science, engineering and health studies

Research Article

Science, Engineering and Health Studies
https://li01.tci-thaijo.org/index.php/sehs
ISSN (Online): 2630-0087

Energy-efficient cloud integrated sensor based on clustering and multihop transmission

Kalyan Das[1,2] and Satyabrata Das[1]*

[1] Veer Surendra Sai University of Technology, Burla 768018, India
[2] Sambalpur University Institute of Information Technology, Burla 768019, India

*Corresponding author: Kalyan Das
kalyandas1983@gmail.com

Received: 7 December 2020
Revised: 2 February 2021
Accepted: 27 July 2021
Published: 13 June 2022

Citation:
Das, K., and Das, S. (2022). Energy-efficient cloud integrated sensor based on clustering and multihop transmission. Science, Engineering and Health Studies, 16, 22040001.

ABSTRACT

An energy-efficient sensor cloud model was proposed using the clustering algorithm and forwarding traffic to the node closet to the destination. Initially, the clustering algorithm created clusters of sensors using the closeness of measurements and distance between them. Following that, the cluster head algorithm selected the cluster head from the group whose measurement values were close to the mean value of all node measurements in that cluster. In the traditional approach, all nodes in the wireless sensor network were active and directly transmitted data to the sink. As a result, energy consumption was higher when compared to multihop transmission. The cluster heads generally communicated on behalf of all the cluster nodes in our proposed approach. That resulted in less number of active nodes that minimized energy consumption. After that, the cluster heads used multihop data transmission to forward the traffic to the destination using the shortest path algorithm. In our simulation, the accuracy of data transmitted by cluster heads was nearly equal to that of data transmitted by nodes. When compared to sending traffic directly to the destination, the multihop transmission of data could save energy. Our simulation showed that our proposed method can save energy, and also, the accuracy of transferred data is acceptable.

Keywords: clustering; wireless sensor network; multihop; cloud

1. INTRODUCTION

Sensors that communicate with each other wirelessly create the wireless sensor network (WSN). One of the most critical applications of WSN is environmental monitoring, which generally needs a lifetime of several months. The battery lifetime is finite, which is a drawback in such applications in the WSN. In cloud systems, the users get storage space and processing resources on a rental basis. The sensor cloud consists of the cloud systems and WSN, in which the user accesses the sensing devices using the cloud (Yuriyama and Kushida, 2010; Alamri et al., 2013; Das et al., 2018). The sensor owner attaches the sensors to the cloud system and removes his sensors according to the requirements. The sensor's owner gets paid if others use his sensors through the cloud. Saving power is challenging for the sensor-cloud system. The energy consumption for running the servers in the cloud system is vast, and the life span of the sensor's battery is finite. Cloud handles many requests on-demand from multiple applications. It takes a lot of effort and time to complete all of the requests. Shi et al. (2014) proposed a task framework in which tasks are classified into three types: periodic, time, and event. By removing redundant sampling tasks, the merge opt-H algorithm reduces the redundancy of tasks requested by different users. Samarah (2015) proposed a data prediction model that reduced energy consumption in sensor-integrated cloud systems. Kumar and Kumar (2018) have discussed a routing algorithm known as energy-efficient link stable routing to improve the life duration and preserve energy for smart devices. It performed better than other available routing algorithms.

Mishra et al. (2022) proposed a cost-sensitive approach to building a hierarchical fat-tree to minimize energy expenditure in the data center. Kurumbanshi and Rathkanthiwar (2018) surveyed the obstacles confronting the increasing life duration of the decentralized wireless network and categorized varieties of solutions. Kirtsaeng et al. (2016) proposed a temperature prediction to predict the maximum and minimum temperature of 24 to 72 h in advance. Panedpojaman and Intarit (2016) predicted maximum temperature using the study of parameters approached. Sethi and Anand (2019) proposed a system to predict disease. Ighravwe et al. (2018) proposed a neural network prediction method to predict the maintenance work's workforce size. Misra et al. (2017) compared the performance of WSN with sensor-cloud. Rao and Kamila (2018) proposed a Bayesian network-based energy-efficient ship monitoring system. Das et al. (2020) designed a data prediction and forecasting-based energy-efficient technique for the sensor cloud that reduces energy expenditure. Booranawong et al. (2018) proposed a model to get the sensor position in the WSN.

