As per UGC guidelines an electronic bar code is provided to seure your paper

International Journal for Modern Trends in Science and Technology, 7(12): 101-108, 2021

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST0712018
Available online at: http://www.ijmtst.com/vol7issue12.html

A Secured and Energy-Efficient Clustering in the Internet of Thinks using ABC

Shreyas J[1] | Mahesh Biradar[2] | Udayaprasad P K[3] | Srinidhi N N[4] | Dharamendra Chouhan[5] | Dilip Kumar S M[6]

[1]Software Engineer, Creencia Technology Private Limited, Bengaluru, Karnataka, India
[2,3,5,6] Department Computer Science, University Visvesvaraya College of Engineering, Bengaluru, Karnataka, India
[4]Department Computer Science and Engineering, Reva University, Bengaluru, Karnataka, India

To Cite this Article
Shreyas J, Mahesh Biradar, Udayaprasad P K, Srinidhi N N, Dharamendra Chouhan and Dilip Kumar S M. A Secured and Energy-Efficient Clustering in the Internet of Thinks using ABC. *International Journal for Modern Trends in Science and Technology* 2021, 7 pp. 101-108. https://doi.org/10.46501/IJMTST0712018

Article Info
Received: 28 October 2021; Accepted: 01 December 2021; Published: 07 December 2021

ABSTRACT

Remote communication on the Web of Things (IoT) requires context-aware information transmission conventions. Creating an energy-efficient clustering instrument is the essential challenge in information transmission over IoT. The existing approaches battle with the brief lifetime of IoT, lopsidedness stack dissemination, and tall transmission delay. This paper master- postures a novel cluster-head determination and clustering instrument on IoT. It is composed of two primary stages. The primary stage chooses the near-optimal cluster-heads utilizing Ant Bee Colony (ABC) calculation. Execution criteria incorporate the remaining vitality of the de- indecencies, the number of neighbors, Euclidean remove between gadgets and the sink, and Euclidean remove between each gadget and its neighbors. The vital objective of the moment stage is to gather gadgets into a few clusters based on Euclidean remove between each cluster-head and its individuals, and the information volume produced by clusters.

KEYWORDS: Artificial bee colony (ABC), Algorithm Clustering, Data transmission, Delay Energy efficiency, Internet of things (IoT), Wireless communication

I.INTRODUCTION

The interconnections of smart devices on the Internet of things (IoT) provides an intelligent landscape for a dynamic world. IoT refers to the interconnection of the networks between heterogeneous wireless sensors and routine physical devices scattered in a monitoring area to sense, gather, share and transmit data throughout the system, and provide a smarter life for human beings. Meanwhile, from the recent advances in technologies and protocols, IoT devices are equipped with sensing, processing, communication, interaction, and corporation capabilities [1] [2].

According to these properties, IoT has penetrated pervasively into the wide-scale aspects of human life, including smart healthcare systems and personal monitoring, smart cities (homes, buildings, traffic monitoring, and urban computing), intelligent commercial, and smart industrial systems. Many IoT devices are powered by batteries with a limited lifetime and deployed in remote areas. Therefore, one of the main concerns while using IoT is energy limitation, as communication and computations on devices might quickly exhaust their battery resources [3].

Thus, providing energy-efficient data processing, aggregation, and transmission mechanisms play a vital role in IoT applications [4]. The amount of energy consumed for wireless communications in internet-based systems is much more than the processing costs. Therefore, one of the significant challenges on IoT is to achieve an energy- efficient data transmission mechanism from the source to the destination devices. Clustering is the primary operation required for appropriate data transmission on IoT. It is defined as a creative process that groups devices in some clusters and determines a cluster-head for each of them to improve resource consumption.

The paper is organized into following sections. Section 2 provides a review of various works with respect to energy conserving and network performance improvement. Section 3 presents the problem statement of the paper. The network architecture is defined in section 4. The proposed methodology and algorithm is presented in section 5 and 6. Simulation and results are discussed section 7 and conclusions along with future work are presented in section 8.

II. RELATED WORK

There are many research works that have been done. In [5], Celal Ozturk et. al., have given the searching mechanism of the DisABC algorithm is improved by the efficient genetic selection and its performance is tested on the dynamic clustering problem, in which the number of clusters is determined automatically i.e. it does not need to be specified in contrast to the classical techniques. Moreover, the superiority of the proposed algorithm (improved discrete binary artificial bee colony) is demonstrated by comparing it with the DisABC, binary particle swarm optimization (BPSO), genetic algorithm (GA), Fuzzy C-means (FCM) and K-means algorithms on benchmark problems.

