

Received November 22, 2021, accepted December 10, 2021, date of publication December 13, 2021, date of current version December 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3135311

Energy Coherent Fog Networks Using Multi-Sink Wireless Sensor Networks

RABIE A. RAMADAN^{1,2}, (Member, IEEE), **MD. HAIDAR SHARIF**¹, (Member, IEEE), **EISSA JABER ALRESHIDI**¹, **MD. HARIS UDDIN SHARIF**³, (Graduate Student Member, IEEE), **IBRAHIM FURKAN INCE**^{4,5}, **HIQMET KAMBERAJ**⁶, AND **AHMED Y. KHEDR**^{1,7}

¹College of Computer Science and Engineering, University of Hail, Hail 2440, Saudi Arabia

²Computer Engineering Department, Faculty of Engineering, Cairo University, Giza 12613, Egypt

³Department of International Graduate Services, University of the Cumberland, Williamsburg, KY 40769, USA

⁴Department of Electronics Engineering, Kyungsung University, Busan 48434, South Korea

⁵Department of Digital Game Design, Nisantasi University, 34485 Istanbul, Turkey

⁶Department of Computer Engineering, Faculty of Engineering, International Balkan University, Skopje 1000, North Macedonia

⁷Systems and Computer Department, Faculty of Engineering, Al-Azhar University, Cairo 11371, Egypt

Corresponding author: Rabie A. Ramadan (rabie@rabieramadan.org)

This work was supported by the Scientific Research Deanship at the University of Ha'il, Saudi Arabia, under Project RG-20 147.

ABSTRACT Fog computing (FC) models the cloud computing paradigm expedient by bridging the breach between centralized data servers and diverse terrestrially distributed applications. It wields various wireless sensor networks (WSNs) that sprawl in the core of any IoT applications. Consequently, the operation of fog network turns on the efficiency of WSNs operation, while the all-inclusive network energy consumption depends on both FC and WSNs operation. This paper addresses how dissimilar organizations of a fog network can influence its effectiveness and energy savings. Chiefly, it appraises whether deploying multi-sink nodes in close enough vicinity to the fog nodes can give in energy savings and foster coherent data communication between WSNs and fog networks. To assess the multi-sink assignment problem the following four criteria are used: (i) Distance from the fog network nodes; (ii) Nodes degree; (iii) Sink nodes energy; and (iv) Sink nodes processing capabilities. This paper suggests four novel solutions to the multi-sink connectivity for some challenges of fog networks deeming: (i) Window Nondominant Set (WNS); (ii) Evaluation Based Approach (EBA); (iii) Harris Hawks Optimizer (HHO); and (iv) Modified HHO (MHHO). Distinct sets of experiments are conducted to check out algorithmic performance. The performance of all algorithms is measured and then compared to each other in terms of power consumption, runtime, packet loss, and localization error. One of the key supremacies of our approaches is the utilization of fog network for sensor networks data processing, principally with the large-scale networks. Yet, the communication challenges could need further study due to the limited communication range of the sensors.

INDEX TERMS Cloud computing, energy, fog computing, fog nodes, wireless sensor networks.

I. PROPOSED ACRONYMS

Abbreviation ⇒ Expansion.

WSNs ⇒ Wireless Sensor Networks.

IoT ⇒ Internet of Things.

WNS ⇒ Window Nondominant Set.

EBA ⇒ Evaluation Based Approach.

HHO ⇒ Harris Hawks Optimizer.

MHHO ⇒ Modified HHO.

CC ⇒ Cloud Computing.

FC ⇒ Fog Computing.

FNs ⇒ Fog Nodes.

SNs ⇒ Sink Nodes.

MST ⇒ Minimum Spanning Tree.

CHs ⇒ Cluster Heads.

LEACH ⇒ Low-Energy Adaptive Clustering Hierarchy.

NLT ⇒ Network Life Time.

II. INTRODUCTION

As the CC utilizes a network of remote resources through the Internet, it brings off organizations the cost of servers and other equipment (e.g., internal servers and personal

The associate editor coordinating the review of this manuscript and approving it for publication was Chien-Ming Chen^{id}.

computers). Most of the data arising from mobile nodes and IoT devices are assembled in realtime. The data-intensive and time-sensitive applications of such devices have challenged the idea that CC operates a high Internet bandwidth. The FC is a good solution to lessen the impediments of WSNs with its capacity to intersect the requirements of IoT applications. Nevertheless, most IoT applications suffer from network latency. Consequently, the processing of large volumes of data would result in a low throughput. The conventional provisioning of virtual machines may extensively cap the number of concurrent applications as well as users within a system. Numerous solutions have been proposed [1]–[4] to alleviate these problems, among them the adoption of serverless computing [5].

The serverless architecture is delegated to third-party providers in charge of dynamically provisioning and allocating resources. Another alternative execution model is the FC (or sometimes referred to as edge computing) [6]–[10]. The FC acts as a mediator between remote servers and hardware. The FC is a distributed decentralized infrastructure, whereas the CC is a centralized system. Most of the FNs are geographically distributed, resource constrained, and heterogeneous. As a result, the FC extends the CC paradigm by bridging the gap between centralized data servers and miscellaneous applications of geographically distributed computing and storage resources. The key eccentricity of the FC is its topology. Explicitly, the geographically distributed nodes that carry out the computations and offer storage as well as network services. An N-tier fog distribution consists of miscellaneous frameworks at various hierarchical levels [11]. It comprises of servers co-located with internet-service-provider gateways and radio base stations from each fog node to each sensor node. The implementation of the FC brings up a long list of challenges including resource handling of FNs [12], the positioning and movement of application equipment with utilities onto fog and cloud nodes [13]–[15], and the offloading from mobile and IoT devices [16]–[21]. Additional adaptations are frequently demanded for almost every single application based on its architecture, implementation process, and service requirement.

Concerning sensor network deployment related issues, most of the research works specify to the placement of a sink node [22]–[31]. In such directional studies, separate nodes are generally deployed on the border of the WSN as an end point to analyze the collected data and make the decision. Nonetheless, in our scenarios, the WSNs are now in a separate layer that is used to communicate with the fog network for data processing. Furthermore, the WSNs transfer their information to the FNs. Besides, in most fog network applications, sensors are deployed randomly and are unattended in the monitoring field. Accordingly, the selection of sensor nodes on the border of the fog networks to operate as gateways is pivotal to keep the WSN operating for a long time and to keep the overall fog and cloud network operating efficiently. In this paper, the selected nodes are referred to as SNs since it is widely used term in WSNs.

Motivation: A comprehensive survey on fog network energy is presented in [32]–[38] where IoT devices are connected to the CC and the FNs are used as gateways between the IoT devices and cloud nodes. The core of the IoT network is the sensor network that incorporates very tiny and limited capability devices. This research focuses on the energy consumption of the fog and cloud networks in terms of energy cost. Yet, there is an important part of the network, which is the sensor network, which keeps track of the environment and delivers its information to the fog points. The energy of this part of the network is critical in it; if one of the sensor nodes connected to a fog node were to fail, the overall network would be dysfunctional. The energy and connectivity of this part of the cloud and fog network cannot attract much research attention even though saving some of the energy used by WSNs would be beneficial to the overall structure of the fog network. Organizing the WSN in a way that fits the requirements of both FC and CC is another fault-finding issue. Henceforth, the aim of this study is to realize the optimal energy saving and connectivity parameters in relation to the monitoring networks (WSNs) and the data analysis and decision-making networks. Based on our previous experience with WSNs, we argue that the sink node could be one of the nodes given that the sink node in WSNs is a changeable node based on the current WSN energy state. In addition, using FNs as SNs is not beneficial in terms of overall energy saving.

One of the advantages of the FC is its closeness to the sink node. Therefore, instead of sending all of the sensed data to the cloud computing which is not energy efficient, the FC could be more closer for data processing and things that needs quick decisions could be handled more faster than sending them to the cloud computing. Besides, some of the data reduction techniques could be applied on the FC before sending it to the cloud.

This paper addresses an energy efficient framework for fog networks based on multi-sink wireless sensor networks. It is an extension to our previous conference paper [1]. The previous paper proposed two greedy algorithms namely WNS and EBA with simple test cases. This paper extends the previous two algorithms by adding more details and extensive experiments for evaluation. In addition, this paper proposes more sophisticated algorithms to solve the multi-sink deployment problem considering both HHO and MHHO. Succinctly, our contributions in this paper include:

- The proposal of a realistic and a simplified framework for fog network involving WSNs as a core network;
- The introduction of the problem of WSNs connectivity with the fog network through a set of SNs (Gateways);
- The proposal of four avant-garde solutions (e.g., Algorithms of 1, modified of 1, 2, and 3) to the multi-sink connectivity for fog network problem.

The rest of the paper is organized as follows: Section III hints some preliminaries; Section IV goes through the relevant literature; Section V discusses miscellaneous algorithms used for this paper; Section VI specifies the problem; Section VII

outlines the approaches used to find a solution to the problem; Section VIII presents the conducted simulation experiments along with their obtained results; Section IX analyzes the obtained results; and, Section XI concludes the paper with few hints of future study.

III. PRELIMINARIES

The FC is a highly virtualized platform that provides computing, storage, and network services between edge devices and cloud-based data servers. The FC harnesses personal devices to speed wireless networks. There are many applications for the FC including the connected vehicle, smart grid, and WSNs. Many applications related to smart cities use controllers that communicate via wireless links [39]. Besides, the FC allows for data collected from video surveillance cameras [40] to be stored and processed in fog nodes. However, according to Bonomi [2], the FC is distinguished by the following characteristics: low latency, location awareness, mobility, geographical distribution, support of a large number of nodes, heterogeneity, strong presence of streaming, and realtime applications. Because of these characteristics, the FC could be an appropriate infrastructure for many IoT applications. The FC can be seen as an extension of the CC from the core to the edge of the network. Nevertheless, the CC suffers from a high latency which is not recommended when working with WSNs. This is one of the reasons that the FC can be beneficial; especially when operating realtime applications that require realtime interactions instead of batch processing (e.g., video streaming, gaming, and augmented reality). Besides, the FC has a distributed control topology; whereas the CC has a centralized control topology and this assists in performing distributed processing on the edge devices. This can be seen in large scale sensor networks used to monitor the environment as the FC can support a large number of nodes. The FC methods are widely wireless but are fixed.

The WSNs are composed of single nodes that are capable of interacting with their environment by sensing and controlling physical tools. To accomplish their tasks, WSN nodes must collaborate using wireless communication. There are several directions conserve energy in constrained energy WSNs including multiple mobile sinks, mobile sinks, mobile sensors, and multiple sinks. Almost all WSN applications need actuators to employ physical actions to either close or open or move. These controlling system actuators could open new dimensions to WSNs. It is worth mentioning that the flow of information between sensors and sink as well as controller node and actuators is bidirectional but not unidirectional. The actuators network and the wireless sensor became an assembly of sensors and actuators connected by a wireless medium to execute distributed sensing and actuation jobs. The sensor network has to be able to interact with other devices. But connecting WSNs with the Internet is difficult due to the lack of standardization in communication protocols. Besides, WSN data cannot be transmitted over long distances due to the restriction of the transmission

protocols of the WSN. Hence, it is necessary that WSN devices exchange data with a gateway or a mobile device that physically provides the connection to the Internet.

IV. RELATED WORKS

While the CC lacks data transmission because of the infrastructure and limitations of networks which enormously decrease its performance, the FC brings computing and storage resources closer to end users. There are many techniques that could be used to save energy in a fog network. Some of the used techniques are based on data caching [41]–[44]. Such techniques depend on retrieving the required data from the cached data [45] to save time and energy. Some approaches are basing the fog operation on 5G technology [46], [47], while other approaches (e.g., [48]) try to optimally localize the virtual machines in the fog network. Additionally, there exists a large body of research focused on multi-sink localization in WSNs [22], [23], [26], [27], [29], [49]–[53]. In one of the studies, the sink placement problem was considered as an optimization problem with an energy saving objective function. For example, an optimal solution was proposed in [22], where different heuristics were used based on Cuckoo [23], Local search [26], Bactria [27], and Backbone information [29]. However, energy saving in WSNs may also depend on the used routing algorithms. One of the most commonly applied algorithms is the LEACH algorithm [54], which can be used as a clustering and routing algorithm simultaneously.

In this section, we aim to comprehensively review the most important up-to-date algorithms in FC related to multi-sink sensor networks energy saving and clustering techniques.

A. COMPREHENSIVE SURVEYS

Habibi *et al.* [55] made a comprehensive architectural survey on FC. Nejad *et al.* [56] aimed to investigate fog-based context-aware systems. Ahammad *et al.* [57] performed an elaborative study on cloud, fog, roof, and dew computing including their interaction, benefits, and limitations in IoT ecosystem. Sharma *et al.* [58] work presented an overview and description of FC in the context of CC and IoT and also shedded light on the key differences between CC and FC. Khalid *et al.* [59] surveyed on privacy and access control schemes in FC. Matrouk *et al.* [60] aimed to comprehensively review and analyze many scheduling algorithms in FC. Singh *et al.* [61] resented a systematic literature review of FC. Merabtine *et al.* [62] reviewed the existing feature-based classifications of clustering protocols and elaborated a more generic and unified classification. Ullah [63] made a survey on hybrid, energy efficient and distributed based energy efficient clustering protocols for WSNs.

B. HHO RELATED ALGORITHMS

Abdel-Basset *et al.* [64] described the high virtualized layered FC model taking into account its heterogeneous architecture. Their normalization and scaling phase aided

the standard HHO algorithm to solve the task scheduling in FC. Hossain *et al.* [65] used a 2-hop routing algorithm based on the multi-objective HHO algorithm that selected the optimal forwarders between the source and destination vehicles. Shanmugam *et al.* [66] proposed cross-layer-based opportunistic routing protocol for WSNs. For optimal clustering, k-medoid with adaptive HHO algorithm was utilized for clustering the sensor nodes. In a different direction, Sharma *et al.* [67] used HHO algorithm for sensor nodes localization problem in WSNs.

C. LIFESPAN RELATED ALGORITHMS

Suzuki *et al.* [68] proposed two schemes for WSNs with multiple sinks for saving battery energy and to improve network lifetime performance. Cha *et al.* [69] proposed an energy efficient clustering algorithm for self-organizing and self-managing WSNs. Their simulation results revealed that their framework allowed smaller energy consumption for network management and longer network lifetime than the existing schemes including LEACH. Khediri *et al.* [70] improved energy efficient clustering protocol for increasing the life time of WSNs. Aydin *et al.* [71] divided the sensor nodes into clusters. In each cluster, the sensor that was the closest to the cluster center and had the highest residual energy, was chosen as the cluster head. To reduce the energy consumption of CHs, a mobile sink was used. Simulation results showed that their models increase the energy efficiency and extend the network lifespan. Koyuncu *et al.* [72] explained several WSN probabilistic routing protocols including LEACH. They selected CHs based on the energy drain pattern and location of the sensor nodes, which increased the lifespan of sensor nodes.

