*I. J. Engineering and Manufacturing,* **2022, 1, 12-28**
Published Online February 2022 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijem.2022.01.02

# Achieving Performability and Reliability of Data Storage in the Internet of Things

**Negar Taheri**
Department of Computer Engineering, Science and Research Branch of Islamic Azad University, Ardabil, Iran
E-mail: ntaheri25@gmail.com

**Shahram Jamali**
Department of Computer Engineering, University of Mohaghegh Ardabili, Ardabil, Iran
E-mail: jamali@uma.ac.ir

**Mohammad Esmaeili**
Department of Computer Engineering, Science and Research Branch of Islamic Azad University, Ardabil, Iran
E-mail: esmaeili.cse@gmail.com

Received: 24 October 2021; Revised: 02 November 2021; Accepted: 27 December 2021; Published: 08 February 2022

**Abstract:** Internet of things (IoT) includes a lot of key technologies; In this emerging field, wireless sensors have a key role to play in sensing and collecting measures on the surrounding environment. In the deployment of large-scale observation systems in remote areas, when there is not a permanent connection with the Internet, the network requires distributed storage techniques for increasing the amount of data storage which decreases the probability of data loss. Unlike conventional networked data storage, distributed storage is constrained by the limited resources of the sensors. In this research, we present a distributed data storage method with the combined K-means and PSO clustering mechanism organized with the binary decision tree C4.5 in the IoT area with considering efficiency and reliability approach. This scheme can provide reliability in responding to inquiries while minimizing the use of energy and computational resources. Simulation results and evaluations show that the proposed approach, due to the distributed data storage with minimal repeat publishing according to the decision tree structure, increases the reliability and availability, reduces the communication costs, and improves the Energy consumption, saving memory consumption without registering the same event and compared to other methods performed in this area have good results.

**Index Terms:** IoT, Distributed Storage, Reliability, Energy Efficiency, PSO Algorithm, K-means, C4.5 Tree.

## 1. Introduction

The phrase of Internet of Things (IoT) at the first time was used by Kevin Ashton in 1999 [1] to describe a world in which everything including (human, animal, inanimate objects) would have a digital identity for itself and are capable to deliver data via communication networks like internet or intranet and objects could be controlled and managed with applications existed in intelligent telephones, tablets, and computers (fig1). The appearance of IoT is one of the thousands of results of internet extension and certainly the development of wireless technologies and micro-electromechanical systems [2]. One of the important features of the internet (IoT) is the Heterogeneity of objects. No need for human-computer interaction in the process of sending data between objects: the data is sent and received automatically [3]. Internet of things(IoT) includes a lot of key technologies; wireless sensor networks are one of them. Wireless sensor networks for IoT surveillance systems generally consist of automatic nodes that sense the surroundings, and a sink node that acts as a data collector and gateways to the internet is considered. Communications between sensor nodes and the sink are typically not instantaneous, especially in isolated WSNs where the sink node is not always present. To avoid this, nodes can cooperate by storing the sensed data in a distributed way. As WSNs tend to be left unattended for long periods, distributed data storage [4] has to be robust also against node failures. To achieve this goal, an attractive approach consists in combining distributed storage [5] with data replication, i.e., by distributing and storing multiple copies of the same data across the WSN. Data storage at a large scale relies heavily on replication [6] as a key technique to address important problems: 1- Data resilience 2-High availability. In addition, Data availability [7], reliability [8], security, data processing, data retrieval [9], network lifetime, and energy efficiency redundancy [10] are the major challenges in data storage in wireless objects [11]. In any connection between two objects, data is sent or received, and this connection does not make sense without exchanging data. So data is one of the most valuable aspects of the Internet of Things. On a large scale, crashes generally occur in a data center. Now if a storage device holds only one copy of a data set and a crash occurs, this leads to data loss. By distributing copies of the data set to multiple storage devices, even if one device fails, the remaining devices can handle requests (same as fault tolerance debate). Distributed data storage across the node, with/without redundancy, is a challenge because it requires the correct choice of data node and requires a communication overhead to transfer data to the node selected for data storage. Therefore, network storage capacity and flexibility have energy costs and this limits the life of the network. Second, high availability; When a data set is repeated in distribution mode, requests can also be distributed among duplicates, which dramatically reduces the number of failed requests due to simultaneous access. This is especially important for data analysis applications that need to process different queries in parallel about the data set. This article**,** discusses a strategy for, distributed data storage mechanism with minimal repeat to increase the resilience, storage capacity, and energy efficiency of an IoT-based surveillance system against node failure and local memory shortage. The goals include: 1) Effective IoT storage using clustered tree structures. 2) Optimal management of energy consumption, reliability, and efficiency. 3) Reducing the cost of communication and increasing the availability of data storage in the Internet of Things with the provided tree structure. The rest of this paper is organized as follows: Section 2 is dedicated to related works. Section 3 brings an introduction to PSO. Section 4 discusses the K-means algorithm. Section 5 Clustering with energy efficiency and storage. Section 6 about the C4.5 Algorithm for formation binary tree of clusters. Section 7 a scheme for distributed data replication will be clarified deeply. Section 8 Results of the simulation of the proposed method are presented. Finally, Section 9 Concludes the paper is presented.

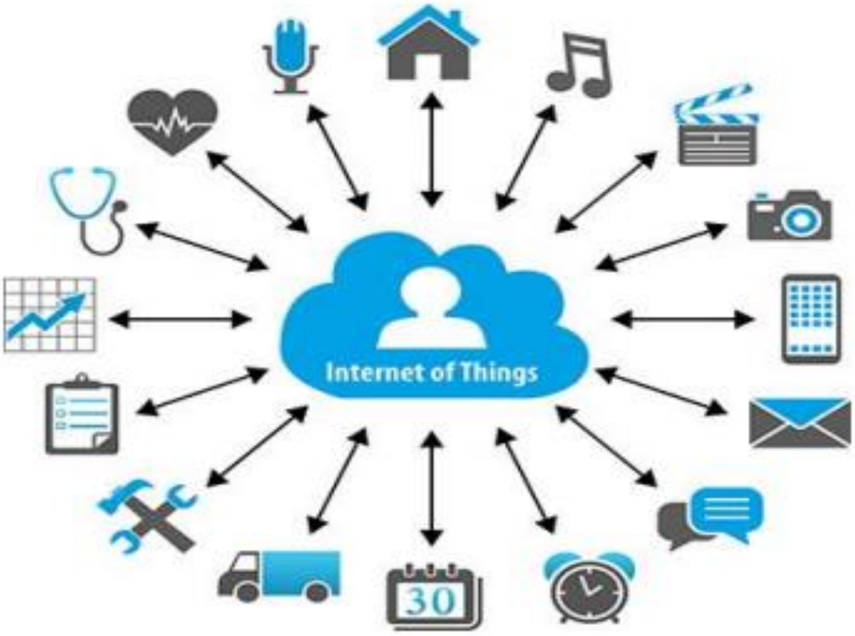


Fig.1. Internet of Things view [2]

## 2. Related Works

In the past years, various schemes to efficiently distribute and replicate data in WSNs have been proposed [12]. In WSN distributed storage schemes, nodes cooperate to efficiently distribute data across the WSN. There are two main approaches: data-centric storage and fully distributed data storage. The data-centric storage approach is described in [13,14]. Here, some distinguished storage nodes, e.g., determined by a hash function, are responsible for collecting a certain type of data. A load-balanced distributed storage approach is proposed in [15], according to which data are preferably stored in densely populated areas of the sensing field to minimize data loss. In [16], data are stored according to their spatial and temporal similarity, in order to reduce the overhead as well as the latency of a query request. Even if data-centric storage approaches are based on node cooperation, they are not fully distributed since specific nodes store all the contents generated by the others. This method, which splits a spatial two-dimensional storage space into objects to distribute information between nodes in a region, is not strong enough in a dynamic environment. Because, in a dynamic environment, the number of nodes is constantly changing. Therefore, nodes may not exist for an object, and some additional nodes may not be fully used to store data. In addition, maintaining two-dimensional temporal-spatial storage space is difficult due to the mobility of the node.