We have considered a sensor-cloud model covering a rectangular region, representing four tuples as $A = \{Loc_1, Loc_2, Loc_3, Loc_4\}$. Here each $Loc_i = \{Lat_i, Long_i\}$, $1 \le i \le 4$. Loc_1, Loc_2, Loc_3, Loc_4 represent the four coordinates of the rectangular area, and Lat_i, $Long_i$ represent the latitude and longitude of Loc_i, respectively. This area covers with n number of sensors, and the set of sensors node represents $S = \{S_1, S_2, S_3, \ldots, \text{and } S_n\}$. Each sensor node represents eight tuples. It can be mathematically expressed as S_j = <*sid*, *stype*, *sowner*, *status*, *scsp*, *slat*, *slong*, *QoS*>. Here, *sid* and *stype* represent serial id and type of sensor, respectively. The *sowner* represents the sensor's owner, and the *status* appears for the sensor's active status. The *status* is a boolean value where 0 shows that the sensor is in sleep mode and 1 shows an active mode. The cloud service provider (CSP) to which the sensor node attaches can represent as *scsp*. Latitude and the longitude of the sensor node represent by *slat* and *slong*, respectively. The CSP user uses the sensor as per his QoS requirement. QoS parameter of the sensor node represents *QoS*. Here each sensor node sends the gathered information to the gateway. A sensor node $S_{j'}$ uses P_j amount of energy to send data to another sensor node $S_{j''}$ with a data rate R_j, in area A. Here $1 \le j', j'' \le n$, and $j' \neq j''$ where j', j'' denotes two distinct nodes and n denotes the maximum number of nodes in the sensor cloud. Here P_j is calculated as follows (Guha et al., 2007):

$$P_j = E_1 + E' \, R_j \, E_{rec} d^{\sigma} \tag{1}$$

where E_1 denotes the ideal power consumption of sensor node $S_{j'}$, i.e., the nodes consume E_1 power even if it doesn't transmit or receive data; E' denotes a constant defined by the physical layer, and it is determined by the power consumed in exchanging the control packets; R_j denotes the data rate of the sensor $S_{j'}$; E_{rec} denotes the minimum energy required for successfully decoding at $S_{j''}$; d denotes the distance between $S_{j'}$ and $S_{j''}$; σ denotes the path loss exponent between 2 and 6, depending upon the environmental factors such as reflection, absorption, etc., and its larger values signify more energy consumption.

The total data rate used by the node $S_{j'}$ to send data to $S_{j''}$ is $R_{j'j''}$, and it is described as follows:

$$R_{j'j''} = O_{j'} + \sum_{j'''=1}^{n} R_{j'''j'}$$

where $1 \le j', j'', j''' \le n$, $j' \neq j'' \neq j'''$; $O_{j'}$ denotes the originating traffic at node $S_{j'}$; $R_{j'''j'}$ denotes the data rate between node $S_{j'''}$ and $S_{j'}$.

The sensor nodes sent the collected data to the gateway, then forwarded it to the cloud service provider. Cloud service users can access the data via the CSP. Using sensor node virtualization, any cloud user can obtain and use a single sensor's data. If the gateway is within the communication range of the sensor node, it can send data directly to it. Otherwise, the data can be sent to the gateway via multihop data communication using other nodes. The total energy consumption of the wireless sensor network is the sum of all the individual active sensors in the communication process. The power consumption of all active nodes can be calculated as follows:

$$P_{Total} = \sum_{i=1}^{n} P_j \tag{2}$$

where *status* $(P_j) = 1$.