D. OPTIMIZATION METHODS RELATED ALGORITHMS

Oprea *et al.* [73] proposed an adaptive direct load optimization and control with IoT architecture for FC. Tuli *et al.* [74] used a gradient based optimization strategy using back-propagation of gradients with respect to input. Singh *et al.* [75] formulated a delay-based task allocation problem which suggested the optimal task allocation among local IoT devices, edge server, and the cloud toward the minimum energy consumption and end to end delay. The problem was then solved using energy-efficient task offloading strategy based on Levy-flight moth flame optimization (LMFO) algorithm. Borujeni *et al.* [76] used FECR (Pegasis-based Routing of FNs) and FEAR (Ant Colony Optimization based Routing of FNs) algorithms for WSNs. The results of their simulation showed that the average amount of energy usage in FECR protocol could be reduced by 9% and by 8% in FEAR. Arafat *et al.* [77] hinted an energy-efficient three-dimensional bio-inspired localization algorithm based on the hybrid gray wolf optimization method. They claimed that their method reduced localization errors, avoided flip ambiguity in bounded distance measurement errors, and achieved high localization accuracy. Abdel-Basset *et al.* [78] demonstrated a multi-objective approach based on integrating

the marine predator's algorithm with the polynomial mutation mechanism for task scheduling in FC environments.

E. DATA UTILITY BASED ALGORITHMS

Abidoeye *et al.* [79] focused on a routing protocol for data transmission in WSNs based on FC. FC was integrated into their scheme due to its capability to optimize the limited power source of WSNs and its ability to scale up to the requirements of the IoT applications. The results of their simulations showed the performance of their approach. Biswas *et al.* [80] focused on reducing the energy loss problem and designing an energy-efficient data transfer scenario between IoT devices and clouds. Bai *et al.* [81] suggested a joint computation offloading, data compression, energy harvesting, and application scenarios algorithm for FC. Maher *et al.* [82] introduced a data backup system based on multi-cloud and FC. Their system utilized the advantages of multi-cloud storage to ensure users' data protection and reliability. Zhang *et al.* [10] designed an auditing system based on two private key cryptographic techniques, namely message authentication code (MAC) and homomorphic MAC, for secure data storage in fog-to-cloud computing. Krishnan *et al.* [83] enhanced a clustering methodology with multiple mobile sinks for efficient data collection. Loganathan *et al.* [84] determined CHs based on the sensor node's weighted metric. The sensor nodes were then self-adaptive by making correct decisions in real-time based on the sensed data, but detected information was often inaccurate due to some mechanical, wireless loss, and battery problems. Barzegaran *et al.* [85] addressed control applications virtualized on a distributed FC platform, which were implemented as tasks running on FNs that exchanged messages over time sensitive networking.

F. DISTANCE RELATED CLUSTERING ALGORITHMS

Suleiman *et al.* [86] proposed an energy-efficient adaptive distance-based clustering in WSNs using an adaptive probability function for formulating clusters. Mehta *et al.* [87] employed an approach based on LEACH and considered network load for electing cluster head in addition to number of links, residual sensor energy and distance. Zhou *et al.* [88] divided the monitor area into several annuluses with different width. By considering the distance between sensor nodes and the base station, sensor nodes were firstly classified into different levels. Then, CHs selection and clusters formation were conducted independently in each annulus. Finally, the simulation results illustrated the effectiveness of their method in terms of energy efficiency and energy balance.

G. NEURAL NETWORKS BASED ALGORITHMS

Aliyu *et al.* [89] employed a lightweight, human immune, and anomaly-based intrusion detection system for the fog layer. They claimed a 10% reduction in the energy consumption of the fog node when compared with deploying a neural network on the fog node. Baccarelli *et al.* [6] designed the main building blocks and supporting services of the

learning-in-the-fog architecture by explicitly accounting for the multiple constraints on the per-exit maximum inference delays of the supported conditional deep neural networks. Kilcioglu *et al.* [9] applied an energy-efficient fine-grained deep neural network partitioning scheme for wireless collaborative FC systems. Farooq *et al.* [90] presented two energy prediction techniques. The first one was based on the recursive least square filter and the second one used the artificial neural network. Both techniques used inputs such as the number of tasks and size of the tasks to predict the energy consumption at different FNs. Wu *et al.* [91] presented an efficient binary convolutional neural network with numerous skip connections for FC to enable real-time smart industrial applications.

H. LATENCY CONSTRAINT ALGORITHMS

Aoun *et al.* [92] addressed the problem of clustering in WSNs, subject to upper bounds on the maximum latency, the energy consumed by intermediate nodes, and clusters size. Hebal *et al.* [93] proposed latency and energy medium access control aware routing for WSNs using information of the time-division multiple-access and exploiting the information of the energy consumed by each node in order to optimize the latency of communications and the energy consumption when relaying information to the sink in a WSN. Essalhi *et al.* [94] presented an approach based on a Fog-IoT architecture with the objective of ensuring smart energy management during communication and processing of offloaded tasks in IoT applications while respecting the latency constraint. Experimental results showed the effectiveness of their approach in terms of energy efficiency. Sutagundar *et al.* [95] utilized the functionalities of sensor cloud and FC to classify and save the information in better way along with minimizing the latency issue.

I. MISCELLANEOUS ALGORITHMS

Ullah-Kifayat [96] implemented and evaluated a fog-assisted link aware and energy efficient protocol for wireless body area networks. Their solution improved the path loss by 24%. Rahman *et al.* [97] suggested trust based virtual machine migration in vehicular FC. They applied fuzzy logic for trust evaluation in vehicular FC. Paris *et al.* [98] investigated edge-facilitated collaborative FC to augment the computing capabilities of individual devices while optimizing for energy-efficiency. Sampaio *et al.* [99] introduced an autonomic system to perform management of energy consumption in IoT devices and FC, including an advanced orchestration mechanisms to manage dynamic duty cycles for extra energy savings. Ijaz *et al.* [100] investigated workflow scheduling in fog-cloud environments to provide an energy efficient task schedule within acceptable application completion times. Lai *et al.* [101] aimed to minimize the total power consumption of a dense small cell network while offering the quality of service to all its user equipment. Kavitha *et al.* [102] determined a near-optimal probability for cluster head selection to reach the maximum effi-

ciency in the energy consumption for large-scale WSNs. Zamry *et al.* [103] applied an energy saving hierarchical network protocol based on low-energy adaptive clustering hierarchy for WSNs. He [104] suggested an energy-saving algorithm and simulation of WSNs based on clustering routing protocol. Vo [105] designed a method to optimize the energy consumption for sensor processing layer in IoT. Debasis *et al.* [106] suggested an energy saving medium access control protocol for wireless sensor networks.

Over the last several years, applications related to IoT have grown one of the most key techniques in the world to enhance the quality of our daily life. The number of devices used in those applications are increasing quickly. In effect, this is leading to the creation of huge amounts of data along with the challenges of low energy efficiency, low latency rate, and low bandwidth. To minimize such issues along with an essential need of a smart computing paradigm, miscellaneous smarter algorithms are being proposed every day. With this vein, we have addressed how dissimilar organizations of a fog network can influence its effectiveness and energy savings.

V. OUR USED ALGORITHMS

In this section, we focus on the algorithms of LEACH, MST, and HHO.

A. THE LEACH ALGORITHM

For any WSN to work efficiently, nodes are grouped in clusters. In this paper, we used a simplified clustering algorithm for the purpose of performance evaluation. We used the LEACH algorithm for its simplicity and short computational time. Besides, the LEACH algorithm is a clustering algorithm designed especially for WSNs that makes the routing process much easier and energy saving [107]–[109]. It is based on the energy consumption of each node. It is also a round-based technique, where CHs should be periodically changed to balance the network energy. Previously selected CHs cannot join the next round of selection. The algorithm divides the nodes into groups, with a cluster head for each group. It has two phases - setup and creation phases. In the setup phase, each node generates a number between 0 and 1. If a node selects a number that is lower than a certain threshold ($T(i)$), it announces itself as a cluster head for the current round. The $T(i)$ can be computed as:

$$T(i) = \begin{cases} \left(\frac{p}{1 - p \times \left(r \bmod \frac{1}{p} \right)} \right), & \text{if } i \in G; \\ 0, & \text{otherwise;} \end{cases} \quad (1)$$

where p is the percentage that is required to become a cluster head and G is the node that was not selected in the $1/p$ round. The nodes deployment could be generated randomly or based on the node's residual energy. The second phase is called the steady state, where nodes join the nearest CHs and start aggregating and transmitting data to their CHs. The CHs can be responsible for transmitting their collected data to the sink node. Following a certain period of time, the network starts

another round of selection to CHs. There are many variations of LEACH including the ones reported in [2].

B. THE MST ALGORITHM

Another routing algorithm that is used to measure performance in this paper named as MST. Prim’s and Kruskal’s algorithms [110], [111] are the famous greedy algorithms. They are used to find the MST of a given graph. To apply them, any given graph should be weighted, connected, and undirected. Kruskal’s algorithm [111] handles edges in order of weight (the smallest to the largest), with each edge not forming a loop with the previously attached edges, but ending after inserting edges to MST. The edges form a forest tree slowly grows into one individual MST. Prim’s algorithm is a Greedy one as well. The process begins with an empty spanning tree. The goal is to keep two groups of vertices separated. The first set comprises vertices that have already been included in the MST, whereas the second set contains vertices that have not yet been included. It evaluates all the edges that connect the two sets at each step and selects the least weighted connection from among them. It inserts the opposite endpoint of the edge to the set containing MST after choosing the edge. So, at each step of Prim’s method, we discover a cut (made up of two sets: one of which includes the vertices already included in MST and the other of which contains the remaining vertices), choose the smallest weight edge from the cut, and add this vertex to the MST Set (the set that contains already included vertices).

C. THE HHO ALGORITHM

The HHO was implemented in [31] that considers the efficient placement of a single sink and multi-SNs in WSN. However, this paper does not take the edge nodes into consideration. Therefore, as part of this paper the HHO has to be modified accordingly. The HHO is based on the hunting activity of the hawks of Harris, also known as dusky hawks. Such birds perch in the air, spot their prey from far away and attack it in a collaborative effort. The hawks’ perching action is the exploration process in the HHO and the exploitation mimics the hawks’ attacking style. The HHO has three major phases: exploring a target, the surprise pounce, and other kinds of hawk attacking strategies. The details of these phases are explained in [31] and [112] as well as implemented using MATLAB in [113]. The algorithm summary is given below.

1) EXPLORATION PHASE

Harris hawks have insightful eyes that can locate and spot predators that are often not simple to find. Harris hawks are patient, they perch and wait quietly, sometimes for hours. The HHO models this behavior at the discovery level, which can be articulated as:

$$X_i^{t+1} = \begin{cases} X_{rand}^t - r_1 |X_{rand}^t - 2r_2 X(t)| & q \geq 0.5; \\ (X_{rabbit} - X_m^t) - r_3(LB + r_4(UB - LB)) & q < 0.5; \end{cases} \quad (2)$$

where X_i^{t+1} is the i-th individual position (t + 1)-th iteration. The X_{rabbit} is the rabbit (prey) position as well as $q, r_1, r_2, r_3,$ and r_4 are random numbers between 0 and 1 (i.e., [0,1]) that are updated with each iteration. The upper and lower bounds of variables are represented by LB and UB, respectively. The X_{rand}^t represents the hawk that is selected randomly out of the hawk population. The X_m^t indicates the average hawk population at iteration t. The average of the hawks’ position is computed using the following equation:

$$X_m(t) = \frac{1}{N} \sum_{i=1}^N X_i(t); \quad (3)$$

where $X_i(t)$ is the location of hawk i at time of t. Fig. 1(a) illustrates HHO stages [113].

2) TRANSITION FROM EXPLORATION TO EXPLOITATION

The HHO may exchange exploration for exploitation on the basis of the rabbit’s escaping energy. The rabbit’s energy can be measured using the following equation:

$$E = 2E_0 \left(1 - \frac{t}{T}\right); \quad (4)$$

where E is the escaping energy of the rabbit, T is the maximum number of iterations, and E_0 is the rabbit’s initial energy, $E_0 \in (-1, 1)$. The energy behavior of the rabbit is shown in Fig. 1 (b) [113].

3) EXPLOITATION PHASE

In the exploitation phase of the process, hawks opt to besiege the prey from several directions. How slowly or quickly they decide to close in on the prey depends on the remaining energy of the prey. So, the idea of the hawks is to try to limit the scape chances for the prey. Therefore, hawks surround the prey from different directions. However, their remaining energy defines their movement speed toward the prey. Escape depends on the prey’s chance r. A successful escape occurs when $r < 0.5$. If $r \leq 0.5$, the algorithm adopts the soft besiege mode; otherwise it will opt for the hard besiege option. The HHO algorithm uses four strategies to simulate the attacking stage: soft besiege, hard besiege, soft siege with progressive rapid dives, and hard siege with progressive rapid dives.

a: CASE 1: SOFT BESIEGE ($r \geq 0.5$ AND $|E| \geq 0.5$)

This behavior is modeled by the following equation:

$$X(t + 1) = \nabla X(t) - E |J X_{rabbit}(t) - X(t)| \quad (5)$$

$$\nabla X(t) = X_{rabbit}(t) - X(t);$$

where $\nabla X(t)$ is the difference between the position of the rabbit and its current location in iteration t and $J = 2(1 - r_5)$ with r_5 is the rabbit random jump intensity during the escape process and $r_5 \in (0, 1)$ is a random number.