A detailed survey on data-centric storage schemes is presented in [17]. In a fully distributed data storage approach, all nodes contribute equally to sensing and storing. All nodes try, first, to store the sensor readings locally and, then, delegate other nodes in the WSN to store newly collected data as soon as their local memories are full. The first significant contribution in this direction is Data Farms [18]. The authors propose a full data distributed storage mechanism with periodic data retrieval. They derive a cost model to measure energy consumption and show how a careful selection of nodes offering storage, denoted as ‘‘donor nodes,’’ optimizes the system capacity at the price of slightly higher transmission costs. They assume a network tree topology, where each sensor node knows the return path to a sink node, which periodically retrieves data. The energy consumption problem is studied also in [19], where data preservation in an isolated WSN is considered. In this context, an energy-efficient data distribution scheme is proposed, based on the dissemination of data from low energy nodes to high energy ones. However, only low-energy nodes generate content. An interesting approach for load balancing is proposed in Environment Store [20]. The authors focus on in-network data redistribution when the remaining storage space of a sensor node exceeds a given threshold. They use a proactive mechanism, where each node maintains a local memory table containing the statuses of the memories of its neighbors. Furthermore, mobile nodes (called mules) are used to carry data from an overloaded area to an offloaded one, as well as to send the collected data to a sink node. The deployment of mobile mules is also addressed in [21], where an efficient distributed data storage mechanism for an isolated WSN with limited storage space is proposed. Data is opportunistically offloaded to mobile mules when the latter are in proximity. Moreover, data are assigned different priorities: high prioritized data are stored closer to areas where the mules will pass more frequently. The main limitation of the above studies consists of a lack of a comprehensive performance evaluation framework, which encompasses analysis, simulations, and experiments. This holistic approach is a key contribution to our work. Data replication strategies have been proposed in the literature, mainly to overcome the problem of node failures. The goal of replication is to copy data at other nodes within the WSN to increase resilience. Authors in [22] propose ProFlex, a distributed data storage protocol for replicating data measurements from constrained nodes to more powerful nodes. The protocol benefits from the higher communication range of such nodes and uses the long link to improve data distribution and replication against the risk of node failures. ProFlex has the advantages of reducing message loss, reducing message overhead, reducing the problem of energy gap and applicable to large scale wireless receiver objects. But the disadvantage is that ensuring data security is the main weakness of ProFlex and we have to solve this problem. If there is no data security, data cannot be accessed and has low reliability. Supple [29] is a flexible data transmission protocol for wireless sensor objects that considers sinks fixed or movable. Unlike ProFLex, Supple uses a unit multiplication structure using a tree topology. Tree construction is started by a central sensor node in the sensing area. The central sensor node is responsible for receiving and repeating data collected in objects. The next step is to assign weight to the node, which indicates the probability of storing node data. But the disadvantage is that this protocol does not consider finding a good central node position. Energy consumption and traffic congestion are high in nodes close to the central node and also include overhead message.

In Tiny DSM [23], a reactive replication approach is presented. Replicas are randomly distributed within a predefined replication range influenced by the specific replica number and density. In mechanism DS [4] the proposed replication-based distributed data storage is greedy: in order to create a replica of stored data, a node selects, according to its neighbors’ memory table, the ‘‘best’’ neighbor the selection criterion will be specified in the following. The selected neighbor becomes a donor node. If no donor node can be chosen and there is no available space in the local memory, then the acquired data is dropped. The greedy distributed storage mechanism consists in creating at most R copies of each data unit generated by a node and distributing them across the network. A shortcoming of this approach that's who This may be inconvenient when the memories of the nodes are almost full and may prevent new data to be stored.

All of the above algorithms, despite some problems, are basically acceptable in terms of performance, but these methods are for storing data in a pervasive environment such as the Internet of Things (due to the large volume of data generated). Considering the number of millions / billion objects) is not responsive to the challenges mentioned. It seems necessary to use some mechanisms that have solved the problems of some protocols and have an acceptable level in terms of efficiency and storage capacity. Therefore, in this article, an optimal method and solution for these challenges are proposed.

## 3. Particle Swarm Optimization

Particle density optimization (PSO) [24] is one of the most common global optimization methods introduced by Kennedy and Abhor in 1995. Crowded intelligence is based on the movement and group behavior of birds and fish. The mass of particles in nature for us represents collective intelligence. Consider the collective movement of fish in water or birds during migration, all members move in perfect harmony with each other. If they are to be hunted, they hunt together, and if they are to be preyed upon by another prey, they escape from the clutches of the hunter by moving a group. In other words, each particle independently looks for the optimal point, each particle moves at the same speed at each step, each particle remembers the best points in it, the particles work together to inform each other of the places they are looking for, each particle is in contact with its neighboring particles, every particle is aware of the particle particles that are in the neighborhood and every particle is known as one of the best particles in its neighborhood.

In the proposed model, when the particle pattern was defined and the initial parameters of the particle swarm optimization algorithm were determined, the K-means algorithm was used randomly to generate the initial population to generate the required number of clustering solutions and then using the fitting function. The suitability of each method is determined and optimized by the PSO algorithm and finally, the optimal solution is selected. In fact, the PSO algorithm is a population-based algorithm, being population-based means that the search algorithm considers a set of solutions as a population for the optimization problem. After generating the initial population (particles) and considering an initial velocity for each particle, the efficiency of each particle is calculated based on its position. Each particle in the search space represents a solution to the problem and changes its speed based on the best answer obtained in the particle group (the best person in the group) and the best place it has ever been. This velocity is obtained by the position of the particle, the new position of the particle, in other words, each particle calculates the value of the target function in the position of the search space in which it is located. In subsequent iterations, the best particle competently assists the other particles and corrects their motion, and after successive iterations, the problem converges toward the optimal answer. Therefore, according to the purpose of the paper, which is to achieve the desired reliability and performance, the particle swarm algorithm gives us the best possible solution. The steps are described below.

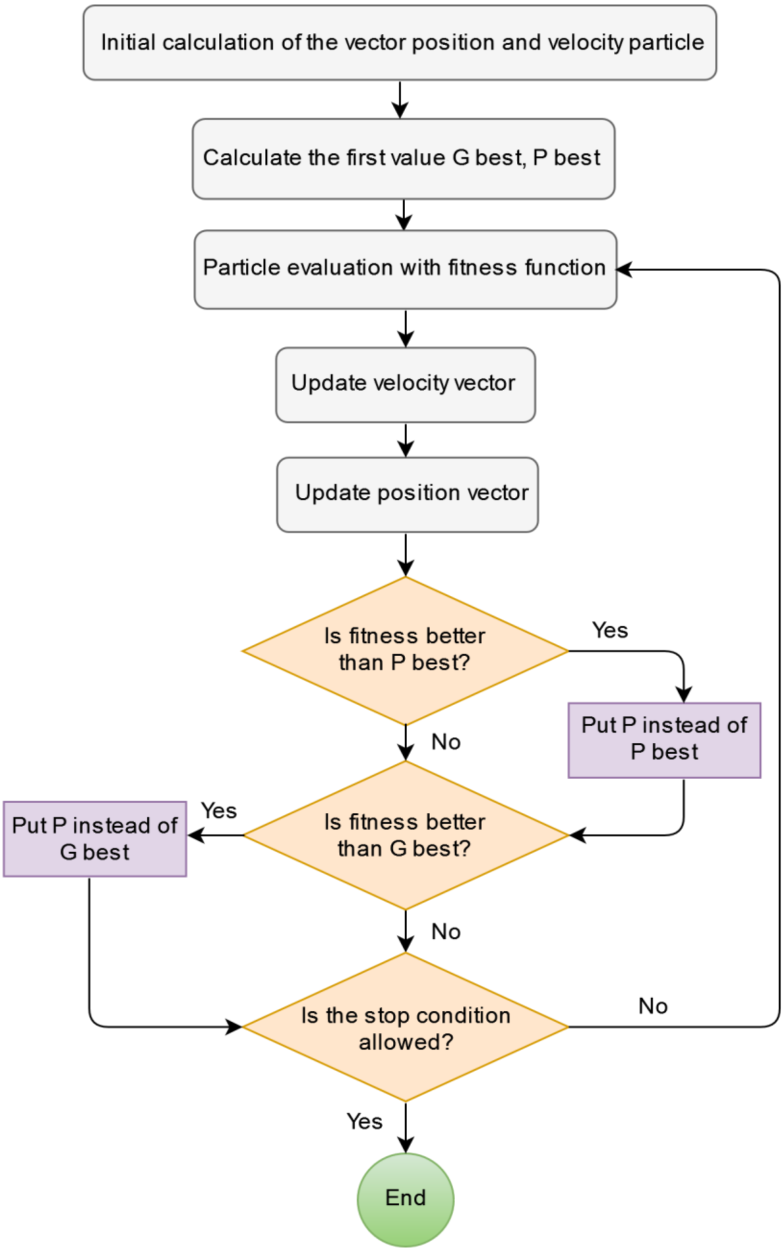

Initial calculation of the vector position and velocity particle
Calculate the first value G best, P best
Particle evaluation with fitness function
Update velocity vector
Update position vector
Is fitness better than P best?
Yes
Put P instead of P best
No
Is fitness better than G best?
Yes
Put P instead of G best
No
Is the stop condition allowed?
No
Yes
End