The similarity measurements of temperature (measured sensor data) and the distance between the nodes were considered to form the clusters of the sensor nodes. Let assume, m numbers of clusters as C_1, C_2, C_3, …, and C_m. Moreover, each cluster having O_1, O_2, O_3, …, and O_m number of nodes, respectively. Each cluster selects a head node as H_1, H_2, H_3, …, and H_m, from the created m clusters. Each head node collects the data from the other nodes presented in that cluster and aggregates the data. After aggregating the data, it was forwarded to the gateway directly or by other clusters' head nodes using the multihop communication process. Here, the main objective was to form the cluster so that the minimum number of groups can be created, resulting in fewer head nodes or active nodes. The nodes' total energy to send the data to the gateway was less with a few active nodes. The energy consumed by a node to transmit or receive the data depended on the nodes' distance. Therefore, our second objective was to find the shortest path between the head node and the gateway. This problem should be addressed so that the root mean square error (RMSE), calculated on the actual data and the sent data, should be minimized. RMSE for a cluster C_k, $1 \le k \le m$ is mathematically represented as follows:

$$RMSE_k = \sqrt{\sum_{j=1}^{O_k} \frac{(T_j - T_k)^2}{O_k}} \tag{3}$$

where T_k is the transmitting sensor head temperature of the cluster C_k and Tj, $1 \le j \le Ok$ is the temperatures of all other sensor nodes. The average RMSE for all the clusters mathematically represents as follows:

$$RMSE = \frac{1}{m} \sum_{k=1}^{m} RMSE_k \tag{4}$$

where $RMSE_k$ is the RMSE of the cluster C_k, $1 \le k \le m$.

2. MATERIALS AND METHODS

The dataset was obtained from the MIT database group website (Bodik et al., 2004). The Intel lab dataset consisted of 54 sensors placed in the Intel Berkeley Research lab from February 28, 2004, to April 5, 2004. End-users connected to sensors through the cloud systems. The clustering of sensor nodes was carried out based on the similarity of measurements and distance. The cluster heads were selected, and they communicated on behalf of all the sensors of that cluster. The cluster heads also forwarded the data to the next cluster head using the shortest path, Dijkstra's routing algorithm, to reach the gateway. The energy-efficient model for the sensor cloud system explains in Figure 1.

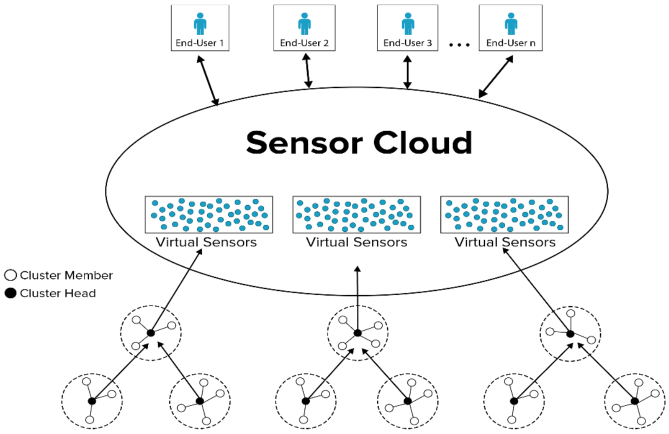

Figure 1. Energy-efficient sensor cloud system

Initially, the clustering algorithm was used to create the clustering of nodes based on the similarity of measurement and distance. We chose a cluster head whose measurement value was close to the mean of all nodes in the cluster. Cluster heads typically participated in the communication process on behalf of all cluster nodes. Our algorithm first reduced the number of nodes actively participating in the communication process, allowing the WSN to consume less energy. After that, we used the shortest path routing algorithm to forward the traffic among the cluster heads to the gateway. Generally, cluster heads directly communicate with the gateway, which consumes more energy due to long-distance transmission. In our approach, the cluster head forwarded the traffic to the nearest cluster head to the destination, which consumes less power.