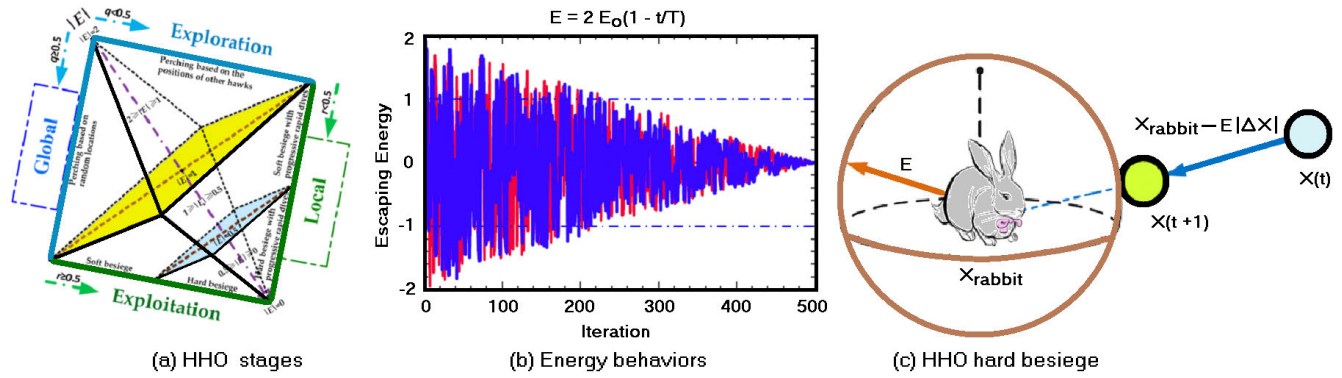


FIGURE 1. Illustration of HHO algorithm [113].

b: CASE 2: HARD BESIEGE ($r \geq 0.5$ AND $|E| < 0.5$)

Here, the current position is updated using the following equation:

$$X(t + 1) = X_{rabbit}(t) - E|\nabla X(t)|. \quad (6)$$

A simple example on this step is shown in Fig. 1 (c) [113].

c: CASE 3: SOFT BESIEGE WITH PROGRESSIVE RAPID DIVES ($r < 0.5$ AND $|E| \geq 0.5$)

Here, hawks decide their next move based on the following equation:

$$Y = X_{rabbit}(t) - E|J\nabla X(t) - X(t)|. \quad (7)$$

It has been assumed that the hawks dive according to the Levy Fight (LF) patterns (named for the French mathematician Paul Pierre Levy) where:

$$Z = Y + (S)(LF(D)); \quad (8)$$

where D is the dimension of the problem and S is a random vector of $1 \times D$ size. The LF patterns are identified using the equation below:

$$LF(D) = \frac{0.01\mu\sigma}{|\nu|^{\frac{1}{\beta}}}$$

$$\sigma = \left(\frac{\Gamma(1 - \beta) \sin(\frac{\pi\beta}{2})}{2^{\frac{\beta-1}{2}} \Gamma(\frac{1+\beta}{2}) \beta} \right)^{\frac{1}{\beta}}$$

$$\beta = 0.5; \quad (9)$$

where μ and ν are random values between 0 and 1. Based on the following equations, the hawks change position according to Eq. 10. Fig. 2 (a) shows an example of this move.

$$X(t + 1) = \begin{cases} Y & \text{if}(F(Y) < F(X(t))); \\ Z & \text{if}(F(Z) < F(X(t))); \end{cases} \quad (10)$$

where Y and Z are computed from Eqs. 7 and 8, respectively.

d: CASE 4: HARD BESIEGE WITH PROGRESSIVE RAPID DIVES ($r < 0.5$ AND $|E| < 0.5$)

Here, the hawk is close to the prey and its behavior is modelled by Eq. 11:

$$X(t + 1) = \begin{cases} Y & \text{if}(F(Y) < F(X(t))); \\ Z & \text{if}(F(Z) < F(X(t))); \end{cases} \quad (11)$$

where Y and Z are computed from Eqs. 12 and 13, respectively.

$$Y = X_{rabbit}(t) - E|J X_{rabbit}(t) - X_m(t)| \quad (12)$$

$$Z = Y + S + LF(D); \quad (13)$$

where $X_m(t)$ is calculated by Eq. 3. The pseudo code is illustrated in Algorithm 1. As it can be seen in Algorithm 1, the three cases of HHO have been represented at the algorithmic lines of 16-21. The HHO algorithm is adapted by [31] for sink node placement in WSNs, where the input of the algorithm is a graph that represents the sensor nodes and their neighbors with energies. For multi-sink placement, the fitness function $f(x)$ is changed as:

$$f(x) = \frac{ACN}{NUM_{nbr}}; \quad (14)$$

where ACN indicates the number of the sink node's sensor neighbors, NUM_{nbr} is the number of the active nodes. The optimization of the location can be expressed as minimum of $f(x)$. Fig. 2 (b) shows an example of hard besiege. Algorithm 1 depicts the pseudocode of HHO.

VI. PROBLEM DEFINITION

With the current emergence of IoT applications, WSNs and fog networks need to communicate in a reliable and efficient manner in terms of data transmission and energy saving. On that account, we deal with two interrelated problems as follows:

- **Problem 1** \Rightarrow There is the need for an efficient framework that combines both fog networks and WSNs. The proposed framework should have efficient communication interfaces between the fog network and WSNs. This leads to the second problem.

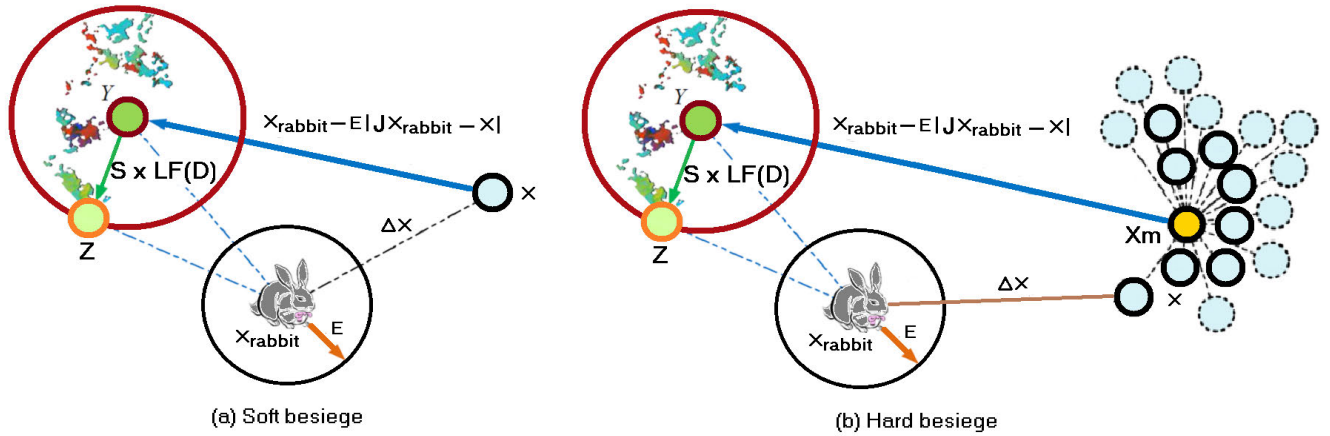


FIGURE 2. Examples of soft and hard besieges [113].

Algorithm 1 Harris Hawks Optimizer (HHO)

```

1: Input  $\Rightarrow$  The population size  $N$  and maximum number of iterations  $T$ .
2: Output  $\Rightarrow$  The location of rabbit and its fitness value.
3: Initialize the random population  $X_i$ , where  $i = (1, 2, \dots, N)$ .
4: Repeat (stopping condition is not met)
5: Calculate the fitness values of hawks.
6: Set  $X_{rabbit}$  as the location of rabbit (best location).
7: for (each hawk ( $X_i$ )) do
8: Update the initial energy  $E_0$  and jump strength  $J$ .
9: Update the  $E$  using Eq. 3.
10: if  $|E| \geq 1$  then
11:   Update the location vector using Eq. 2.
12: end if
13: if  $|E| < 1$  then
14:   if  $r \geq 0.5$  and  $|E| \geq 0.5$  then
15:     Update the location vector using Eq. 5.
16:   else if  $r \geq 0.5$  and  $|E| < 0.5$ 
17:     Update the location vector using Eq. 6.
18:   else if  $r < 0.5$  and  $|E| \geq 0.5$ 
19:     Update the location vector using Eq. 10.
20:   else if  $r < 0.5$  and  $|E| < 0.5$  then
21:     Update the location vector using Eq. 11.
22:   end if
23: end if
24: end Repeat
25: return  $X_{rabbit}$ 

```

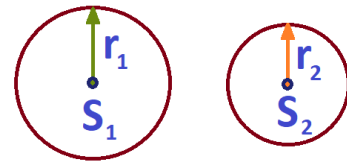


FIGURE 3. Two sensor nodes with communication ranges r_1 and r_2 .

In addition, sensor nodes consist of many components including microcontrollers or microprocessors, sensor(s), transceiver, and energy source. Therefore, to model a sensor node's energy, components such as energy consumption need to be considered. We follow the footsteps of Zhou *et al.* [114] in modelling the energy of sensors, taking into consideration Eqs. 3, 4, 5, 6, and 7 during the simulation.

$$E_{cpu} = E_{cpu-state} + E_{cpu-change} \quad (16)$$

$$E_{sensor} = E_{on-off} + E_{off-on} + E_{sensor-run} \quad (17)$$

$$E_{trans-state} = E_{TX} + E_{RX} + E_{Idle} + E_{sleep} + E_{CCA} \quad (18)$$

$$E_{trans-transition} = \sum_{i=1}^n (N_{trans-change}(j)) (e_{trans-change}(j)) \quad (19)$$

$$e_{trans-change}(j) = (E_{init-end}(j)) \left(\frac{P_{init}(j) + P_{end}(j)}{2} \right); \quad (20)$$

where E_{cpu} is the sum of the CPU energy consumption as a function of $E_{cpu-state}$ and $E_{cpu-change}$; $E_{trans-state}$ is the sum of transceiver based on its state; $E_{trans-transition}$ is the transmitter transition energy; and, E_{sensor} is the sensor consumed energy in terms of running and state transition.

It can be seen from the aforementioned equations that energy consumption in WSN is a critical issue, especially when it is left unattended and it is required to work for a long period of time. Sensors sense the environment and send their data to a sink node. The sink node is one of the sensor nodes that should be elected to collect data of sensors for further transmission to the cloud or for internal analysis. Based on our proposed framework, sensed data must travel

- **Problem 2** \Rightarrow As stated above, sensor nodes suffer from different limitations including memory and processing capabilities. They also have security issues that will be considered in a different article.

The WSNs could be modelled as an undirected graph $G(N, E)$, where sensors (N) represent the nodes and connectivity represents the edges (E). Two nodes are connected, when they can reach each other wirelessly. If we consider the communication range as r_i , where i is the node index and the node's location is (x_i, y_i) , two nodes are considered connected if the distance d_{ij} between node i and node j equals the $\min(r_i, r_j)$, as shown in Fig. 3. The distance d_{ij} would be simply computed by Euclidian distance as:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \quad (15)$$

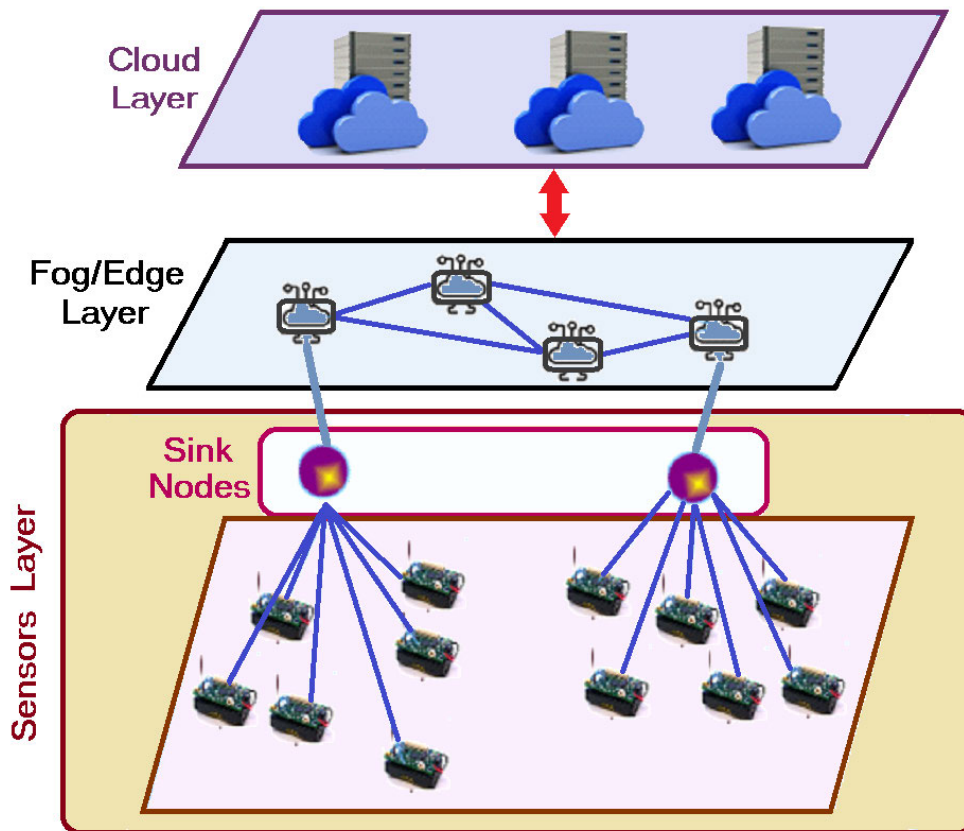


FIGURE 4. The proposed energy efficient fog network.

through a fog network to a cloud server for analysis and decision making. To increase the NLT [115] and save energy, we propose using a multi-sink to transmit the sensed data to the fog network. The key problem is the selection of the SNs that can represent the edge of the framework as depicted in Fig. 4. Such SNs should be selected carefully based on the following four criteria.

- 1) **Energy source (E)** \Rightarrow Nodes should have the best energy for a longer lifetime.
- 2) **Efficient processing capabilities (P)** \Rightarrow Nodes will be used to process the sending and receiving from a large number of nodes.
- 3) **Nodes degrees (D)** \Rightarrow Selected nodes have to have a sufficient number of neighbors to serve.
- 4) **Closeness to the fog network nodes (C)** \Rightarrow This reduces the required communication energy which diminishes the number of dropped messages. This also limits the selected nodes to the nodes on the border of the fog network.

There are some other characteristics that can be included, for example, the reliability of the selected node and security. These are, however, not within the scope of this paper. Another problem is the number of SNs to be selected. This is important because selecting a large number of SNs would not make sense and selecting a small number of sinks might cause

a network congestion problem. Henceforth, it is a tradeoff between these two extremes. For the purpose of the evaluation of our proposed algorithms, we consider the number of SNs as the number of FNs on the border of the fog network. The relation between the sink node and the fog node could be one to one or many to one. To wit, each sink node could have a dedicated fog node, or a group of sink nodes could communicate with a certain fog node based on the fog nodes availability and their locations.

VII. SOLUTION APPROACH

This section is dedicated to the suggested solution approaches for the aforementioned problems. In this section, we explain the proposed framework and solutions in detail.

A. FOG-WSNs FRAMEWORK

In this subsection, we present an efficient framework for fog and WSNs. It is noticeable from Fig. 4, the proposed framework consists of three main layers, from the in-out, the cloud computing layer, the fog layer, and the edge layer. At the same time, since we are dealing with WSNs, there exists an extra layer that is referred to as the sensor nodes layer. The cloud and fog layers have been explained in the previous sections. The edge layer is a layer of interface between the WSN and the fog network as well as SNs. Those

SNs are carefully selected to minimize the energy consumed by WSNs and prolong their lifetimes. At the same time, it guarantees easy and fast transfer of the sensed data to the fog network. Those edge nodes are carefully selected on the basis of certain criteria, as mentioned previously.