In [6], Sung Y et al. presents an artificial bee colony clustering algorithm to optimally partition N objects into K clusters. The Deb's rules are used to direct the search direction of each candidate. This algorithm has been tested on several well-known real datasets and compared with other popular heuristics algorithm in clustering, such as GA, SA, TS, ACO and the recently proposed K–NM–PSO algorithm. The computational simulations reveal very encouraging results in terms of the quality of solution and the processing time required.

In [7], Dervis Karaboga et al. think about Artificial Bee Colony (ABC) algorithm which is one of the most recently introduced optimization algorithms, simulates the intelligent foraging behavior of a honey bee swarm. Clustering analysis, used in many disciplines and applications, is an important tool and a descriptive task seeking to identify homogeneous groups of objects based on the values of their attributes. In this work, ABC is used for data clustering on benchmark problems and the performance of ABC algorithm is compared with Particle Swarm Optimization (PSO) algorithm and other nine classification techniques from the literature. Thirteen of typical test data sets from the UCI Machine Learning Repository are used to demonstrate the results of the techniques. The simulation results indicate that the ABC algorithm can efficiently be used for multivariate data clustering.

In [8], Xiaohui Yan et al. presents a Hybrid Artificial Bee Colony(HABC) algorithm for data clustering. The incentive mechanism of HABC is enhancing the information exchange(social learning) between bees by introducing the crossover operator of Genetic Algorithm(GA) to ABC. With a test on ten benchmark functions, the proposed HABC algorithm is proved to have significant improvement over canonical ABC And several other comparison algorithms. The HABC algorithm is then employed for data clustering. Six real data sets selected from the UCI machine learning repository are used. The results show that the HABC algorithm achieved better results than other algorithms and is a competitive approach for data clustering.

In [9] Najjar-Ghabel S et al., proposes a novel binary version of the artificial bee colony algorithm based on genetic operators (GB-ABC) such as crossover and swap to solve binary optimization problems. Integrated to the neighbourhood searching mechanism of the basic ABC algorithm, the modification comprises four stages: (1) In neighbourhood of a (current) food source, randomly select two food sources from population and generate a solution including zeros (Zero) outside the population; (2) apply two-point crossover operator between the current, two neighbourhood, global best and Zero food sources to create children food sources; (3) apply swap operator to the children food sources to generate grandchildren food sources; and (4) select the best food source as a neighbourhood food source of the current solution among the children and grandchildren food sources. In this way, the global–local search ability of the basic ABC algorithm is improved in the binary domain. The effectiveness of the proposed algorithm GB-ABC is tested on two well-known binary optimization problems: dynamic image clustering and 0–1 knapsack problems. The obtained results clearly indicate that GB-ABC is the most suitable algorithm in binary optimization when compared with the other well-known existing binary optimization algorithms. In addition, the achievement of the proposed algorithm is supported by applying it to the CEC2005 benchmark numerical problems.

In [10] D.T. Pham et al. talks about Clustering is concerned with partitioning a data set into homogeneous groups. One of the most popular clustering methods is k-means clustering because of its simplicity and computational efficiency. K-means clustering involves search and optimization. The main problem with this clustering method is