2.1 Clustering algorithm

The clustering algorithm was used to group sensor nodes based on measurement similarity and distance. The temperature was treated as measured sensor data in algorithm. The clustering algorithm first computed the distances and differences between the measured temperature data. The greatest distance between all nodes and the greatest difference in measured temperature data was computed. The min-max normalization technique was used to normalize distance and temperature difference. The final normalized values were computed by assigning equal weight to the distance and temperature difference, 0.5. If the final normalized values are less than the threshold value for all nodes, group the nodes into clusters. If there are common nodes within different clusters, then merge these clusters. Final clusters of nodes are found without repetition of nodes. Table 1 explains the clustering algorithm.

Table 1. Pseudo code for clustering algorithm

Input: 2D matrix - *POS*, 1-D matrix - *TEMP*
Output: *m* number of clusters

```
for j = 1, 2, 3, …, n
  for j′ = 1, 2, 3, …, n
    DIS[j, j′]  = SQRT((POS[j, 1] - POS[j′, 1])^2 + (POS[j, 2] - POS[j′, 2])^2)
    TEMP_DIFF = TEMP[j] - TEMP[j′]
  end
end
max_distance = MAX (DIS)
max_temp_diff = MAX(TEMP_DIFF)
for j = 1, 2, 3, …, n
  for j′ = 1, 2, 3, …, n
    NORM_DIS[j, j′] = DIS[j, j′] / max_distance
    NORM_TEMP_DIFF[j, j′] = TEMP[j, j′] /max_temp_diff
    FINAL_NORM[j, j′] = NORM_DIS × 0.5 + NORM_TEMP_DIFF × 0.5
  end
end
for j = 1, 2, 3, …, n
  for j′ = 1, 2, 3, …, n
    if (FINAL_NORM[j, j′]) ≤ threshold
      cluster_no_node [j] = cluster_no_node [j] + 1
      cluster[j, cluster_no_node[j]]  = j
    end
  end
end
```

Table 1. Pseudo code for clustering algorithm (Continued)

```
Input: 2D matrix - POS, 1-D matrix - TEMP
Output: m number of clusters

for j′ = 1, 2, 3, …, n
  x = cluster(j, :)   // Copy the nodes of the jth cluster to x
  for j′ = j + 1, j + 2, j + 3, …, n
    y = cluster(j′, :); match = 0
    sx = LENGTH(x)
    sy = LENGTH(y)
    for j1 = 1, 2, 3, …, sx
      for j2 = 1, 2, 3, …, sy
        if (x[j1] == y[j2] && x[j1] ≠ 0 && y[j2] ≠ 0)
          match = 1
        end
      end
    end
    if match == 1
      more_node = cluster_no_node[j′]
      for j″ = 1, 2, 3, …, more_node
        cluster[j, cluster[j] + j″] = cluster[j′, j″]
      end
      cluster_no_node[j] = cluster_no_node[j] + more_node
    end
  end
end
final_cluster_nonode[] = 0
final_cluster = 0
for j = 1, 2, 3, …, n
  for j′ = 1, 2, 3, …, MAX(cluster_no_node)
    if ISMEMBER(cluster[j, j′], final_cluster) || cluster(j, j′) == 0
      continue
    else
      final_cluster_nonode[j] = final_cluster_nonode[j] + 1
      final_cluster[j, final_cluster_nonode[j]] = cluster[j, j′]
    end
  end
End
```

2.2 Cluster head selection algorithm

The nodes, which were nearer to each other, had nearly equal measured temperature values form clusters using the clustering algorithm. The cluster head selection algorithm was implemented after the clustering algorithm created clusters of nodes. The length of each cluster was first calculated in the cluster head selection algorithm. The sum of all measured temperature values and temperature means were computed. The cluster head selection algorithm selected a cluster head whose measurement value was close to the mean of all nodes in the cluster. Cluster heads typically participated in the communication process on behalf of all cluster nodes. Because it was simple to implement and minimized the RMSE of sensor and cluster head data, the cluster head selection algorithm only considered the closeness of measured temperature. Table 2 explains the cluster head selection algorithm.