Fig.2. Flowchart of PSO Algorithm

Suppose that the problem space is d-dimensional. In this regard, the i-th particle of the population can be represented by a position vector (relation (1)) and a velocity vector (relation (2)). Changing the position of each particle is possible by changes in the structure of the position and the previous velocity.

$$X_i = \left[x_{i،1}،x_{i،2}، \dots ،x_{i،d}\right] \tag{1}$$

$$V_i = \left[v_{i،1}،v_{i،2}، \dots ،v_{i،d}\right] \tag{2}$$

Each particle is containing the best value ever achieved (p-best), the relationship (3), and the xi position. This information comes from the efforts made by each particle to find the best answer, or, in other words, from the situations in which a particle has ever been located. On the other hand, the best answer ever seen by all the particles in the group (of the amount of p-best) is shown by g-best, relationship (4)

$$pbest \ = [p\_(b،1)،p\_(b،2)، \dots ،p\_(b،d) \ ] \tag{3}$$

$$gbest = \left[g_{b،1}، \ g_{b،2} \ ، \dots ، \ g_{b،d}\right] \tag{4}$$

In this regard, every particle is always trying to correct its current position in line with the current movement, the best position obtained by itself and the best position obtained by the whole of the particles and is transferred to a new position, relations (5) and (6).

$$v_{i،j}(t+1) = w.v_{i،j}(t) + c_1.r_1.\left(pbest_{i،j}(t) - x_{i،j}(t)\right) + c_2.r_2.(\ gbest_j(t) - x_{i،j}(t)) \tag{5}$$

$$x_{i،j}(t+1) = \quad x_{i،j}(t) + v_{i،j}(t+1) \tag{6}$$

So the new position is obtained based on the current position of the particle **xi**, the current particle velocity **vi,** the distance between the current position and the best response experienced by the p-best particle, as well as the distance between the current position and the best position obtained in the total g-best. Thus, the velocity of each particle is changed according to (5), the new position of each particle is determined by the total position of the past and the new velocity, using the relation (6), Where dimension j-th of each particle is in repeating t-th. **ω** is the inertial weight. **c1** and **c2**, respectively, are the Weight Factor Learning coefficient personal experience of each particle and the collective experiences of all particles (groups) in the range of **[0,4]**, **r1** and **r2** are random numbers between **0** and **1** which causes diversity and the difference in the answers (solutions), **()** is the velocity of the j-th dimension of each particle in repeating t-th, is equal to p-best dimension j-th each particle and is also g-best in the group.

## 4. K-Means Algorithm

K-means clustering is one of the most popular split clustering methods that can be easily implemented and is one of the mechanisms that, while increasing scalability, the ability to access various types of objects in Provides IoT with minimizing energy consumption. Despite the simplicity of this algorithm, a method is a basis for many other clustering methods [25]. This is exclusive and flat. The algorithm is expressed in different forms. But all of them are routine for a fixed number of clusters tried to estimate the following:

A) Obtain points as cluster centers. These points are the average of points belonging to each cluster.

B) Assigning each data sample to a given cluster that has the least distance to the cluster center.

These function as the objective function in this algorithm:

$$J = \sum_{j=1}^{K} \sum_{i=1}^{K} \left\| x_I^{(J)} - c_j \right\|_{,.}^{2} \tag{7}$$

Where $\|.\|$ is the measure of the distance between points and $x_i$ $^{(j)}$ is the J-th cluster center. The following algorithm is the basic algorithm for this method:

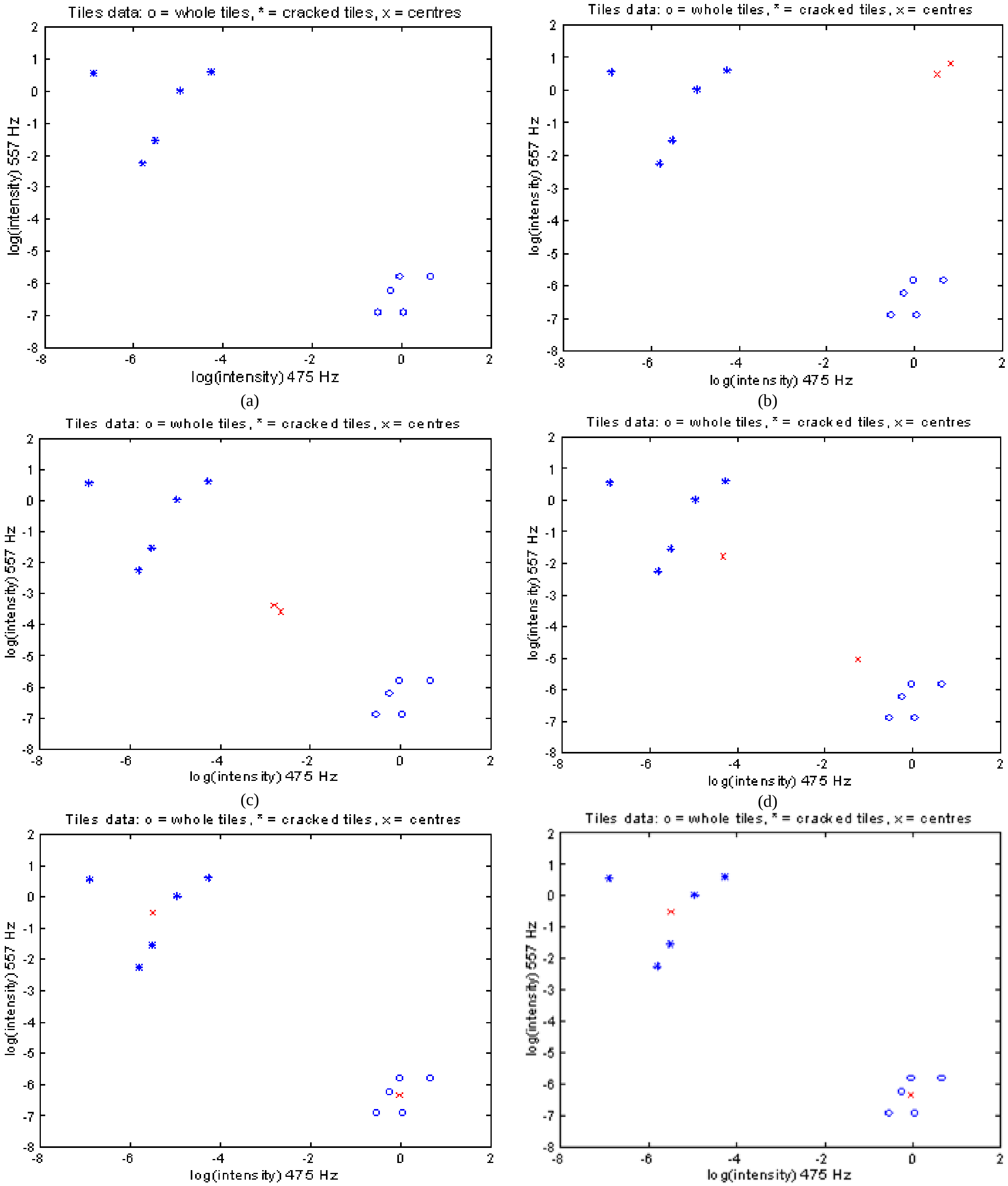
A) First K points are selected as the cluster centers.

B) Each instance of the data to the cluster is assigned the minimum distance to the cluster's center.

C) Dependence of total data to one of the clusters, for each cluster a new point is calculated as the center (Average points in each cluster).

D) B and C steps are repeated until no change in cluster centers is not achieved.

Figure 3 how to apply this algorithm on a data set that contains the two groups is shown (Steps A, B, C, D). The process to reach areas that do not change, has continued.



(a)

(b)

(c)

(d)

(e)

(f)

Fig.3. Example of K-Means Clustering Method (a, b, c, d, e, f) [13]

## 5. Clustering

Clustering partitions, the network into groups of sensors nodes that are geographically close to each other. Each cluster will have a cluster head that is responsible for controlling all the activities of the group like transmission, aggregation, management, and maintaining structure. With clustering in WSNs [2, 26, 30, 31, 32], energy consumption, the lifetime of a network, and scalability can be improved. Currently, the accepted and mostly used topology for clustering in WSNs is where each cluster has a cluster head. The sensor nodes transfer their data directly to their associated cluster head nodes (relay nodes) and then cluster head nodes perform the initial data aggregation and send it to the designated route. How clustering of objects is one of the challenges [11] ahead of the internet because it is a distributed and dynamic environment (homogeneous and heterogeneous, battery-powered, and power-coupled), and the clustering of such environments, due to the large scale of the devices, is a matter of NP-Complete. K-means clustering is one of the well-known clustering techniques that can be easily implemented and is one of the mechanisms that, while increasing scalability, enables access to various types of objects in the IoT provides with minimal communication energy consumption. In the flowchart below how to implement the structure and how to store it is shown.