B. MULTI-SINK ASSIGNMENT APPROACHES

To select the best SNs, we propose two solutions namely WNS and EBA as hinted in algorithms of 2 and 3, respectively.

Algorithm 2 Window Nondominated Set (WNS)

```

1: Setup Phase:
2: Define the number of nodes to be deployed ( $S$ ).
3: Define the number of required SNs ( $N$ ).
4: Compute the nodes parameters ( $E, P, D, C$ ).
5: Define the size of the monitored area ( $A$ ).
6: Define the window size ( $N$ ).
7: Deploy the nodes on the monitored area either randomly or deterministically.
8: Repeat
9:   if  $W$  is not full then
10:     Add( $W_N, s_i$ )
11:   end if
12: end Repeat
13: Sink Selection Phase:
14: Repeat for all  $s_i$ 
15:   Compute ( $s_i, E, P, D, C$ )
16:   if (Nondominated( $W_N$ ) $\neq 0$  then
17:     Remove ( $D(s_i)$ )
18:     Repeat
19:     if  $W$  is not full then
20:       Add( $W_N, s_i$ )
21:     end if
22:   end Repeat
23:   else if ((Nondominated( $W_N$ ) $== 0$  &&  $i \neq N$ )
24:     Expand( $W, 1$ )
25:     Add( $W_N, s_i$ )
26:   end if
27: end Repeat
28: return  $W$ 

```

1) WNS ALGORITHM

This solution is a strict solution where the best SNs are chosen in terms of the four-selection criteria stated in the section VI at item 1, item 2, item 3, and item 4. Selecting the best SNs could be formulated as a multi-objective optimization problem. In addressing this problem, we propose to use a Pareto Optimality [116], [117], named after the Italian engineer and economist Vilfredo Pareto (1848–1923), as a multi-objective optimization technique. Nevertheless, it is well-known that the Pareto Optimality with more than two parameters is an NP-Hard problem [118]. Therefore, we try to simplify the problem as much as possible by introducing the concept of the *window*. The window concept is inspired from the sliding window communication protocol, where we limit the number of elements to be processed to a window size n , where all nodes need to be processed. One may suggest processing only the nearest nodes to the fog network. Based on the criteria that need to be optimized, that might not be an optimal solution as some may have a lower level of energy than other nodes that are a little far from the fog border.

As it can be observed in Fig. 5, the nodes in the window are the ones with the highest IDs. Within this window, the Pareto Optimality is applied. Once the first round is completed, the dominated nodes are removed from the window and the window is moved to include other nodes within the window limit n . This process is repeated until the last node is reached. The window size could be identified by the fog network designer or could simply be the number of FNs that will be connected to the sensor network. Explicitly, one sink node must correspond to one fog node. There are two special cases that occur when applying the window concept along with the Pareto Optimality concept:

- The window is full, and the last node was not reached. In this case, the window is extended by one space and then the algorithm is retracked. This might lead to the next problem.
- Too many nondominated nodes. In this case, nodes are sorted according to their distance from the FNs and a certain number has to be selected.

By the end of the process, the window is left with the best selected SNs in terms of the optimized criteria. Algorithm 2 shows how the Window Non-dominated Set approach works. As shown in the algorithm, a window is treated as a queue data structure. The output of the algorithm is the window elements (W) that involve the selected SNs. The selected nodes will work as gateways between WSN and FNs.

2) ALGORITHM OF EBA

This is a simplified approach, where the optimization parameters are normalized and prioritized on the basis of the equation:

$$E_{vli} = W_1 E_i + W_2 P_i + W_3 D_i + W_4 C_i \quad (21)$$

where

$$W_1 + W_2 + W_3 + W_4 = 1. \quad (22)$$

The E_{vli} value is computed for each node and then the overall values are sorted accordingly. A number of nodes are then selected to be the SNs based on the normalized values. The advantage of this approach is that the optimized criteria can be prioritized and this approach is quite simple to process. Algorithm 3 shows the steps required for the compute function that is responsible for computing the normalized value for all of the sensor nodes. It is noticeable that EBA algorithm is based on equations. It is a very light algorithm in terms of computation, where its complexity is $O(1)$.

3) ALGORITHM OF MHHO

In the previously described HHO [31] approach, a number of sinks are chosen with the fitness function that depends on the sink node degree, sensors with high energy connected to the sink, and closeness to the center of the deployment field as given in the following Eq. 23:

$$\min f(x) = \frac{1}{\alpha_1 \sum_{i=1}^{N_{nbr}} E_{nbr}(i) + E_x + \alpha_2 N_{nbr} + \alpha_3 d_x} \quad (23)$$

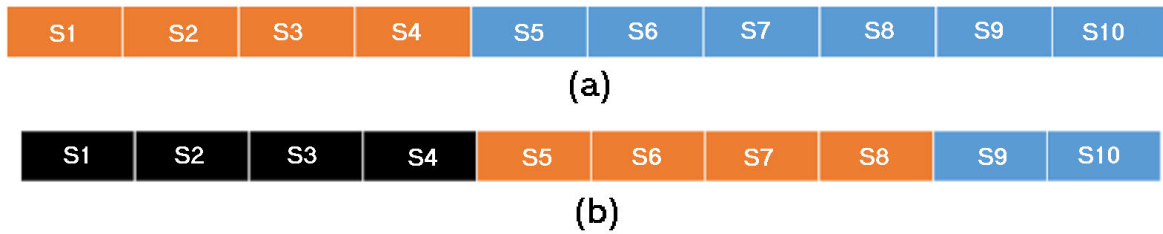


FIGURE 5. Window dominated set example.

Algorithm 3 Evaluation Based Approach (EBA)

1: *Setup Phase:*
 2: Define the number of nodes to be deployed (S).
 3: Define the number of required SNs (N).
 4: Define the size of the monitored area (A).
 5: Deploy the nodes on the monitored area either randomly or deterministically.
 6: *Sink Selection Phase:*
 7: **Repeat** for all s_i
 8: Compute (s_i, E, P, D, C)
 9: List[S] = Compute(s_i, E, P, D, C)
 10: **end Repeat**
 11: Sort(List[S])
 12: Return the top N nodes.

where N_{nbr} is the number of sensor neighbors served by the sink node; E_{nbr} is the energy of each neighbor to the sink; E_x is the residual energy of node x ; d_x is the distance between node x and the center of the deployment field; and, $\alpha_1, \alpha_2, \alpha_3$ are random numbers between 0 and 1.

For HHO to fit the purpose of WSN connectivity as part of a fog network, there must be two variations, as follows:

- No change to the selected SNs is made where a sink node operation is only modified to send its collected data to the nearest FN. This variation is referred to as the HHO algorithm.
- The fitness function is modified to fit the purpose of this paper to include the required parameters of initial energy (E), high processing capabilities (P), node degree (D), and closeness to the FNs (C) as given in Eq. 24. This modification is referred to as MHHO.

$$\min f(x) = \frac{1}{\alpha_1 \sum_{i=1}^{N_{nbr}} D_{nbr}(i) + E_x + \alpha_2 P_x + \alpha_3 \sum_{j=1}^N C(j)} \quad (24)$$

where D_{nbr} is the number of sensor neighbors served by the sink node; E_x is the residual energy of node x ; P_x is the processing capability of nodes x ; $C(j)$ is the distance between node x and the fog node j ; and $\alpha_1, \alpha_2, \alpha_3$ are random numbers between 0 and 1.

VIII. EVALUATION CRITERIA AND SIMULATION ENVIRONMENT CONFIGURATION

The evaluation criteria for the proposed algorithms and the environment setup are explained in this section. This section also presents the simulation results for WNS, EBA, HHO, and

MHHO algorithms. Our simulation strategy is to experiment with a different number of nodes using LEACH followed by the results of MST along with the different number of FNs and present the results of each evaluation criterion.

A. EVALUATION CRITERIA

The proposed evaluation criteria are described in this subsection. In addition, we follow the footsteps of Donta *et al.* [119] in mathematical description to the evaluation criteria. Our proposed evaluation criteria are stated below.

- **Energy Consumption** \Rightarrow This is the percentage of the sum of the energy consumed by all the nodes in transmitting messages within a predetermined time of operation over the sum of the initial energy of all nodes. This could be measured by the Eq. 25 as:

$$APC = \sum_{i=1}^N \left(\frac{sn_i CE}{sn_i IE} \right); \quad (25)$$

where sn_i is the sensor node i ; N is the number of nodes in the sensor network; and CE is the consumed energy and IE is the initial energy.

- **NLT** \Rightarrow This is the time that elapses from the start of the simulation to when the first node dies due to energy depletion. NLT could be computed by the Eq. 26 as:

$$NLT = t_{end} - t_0; \quad (26)$$

where t_0 and t_{end} belong the start time and the end time of the simulation, respectively.

- **Packet Loss** \Rightarrow The definition of the packet loss is the ratio of the sum of the received messages by the FNs to the sum of all messages sent by the nodes. This indicates how efficient the sink node should be placed. Packet loss could be computed by the Eq. 27 as:

$$PacketLoss = \sum_{i=0}^n \frac{Msg_{i, FN}}{Msg_{i, sent}}; \quad (27)$$

where $Msg_{i, FN}$ is the messages received by FN from node i , $Msg_{i, sent}$ is the number of messages sent by node i , and n is the total number of nodes.

- **Localization Error** \Rightarrow This is an indicator to how much of the required parameters are satisfied by the algorithms. In other words, the algorithms are required to find at least one sink node that is corresponding

and connected to each fog node. For example, if there are 5 FNs, it is assumed that the algorithm is able to find at least 5 corresponding SNs that are directly connected to the 5 FNs. If an algorithm produces at least the number of SNs that corresponds to the number of FNs, the localization error is equal to 100%. The localization error could be computed by the Eq. 28 as:

$$\text{LocalizationError} = \frac{FSN}{SN}; \quad (28)$$

where FSN is the number of sink nodes discovered by the algorithm and SN is the total number of sink nodes.

B. SIMULATION ENVIRONMENT

This section outlines the set of experiments that were conducted to verify the effectiveness of the proposed algorithms in selecting SNs for fog network purposes. Our algorithms were implemented and tested using the CupCarbon [120] simulator. The simulator was extended to include modules for our proposed algorithms. Only the HHO and MHHO algorithms were implemented using MATLAB benefiting from the work done in [113]. However, to enable a fair comparison, the same simulation environments and parameters were used and the same types of networks simulated. Table 1 shows the simulation parameters that were used with all of the algorithms. Different sets of experiments were conducted using a different number of nodes (e.g., 500, 1000, and 5000), a different number of FNs (e.g., 5, 10, and 15) and different routing methodologies (e.g., LEACH and MST). The sensors energy must be a value other than 0. If the initial energy is 0, the sensor will not be able to handle any of the required tasks. For this reason, in the simulation scenario, we have assumed initial energy as 1J, but not 0J. In addition, the carrier-sense multiple access with collision avoidance (CSMA/CA) of the MAC (Medium Access Control) layer protocol was used in all of the experiments for simplicity. There is no specific topology that was used during the data collection. The nodes are distributed randomly with uniform distribution. It is noted that this paper is based on specific module for data generation, implemented by authors, used in all of the algorithms for performance measure. Hence, there is no need for specific data generation module on MATLAB. The results presented in this paper relate to small scale, medium-scale, and large-scale networks as well as different routing methodologies. Only for clarity, Fig. 6(a) hints a network with 130 nodes to demonstrate the nodes' connectivity and distribution on certain terrain. Fig. 6(b) shows a sample of 500 nodes deployed randomly within the monitored areas representing a small-scale network. The computing capabilities of 500 nodes, the degree of 500 nodes, and the distance of 500 nodes from the nearest FN node have been demonstrated in Fig. 6 (c), (d), and (e), respectively. This indicates the variations of the sensor nodes used as parameters.

TABLE 1. Simulation environment parameters.

Parameters	Values
Deployment area	500 m^2 - 1000 m^2
Number of nodes	500, 1000, 5000
Initial energy of normal sensor nodes	1J
Maximum transmission range of a node	80 m
Electronic circuit's energy	50 nJ
Data aggregation energy	5 nJ/bit/message
Signaling packet size	200 bits
Data packet size	4000 bits
Free space communication energy	10 pJ/bit/ m^2
Routing protocols	LEACH and Prim's MST
Number of FNs	5, 10, 15

IX. SIMULATION RESULTS

This section presents the simulation results relating to the evaluation criteria. TDMA is adopted in this paper. TDMA is a channelization technique that divides a channel's bandwidth into many stations on a time basis. Each station is assigned a time slot, and it may only transmit data during that time slot.

A. POWER CONSUMPTION

Power consumption is used as a measure of the quality of the sink assignment based on the assumption that a specific number of packets/messages were sent per node. We feel that assessing the energy consumed by the network on the basis of a specific number of messages sent from each node to the selected SNs is a justifiable method to use. It also allows investigation of all of the routes from all of the sensor nodes to the assigned SNs. The reason behind this set of experiments is not to exhaust the network until it dies, but to look at the network while it is operating and compute the average consumed energy under the operation of the proposed algorithms. Fig. 7 shows the energy consumption of 500 sensor nodes, where LEACH is used as a clustering and routing protocol. Fig. 7 (a), (b), and (c) depict the energy consumption for 5, 10, and 15 FNs, respectively. It is noticeable that the four algorithms reveal a slight difference in terms of energy consumption. WNS results in the best level of energy consumption in all of the cases, followed by EBA and MHHO. The results obtained for HHO are also encouraging, showing only 5% more consumption than WNS, 4% more than EBA, and 3% more than MHHO.

Fig. 8 shows the energy consumption for 500 sensor nodes, where MST is used as a clustering and routing protocol. The energy consumption increases for all of the algorithms because of the routing algorithm used. WNS still obtained the best results in all of the cases at an average of 43%. It is worth mentioning that when FNs equal 15, the MHHO performs almost the same as HHI with only 3% difference. An investigation into why there was an increase of energy consumption showed that MST fixes the routes all of the time and some of the nodes die due to energy depletion. A resending process takes places in such cases in addition to the sink positions that are set differently by the different algorithms.

