tunisia, 26–28 June 2012. New York: IEEE.
17. Bandyopadhyay S and Coyle EJ. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In: *INFOCOM 2003: twenty-second annual joint conference of the IEEE computer and communications societies*, San Francisco, CA, 30 March–3 April 2003, pp.1713–1723. New York: IEEE.
18. Shivapur S, Kanakaraddi SG and Chikaraddi AK. Load balancing techniques in wireless sensor networks: a comparative study. *Int J Emerg Technol Comp Sci Electron* 2015; 14(2): 1–6.
19. Kiani F. A survey on management frameworks and open challenges in IoT. *Wirel Commun Mob Comput* 2018; 2018: 9857026.
20. Heinzelman WR, Chandrakasan A and Balakrishnan H. Energy efficient communication protocol for wireless microsensor networks. In: *Proceedings of the 33rd annual Hawaii international conference on system sciences*, Maui, HI, 7 January 2000. New York: IEEE.
21. Younis O and Fahmy S. HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Trans Mob Comput* 2004; 3(4): 366–379.
22. Qing L, Zhu Q and Wang M. Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks. *Comput Commun* 2006; 29(12): 2230–2237.
23. Smaragdakis G, Matta I and Bestavros A. SEP: a stable election protocol for clustered heterogeneous wireless sensor networks. Technical report, Computer Science Department, Boston University, Boston, MA, July 2004.
24. Vançin S and Erdem E. Threshold balanced sampled DEEC model for heterogeneous wireless sensor networks. *Wirel Commun Mob Comput* 2018; 2018: 4618056.
25. Tukisi TW, Mathaba TND and Odhiambo MO. Multi-hop PSO based routing protocol for wireless sensor networks with energy harvesting. In: *Conference on information communication technology and society (ICTAS)*, Durban, South Africa, 6–8 March 2019. New York: IEEE.
26. Uplap P and Sharma P. Review of heterogeneous/homogeneous wireless sensor networks and intrusion detection system techniques. In: *Proceedings of fifth international conference on recent trends in information, telecommunication and computing (ITC 2014)*, Chandigarh, India, 21 March 2014. Chandigarh, India: Association of Computer Electronics and Electrical Engineers.
27. Hasnat MA, Akbar M, Iqbal Z, et al. Bio inspired distributed energy efficient clustering for wireless sensor networks. In: *5th National Symposium on Information Technology: Towards New Smart World (NSITNSW)*, Riyadh, Saudi Arabia, 17–19 February 2015, pp.1–7. New York: IEEE.
28. Al-Janabi TA and Al-Raweshidy HS. Optimised clustering algorithm-based centralised architecture for load balancing in IoT network. In: *International symposium on wireless communication systems (ISWCS)*, Bologna, 28–31 August 2017, p.6. New York: IEEE.
29. Agbehadji IE, Millham R, Fong SJ, et al. Kestrel-based Search Algorithm (KSA) for parameter tuning unto Long Short Term Memory (LSTM) network for feature selection in classification of high-dimensional bioinformatics datasets. In: *Proceedings of the federated conference on computer science and information systems*, Poznan, 9–12 September 2018, p.15–20. New York: IEEE.
30. Kennedy J and Eberhart RC. Particle swarm optimization. In: *Proceedings of IEEE international conference on neural networks*, Piscataway, NJ, 27 November–1 December 1995, pp.1942–1948. New York: IEEE.
31. Stützle T and Dorigo M. *Ant colony optimization*. Cambridge, MA: The MIT Press, 2002.
32. Tang R, Fong S, Yang X-S, et al. Wolf search algorithm with ephemeral memory. In: *Seventh international*

- conference on digital information management (ICDIM), Macau, China, 22–24 August 2012. New York: IEEE.
33. Yang X-S. A new metaheuristic bat-inspired algorithm. In: González JR, Pelta DA, Cruz C, et al. (eds) *Nature inspired cooperative strategies for optimization (NICSO 2010)*. Berlin: Springer, 2010, pp.65–74.
 34. Latiff NMA, Tsimenidis C and Sharif BS. Energy-aware clustering for wireless sensor networks using particle swarm optimization. In: *IEEE 18th international symposium on personal, indoor and mobile radio communications (PIMRC 2007)*, Athens, 3–7 September 2007, pp.1–5. New York: IEEE.
 35. Ari AAA. *Bio-inspired solutions for optimal management in wireless sensor networks. Artificial intelligence [cs.AI]*. Paris: Université Paris-Saclay, 2016, p.139.
 36. Jadhav AR and Shankar T. Whale optimization based energy-efficient cluster head selection algorithm for wireless sensor networks. *Neural Evolut Comput* 2017; 22: 1–22.
 37. Liu J-L and Ravishankar CV. LEACH-GA: genetic algorithm-based energy-efficient adaptive clustering protocol for wireless sensor networks. *Int J Mach Learn Comput* 2011; 1(1): 79–85.
 38. Bui K-HN and Jung JJ. Computational negotiation-based edge analytics for smart objects. *Inform Sci* 2018; 480: 222–236.
 39. Varghese B, Wang N, Barbhuiya S, et al. Challenges and opportunities in edge computing. In: *IEEE international conference on smart cloud*, New York, 18–20 November 2016. New York: IEEE.
 40. Omoniwa B, Hussain R, Javed MA, et al. Fog/edge computing-based IoT (FECIoT): architecture, applications, and research issues. *IEEE Internet Things J* 2019; 6(3): 4118–4149.
 41. Kuila P and Jana PK. Energy efficient load-balanced clustering algorithm for wireless sensor networks. In: *2nd International conference on communication, computing & security*, Rourkela, India, 6–8 October 2012, pp.771–777. Amsterdam: Elsevier.
 42. Halgamuge MN, Zukerman M, Ramamohanarao K, et al. An estimation of sensor energy consumption. *Prog Electromagn Res B* 2009; 12: 259–295.
 43. Zhao Z, Xu K, Hui G, et al. An energy-efficient clustering routing protocol for wireless sensor networks based on AGNES with balanced energy consumption optimization. *Sensors* 2018; 18(11): 3938.
 44. Agbehadji IE, Millham RC, Fong SJ, et al. Bioinspired computational approach to missing value estimation. *Math Probl Eng* 2018; 2018: 9457821.
 45. Abo-Zahhad M, Amin O, Farrag M, et al. Survey on energy consumption models in wireless sensor networks. *Open Trans Wirel Commun* 2014; 1(1): 63–79.
 46. Agbehadji IE, Millham RC and Fong S. Kestrel-based search algorithm for association rule mining and classification of frequently changed items. In: *IEEE international conference on computational intelligence and communication networks*, Dehradun, India, 23–25 December 2016. New York: IEEE.
 47. Agbehadji IE, Millham RC, Fong SJ, et al. Integration of Kestrel-based search algorithm with artificial neural network (ANN) for feature subset selection. *Int J Bio Inspir Comput* 2019; 13(4): 222–233.
 48. Anzola J, Pascual J, Tarazona G, et al. A clustering WSN routing protocol based on k-d tree algorithm. *Sensors* 2018; 18(9): 2899.
 49. Behzad M and Ge Y. Performance optimization in wireless sensor networks: a novel collaborative compressed sensing approach. In: *International conference on advanced information networking and applications*, Taipei, 27–29 March 2017, pp.749–756. New York: IEEE.
 50. Liaqat M, Gani A, Anisi MH, et al. Distance-based and low energy adaptive clustering protocol for wireless sensor networks. *PloS ONE* 2016; 11(9): e0161340.