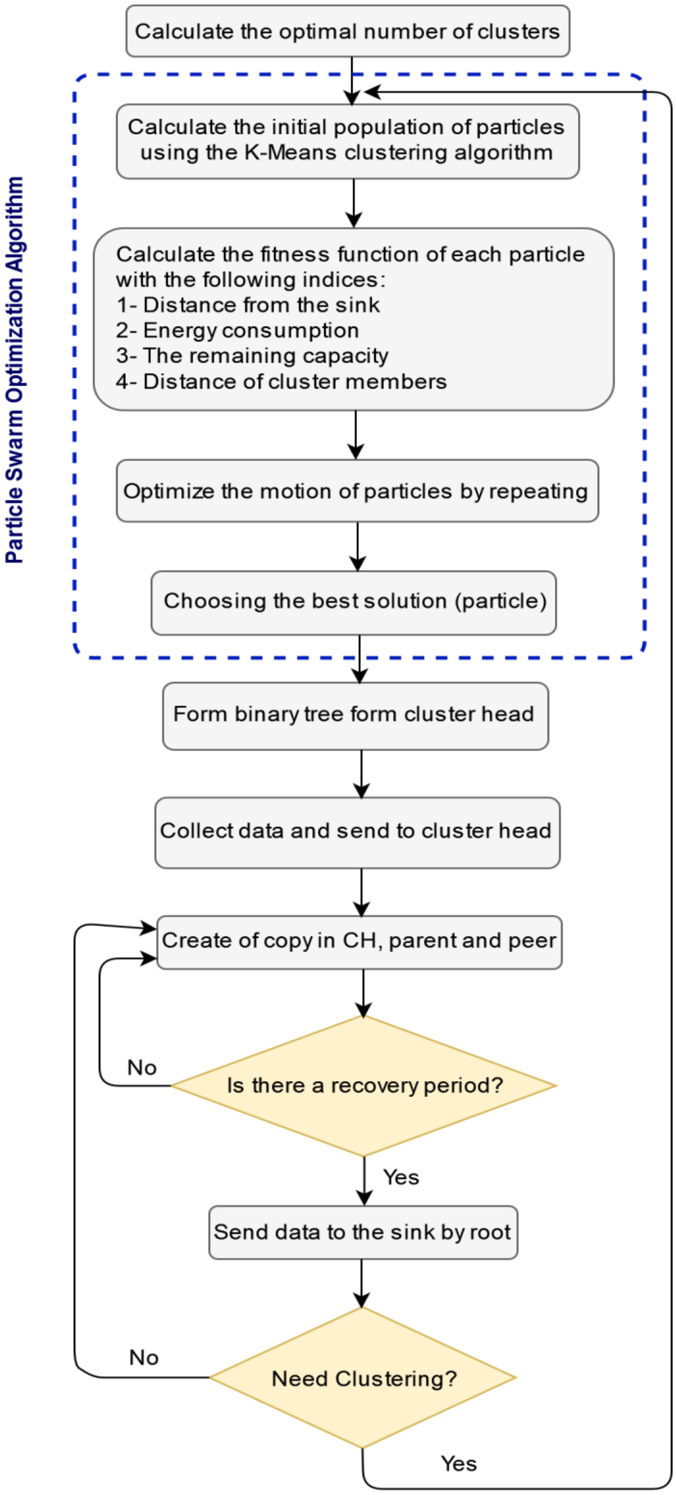

Calculate the optimal number of clusters
Particle Swarm Optimization Algorithm
Calculate the initial population of particles using the K-Means clustering algorithm
Calculate the fitness function of each particle with the following indices:
1- Distance from the sink
2- Energy consumption
3- The remaining capacity
4- Distance of cluster members
Optimize the motion of particles by repeating
Choosing the best solution (particle)
Form binary tree form cluster head
Collect data and send to cluster head
Create of copy in CH, parent and peer
No
Is there a recovery period?
Yes
Send data to the sink by root
No
Need Clustering?
Yes

Fig.4. Flowchart of Clustering and storage structure

### *A. Calculate the optimal number of clusters ($K_{opt}$)*

The first step in structure is to calculate the optimal number of clusters for this, the relation (8) given in [27] is also usedin this study. Used notations and symbols of equation (8) are described as in table (1).

$$K_{opt} = \sqrt{\frac{n_0 \beta T r_H^2 (2l+\mu r^k)}{r^2(\alpha_1 + c\beta T\left(l^1+\mu^1 d^{k^1}\right))}} \tag{8}$$

Table 1. Parameters for calculating the optimal number of clusters

| Symbols | Description |
|---|---|
| $n_0$ : | The default value for the number of clusters. |
| β: | Knot energy level |
| T: | The amount of time needed to transfer a data unit |
| $r_H$: | Node radio rang |
| l: | Data volume for transfer between two nodes (inside a cluster) |
| $\mu r^k$: | Amount of energy used for propagation losses |
| a1: | Node hardware cost average ci |
| ci: | Is the degree of the relationship between a node with its neighboring nodes (inside a cluster) |
| $l^1 + \mu^1 d^{k^1}$: | Energy consumed in transmitting data from one node to BS |

### *B. Particle Definition Pattern*

Each particle in the PSO algorithm means a solution. In the present study, each solution means clustering with the number of known clusters as described in (Eq.8) how it is calculated. The initial position of each particle is a vector, including the index of the nodes selected in the intended solution as the cluster header.

*Phase I:  Primary population generation by K-means algorithm*

In this method, when the particle pattern was defined and the initial parameters of the particle swarm optimization algorithm were determined, the K-means algorithm was used to generate the initial population. To do this, the number of clusters obtained from equation (8) is applied to the K-means algorithm so that the nodes in the cluster **kept** are grouped. K-means is used to generate particles, which means that each particle is a solution (cluster) and the particle's initial population is calculated for the header selection (clustering with distance logic) and finally delivered to the PSO.

*Phase II: Optimal cluster selection using the Particle Swarm Optimization (PSO) algorithm*

In this algorithm, the best solutions are identified and presented using the function of four indicators, which based on the parameters and indicators of goals are designed. Using the fitness function, the fitness of each solution is determined using 4 parameters and is optimized by the PSO algorithm and finally, the optimal solution is chosen. The four indicators used to fitness are:

1- Cluster head distance from the sink.
2- Energy consumption of cluster heads.
3- Cluster Consumption Memory.
4-Distance of cluster members from cluster head that can affect in energy consumption the whole cluster.

The PSO algorithm runs for each cluster separately to find the optimal cluster and moves the particles to the desired location. The steps are explained below.

### *C. Fitness function*

In this research, for achieving the ultimate goal, 4 parameters have been defined which attempt to optimize in structure. The first parameter is the cluster's distance from the sink that should be less. To calculate and evaluate it, the usual geometric spacing given in (9) is used.

$$dis_{h(i)}^{BS} = \sqrt{(x_{h(i)} - x_{BS})^2 + (y_{h(i)} - y_{BS})^2} \tag{9}$$

In which i is an index of cluster head, h (i) means the cluster-head i, $^{x}{}_{h\,(i)}$ represents the longitudinal coordinates of the cluster head and $^{x}{}_{BS}$, the coordinates of the longitudinal sink or the central station and $^{y}{}_{h\,(i)}$, $^{y}{}_{BS}$ are respectively the cross-coordinates of the cluster head and sink and $\mathrm{dis}^{BS}{}_{h\,(i)}$ represent the cluster heads distance from the sink. The second selected parameter for applying in the evaluation function is the energy used to transfer data between nodes, which is dependent on the distance which is obtained from the following equation.

$$\mathrm{Energy}_{h(i)} = \sum_{j=1}^{m} \mathrm{E}_{tr} * dis_{h(i)}^{s(j)} \tag{10}$$

Where m is the number of cluster members, $\mathrm{E}_{tr}$ is the base energy level defined for the data unit transfer,$dis_{h(i)}^{S(j)}$ which shows the distance of the header from the members of the cluster, which is similar to (9) and $Energy_{\boldsymbol{h(i)}}$ represents the amount of energy consumed in the desired cluster. The next parameter in the proposed method is the amount of space occupied, in other words, the amount of space required to store the data of a cluster in the header. Therefore, reversing the amount of space required in Equation (11) has been used.