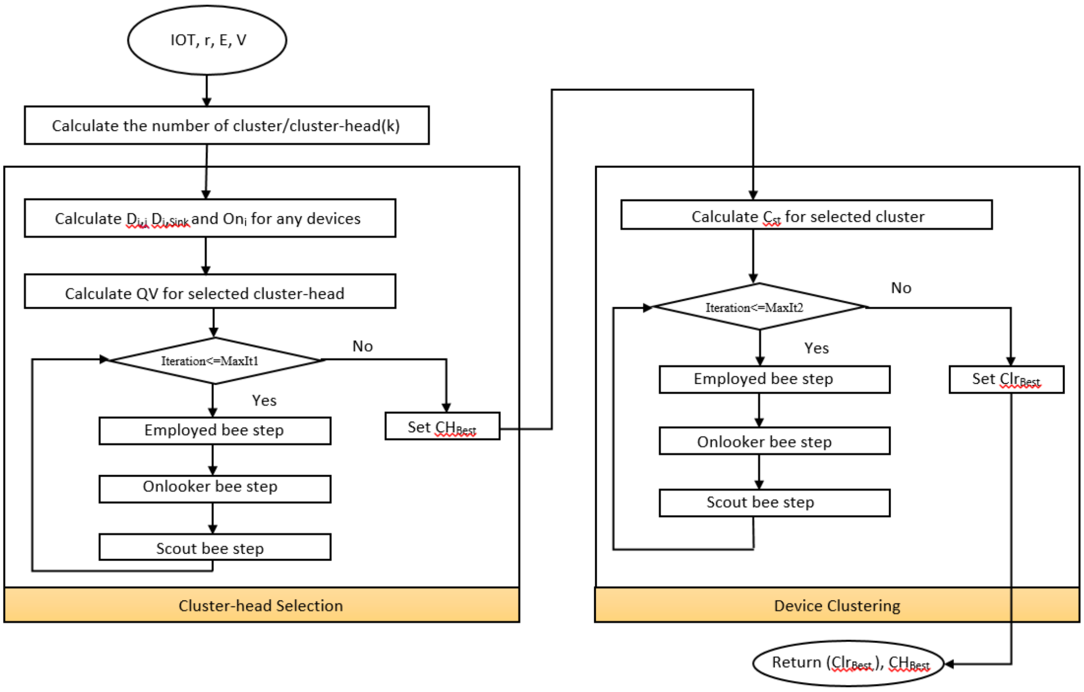

Figure 1. Block diagram of ABC-DC: details of cluster-head selection and device clustering phases

its tendency to converge to local optima. The authors' team have developed a new population-based search algorithm called the Bees Algorithm that is capable of locating near optimal solutions efficiently. This paper proposes a clustering method that integrates the simplicity of the k-means algorithm with the capability of the Bees Algorithm to avoid local optima. The paper presents test results to demonstrate the efficacy of the proposed algorithm.

III. PROBLEM STATEMENT AND OBJECTIVE

An energy-efficient clustering mechanism is the primary challenge in data transmission over IoT. Many IoT devices are powered by batteries with a limited lifetime and deployed in remote areas. Therefore, one of the main concerns while using IoT is energy limitation, as communication and computations on devices might quickly exhaust their battery resources.

Performance criteria include the residual energy of the de- vices, the number of neighbors, Euclidean distance between devices and the sink, and Euclidean distance between each device and its neighbors. The principal objective of the second phase is to group devices into some clusters based on Euclidean distance between each cluster-head and its members, and the data volume generated by clusters.

Proposes a novel cluster-head selection and clustering mechanism on IoT. It is composed of two main phases. The first phase selects the near-optimal cluster-heads using Artificial Bee Colony (ABC) algorithm. Artificial Bee Colony (ABC) is one of the well-known metaheuristic algorithms, which does not need the parameter-setting to address the issues. Also, it yields better performance in comparison to others. This paper proposes a novel mechanism for cluster-head selection and device clustering on IoT, which is named *ABC-based Device Clustering (ABC-DC)*. The presented mechanism consists of two phases. The first phase aims to select the near-optimal cluster-heads using the ABC. It considers the residual energy of devices, the number of one-hop neighbors, Euclidean distance between devices and sink, and Euclidean distance between each device and its one-hop neighbors as the performance criteria. The Euclidean distance between each cluster-head and its members and the data volume generated by each cluster are employed as the performance criteria. On move the IoT nodes broadcast the check neighbour message and nodes within the transmission range receives the message and updates the neighbour list. Based on the current position of the IoT nodes the cluster head changes. Each sensor node collect data and forward towards the sink. The path taken to reach the data to the sink may be via cluster head or may not. If via cluster head then once the data reaches the cluster head, it forward the data to the sink.