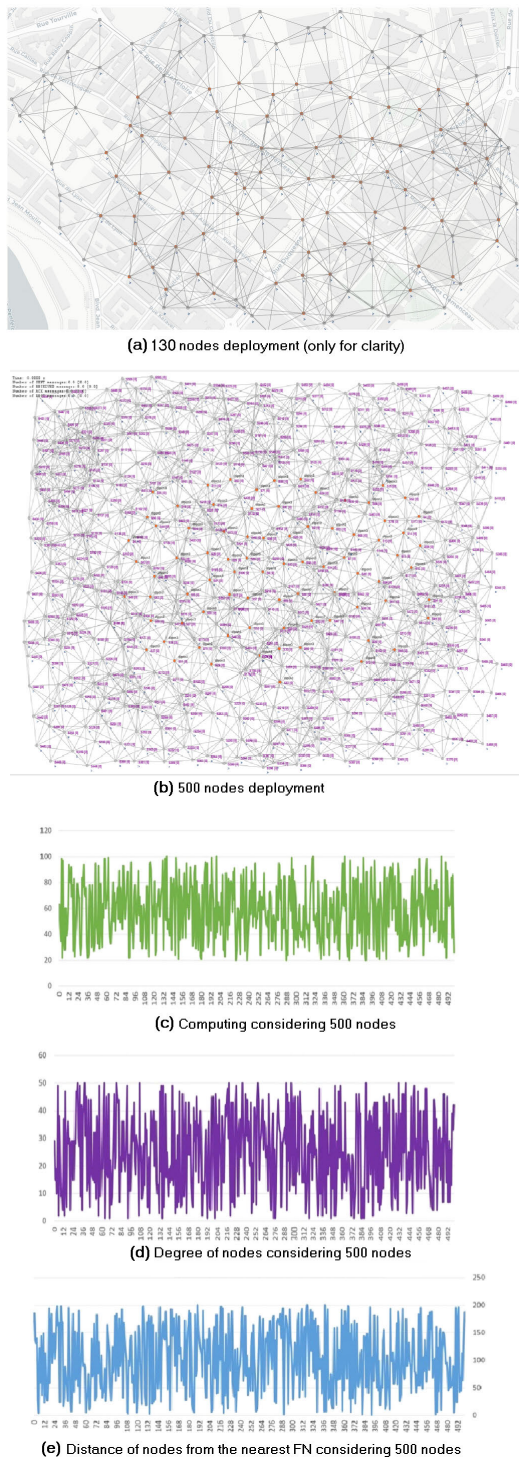


FIGURE 6. Representation of small-scale networks.

Fig. 9 presents the same information from a different point of view, where the consumed energy is presented for the four algorithms with a different number of FNs, namely 5, 10, and 15. It is evident that HHO level of energy consumption increases as the number of FNs increases; especially when LEACH is used as a routing algorithm. WNS performs well with both LEACH and MST in all of the cases with only

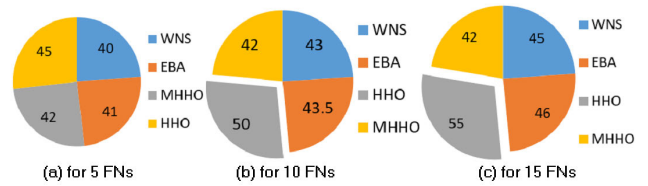


FIGURE 7. Consumed energy of 500 nodes using LEACH.

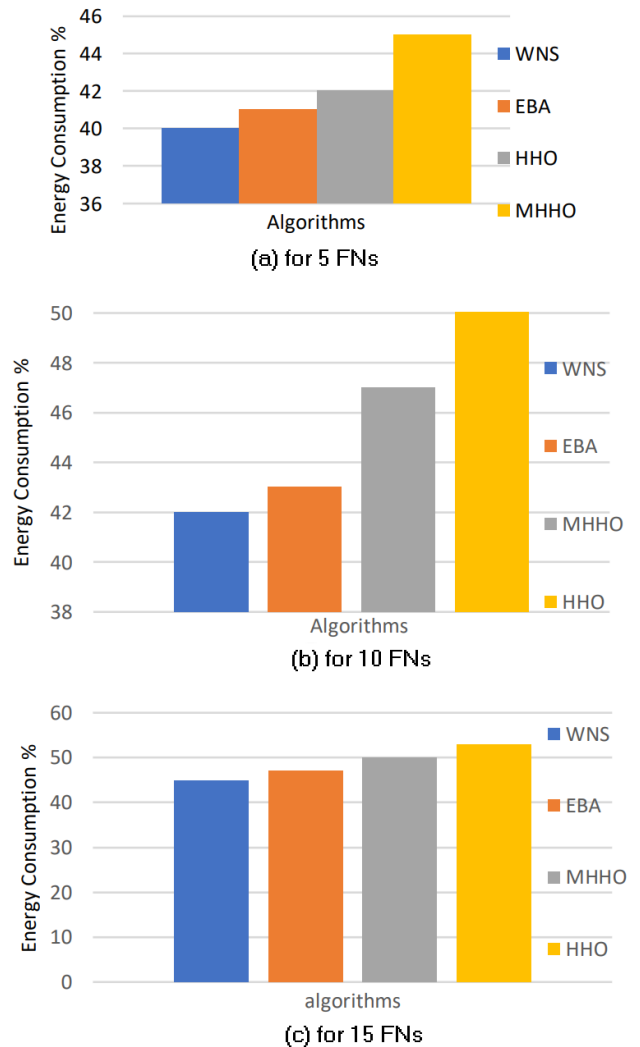


FIGURE 8. Consumed energy of 500 nodes using MST.

a small difference in the energy consumption figures when MST is used. The MHHO also shows a reasonable level of energy consumption in most of the cases with both the routing algorithms.

The energy consumption results were confirmed using a different number of nodes, specifically 1000 and 5000 nodes. Fig. 10 elaborates on the energy consumption, when 1000 nodes are deployed with different FNs. Fig. 10 (a) shows the energy consumption, when LEACH routing is used; and Fig. 10 (b) shows the energy consumption, when the MST

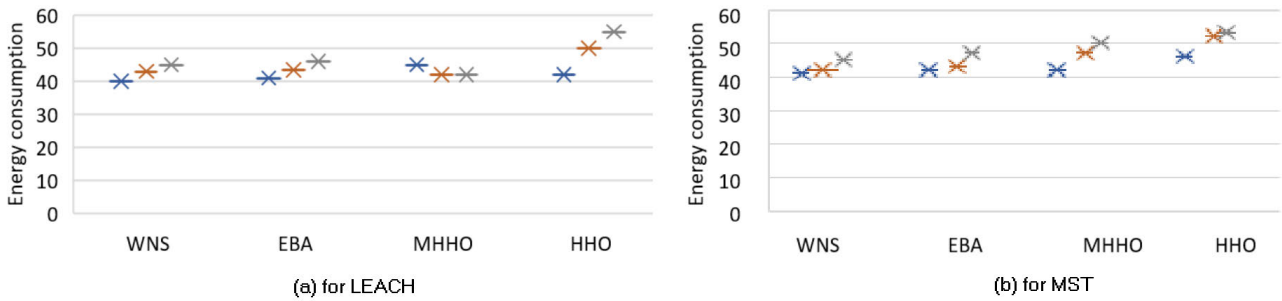


FIGURE 9. Consumed energy of 500 nodes using both LEACH and MST from a different point of view.

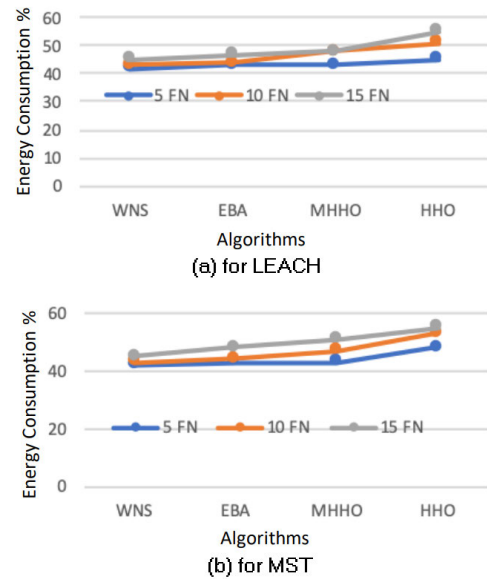
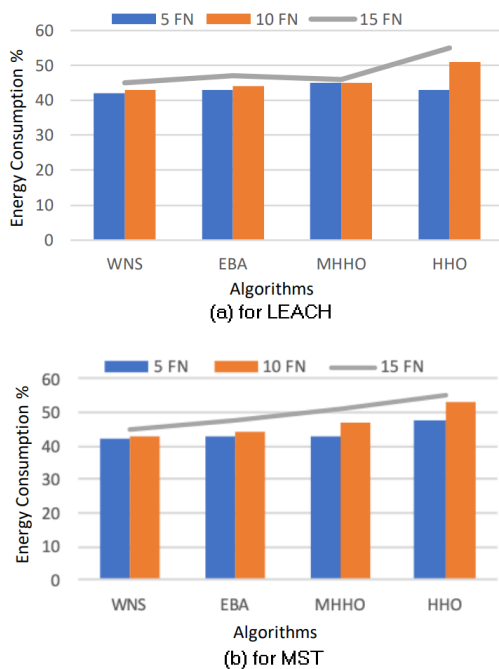


FIGURE 11. Consumed energy of 5000 nodes using both LEACH and MST.

FIGURE 10. Consumed energy of 1000 nodes using both LEACH and MST.

algorithm is used. Both figures confirm that WNS led to the best results, followed by EBA and MHHO. The MHHO energy consumption level was close to HHO with only 1% difference.

Fig. 11 shows the energy consumption for a large-scale network with 5000 nodes with different FNs, namely 5, 10, and 15. As it can be noticed from Fig. 11(a) and Fig. 11(b), although the results look similar to the ones obtained using other networks (e.g., Fig. 10) the difference between the energy consumption of algorithms increases with the increasing number of FNs. For instance, the difference between WNS and HHO is, on average, 8% and reaches 10% when 15 FNs are used.

B. THE NLT

The NLT was examined for all the algorithms. As stated above, the NLT is the runtime from the start of the simulation until the first node dies. Some researchers calculate the

lifetime of the network from the start of the simulation until the last node dies; however, if this definition were used the measurements in this study would not be accurate since some of the monitored areas are left without coverage; and in some cases, one node stays alive in the network for a long time after the others have died.

Fig. 15 shows the lifetime of the 500 nodes network with different FNs, namely 5, 10, and 15. As it can be observed from Fig. 15(a), the WNS algorithm gives the largest lifetime in the three cases followed by EBA. The performance of MHHO and HHO lags far behind that of the WNS. The difference is almost 8% on the average. The difference became significant, when MST was used (Fig. 15(b)) yielding an 11% increase, on the average.

The same results were confirmed, when 1000 nodes are used, as shown in Figs. 16 and 17. The difference between the algorithms increased to 11% on the average between WNS and HHO in both LEACH and MST. The performance of four algorithms (WNS, EBA, MHHO and HHO) remained the same in terms of lifetime. It is worth mentioning that the difference in NLT between the algorithms got smaller as

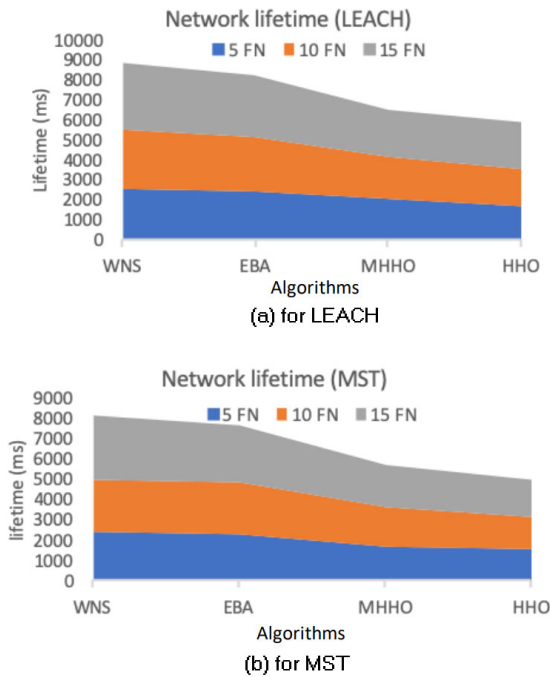


FIGURE 12. NLT of 500 nodes using both LEACH and MST.

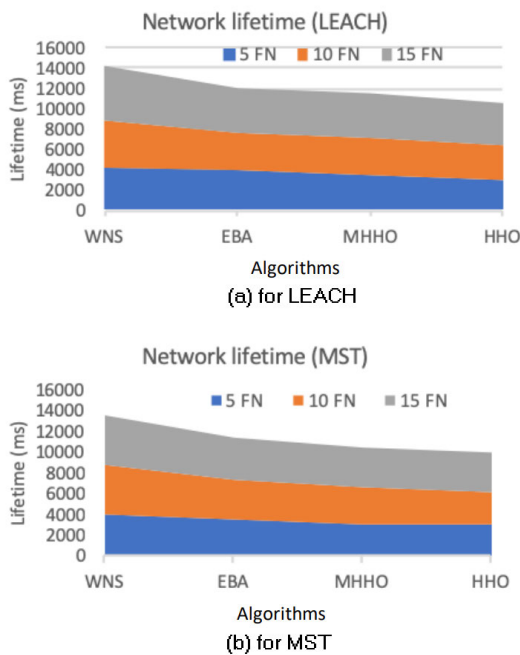


FIGURE 13. NLT of 1000 nodes using both LEACH and MST.

the number of nodes increased when using both LEACH and MST.

C. PACKET LOSS

The idea behind the packet loss is to compute the ratio between the sum of the received packets by all of the FNs to all of the packets sent by the sensor nodes. This ratio is

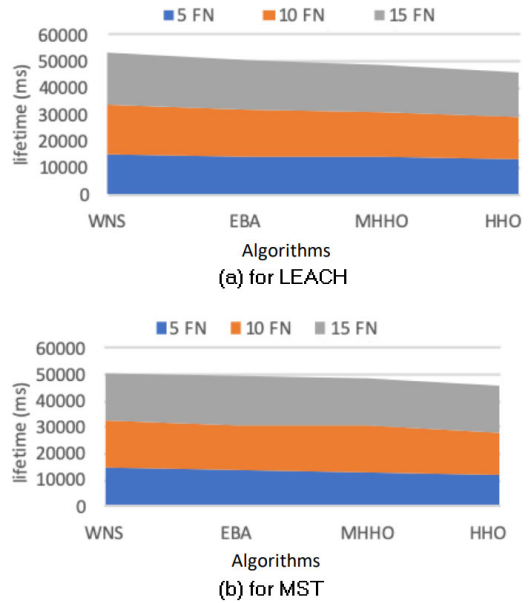


FIGURE 14. NLT of 5000 nodes using both LEACH and MST.