$$Storage_{h(i)} = \frac{m_{h(i)}}{M_{h(i)}} \tag{11}$$

Where is the amount of memory the header and number of cluster members and also represents the index of memory. The last parameter to determine the fitness is the distance between the members of a cluster of cluster heads which is obtained from Equation (12).

$$Dis_{h(i)} = \sum_{j=1}^{m} dis^{s(j)}{}_{h(i)} = \sum_{j=1}^{m} \sqrt{(\mathrm{X}_{\mathrm{h(i)}} - x_{s(j)})^2 + (y_{h(i)} - y_{s(j)})^2} \tag{12}$$

To determine the overall fit function of the weighted average, the four defined parameters were used after their normalization and the weight of all of them was considered to be 0.25. The function used to normalize the relationship is (13).

$$norm\big(f(t)\big) = \frac{(f(t) - \min(f))}{(\max(f) - \min(f))} \tag{13}$$

As a result, the general Fit function is defined as follows:

$$\begin{aligned} Fit_{h(i)} = {} & w_{dis}.norm\big(dis_{h(i)}^{BS}\big) + w_{Energy.}norm\big(Energy_{h(i)}\big) \\ & + W_{Storage}.norm\big(Storage_{h(i)}\big) + W_{Dis}.norm\big(Dis_{h(i)}\big) \end{aligned} \tag{14}$$

where in:

$$w_{dis} = w_{Energy} = w_{Storage} = w_{Dis} = 0.25 \tag{15}$$

The weight of each of the four parameters is. The goal of the proposed method is to realize the relation (16) and find the appropriate response to it.

$$Object = minimize\big(Fit_{h(i)}\big) \tag{16}$$

## 6. C4.5 Tree-Construction Algorithm

In short, the decision tree is a map of the possible outcomes of a series of related choices or options that allow an individual or organization to measure probable actions in terms of cost, probability, and benefit. A decision tree typically begins with an initial node, after which the possible consequences branch off into branches, each of which leads to other nodes, which in turn create branches of other possibilities. This branch-to-branch structure eventually turns into a tree-like diagram. A decision tree is used the classify a case, i.e. to assign a class value to a case depending on the values of the attributes of the case. The class specified at the leaf is the class predicted by the decision tree. A performance measure of a decision tree over a set of cases is called classification error. It is defined as the percentage of miss-classified cases, i.e. of cases whose predicted classes differ from the actual classes.

**The Tree-Construction Algorithm.** The C4.5 algorithm constructs the decision tree with a divide and conquers strategy. In c4.5, each node in a tree is associated with a set of cases. Also, case on assigned weights to take into account unknown attribute values. In the beginning, only the root is present, with associated the whole training set and with all case weights equal to 1.0. At each node the following *divide* and *conquer* algorithm (see program 1) is executed, trying to exploit the locally best choice, with no backtracking allowed [28].

Let *T* be the set of cases associated with the node. The weighted frequency *freq (Ci, T)* is computed (step (1)) of cases in *T* whose class is *Ci*, for I € [1, N Class].

If all cases (step (2)) in *T* belong to the same class *Cj* (or the number of cases in *T* is less than a certain value) then the node is a leaf, with associated class *Cj* (resp., the most frequent class).

If *T* contains cases belonging to two or more classes (step (3)), then the *information gain (*weighted average function*)* of each attribute is calculated. For discrete attributes, the *information gain* is relative to the splitting of cases in *T* into sets with distinct attribute values. For continuous attributes, the *information* gain is relative to the splitting of *T* into two subsets, namely cases with attribute value *greater than* a certain *local threshold*, which is determined during information gain calculation.

The attribute with the highest information gain (step (4)) is selected for the test at the node. Moreover, in case a continuous attribute is selected, the *threshold* is computed (step (5)) as the greatest value of the *whole* training set that is below the local threshold.

A decision node has *S* children if T1, ..., Ts are the sets of the splitting produced by the test on the selected attribute (step (6)). S=2 when the selected attribute is continuous, set *s*=*h* for discrete attributes with each known value.

For *i* = [*1, s*], if *Ti* is empty (step (7)) the child node is directly set to be a leaf, with the associated class the most frequent class at the parent node and classification *error 0*.

If *Ti* is not empty, the divide and conquer approach consist of recursively applying the same operations (step (8)) on the set consisting of *Ti* plus those cases in *T* with the unknown value of the selected attribute.

Note that cases with an unknown value of the selected attribute are replicated in each child with their weights proportional to the proportion of cases in *Ti* over cases in *T* with the known value of the selected attribute.

Finally, the classification error (step (9)) of the node is calculated as the sum of the errors of the child nodes. If the result is greater than the error of classifying all cases in *T* as belonging to the most frequent class in *T*, then the node is set to be a leaf, and all sub-trees are removed [16].

### *A. Formation binary tree (organized with k-means and PSO algorithms Based on their weight)*

The c4.5 decision tree operates based on a top-down greedy search in the binary tree and creates a decision tree with a strategy of division and conquest. The goal is to optimize, so the search is done using greedy decisions and the best choice is. The decision tree selects the best feature using an index as the header capability that is the equation (17) is used to determine the node level in the tree. The condition for making a tree is that the cluster with more weight is considered as the root of the tree and for the rest of the levels, the clusters are calculated separately, whichever is more capable in terms of weight is selected. Then, for each node, two child nodes are selected, this time the distance between them with the parent of that child is calculated. This process continues until the end of the number of clusters.

$$HDR\_ability_{h(i)} = W_s * \left(\frac{M_{h(i)}}{\max(M_h)}\right) + W_E \left(\frac{E_{h(i)}}{\max(E_h)} + \frac{E_{h(i)}}{dis_{h(i)} * \max(E_{h(\dots)})}\right) \tag{17}$$

Where in $\frac{\mathrm{M}_{\mathrm{h(i)}}}{\max(\mathrm{M_h})}$ Normalized amount of memory, $\frac{\mathrm{E}_{\mathrm{h(i)}}}{\max(\mathrm{E_h})}$ Normalized value of absolute energy, $\frac{\mathrm{E}_{\mathrm{h(i)}}}{\mathrm{dis}_{\mathrm{h(i)}} \times \max(\mathrm{E}_{\mathrm{h(\dots)}})}$ Normalized value of energy relative to distance and $\mathrm{W_s}$ ‹$\mathrm{W_E}$ Accordingly, the weights are determined for the effect of energy and memory whose values are determined in simulation.

## 7. Distributed Data Storage Mechanism

Suppose the storage sensor nodes (the wireless objects) hold the collected data. Periodically, the data is retrieved by the sink and removed from memory nodes. This periodic retrieval is a tool that allows the use of processor limited memory. Data recovery is based on the transfer of data collected by the network to the central station for further processing. To prevent data loss due to node failure or memory loss, nodes work in the following way. The data generated by the members of the cluster sensors to create R = 3 copies are distributed and stored in their header and the header of several neighboring nodes (parent and peer). In Figure 5. We show the scenario with R = 3, where R = 3 copies for two different modes. In the first case, three copies of the total data unit D1(cluster) generated by members of the cluster are stored in their cluster and cluster heads of parent and peer (2 and 5), respectively. In the second case, the repetition process stops if there is no existing node or there is no space in the neighboring node (memory limit) as well as energy limit. In this case, the last copy will not be saved.

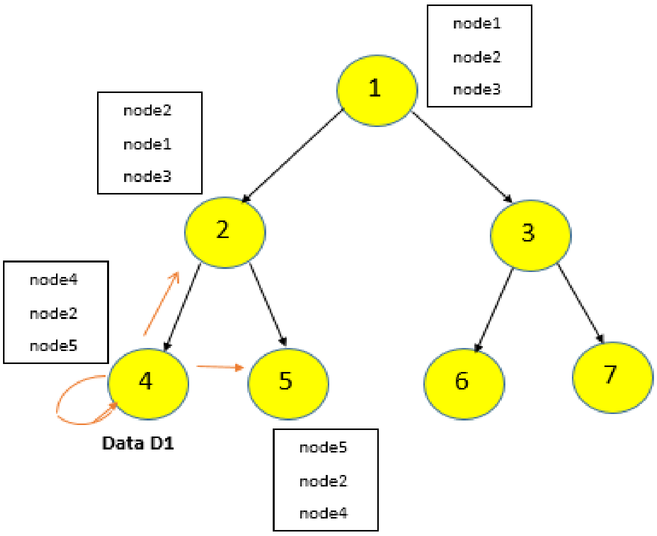


Fig.5. replication in the case of R = 3 copy

## 8. Results of Simulation and Evaluation of Proposed Method

The simulation program was implemented using the 2017 MATLAB software and implemented on the storage method distributed in the object-oriented internet environment using the combined PSO and K-means algorithm. This software is able to quickly and easily analyze your data in a distributed environment and also saves time and money. The computer used was selected with Intel ci5 processor specifications with 4 GB of memory and running Windows 7 operating system. Our simulation consists of a base station outside the field of measurement and the number of measurement nodes is evenly distributed in the environment with 60, 120, and 180 nodes. The dimensions of the network are 600 × 2000 meters and the connection mode of the nodes is in the form of a mesh network. The following result was obtained. In the simulation, the nodes are the same and have a capacity of 100 units. Therefore, if the simulation is performed with N nodes, the total available storage space is N × 100. And citation memory ceiling in the simulation, this amount will be and before completing it the data retrieval operation in the sink should take place. Figure6, which holds the total network data, shows how much memory is occupied. According to numerous experiments, 3 copies were selected for the simulation. The proposed method with three scenarios: 60, 120, and 180 knots with the previous method is 60 nodes compared.