IV. THE PROPOSED SUBSTRUCTURE COMPRISES OF ARE A FEW TECHNIQUES OR MODULES OF THE ACCOMPANYING STRIDES.

A. Cluster-head selection

The first challenge in the proposed mechanism is to find the appropriate value of k (the number of clusters/cluster-heads) as the number of random solutions are generated in the initialization step of ABC. The optimal number of cluster-heads (k) for the assumed monitoring area is computed as below

$$n = \sqrt{\frac{|N| e_{ta} A}{2\pi(e_{ta}\frac{A^2}{6} - e_{rt}}} \tag{1}$$

where A, e_{ta}, and Ert are the side of the monitoring area (square), the energy consumption of the transmission amplifier, and energy used for receiving/transmitting 1-bit data, respectively. At this point, the main steps of ABC are performed with *Maximum Iterations (MaxIt1)* to provide efficient cluster-heads. The steps of the first phase of the ABC-DC are explained as follows:

B. Initialization Step:

In the first step, k initial population of the solutions (cluster-heads) is selected randomly among the index of devices to start solving the optimization problem (it is considered that each IoT device has a unique index). Each selected cluster-head is denoted by an integer, which identifies its index. At the end of the initialization step, a k member array of selected cluster-heads is generated as the initial solution of ABC. It is worth mentioning that there should not exist recurring members in this array because a cluster-head could not manage multiple clusters. We use a repair function to replace the recurring cluster-heads with non-recurring ones.

C. Employed bee step:

After generating initial population (selecting initial cluster-heads), each employed bee modifies a cluster-head in its memory in some rounds for finding a more efficient one. In this case, the size of the initial population *(k)* is equal to the number of employed bees. Each employed bee improves its corresponding cluster-head by exchanging it with another non-duplicate index. Then, it uses the local information to check the quality of the new solution; if this modification enhances the quality value, the bee remembers the new cluster-head as a member of the population array. At the end of the employed bee step, there should not exist recurring members in the array too. Thus, the repair function is called again to replace the recurring cluster-heads with non-recurring ones.

D. Onlooker bee step:

Onlooker bees increase the quality value of some cluster-heads in a probabilistic manner. They select high-quality cluster-heads (based on the information that is shared by the employed bees) with a higher probability. In the ABC-DC, the cluster-head selection is assumed as a maximization problem. So, the probability of selecting the cluster-head *ch* (*Pro ch*) is defined as below

$$P_{ch} = \frac{Q_{tch}}{\sum_{i=1}^{k} Q_{tch}} \tag{2}$$

where Q_{tch} denotes the quality value of the cluster-head *ch*. After calculating the selection probability, each onlooker bee selects a cluster-head using the Roulette Wheel Selection method. At this point, like the employed bee step, the bee updates its selected cluster-head; it remembers the new solution if the quality value increases compared to the old one.

Finally, the repair function is called to replace the recurring cluster-heads with non-recurring ones.

E. Scout bee Step:

In the final step, the scout bees search the area to find new cluster-heads randomly and replace a predetermined number of them with the low-value ones.

V. DEVICE CLUSTERING

In this section, we describe how ABC can be exploited to solve the clustering problem on IoT. The second phase of ABC- DC aims to assign IoT devices to the selected cluster-heads in the previous section, based on the Euclidean distance between each cluster-head and its members, and the data volume generated by each cluster. We model the device clustering phase of our mechanism as a minimizing problem because the performance criteria of it are in the form of cost. The main steps of ABC are executed with *Maximum Iterations (MaxIt2)* to provide efficient clusters. The steps of the second phase of ABC-DC are described as follows:

A.Initialization step:

In the first step of the device clustering process, the initial population of the solutions (clusters) is generated randomly to start resolving the optimization problem. It is assumed that a unique index recognizes each de- vice. So, each cluster is demonstrated by an array of integers, which identifies the index of its members. At the end of the initialization step, we have a three-dimensional array as the initial solution space of ABC, in which each row represents the members of a specific cluster (each two-dimensional array denotes a solution). There should not exist recurring members in the two-dimensional arrays because a device could not be allocated to multiple clusters. We use a repair function to replace/delete the recurring devices.

B. Employed bee step:

After generating initial clusters, each employed bee exchanges a member of the two-dimensional array in its memory in some rounds for reaching more effective solutions. In this scenario, the size of the initial population *(k)* is also considered equal to the number of employed bees. Each employed bee improves its corresponding cluster set by replacing a device with another non-duplicate one (also, a device could be added to/removed from a cluster). At this point, it exploits the local information to check the cost of the new cluster; if this modification decreases the cost value, the bee stores the new cluster as a member of the population array. After employing bee steps, there should not exist recurring members in the solutions too. We call the repair function again to replace/delete the recurring devices.