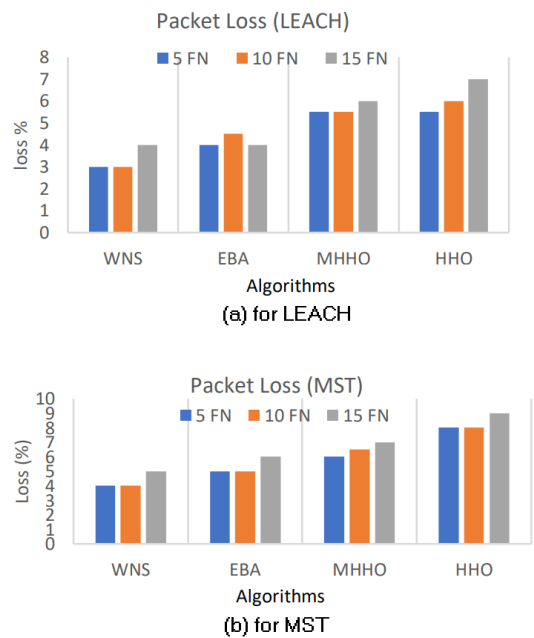


FIGURE 15. Packet loss of 500 nodes using both LEACH and MST.

computed over the lifetime of the network. This is an indicator of the efficiency of our proposed algorithms in delivering the required messages to the edge of the fog networks. The results are summarized in Figs. 15, 16, and 17. As it can be seen in Fig. 15, 500 nodes were deployed in the monitored fields with a different number of FNs. The packet loss overall reached 10% in the worst case and 3% in the best case. As it can be noted in Fig. 15 (a) and Fig. 15 (b), WNS resulted in a range of 3% to 5% packet loss while EBA packet loss ranged between 4% and 6%. MHHO packet loss was only 1% different from

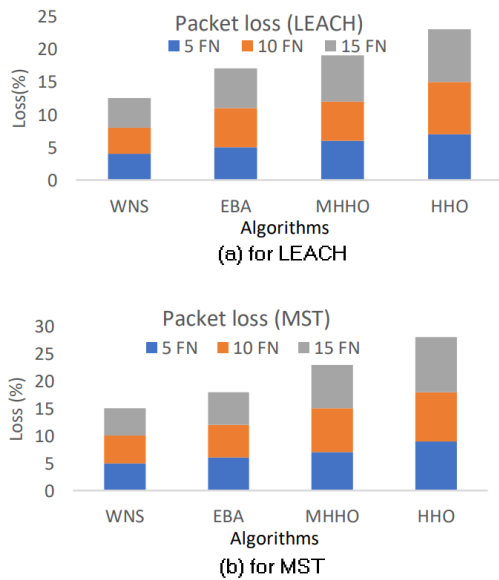


FIGURE 16. Packet loss of 1000 nodes using both LEACH and MST.

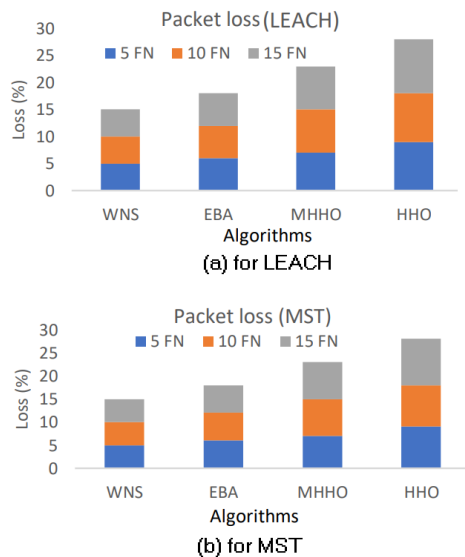


FIGURE 17. Packet loss of 5000 nodes using both LEACH and MST.

EBA's and only 2% from HHO's. Fig. 16 shows the packet loss for a 1000 node network deployed in the monitored field. Fig. 16 (a) examines the packet loss following the use of LEACH and Fig. 16 (b) examines the network packet loss with MST routing. It is noticeable that there is no huge difference in the packet loss from the previous experiments carried out with 500 nodes. The packet loss increased by almost 1% in connection with both LEACH and MSN routing algorithms. Interestingly, as indicated in Fig. 17, in a large-scale network (e.g., 5000 nodes) the packet loss does not increase substantially from the previous deployment with 1000 nodes. There is only an exceedingly small difference in packet loss (0.5%). These results confirm the energy loss

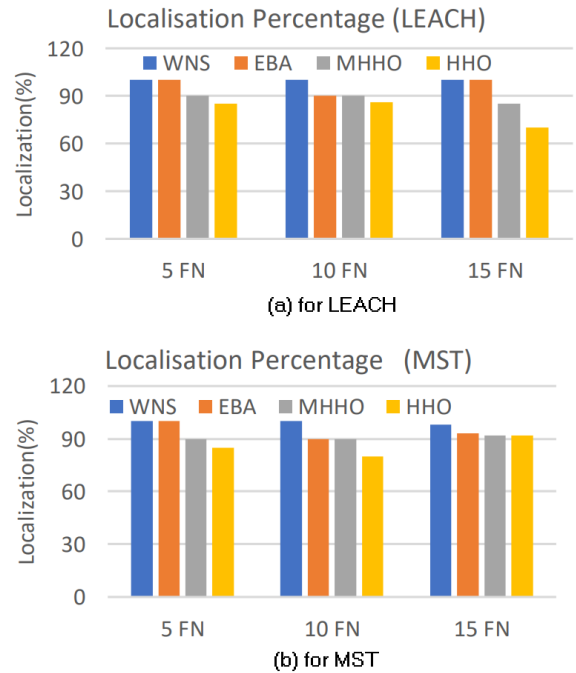


FIGURE 18. Average localization percentage of 500, 1000, and 5000 nodes using both LEACH and MST.

of 1000 networks presented in Fig. 11, where the energy loss difference was also minimal.

D. LOCALIZATION ERROR

The localization error is an indicator of how efficient the algorithms are in finding a sink node connected to each of the FNs. The results of this set of experiments are summarized in Fig. 18. This time, the results are presented differently, with the four algorithms each being compared with each different number of FNs, namely 5, 10, and 15. The results show the average over the different number of nodes, namely 500, 1000, and 5000 nodes. It is evident that the performance of the algorithms differs according to the required number of FNs. However, in most of the cases, WNS satisfied the localization requirements 100% of the time, but with 15 FNs MST was deployed. The performance of EBA was between 90% and 100% accurate while the performance of MHHO ranged between 85% and 92% accuracy which is also a reasonable performance. The performance of HHO fell in the range of 70% and 100% accuracy.

X. DISCUSSION

This section presents a discussion on the results obtained following an investigation of the logs of experiments. First, WNS emerged as the optimal solution as it scans the deployed nodes one by one and retains only the best nodes that meet the set parameters. EBA bases its selection on the weight multiplied by each parameter. Therefore, increasing the weight of one of the parameters affects the performance of the algorithm. In the previous set of experiments, the

weights were even across all of the parameters and were not prioritized. However, even after normalizing the parameters, there are some parameters that might bias the algorithm sink assignment.

A. SOME INFLUENTIAL FACTORS

The performance of HHO depends on the number of iterations implemented. In the experiments, the number of iterations was limited to 1000 due to the time taken by the algorithm to finalize the results, especially with large-scale networks (5000 nodes). The algorithm performance and conversion rate increased with the number of iterations. Further more, the algorithm operation was affected by other parameters e.g., initialization, function evaluation, and position update as follows:

$$\begin{aligned}
 O(HHO) &= O(\text{initialization}) + O(\text{evaluation}) + O(\text{position update}) \\
 &= O(N) + O((t_{max})(N)(Cof)) + O((t_{max})(N)(d)) \\
 &= O(N(1 + t_{max}(Cof + d))) \quad (29)
 \end{aligned}$$

where N is the population size; Cof is the function evaluation cost; d is the optimization problem dimension; and, t_{max} is the maximum number of iterations. Therefore, HHO performance is clearly affected by those parameters. This fact reflects the performance of the HHO algorithm in all of the experiments. To soften the complexity of HHO, the MHHO was used with a smaller number of nodes to be examined. The nodes examined were limited to the ones selected as CHs. Another important variable was the type of routing algorithm used, in this case either LEACH or MST. The LEACH works in rounds where in each round new CHs are chosen. This balances the network energy. In the experiments, the number of rounds is limited to 100ms of the simulation running time. In the case of MST, the routes are chosen once and remain fixed during NLT. This means that when the energy of some of the nodes depletes with time, some routes break up. This explains why the performance of MST is lower than that of LEACH in terms of consumed energy, lifetime, and packet loss. However, localization depends mainly on the proposed algorithms.

B. LIMITATIONS OF OUR WORK

The limitations of our proposed framework falls in the following points:

- For WNS approach, selecting the appropriate window size needs trial and error experiments which might not be practical.
- The complexity of HHO algorithm might increase with the number of hawks and iterations.
- More studies are needed to handle the communication issues between the sensor nodes and the sink node as well as between the sink nodes and the edge network.

XI. CONCLUSION

We proposed a novel framework for both fog and cloud computing considering large-scale networks. The focus was on investigating the energy efficiency of the external networks falling outside the fog network. Four different algorithms for energy saving in WSNs were proposed based on the smart positioning of SNs as near as possible to the FNs. The algorithms sought to locate the SNs that satisfied the parameters of high energy nodes, located nearby the FNs, node degree, and node processing capabilities. Different algorithms were used including Pareto Optimality, EBA with weights normalization, and MHHO. An extensive set of experiments was conducted. Different criteria were used for the evaluation to show that the proposed algorithms were superior to HHO. Future work in this area could focus on trying different routing techniques and involving different types of nodes, including mobile nodes, vehicle nodes, computer nodes and other types of devices.

ACKNOWLEDGMENT

The authors thank to the anonymous reviewers for their appreciative and constructive comments on the draft of this paper.

REFERENCES

- [1] R. A. Ramadan, E. J. Alreshidi, and M. H. Sharif, "Energy efficient framework for fog network based on multisink wireless sensor networks," in *Proc. 2nd Int. Conf. Comput. Inf. Sci. (ICIS)*, Sakaka, Saudi Arabia, Oct. 2020, pp. 1–5.
- [2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st Ed. Workshop Mobile Cloud Comput. (MCC)*, M. Gerla and D. Huang, Eds., Helsinki, Finland, 2012, pp. 13–16.
- [3] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.
- [4] T. Taleb and A. Ksentini, "Follow me cloud: Interworking federated clouds and distributed mobile networks," *IEEE Netw.*, vol. 27, no. 5, pp. 12–19, Sep./Oct. 2013.
- [5] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, "Mobile-edge computing architecture: The role of MEC in the Internet of Things," *IEEE Consum. Electron. Mag.*, vol. 5, no. 4, pp. 84–91, Oct. 2016.
- [6] E. Baccarelli, M. Scarpiniti, A. Momenzadeh, and S. S. Ahrabi, "Learning-in-the-fog (LiFo): Deep learning meets fog computing for the minimum-energy distributed early-exit of inference in delay-critical IoT realms," *IEEE Access*, vol. 9, pp. 25716–25757, 2021.
- [7] L. Shu, J. Zhao, Y. Gong, C. Luo, and T. Gordon, "IEEE access special section editorial: Communication and fog/edge computing toward intelligent connected vehicles (ICVS)," *IEEE Access*, vol. 9, pp. 63740–63744, 2021.
- [8] T. A.-N. Abdali, R. Hassan, A. H. M. Aman, and Q. N. Nguyen, "Fog computing advancement: Concept, architecture, applications, advantages, and open issues," *IEEE Access*, vol. 9, pp. 75961–75980, 2021.
- [9] E. Kilcioglu, H. Mirghasemi, I. Stupia, and L. Vandendorpe, "An energy-efficient fine-grained deep neural network partitioning scheme for wireless collaborative fog computing," *IEEE Access*, vol. 9, pp. 79611–79627, 2021.
- [10] X. Zhang and W. Si, "Efficient auditing scheme for secure data storage in fog-to-cloud computing," *IEEE Access*, vol. 9, pp. 37951–37960, 2021.
- [11] O. Consortium. (2017). *IoT Journal—OpenFog Consortium Releases Reference Architecture for Fog Computing*. [Online]. Available: https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf
- [12] N. Wang, B. Varghese, M. Matthaiou, and D. S. Nikolopoulos, "ENORM: A framework for edge node resource management," *CoRR*, vol. abs/1709.04061, pp. 1–14, Sep. 2017.