### *A. Evaluation of stored data*

Considering that in the previous method [4] triple copy operations are performed by all nodes, therefore the data exchange rate is high and consumes a lot of energy. The previous method fills the memory very early because all the nodes are copied and additional duplicates will not be deleted so the volume of data stored by it and the occupancy of the available space quickly grow and over time its intensity is reduced due to the loss of data due to the completion of the capacity of the nodes but in our proposed approach, given that the data are not copied at nodes level at the data aggregation stage instead of that copy operation conducted in a limited number of cluster nodes after aggregation. Our method uses a header those nodes that are stronger are doing the copying, and all nodes do not copy operation. Therefore, the process of memory occupancy in all three scenarios of the proposed method is uniformly accomplished until the completion of the 600 second simulation period. low memory consumption and uniform growth the amount of memory consumption in the proposed method is quite evident. In the first scenario, the proposed method with 60 nodes is the amount of data stored in it compared to the previous method [4] and its superiority is known. Percentage improvement in the proposed method compared to the base method was 3.4%.

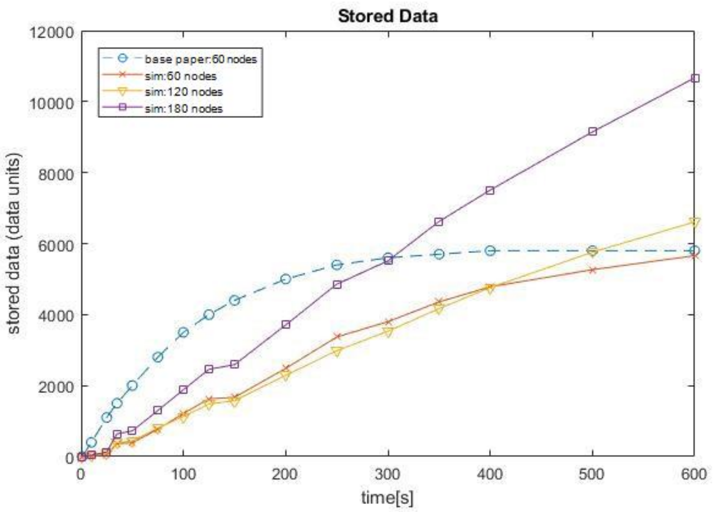


Fig.6. Evaluation of used memory (stored data)

### *B. Evaluation of data deletion due to memory limitation*

The main concern in the subject of the present research is the optimal use of memory, due to memory limitation, data deletion will be possible. Further, in Fig. 7, the number of units of data deleted during the simulated period is shown. as you can see in the figure, the previous method performed with 60 nodes [4] compared to the proposed method with 60 nodes in the beginning simulation has a better situation However, the status of deletion of data in the form of multiplication increases. due to the above figure in implementing with 120 nodes and 180 nodes, given that the memory available in each node is constant and equal to 100 units of data. Therefore, with the increasing number of nodes and network expansion, we are faced with an increase in the number of deleted data; when the number of nodes is high and the memory capacity is constant, the drop is rising because the grid is enlarged and the data has increased. In general, can be claimed that the proposed method is better suited to the results obtained from the basic method. The amount of data deletion along with the number of unique stored data reviewed, not copy data. In other words, deleting data that is no copy of which has been done is important. The percent recovery of data loss due to memory limitation, the proposed method compared to basic research 86.32 percent.

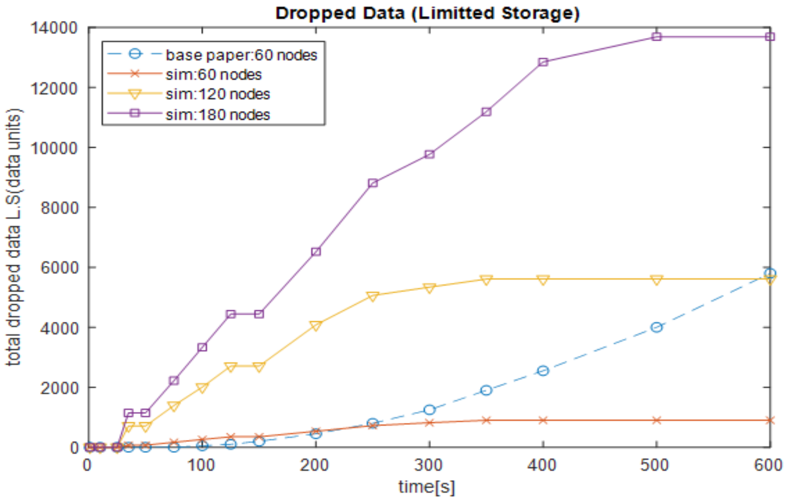


Fig.7. Deleted data due to memory limitation

### *C. Evaluation of Unique Stored Data*

One of the primary conditions and other goals in this research is to store purely stored data and how to store data for later use so that they do not hurt the network performance and keep as much information as possible in the shortest possible time. Be figure 8 shows this. As long as the sensors are not dead and their energy and battery are not over, they will be counted as Unicode data. comparing the previous method with the proposed method in 60 nodes is that the previous method [4] because of the use of the whole network storage space causing little unique data stored in the network, while the proposed method is due to better memory usage has caused a lot of data to be stored on the network. Due to fixed network conditions, 60 nodes are better than 120 nodes and 180 nodes.

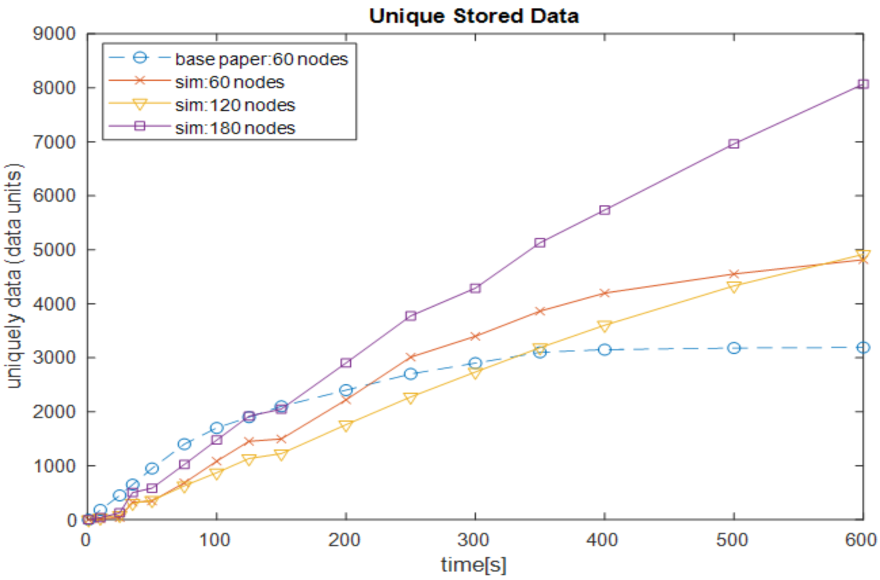


Fig.8. Unicode stored data

### D. Estimation of network lifetime and energy consumption

Fig. 9 shows the number of nodes lost due to the completion of energy during the simulation. As shown in the figure, with increasing the number of nodes, while keeping other conditions constant, the number of dead nodes increases. Also, the time of the first death is reduced. death is the growth nodes are almost identical. to reduce node deaths and increase network lifetimes, the periodicity of collecting data by sensors and the recovery period can be increased.