C. Onlooker bee step:

Onlooker bees behave similarly to employed ones, but their operation is based on a probabilistic manner using the information generated in the previous step. Indeed, onlooker bees choose low-cost clusters with a higher probability. In ABC-DC, device clustering is assumed as a minimization problem. So, the probability of selecting the cluster *C* (*Pro C*) is calculated as

$$P_c = \frac{\sum_{u=1}^{k} CV_u - CV_c}{\sum_{j=1}^{k}\sum_{u=1}^{k} CV_u - CV_j} \quad (3)$$

where *CVc* defines the cost value of the cluster set *C*. After calculating the selection probability, each onlooker bee chooses a set of clusters using the Roulette Wheel Selection method. Then, the bee updates its clusters; it stores the new cluster if the cost value decrease compared to the old one. Finally, it calls the repair function to replace/delete the recurring devices.

D. Scout bee step:

In the final step, the scout bees prob the area to discover new clusters randomly and change a predetermined number of them with the high-cost ones. To define the cost value of the set of the clusters, which has already been mentioned in the employed bee, onlooker bee, and scout bee steps, we also use the weighted sum of the considered performance criteria. The clustering problem on IoT aims to minimize the amount of the linear combination of the considered criteria (the Euclidean distance between each cluster-head and its members, and the data volume generated by each cluster). Accordingly, *CV* is defined as

$$CV = W_5 \frac{\sum_{u=1}^{k}\sum_{j=1}^{k} E_{uj} - E_{max}}{D_{min} - D_{max}} + W_6 \frac{\sum_{u=1}^{k}\left(\sum_{j=1}^{m} F_u\right)^2 - (F_{max})^2}{(F_{min})^2 - (F_{max})^2} \quad (4)$$

Where ∑Euj is the total Euclidean distance between cluster-head *j* and its members (*A* denotes the maximum number of clusters' members). Our proposed mechanism aims to cluster devices so that the Euclidean distance between each cluster head and its members is minimized to improve the energy consumption of them. *D min* and *D max* are considered as the best and worst total Euclidean distance between cluster-heads and their members. In the best case, each cluster-head and its members stay in the same geographic coordinates.

VI. ALGORITHM

A. Algorithm and example

Algorithm 1: CH selection
Input: IoT Nodes = (N, L), E, r1;
Output: CH_B;
Start
for (i = 1: N) ***do***
$ON_i = \emptyset$; //Set null as the preliminary significance of one-hop neighbors set for any cluster-heads
$D_{i,Sink} = \sqrt{(y_{sink} - y_i)^2 + (x_{sink} - x_i)^2}$ // Euclidean distance between d_i and sink
for (j = 1: N) **do**
$D_{i,j} = \sqrt{(y_j - y_i)^2 + (x_j - x_i)^2}$; // Euclidean distance between di and d_j
if ($D_{i,j} < r_i$) **then**
$ON_i = ON_i \cup Dj$; //Adding a new member to the set of one-hop neighbors of D_i
end
end
end
Initial_Pop = Initial_Population(IoT); //Preliminary population of nominated CHs
[QV, CH Best] = Fitness (Initial_Pop); //Estimate QV for all affiliates of Initial_Pop expending (5) and set the elucidation with the utmost QV as the CH Best
Current_Population = Initial_Pop;
while (iteration ≤ MaxIt1) **do**
[Current+Population, QV, CH Best] = Employed_Bee(IoT, Current_Population, QV);
[Current_Population, QV, CH Best] = Onlooker_Bee(IoT, Current_Population, QV);
[Current_Population, QV, CH Best] = Scout_Bee(IoT, Current_Population, QV);
end
return (CH Best) //The set of best cluster-heads
End

B. ***Algorithm 2*** *Device clustering*

Input: IoT, *V*, *CH_Best*;
Output: Clr best;
Begin
Initial_Pop = Initial_Population(IoT); *//Initial populace of constellations*
[Cst, Clr Best] = Fitness (Initial_Pop); *//Evaluate Cst for all participants of Initial_Pop and set the resolution with the bottommost Cst as the Clr best*
Current_Population = Initial_Pop;
while (*iteration ≤ MaxIt2*) **do**
[Current_Population, Cst, Clr Best] = Employed_Bee(IoT, Current_Population, Cst);
[Current_Population, Cst, Clr Best] = Onlooker_Bee(IoT, Current_Population, Cst);
[Current_Population, Cst, Clr Best] = Scout_Bee(IoT, Current_Population, Cst);
end
return (Clr best) *//The established of best CHs*
End