- [13] S. Wang, R. Uргаonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge-clouds," in *Proc. IFIP Neww. Conf. (IFIP Neww.)*, R. Kacimi and Z. Mammeri, Eds., May 2015, pp. 1–9.
- [14] S. Wang, M. Zafer, and K. K. Leung, "Online placement of multi-component applications in edge computing environments," *IEEE Access*, vol. 5, pp. 2514–2533, 2017.
- [15] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, "Live service migration in mobile edge clouds," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 140–147, Feb. 2018.
- [16] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, Jul. 2016, pp. 1451–1455.
- [17] G. Orsini, D. Bade, and W. Lamersdorf, "CloudAware: A context-adaptive middleware for mobile edge and cloud computing applications," in *Proc. IEEE 1st Int. Workshops Found. Appl. Self Syst. (FAS W)*, S. Elnikety, P. R. Lewis, and C. M. Schloer, Eds., Augsburg, Germany, 2016, pp. 216–221.
- [18] M. H. Sharif, I. Despot, and S. Uyaver, "A proof of concept for home automation system with implementation of the Internet of Things standards," *Periodicals Eng. Natural Sci.*, vol. 6, no. 1, pp. 95–106, 2018.
- [19] J. Du, W. Liu, G. Lu, J. Jiang, D. Zhai, F. R. Yu, and Z. Ding, "When mobile-edge computing (MEC) meets nonorthogonal multiple access (NOMA) for the Internet of Things (IoT): System design and optimization," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 7849–7862, May 2021.
- [20] K. Haseeb, I. Ud Din, A. Almogren, N. Islam, and A. Altameem, "RTS: A robust and trusted scheme for IoT-based mobile wireless mesh networks," *IEEE Access*, vol. 8, pp. 68379–68390, 2020.
- [21] B. Safaei, A. Mohammadsalehi, K. T. Khoosani, S. Zarbaf, A. M. H. Monazzah, F. Samie, L. Bauer, J. Henkel, and A. Ejlali, "Impacts of mobility models on RPL-based mobile IoT infrastructures: An evaluative comparison and survey," *IEEE Access*, vol. 8, pp. 167779–167829, 2020.
- [22] T. N. D. Mathaba, "Optimal sink-node placement and routing for an energy efficient two-tier wireless sensor network," in *Proc. Int. Conf. Adv. Big Data, Comput. Data Commun. Syst. (iABCD)*, Aug. 2018, pp. 1–6.
- [23] S. Tahmasebi, M. Safi, S. Zolfi, M. R. Maghsoudi, H. R. Faragardi, and H. Fotouhi, "Cuckoo-PC: An evolutionary synchronization-aware placement of SDN controllers for optimizing the network performance in WSNs," *Sensors*, vol. 20, no. 11, p. 3231, Jun. 2020.
- [24] R. Ramadan, H. El-Rewini, and K. Abdelghany, "Optimal and approximate approaches for deployment of heterogeneous sensing devices," *EURASIP J. Wireless Commun. Netw.*, vol. 2007, no. 1, pp. 1–14, Dec. 2007.
- [25] A. E. El-Din, R. A. Ramadan, A. A. Elmagid, and S. A. Aly, "Novel scheme of fuzzy based concealing sink node with fake holes (F-CSH)," in *Proc. Int. Conf. Ambient Syst., Netw. Technol. (ANT), 4th Int. Conf. Sustainable Energy Inf. Technol. (SEIT)*, E. M. Shakshuki and A. Yasar, Eds., Hasselt, Belgium, vol. 32, 2014, pp. 1174–1179.
- [26] S. Bhattacharjee and K. Agarwal, "Energy efficient multiple sink placement in wireless sensor networks," in *Proc. 4th Int. Conf. Netw., Syst. Secur. (NSysS)*, Dhaka, Bangladesh, Dec. 2017, pp. 1–7.
- [27] P. Bose and M. Gurusamy, "Bacteria foraging algorithm based optimal multi sink placement in wireless sensor networks," *J. Intell. Syst.*, vol. 27, no. 4, pp. 609–618, Oct. 2018.
- [28] M. Sharif and O. Gursoy, "Parallel computing for artificial neural network training using Java native socket programming," *Periodicals Eng. Natural Sci.*, vol. 6, no. 1, pp. 1–10, 2018.
- [29] I. Snigdha, D. Gosain, and N. Gupta, "Optimal sink placement in backbone assisted wireless sensor networks," *Egyptian Inform. J.*, vol. 17, no. 2, pp. 217–225, 2016.
- [30] S. Sun, J. Zhao, X. Feng, J. Zhang, and J. Luo, "Mobile multi-sink nodes path planning algorithm concerned with energy balance in wireless sensor networks," *IEEE Access*, vol. 7, pp. 96942–96952, 2019.
- [31] E. H. Houssein, M. R. Saad, K. Hussain, W. Zhu, H. Shaban, and M. Hassaballah, "Optimal sink node placement in large scale wireless sensor networks based on Harris' hawk optimization algorithm," *IEEE Access*, vol. 8, pp. 19381–19397, 2020.
- [32] K. Hinton, F. Jalali, and A. Matin, "Energy consumption modelling of optical networks," *Photonic Netw. Commun.*, vol. 30, no. 1, pp. 4–16, Aug. 2015.
- [33] J. Delsing, "Local cloud Internet of Things automation: Technology and business model features of distributed Internet of Things automation solutions," *IEEE Ind. Electron. Mag.*, vol. 11, no. 4, pp. 8–21, Dec. 2017.
- [34] F. Jalali, A. Vishwanath, J. de Hoog, and F. Suits, "Interconnecting fog computing and microgrids for greening IoT," in *Proc. IEEE Innov. Smart Grid Technol. Asia (ISGT-Asia)*, Nov. 2016, pp. 693–698.
- [35] N. Gaouar and M. Lehsaini, "Toward vehicular cloud/fog communication: A survey on data dissemination in vehicular ad hoc networks using vehicular cloud/fog computing," *Int. J. Commun. Syst.*, vol. 34, no. 13, pp. 1–27, Sep. 2021.
- [36] R. Verma and S. Chandra, "A systematic survey on fog steered IoT: Architecture, prevalent threats and trust models," *Int. J. Wireless Inf. Netw.*, vol. 28, no. 1, pp. 116–133, Mar. 2021.
- [37] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet," *ACM Comput. Surv.*, vol. 52, no. 6, pp. 125:1–125:36, Oct. 2019.
- [38] W. Almbaideen and M. Altarawneh, "Fog computing: Survey on decoy information technology," *Int. J. Secur. Netw.*, vol. 15, no. 2, pp. 111–121, 2020.
- [39] H. A. Shehu, M. H. Sharif, and R. A. Ramadan, "Distributed mutual exclusion algorithms for intersection traffic problems," *IEEE Access*, vol. 8, pp. 138277–138296, 2020.
- [40] M. H. Sharif, "Laser-based algorithms meeting privacy in surveillance: A survey," *IEEE Access*, vol. 9, pp. 92394–92419, 2021, doi: 10.1109/ACCESS.2021.3092687.
- [41] M. H. Shahid, A. R. Hameed, S. U. Islam, H. A. Khattak, I. U. Din, and J. J. P. C. Rodrigues, "Energy and delay efficient fog computing using caching mechanism," *Comput. Commun.*, vol. 154, pp. 534–541, Mar. 2020.
- [42] L. N. T. Huynh, Q.-V. Pham, T. D. T. Nguyen, M. D. Hossain, Y.-R. Shin, and E.-N. Huh, "Joint computational offloading and data-content caching in NOMA-MEC networks," *IEEE Access*, vol. 9, pp. 12943–12954, 2021.
- [43] W. Fan, J. Han, J. Chen, Y. Liu, and F. Wu, "Probabilistic computation offloading and data caching assisted by mobile-edge-computing-enabled base stations," *Ann. Télécommun.*, vol. 76, nos. 7–8, pp. 447–465, Aug. 2021.
- [44] C. Pu, "Pro NDN: MCDM-based interest forwarding and cooperative data caching for named data networking," *J. Comput. Netw. Commun.*, vol. 2021, Mar. 2021, Art. no. 6640511.
- [45] M. H. Sharif, "High-performance mathematical functions for single-core architectures," *J. Circuits, Syst. Comput.*, vol. 23, no. 4, Apr. 2014, Art. no. 1450051.
- [46] S. Kitanov and T. Janevski, "Energy efficiency of fog computing and networking services in 5G networks," in *Proc. 17th Int. Conf. Smart Technol. (IEEE EUROCON)*, L. Karadzinov, G. Cvetkovski, and P. Latkoski, Eds., Jul. 2017, pp. 491–494.
- [47] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974–117017, 2020.
- [48] H. A. Alharbi, T. E. H. Elgorashi, and J. M. H. Elmihghani, "Energy efficient virtual machines placement over cloud-fog network architecture," *IEEE Access*, vol. 8, pp. 94697–94718, 2020.
- [49] H. Barani, Y. Jaradat, H. Huang, Z. Li, and S. Misra, "Effect of sink location and redundancy on multi-sink wireless sensor networks: A capacity and delay analysis," *IET Commun.*, vol. 12, no. 8, pp. 941–947, May 2018.
- [50] M. Carlos-Mancilla, E. Lopez-Mellado, and M. Siller-Gonzalez, "A localized multi-sink multi-hop algorithm for wireless sensor networking," in *Proc. Global Inf. Infrastructure Netw. Symp. (GIIS)*, Guadalajara, Mexico, Oct. 2015, pp. 1–6.
- [51] T. Luo, H. Li, S. Ru, W. Tong, and Y. Xu, "Multiple sink location problem in path networks with a combinational objective," *Optim. Lett.*, vol. 15, no. 2, pp. 733–755, Mar. 2021.
- [52] Z. W. Hussien, D. S. Qawasmeh, and M. Shurman, "MSCLP: Multi-sinks cluster-based location privacy protection scheme in WSNs for IoT," in *Proc. 32nd Int. Conf. Microelectron. (ICM)*, Aqaba, Jordan, Dec. 2020, pp. 1–4.
- [53] S. S. Kashi, "A load balanced location service for location information management of multi-sink wireless sensor networks," *Computing*, vol. 100, no. 2, pp. 93–117, Feb. 2018.
- [54] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Hawaii Int. Conf. Syst. Sci. (HICSS)*, 2000, pp. 1–10.

- [55] P. Habibi, M. Farhoudi, S. Kazemian, S. Khorsandi, and A. Leon-Garcia, "Fog computing: A comprehensive architectural survey," *IEEE Access*, vol. 8, pp. 69105–69133, 2020.
- [56] H. V. Nejad, A. Tavakolifar, C. Bhatt, N. Hanafi, N. Gholizadeh, R. Khatooni, and H. Behzadian, "A survey on context-aware fog computing systems," *Computación y Sistemas*, vol. 25, no. 1, pp. 5–12, Feb. 2021.
- [57] I. Ahammad, A. R. Khan, and Z. U. Salehin, "A review on cloud, fog, roof, and dew computing: IoT perspective," *Int. J. Cloud Appl. Comput.*, vol. 11, no. 4, pp. 14–41, Oct. 2021.
- [58] S. Sharma and M. Sajid, "Integrated fog and cloud computing issues and challenges," *Int. J. Cloud Appl. Comput.*, vol. 11, no. 4, pp. 174–193, Oct. 2021.
- [59] T. Khalid, M. A. K. Abbasi, M. Zuraiz, A. N. Khan, M. Ali, R. W. Ahmad, J. J. P. C. Rodrigues, and M. Aslam, "A survey on privacy and access control schemes in fog computing," *Int. J. Commun. Syst.*, vol. 34, no. 2, pp. 1–39, Jan. 2021.
- [60] K. Matrouk and K. Alatoun, "Scheduling algorithms in fog computing: A survey," *Int. J. Netw. Distrib. Comput.*, vol. 9, no. 1, pp. 59–74, 2021.
- [61] J. Singh, P. Singh, and S. S. Gill, "Fog computing: A taxonomy, systematic review, current trends and research challenges," *J. Parallel Distrib. Comput.*, vol. 157, pp. 56–85, Nov. 2021.
- [62] N. Merabtine, D. Djenouri, and D.-E. Zegour, "Towards energy efficient clustering in wireless sensor networks: A comprehensive review," *IEEE Access*, vol. 9, pp. 92688–92705, 2021.
- [63] Z. Ullah, "A survey on hybrid, energy efficient and distributed (HEED) based energy efficient clustering protocols for wireless sensor networks," *Wireless Pers. Commun.*, vol. 112, no. 4, pp. 2685–2713, Jun. 2020.
- [64] M. Abdel-Basset, D. El-Shahat, M. Elhoseny, and H. Song, "Energy-aware Metaheuristic algorithm for industrial-Internet-of-Things task scheduling problems in fog computing applications," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12638–12649, Aug. 2021.
- [65] M. A. Hossain, R. M. Noor, K.-L.-A. Yau, S. R. Azzuhri, M. R. Z'Abbar, I. Ahmady, and M. R. Jabbarpour, "Multi-objective Harris hawks optimization algorithm based 2-Hop routing algorithm for CR-VANET," *IEEE Access*, vol. 9, pp. 58230–58242, 2021.
- [66] R. Shanmugam and B. Kaliaperumal, "An energy-efficient clustering and cross-layer-based opportunistic routing protocol (CORP) for wireless sensor network," *Int. J. Commun. Syst.*, vol. 34, no. 7, pp. 1–31, May 2021.
- [67] R. Sharma and S. Prakash, "HHO-LPWSN: Harris hawks optimization algorithm for sensor nodes localization problem in wireless sensor networks," *ICST Trans. Scalable Inf. Syst.*, vol. 8, no. 31, Jul. 2018, Art. no. 168807.
- [68] T. Suzuki, M. Bandai, and T. Watanabe, "DispersiveCast: Dispersive packets transmission to multiple sinks for energy saving in sensor networks," in *Proc. IEEE 17th Int. Symp. Pers., Indoor Mobile Radio Commun.*, Helsinki, Finland, Sep. 2006, pp. 1–5.
- [69] S. Cha, M. Jo, J.-E. Lee, D.-Y. Kim, S. Kang, K.-H. Cho, N. Lee, and Y. Kim, "Hierarchical node clustering approach for energy savings in WSNs," in *Proc. 5th ACIS Int. Conf. Softw. Eng. Res., Manage. Appl. (SERA)*, Busan, South Korea, Aug. 2007, pp. 253–259.
- [70] S. E. Khediri, N. Nasri, R. U. Khan, and A. Kachouri, "An improved energy efficient clustering protocol for increasing the life time of wireless sensor networks," *Wireless Pers. Commun.*, vol. 116, no. 1, pp. 539–558, Jan. 2021.
- [71] M. A. Aydin, B. Karabekir, and A. H. Zaim, "Energy efficient clustering-based mobile routing algorithm on WSNs," *IEEE Access*, vol. 9, pp. 89593–89601, 2021.
- [72] H. Koyuncu, G. S. Tomar, and D. Sharma, "A new energy efficient multitier deterministic energy-efficient clustering routing protocol for wireless sensor networks," *Symmetry*, vol. 12, no. 5, p. 837, May 2020.
- [73] S.-V. Oprea and A. Băra, "Edge and fog computing using IoT for direct load optimization and control with flexibility services for citizen energy communities," *Knowl.-Based Syst.*, vol. 228, Sep. 2021, Art. no. 107293.
- [74] S. Tuli, S. R. Poojara, S. N. Srirama, G. Casale, and N. R. Jennings, "COSCO: Container orchestration using co-simulation and gradient based optimization for fog computing environments," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 1, pp. 101–116, Jan. 2022.
- [75] P. Singh and R. Singh, "Energy-efficient delay-aware task offloading in fog-cloud computing system for IoT sensor applications," *J. Netw. Syst. Manage.*, vol. 30, no. 1, p. 14, Jan. 2022.
- [76] E. M. Borujeni, D. Rahbari, and M. Nickray, "Fog-based energy-efficient routing protocol for wireless sensor networks," *J. Supercomput.*, vol. 74, no. 12, pp. 6831–6858, Dec. 2018.
- [77] M. Y. Arafat and S. Moh, "Bio-inspired approaches for energy-efficient localization and clustering in UAV networks for monitoring wildfires in remote areas," *IEEE Access*, vol. 9, pp. 18649–18669, 2021.
- [78] M. Abdel-Basset, N. Moustafa, R. Mohamed, O. M. Elkomy, and M. Abouhawah, "Multi-objective task scheduling approach for fog computing," *IEEE Access*, vol. 9, pp. 126988–127009, 2021.
- [79] A. P. Abidoye and B. Kabaso, "Energy-efficient hierarchical routing in wireless sensor networks based on fog computing," *EURASIP J. Wireless Commun. Netw.*, vol. 2021, no. 1, p. 8, Dec. 2021.
- [80] A. Biswas, A. Majumdar, S. Das, and K. L. Baishnab, "OCSO-CA: Opposition based competitive swarm optimizer in energy efficient IoT clustering," *Frontiers Comput. Sci.*, vol. 16, no. 1, Feb. 2022, Art. no. 161501.
- [81] W. Bai, Z. Ma, Y. Han, M. Wu, Z. Zhao, M. Li, and C. Wang, "Joint optimization of computation offloading, data compression, energy harvesting, and application scenarios in fog computing," *IEEE Access*, vol. 9, pp. 45462–45473, 2021.
- [82] R. Maher and O. A. Nasr, "DropStore: A secure backup system using multi-cloud and fog computing," *IEEE Access*, vol. 9, pp. 71318–71327, 2021.
- [83] M. Krishnan, S. Yun, and Y. M. Jung, "Enhanced clustering and ACO-based multiple mobile sinks for efficiency improvement of wireless sensor networks," *Comput. Netw.*, vol. 160, pp. 33–40, Sep. 2019.
- [84] S. Loganathan, J. Arumugam, and V. Chinnababu, "An energy-efficient clustering algorithm with self-diagnosis data fault detection and prediction for wireless sensor networks," *Concurrency Comput., Pract. Exper.*, vol. 33, no. 17, pp. 1–27, Sep. 2021.
- [85] M. Barzegaran and P. Pop, "Communication scheduling for control performance in TSN-based fog computing platforms," *IEEE Access*, vol. 9, pp. 50782–50797, 2021.
- [86] H. Suleiman and M. Hamdan, "Adaptive probabilistic model for energy-efficient distance-based clustering in WSNs (Adapt-P): A LEACH-based analytical study," *J. Wireless Mob. Netw. Ubiquitous Comput. Dependable Appl.*, vol. 12, no. 3, pp. 65–86, 2021.
- [87] D. Mehta and S. Saxena, "Load-based node ranked low-energy adaptive clustering hierarchy: An enhanced energy-efficient algorithm for cluster head selection in wireless sensor networks," *Concurrency Comput., Pract. Exper.*, vol. 33, no. 21, pp. 1–7, Nov. 2021.
- [88] Z. Zhou and Y. Niu, "An energy efficient clustering algorithm based on annulus division applied in wireless sensor networks," *Wireless Pers. Commun.*, vol. 115, no. 3, pp. 2229–2241, Dec. 2020.
- [89] F. Aliyu, T. Sheltami, M. Deriche, and N. Nasser, "Human immune-based intrusion detection and prevention system for fog computing," *J. Netw. Syst. Manage.*, vol. 30, no. 1, p. 11, Jan. 2022.
- [90] U. Farooq, M. W. Shabir, M. A. Javed, and M. Imran, "Intelligent energy prediction techniques for fog computing networks," *Appl. Soft Comput.*, vol. 111, Nov. 2021, Art. no. 107682.
- [91] L. Wu, X. Lin, Z. Chen, J. Huang, H. Liu, and Y. Yang, "An efficient binary convolutional neural network with numerous skip connections for fog computing," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11357–11367, Jul. 2021.
- [92] B. Aoun and R. Boutaba, "Clustering in WSN with latency and energy consumption constraints," *J. Netw. Syst. Manage.*, vol. 14, no. 3, pp. 415–439, Sep. 2006.
- [93] S. Hebal, L. Louail, and S. Harous, "Latency and energy optimization using MAC-aware routing for WSNs," *Int. J. Bus. Data Commun. Netw.*, vol. 16, no. 1, pp. 19–27, Jan. 2020.
- [94] S. E. Essalhi, M. R. El Fenni, and H. Chafnaji, "Smart energy management for fog-enabled IoT network," in *Proc. 13th Int. Conf. Intell. Syst., Theories Appl.*, Rabat, Morocco, Sep. 2020, p. 43.
- [95] A. Sutagundar and P. Sangulagi, "Fog computing based information classification in sensor cloud-agent approach," *Expert Syst. Appl.*, vol. 182, Nov. 2021, Art. no. 115232.
- [96] K. Ullah and H. Khan, "Fog-LAEEBA: Fog-assisted link aware and energy efficient protocol for wireless body area network," *Acta Universitatis Sapientiae, Inf.*, vol. 13, no. 1, pp. 180–194, Jun. 2021.
- [97] F. H. Rahman, S. H. S. Newaz, T.-W. Au, W. S. Suhaili, M. A. P. Mahmud, and G. M. Lee, "EnTruVe: ENergy and TRUst-aware virtual machine allocation in VEHICLE fog computing for catering applications in 5G," *Future Gener. Comput. Syst.*, vol. 126, pp. 196–210, Jan. 2022.