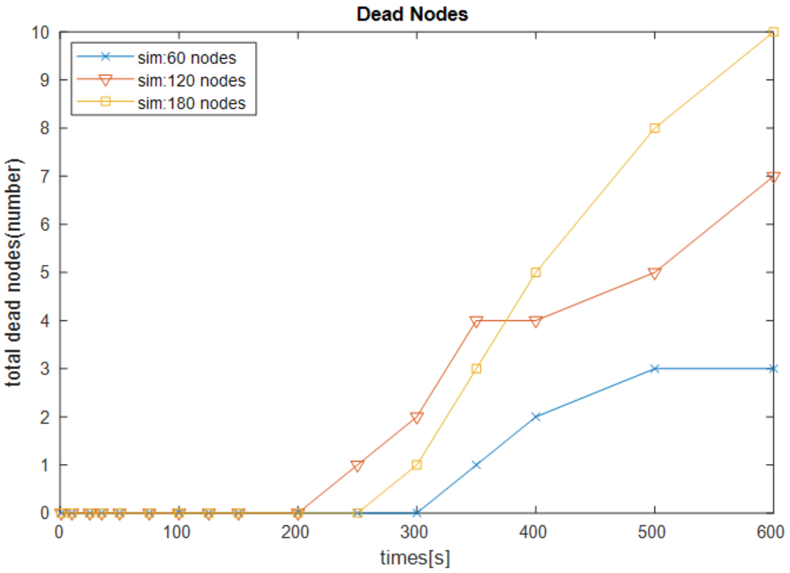


Fig.9. Number of lost nodes relative to time

Table II shows the amount and percentage of energy consumed in each of the three scenarios of the proposed method. As it is known, the large number of nodes is a lot of activity and high energy consumption but the third column shows the percentage of energy consumed in total energy of the grid; we can see from the perspective of energy consumption spacing is near so low energy consumption, and this becomes the percentage of total energy consumption of the network. And also the fourth column, which shows the ratio of energy consumption to the number of nodes (average per capita energy consumption). energy consumption to the number of nodes (average per capita energy consumption), indicates that with the increasing number of nodes, per capita consumption of energy decreases.

Table 2. Amount, The percentage and per capita energy consumption in simulation

| Energy consumption ratio to node number | Percentage of consumption | Energy consumption (energy unit) | Suggested scenario |
|---|---|---|---|
| 0.787425 | 15.75 | 47.2455 | Sim: 60 nodes |
| 0.7412858 | 14.83 | 88.9543 | Sim: 120 nodes |
| 0.6644272 | 13.29 | 119.5969 | Sim: 180 nodes |

Table 3. Relative index nodes death

| Missed nodes percentage | Number of dead knots | Total number of nodes | Suggested scenario |
|---|---|---|---|
| 0.05 | 30 | 60 | Sim: 60 nodes |
| 0.05833 | 7 | 120 | Sim: 120 nodes |
| 0.5556 | 10 | 180 | Sim: 180 nodes |

### *E. Assessment of network access capability*

One of the important assessments is accessibility, which means that it is known how many percent of the nodes are available and active over the lifetime of the network. For this purpose, the relation (8-1) has been used.

$$Availability(t) = 100 * \frac{(n-dead(t))}{n} \tag{18}$$

Where denotes the total number of nodes and representing the number of dead nodes and shows the time. Based on the results, accessibility is not so different from each other, and these lower fluctuations are more related to the clustering and data collection method, which is either random or based on non-deterministic and exploratory methods. It can be concluded from the results that the proposed method about accessibility is Scalable, and the number of nodes does not have much effect on accessibility due to the constant of other network conditions.

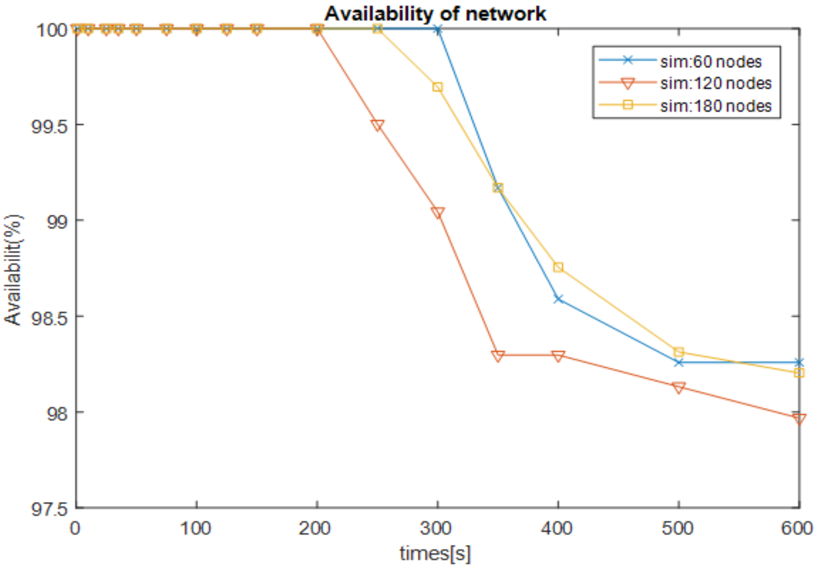


Fig.10. The availability of nodes in the network

### *F. Evaluation of network reliability*

The results indicate that reliability in the proposed first scenario with 60 is reduced to 98.35, and the safe value for the second scenario with 120 nodes is 98 and for the third scenario with 180 nodes is 98.18%. Increasing the nodes by fixed network conditions reduces reliability but the reliability reduction is minor and 98% is considered to be good network reliability. Increasing the period sensing the nodes and the recovery period can increase reliability by keeping the memory and energy constant. The reason for the loss of reliability is due to the increase in the number of nodes, the increase in data volumes, and also the fixed amount of memory nodes. So in using a tree plan and three copies, can define the relationship between the number of nodes and the node's properties to increase reliability.

$$Rel_{sys}(t) = \sum_{l=1}^{X(t)} \frac{Rel_l(t)}{X(t)} * 100 \qquad (19)$$

$X(t)$ the amount of data stored in the system at time $t. Rel_l(t)$ is equivalent to 1 if at least two copies of the lth data are stored in the system otherwise $Rel_l(t)$=0.

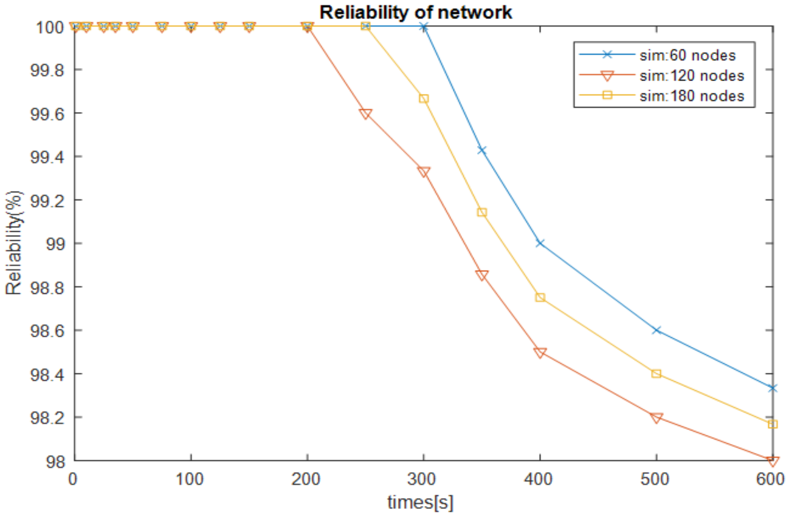


Fig.11. Reliability in the network

## 9. Conclusion

This paper describes a newly proposed method that is an optimal method for clustering and copying to increase the reliability and performance of data retention in the Internet of Things. The mentioned method tries to achieve the mentioned goals by applying the four parameters (Cluster head distance from the sink, Energy consumption of cluster heads, Cluster Consumption Memory, Distance of cluster members from cluster head) and using the combined algorithm of PSO, k-means, and decision tree. Based on the proper routing architecture of clusters and tree formation, this structure can reduce data transmission distance between measurement nodes, scalability, and communication load balance, reduce energy consumption and increase the life of measurement nodes and ultimately availability. Secure data on the IoT network. In addition, fast data retrieval is another requirement for data storage plans in critical times of applications, and the above method provides reliability in answering queries. The simulation results of the proposed method for distributed storage with three copies were increased to increase reliability and also to reduce energy consumption, and the results were evaluated in three proposed scenarios using the reference method. In general, according to the obtained results, the efficiency of the proposed method and the method of using the hybrid particle swarm meta-exploration algorithm with Kmeans were investigated and its efficiency for the proposed idea was proved. The issue of data redundancy, especially in the relations between objects of IoT is a major challenge. That's where we need methods to introduce sufficient redundancy with minimal communication cost to a network such that can the entire network data be retrieved after a failure, which is an important goal in this context. This article presented a data distribution scheme that reduces the communication cost and ensures data availability in wireless objects. In addition, due to the high consumption of memory and energy in the cluster nodes, especially the cluster nodes near the root of the binary tree, increasing the amount of memory and energy of nodes it is possible to increase the lifetime and capacity of the network.