VII. PERFORMACE RESULTS AND ANALYSIS

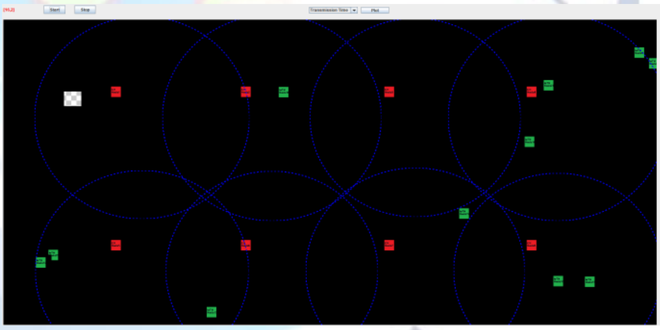

Figure 2. Node and cluster heads are in the network space

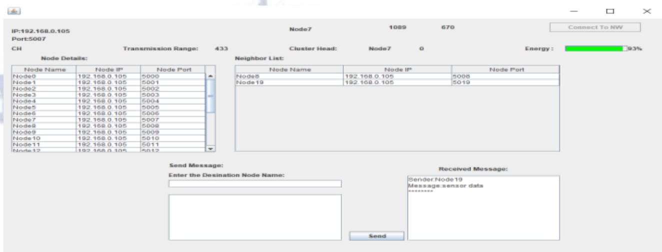

Figure 3. Destination node receives the information from the sender

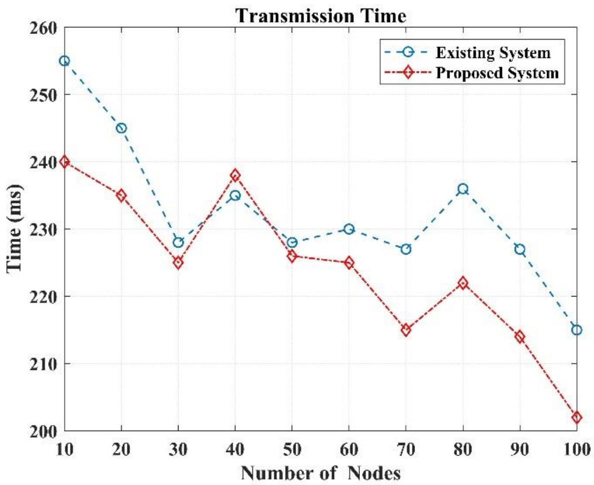

Figure 4. Transmission time of data transmission via multiple nodes

Tranmission time of data transmission via multiple nodes has been plotted. We have observed that the time taken in the proposed system is comparitively less than the existing work. This is because we have agregated the data before thasmission. As a result the network congession also reduced.

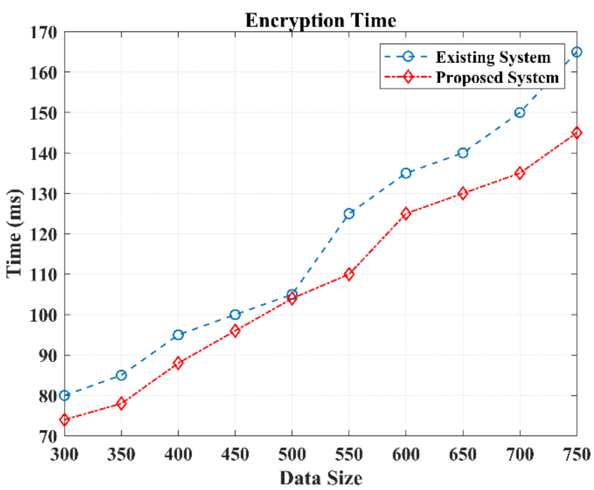

Figure 5. Encryption time of data

Encryption time of data based on the data size has been plotted. We have observed that the time taken in the proposed system is comparitively less than the existing work.