Table 2. Pseudo code for cluster head selection algorithm

```
Algorithm - Pseudo code for cluster head selection algorithm

for k = 1, 2, 3, …, no_clustsers
  x1 = final_cluster[k, :]; sum = 0; len = 0
  for o = 1, 2, 3, …, LENGTH(x1)
    if x1[o] ≠ 0
      len = len + 1
    end
  end
  for o = 1, 2, 3, …, len
    sum = sum + TEMP[x1[o]]
  end
  mean_temp = sum / len
  diff = mean_temp - TEMP[x1[1]]
  NODE_HEAD[k] = x1[1]
  for o = 1, 2, 3, …, len
```

Table 2. Pseudo code for cluster head selection algorithm (Continued)

Algorithm - Pseudo code for cluster head selection algorithm

```
      if diff > (mean_temp - TEMP[x1[o]])
        diff = mean_temp - TEMP[x1[o]]
        NODE_HEAD[k] = x1[o]
      end
    end
  End
```

3. RESULTS

The parameters for the simulation are described in Table 3.

Table 3. List of parameters

Parameters	Values/unit
Number of sensor nodes	54
Area of simulation	41* 31 m^2
Simulation time	1 h
Nodes distribution	Random
Data rate	1Kbps
Transmission range	25/35 m
σ	4
Gateway position	Coordinate (0, 0)
Types of communication	Multihop

The threshold values for simulation were considered as 0.1, 0.15, 0.2, 0.25, 0.3 and 0.35 for the creation of clusters. MATLAB was used for simulation. The Intel lab dataset was used, in which the positions of the 54 sensor nodes were static and were distributed in a rectangular area of 41*31 m^2. The data transfer rate was 1 Kbps, and the sink position coordinated (0, 0). The transmission range was either 25 or 35 m. The path loss exponent, $\sigma = 4$, which arose in simulation's closed environment, was used. From every cluster, one cluster head was selected, which took part in data transmission. Multihop transmission was used among the cluster head to forward the traffic to the sink. Table 4 explains the threshold, RMSE, number of clusters, power consumption of the proposed approach using different transmission ranges.

Table 4. Threshold, RMSE, number of clusters, and power consumption of the proposed approach using different transmission range

Threshold	RMSE	Number of clusters	Power consumption (Joule)	Transmission range
0	0.0	54	42.12	25
0.1	0.0539	28	27.57	25
0.15	0.1252	13	8.64	25
0.2	0.1419	8	5.184	25
0.25	0.5394	3	7.92	35
0.3	0.6210	2	6.48	35
.35 or above	0.8685	1	2.88	35

The threshold and its corresponding number of clusters are explained in Figure 2. If the threshold value increased, the corresponding number of clusters decreased. Increasing threshold value allowed more dissimilar nodes to group into a single group in distance and measurement. A large threshold allowed far distance nodes and more differences in measurement nodes to group into a single group. The threshold and its corresponding RMSE are explained in Figure 3.