## References

[1] Taheri, Negar, AND Jamali, Shahram. "Distributed Data Storage in the IoT: A Performance and Reliability Approach" Networking and Communication *Engineering*, 2020.

[2] Esmaeili, Mohammad, AND Jamali, Shahram, "A Survey: Optimization of Energy Consumption by using the Genetic Algorithm in WSN based Internet of Things", CiiT International Journal of Wireless Communication, 2016.

[3] L. Atzori, A. Iera, G. Morabito. "The Internet of Things: A survey". Journal of Computer Network, 2010.

[4] Pietro Gonizzi, Gianluigi Ferrari, Vincent Gay, Jérémie Leguay. ‘‘Data Dissemination Scheme for Distributed Storage for IoT Observation Systems at Large Scale". Journal of Information Fusion, 2013.

[5] Neenu M. Nair, J. Sebastian Terence. ‘‘Survey on Distributed Data Storage Schemes in Wireless Sensor Networks". Indian Journal of Computer Science and Engineering (IJCSE), 2014.

[6] Nukarapu Dharma, Tang Bin, Wang Liqiang, Lu Shiyong, "Data Replication in Data Intensive Scientific Applications with Performance Guarantee", IEEE Transactions on Parallel and Distributed Systems", 2011.

[7] B. Meroufel, G. Belalem ‘‘Managing Data Replication and Placement based on Availability", 2013.

[8] K. PIoTrowski, P. Langendoerfer, S. Peter. ‘‘tinyDSM: A Highly Reliable Cooperative Data Storage for Wireless Sensor Networks", 2009.

[9] A. Omotayo, M. Hammad, K. Barker, ‘‘a cost model for storing and retrieving data in wireless sensor networks)", 2007.

[10] Jamali S, Taheri N, Esmaeili M. A hybrid method for energy efficient data storage in the internet of things.J Commun Technol ElectronComput Sci. 2021;26:1-5.

[11] B. Zerhari, A. Lahcen, S.Mouline, ‘‘Big Data Clustering: Algorithms and Challenges" Conference Paper, 2015.

[12] Jana Neumann, Christoph Reinke, Nils Hoeller, Volker Linnemann. ‘‘Adaptive Quality-Aware Replication in Wireless Sensor Networks. In International Workshop on Wireless Ad Hoc, Mesh and Sensor Networks", 2009.

[13] Ángel Cuevas Rum ń, Manuel Urue ña Pascual, Ricardo Romeral Ortega, David Larrabeiti L ópez. ‘‘Data Centric Storage Technologies: Analysis and Enhancement. Sensors", 2010.

[14] A. Awad, R. Germany, F. Dressler. ‘‘Data-Centric Cooperative Storage in Wireless Sensor Network. 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL)", 2009.

[15] M. Albano, S. Chessa, F. Nidito, S. Pelagatti, ‘‘Dealing with nonuniformity in data centric storage for wireless sensor networks", IEEE Transactions on Parallel and Distributed Systems, 2011.

[16] H. Shen, L. Zhao, Z. Li, ‘‘A distributed spatial–temporal similarity data storage scheme in wireless sensor networks, IEEE Transactions on Mobile Computing", 2011.

[17] A. Awad, R. Germany, F. Dressler. ‘‘Data-Centric Cooperative Storage in Wireless Sensor Network. 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL)", 2009.

[18] A. Omotayo, M. Hammad, K. Barker, ‘‘a cost model for storing and retrieving data in wireless sensor networks, in: IEEE 23rd International Conference on Data Engineering Workshop (ICDE)", 2007.

[19] M. Takahashi, B. Tang, N. Jaggi, ‘‘Energy-efficient data preservation in intermittently connected sensor networks, in: IEEE 30th Conference on Computer Communications Workshops", 2011.

[20] L. Luo, C. Huang, T. Abdelzaher, J. Stankovic, Envirostore: ‘‘a cooperative storage system for disconnected operation in sensor networks, in: 26th IEEE International Conference on Computer Communications",2007.

[21] Y.-C. Tseng, F.-J. Wu, W.-T. Lai, ‘‘Opportunistic data collection for disconnected wireless sensor networks by mobile mules, Ad Hoc Networks, 2013.

[22] G. Maia, D.L. Guidoni, A.C. Viana, A.L. Aquino, R.A. Mini, A.A. Loureiro. ‘‘A Distributed Data Storage Protocol for Heterogeneous Wireless Sensor Networks with Mobile Sinks. Journal of Ad Hoc Networks", 2013.

[23] K. Piotrowski, P. Langendoerfer, S. Peter. ‘‘tinyDSM: A Highly Reliable Cooperative Data Storage for Wireless Sensor Networks. International Symposium on Collaborative Technologies and Systems", 2009.

[24] Raghavendra V. Kulkarni, Senior Member, ‘‘Particle Swarm Optimization in Wireless Sensor Networks:" A Brief Survey, 2008.

[25] Youguo Li, Haiyan Wu, ‘‘A Clustering Method Based on K-Means Algorithm", 2012.

[26] M. Esmaeili, S. Jamali, and H. S. Fard, "Energy-Aware Clustering in the Internet of Things by Using the Genetic Algorithm," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 12, no. 2, pp. 29-37, 2020.

[27] V. Kumar, S. B. Dhok, R. Tripathi, and S. Tiwari, ‘‘A review study on analytical estimation of optimal".2008

[28] S. Ruggieri,Dipartimento di information, Universita di Pisa. ‘‘Efficient C4.5",2010.

[29] C. Viana, T. Herault, T. Largillier, S. Peyronnet, F. Zaı¨di, Supple: a flexible probabilistic data dissemination protocol for wireless sensor networks, in: 13th ACM International Conference on Modeling, analysis, and simulation of wireless and mobile systems,2010.

[30] Amin Rezaeipanah, Hamed Nazari, MohammadJavad Abdollahi, " Reducing Energy Consumption in Wireless Sensor Networks Using a Routing Protocol Based on Multi-level Clustering and Genetic Algorithm ", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.10, No.3, pp. 1-16, 2020.DOI: 10.5815/ijwmt.2020.03.01

[31] Atul Kumar Pandey, Nisha Gupta, "An Energy Efficient Clustering-based Load Adaptive MAC (CLA-MAC) Protocol for Wireless Sensor Networks in IoT", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.9, No.5, pp. 38-55, 2019.DOI: 10.5815/ijwmt.2019.05.04

[32] Md. Imran Hossain, M. Mahbubur Rahman, Tapan Kumar Godder, Mst. Titasa Khatun,"Improving Energy Efficient Clustering Method for Wireless Sensor Network", International Journal of Information Technology and Computer Science(IJITCS), vol.5, no.9, pp.73-79, 2013. DOI: 10.5815/ijitcs.2013.09.07

## Authors' Profiles

**Negar Taheri** received the B.S. degree in Software Engineering from UCASJ (Applied science of Jahad) University, Tabriz, Iran, in 2014. She received her M.Sc. degree in Computer Engineering from the Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Ardabil, Iran, in 2018. Her research interests include Internet of Things and Wireless Sensor Networks.

**Shahram Jamali** is professor leading the Autonomic Networking Group at the Department of Engineering, University of Mohaghegh Ardabili. He teaches on computer networks, network security, computer architecture and computer syetems performance evaluation. Dr. Jamali received his M.Sc. and Ph.D. degree from the Dept. of Computer Engineering, Iran University of Science and Technology in 2001 and 2007, respectively. Since 2008, he is with Department of Computer Engineering, University of Mohaghegh Ardabil and has published more than 150 conference and journal papers.

**Mohammad Esmaeili** received the BSc degree in software engineering from the University College of Nabi Akram, Tabriz, Iran, and the MSc degree in software engineering from Science and Research Branch, Islamic Azad University, Ardabil, Iran, respectively in 2009, and 2016. Since March 2015, he is a member of the Computer Society of Iran. Currently, his research is focused on the Internet of Things.

**How to cite this paper:** Negar Taheri, Shahram Jamali, Mohammad Esmaeili, " Achieving Performability and Reliability of Data Storage in the Internet of Things ", International Journal of Engineering and Manufacturing (IJEM), Vol.12, No.1, pp. 12-28, 2022. DOI: 10.5815/ijem.2022.01.02