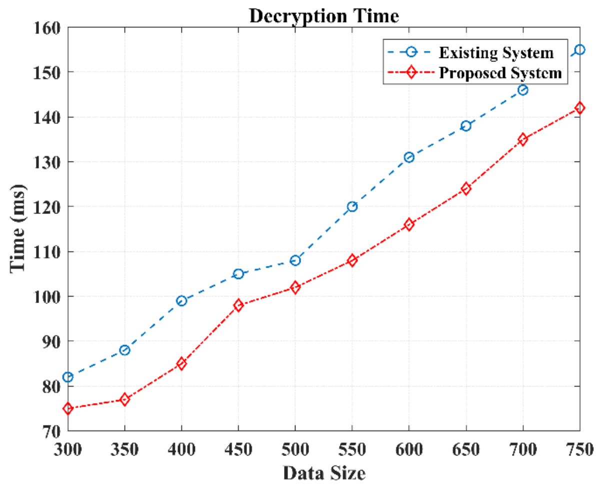

Figure 6. Decryption time of data

Decryption time of data based on the data size has been plotted. We have observed that the time taken in the proposed system is comparitively less than the existing work.

VIII. CONCLUSION

Data transmission is one of the significant concerns about IoT. Device clustering is the primary operation to satisfy the requirements of an energy-efficient communication in such systems. In this paper, we presented the ABC-DC mechanism that selects appropriate cluster-heads and cluster devices on IoT. The proposed mechanism is composed of two phases. The first phase exploits the ABC algorithm to determine the near-optimal cluster-heads considering the residual energy of devices, the number of one-hop neighbors, Euclidean distance between devices and the sink, and Euclidean distance between each device and its one-hop neighbors as the performance criteria. In the next phase, ABC-DC employs ABC again with the criteria, including Euclidean distance between each cluster-head and its members, and the data volume generated by each cluster to group devices in some clusters. Performance evaluations verified that the proposed mechanism improves energy consumption, lifetime, and data transmission delay on IoT in comparison with recent state-of-the-art techniques. As future work, we will try to suggest an energy-efficient and delay-aware routing mechanism to send aggregated data from cluster-heads to the sink, for maximizing the lifetime of IoT.

REFERENCES

[1] Shreyas, J., & Kumar, S. M. (2019, November). A survey on computational intelligence techniques for internet of things. In International Conference on Communication and Intelligent Systems (pp. 271-282). Springer, Singapore.

[2] Shreyas, J., Jumnal, A., Kumar, S. D., & Venugopal, K. R. (2020). Application of computational intelligence techniques for internet of things: an extensive survey. International Journal of Computational Intelligence Studies, 9(3), 234-288.

[3] Shreyas, J., Chouhan, D., Akshatha, A. R., Udayaprasad, P. K., & SM, D. K. (2020, March). Selection of Optimal Path for the Communication of Multimedia Data in Internet of Things. In 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS) (pp. 477-481). IEEE.

[4] Shreyas, J., Chouhan, D., Rao, S. T., Udayaprasad, P. K., Srinidhi, N. N., & Kumar, S. D. (2021). An energy efficient optimal path selection technique for IoT using genetic algorithm. International Journal of Intelligent Internet of Things Computing, 1(3), 230-248.

[5] Harfouchi, F., Habbi, H., Ozturk, C., & Karaboga, D. (2018). Modified multiple search cooperative foraging strategy for improved artificial bee colony optimization with robustness analysis. Soft Computing, 22(19), 6371-6394.

[6] Sung Y, Lee S, Lee M. A multi-hop clustering mechanism for scalable IoT networks. Sensors 2018;18:961. doi: 10.3390/s18040961 .

[7] Karaboğa, D., ASLAN, S., & AKSOY, A. (2018, September). Finding DNA Motifs with Collective Parallel Artificial Bee Colony Algorithm. In 2018 International Conference on Artificial Intelligence and Data Processing (IDAP) (pp. 1-7). IEEE.

[8] Zhao, Z., Zhang, Z., Xiao, S., Xiao, Z., Yan, X., Yu, J., ... & Wu, F. (2019). Long-form video question answering via dynamic hierarchical reinforced networks. IEEE Transactions on Image Processing, 28(12), 5939-5952.

[9] Najjar-Ghabel S, YousefiS, Farzinvash L. Reliable data gathering in the Internet of Things using artificial bee colony. Turkish J Electr Eng Comput Sci 2018;26:1710–23. doi: 10.3906/elk- 1801-100 .

[10] Pham, D. T., Baronti, L., Zhang, B., & Castellani, M. (2018). Optimisation of engineering systems with the bees algorithm. International Journal of Artificial Life Research (IJALR), 8(1), 1-15.