- [98] A. Paris, H. Mirghasemi, I. Stupia, and L. Vandendorpe, "Leveraging user-diversity in energy-efficient edge-facilitated collaborative fog computing," *IEEE Access*, vol. 9, pp. 95636–95650, 2021.
- [99] H. V. Sampaio, C. B. Westphall, F. Koch, R. do Nascimento Boing, and R. N. S. Cruz, "Autonomic energy management with fog computing," *Comput. Electr. Eng.*, vol. 93, Jul. 2021, Art. no. 107246.
- [100] S. Ijaz, E. U. Munir, S. G. Ahmad, M. M. Rafique, and O. F. Rana, "Energy-makespan optimization of workflow scheduling in fog-cloud computing," *Computing*, vol. 103, no. 9, pp. 2033–2059, Sep. 2021.
- [101] W. K. Lai, C.-S. Shieh, C.-S. Ho, and Y.-R. Chen, "A clustering-based energy saving scheme for dense small cell networks," *IEEE Access*, vol. 7, pp. 2880–2893, 2019.
- [102] A. Kavitha and R. L. Velusamy, "Simulated annealing and genetic algorithm-based hybrid approach for energy-aware clustered routing in large-range multi-sink wireless sensor networks," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 35, no. 2, pp. 96–116, 2020.
- [103] N. M. Zamry, A. Zainal, and M. A. Rassam, "LEACH-CR: Energy saving hierarchical network protocol based on low-energy adaptive clustering hierarchy for wireless sensor networks," in *Proc. 3rd Int. Cyber Resilience Conf. (CRC)*, Langkawi Island, Malaysia, Jan. 2021, pp. 1–6.
- [104] W. He, "Energy-saving algorithm and simulation of wireless sensor networks based on clustering routing protocol," *IEEE Access*, vol. 7, pp. 172505–172514, 2019.
- [105] H. Vo, "Implementing energy saving techniques for sensor nodes in IoT applications," *EAI Endorsed Trans. Ind. Netw. Intell. Syst.*, vol. 5, no. 17, Dec. 2018, Art. no. 156076.
- [106] K. Debasis, M. P. Singh, and R. Gupta, "An energy saving medium access control protocol for wireless sensor networks," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 1, pp. 1435–1448, Jan. 2021.
- [107] A. E. El-Din, R. A. Ramadan, and M. B. Fayek, "VEGK: Virtual ECC group key for wireless sensor networks," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, San Diego, CA, USA, Jan. 2013, pp. 364–368.
- [108] R. A. Ramadan, "Efficient intrusion detection algorithms for smart cities-based wireless sensing technologies," *J. Sensor Actuator Netw.*, vol. 9, no. 3, p. 39, Aug. 2020.
- [109] F. Medhat, R. A. Ramadan, and I. Talkhan, "CAM: Clustering algorithms for multimodal WSN," *Int. J. Syst. Dyn. Appl.*, vol. 2, no. 4, pp. 47–67, Oct. 2013.
- [110] R. C. Prim, "Shortest connection networks and some generalizations," *Bell Syst. Tech. J.*, vol. 36, no. 6, pp. 1389–1401, 1957.
- [111] J. B. Kruskal, Jr., "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proc. Amer. Math. Soc.*, vol. 7, no. 1, pp. 48–50, Feb. 1956.
- [112] P. Du, J. Wang, Y. Hao, T. Niu, and W. Yang, "A novel hybrid model based on multi-objective Harris hawks optimization algorithm for daily PM_{2.5} and PM₁₀ forecasting," *CoRR*, vol. abs/1905.13550, pp. 1–24, Nov. 2019.
- [113] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.
- [114] H. Zhou, D. Luo, and D. Zuo, "Modeling of node energy consumption for wireless sensor networks," *Wireless Sensor Netw.*, vol. 3, no. 1, pp. 18–23, Jan. 2011.
- [115] M. Singh and S. K. Soni, "Network lifetime enhancement of WSNs using correlation model and node selection algorithm," *Ad Hoc Netw.*, vol. 114, Apr. 2021, Art. no. 102441.
- [116] C. Cobos-Sánchez, J. A. Vilchez-Membrilla, A. Campos-Jiménez, and F. J. García-Pacheco, "Pareto optimality for multiobjective optimization of continuous linear operators," *Symmetry*, vol. 13, no. 4, p. 661, Apr. 2021.
- [117] S. Duleba and S. Moslem, "Examining Pareto optimality in analytic hierarchy process on real data: An application in public transport service development," *Expert Syst. Appl.*, vol. 116, pp. 21–30, Feb. 2019.
- [118] W. Li, Y. Ding, Y. Yang, R. S. Sherratt, J. H. Park, and J. Wang, "Parameterized algorithms of fundamental NP-hard problems: A survey," *Hum.-Centric Comput. Inf. Sci.*, vol. 10, no. 1, p. 29, Dec. 2020.
- [119] P. K. Donta, T. Amgoth, and C. S. R. Annavarapu, "An extended ACO-based mobile sink path determination in wireless sensor networks," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 10, pp. 8991–9006, Oct. 2021.
- [120] Cupcarbon. (2020). *CupCarbon*. [Online]. Available: <http://cupcarbon.com>



RABIE A. RAMADAN (Member, IEEE) received the Ph.D. degree in computer engineering from Cairo University and Southern Methodist University (SMU), Dallas, TX, USA, in 2007. He is currently the Chair of the Computer Engineering Department, Hail University, Hail, Saudi Arabia. He is also an Associate Professor at Cairo University, Cairo, Egypt, and Hail University. He is also a Co-Founder of the IEEE Computational Intelligence, Egypt Chapter. He is also the Director of Industrial Partnership Program (CISCO, Oracle, and Microsoft) with the College of Computer Science and Engineering, Hail University. In addition, he is the Founder of FabLab and the Center of Programming and Applications, Hail University, sponsored by the Ministry of Education, Saudi Arabia. Moreover, he is the Founder of TripleTech expert house sponsored by Hail University. He is the author of more than 125 articles in the field of big data, cloud computing, the IoT, and computational intelligence. He served as a general chair, a program committee chair, and a TPC for many of the conferences and journals, including *Web of Science* and IEEE TRANSACTIONS journals. In addition, he served as the Co-Chair of the International Conference on Recent Advances in Computer Systems (RACS-2015) and the 2nd National Computing Colleges Conference (NC3 2017) held at Hail University. He is also the General Chair of the International Conference on new Computer Science and Engineering Trends (NCSET2020). In addition, he was the Chair of the ACM Programming Competition in Saudi Arabia associated with NC3 2017 Conference.



MD. HAIDAR SHARIF (Member, IEEE) was born in Jessore, Bangladesh, in 1977. He received the B.Sc. degree in electronics and computer science from Jahangirnagar University, Bangladesh, in 2001, the M.Sc. degree in computer engineering from Duisburg-Essen University, Germany, in 2006, and the Ph.D. degree in computer science from the University of Lille 1, France, in 2010. From January 2011 to January 2016, he had been working at Izmir University Bakırçay (old Gediz), Turkey, as an Assistant Professor. From April 2016 to June 2017, he had been working at the International University of Sarajevo, Bosnia and Herzegovina, as an Assistant Professor. From October 2017 to September 2018, he had been working at International Balkan University, North Macedonia, as an Associate Professor. He has been working as an Associate Professor with University of Hail, Saudi Arabia, since November 2018. He is the author of the book entitled *Sundry Applications and Computations of sin() & cos()*. His research interests include applications of computer vision, deep learning, BCI (brain-computer interface) technologies, computer architecture, and computational intelligence algorithms.



EISSA JABER ALRESHIDI was born in Hail, Saudi Arabia, in February 1986. He received the master's degree from the University of Birmingham, U.K., and the Ph.D. degree from Cardiff University, U.K. He is involved in various scientific research activities at the Department of Computer Science and Engineering, University of Hail. He is specialized in computer sciences and engineering with an area interest in smart sustainable systems, cloud computing, artificial intelligence, knowledge-base management, information modeling, and software engineering. Currently, he is also involved in various research and development activities within the University of Hail and abroad as a consultant as well as a project leader for several projects.



MD. HARIS UDDIN SHARIF (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree in information technology with the University of the Cumberlands, Williamsburg, KY, USA. Besides, he has solid hands-on programming experience in developing enterprise software applications. Since last few years, he has been leading a team of developers towards successful project accomplishments, establishment of good practices, and production of high-quality software applications. He possesses exceptional project and resource management skills. His research interests include cyber security, artificial intelligence, deep learning, computer vision, pattern recognition, advanced software engineering, blockchain, and cryptography.



IBRAHIM FURKAN INCE received the Ph.D. degree from the Department of Information Technologies (IT) Convergence Design, Kyungsoong University, South Korea, in August 2010. Immediately after, he did postdoctoral studies at The University of Tokyo, Japan, from October 2010 to April 2012. Later, from May 2012 to May 2014, he worked in an IT company named Hanwul Multimedia Communication Company Ltd. (HMC), South Korea, as a Chief Research Engineer. Then, he made a definite return to Turkey and worked as an Assistant Professor with the Department of Computer Engineering, Gediz University, from November 2014 to July 2016. Later, after taking part in joint projects with the Department of Electronics Engineering, Kyungsoong University, as a Freelance Researcher, for a period of three years, he started working as an Assistant Professor with the Department of Management Information Systems (MIS), Nişantaşı University, in October 2019, and after working there as a full-time Ph.D. Faculty Member, for a period of six months, he returned back to South Korea after taking an unpaid leave permission from Nişantaşı University. He is currently a Ph.D. Faculty Member with the Department of Digital Game Design, Nişantaşı University, and since March 2020, he has been actively working as an Assistant Research Professor with the Department of Electronics Engineering, Kyungsoong University. His research interests include image processing, computer vision, machine learning, pattern recognition, and human-computer interaction.



HIQMET KAMBERAJ received the Bachelor of Science degree in physics from the University of Tirana, in 1996, the Master of Science degree in physics from the University of Siegen, in 2000, and the Ph.D. degree in computational physics from Manchester Metropolitan University, in 2005. He completed his postdoctoral studies from the University of Minnesota, Arizona State University, and the National Institute of Nanotechnology (University of Edmonton). He acted as the Dean of the Faculty of Engineering, from January 2017 to September 2019. He is currently a Professor at International Balkan University, Skopje, North Macedonia. He has published more than 60 articles and book chapters in reputed journals and served as the Editor-in-Chief, an editorial board member, and an ad-hoc reviewer of repute. In April 2020, he published the first book at Springer Nature in Computation Science Series entitled *Molecular Dynamics Simulations in Statistical Physics: Theory and Applications*. In September 2021, he published his second book at De Gruyter on Classical Mechanics. His articles focus on understanding the structure, dynamics, and thermodynamics of macromolecular systems using the laws of physics, biochemistry, applied mathematics, and machine learning approaches. He has participated in more than 30 international conferences presenting his work.



AHMED Y. KHEDR is currently an Associate Professor of systems and computer engineering at Al-Azhar University, Cairo, Egypt. He is also working in computer science and engineering with Hail University, Hail, Saudi Arabia. He was funded by the Egyptian Government to visit SMU at USA and conduct research in mobile computing with the PDA Mobile Research Group. His research interests include wireless sensor networks, cloud computing, big data, and e-learning algorithms.

• • •