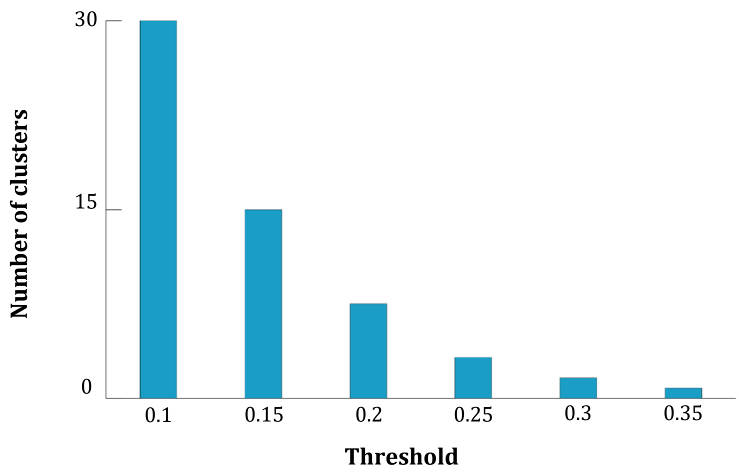

Figure 2. Threshold and its corresponding no of clusters

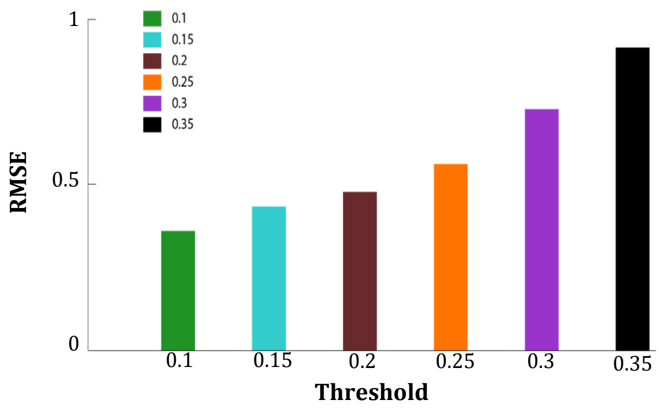

Figure 3. Threshold and its corresponding RMSE

4. DISCUSSION

The low threshold means the nodes are closer in the distance and nearly the same in measurement values. If the threshold is low, the number of cluster heads will be more similar in measurement, and closer nodes must form the cluster. If the threshold increases, no cluster nodes decrease. The RMSE increases if the threshold increases. We assume that the cluster heads communicate on behalf of the entire cluster. If we increase the threshold, then no cluster heads decrease. As a result, fewer nodes take part in communication. Few nodes will take part in transmission, and fewer data will be communicated; thus, less energy will be consumed. If the range of transmission is constant, then power consumption decreases if fewer nodes communicate. More transmission range means long-distance communication, which consumes more energy. The comparison of energy consumption in the proposed approach and ACOSIM (Lemos et al., 2019) is explained in Figure 4.

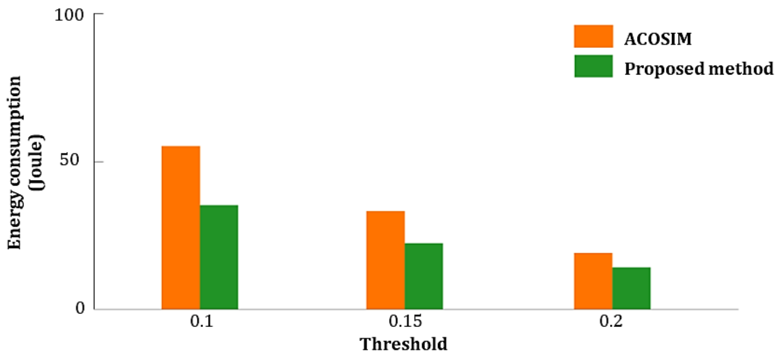

Figure 4. Energy consumption in the proposed approach and ACOSIM

According to the findings, sending data to a cluster head near the destination can save energy, compared to sending directly to the sink. The cluster heads directly communicate with the sink in the ACOSIM algorithm, which consumes more power due to the long-distance transmission. Long-distance transmission consumes more energy because the distance between the source and the sink is directly proportional to the energy consumption. In our proposed approach, the cluster head transmitted data to the nearest cluster head to reach the sink using the shortest path algorithm if the distance between the cluster head and the sink was more than the range of transmission of the sensors. The cluster heads can also directly transfer the data to the sink if the distance between the source and the sink was less than the transmission range of the sensor. As a result, the distance of transmission was less, and the energy consumption is also less. The RMSE for the thresholds of 0.1, 0.25, and 0.2 are 0.0539, 0.1252, and 0.1419, respectively. Hence, it signifies the high accuracy of the proposed method. In the proposed approach, the actual sensor measured data and transmitted cluster head data were nearly equal.

5. CONCLUSION

The clustering method grouped the sensor nodes into clusters based on the nodes' temperature values and distance. The cluster head selection algorithm chose a cluster head whose temperature value was close to the mean of all node temperatures in that cluster. Only the cluster heads were active, and they transferred data on behalf of the entire cluster. As a result, there were fewer active nodes, resulting in lower energy consumption. Following that, cluster heads used the shortest path algorithm to forward traffic to the destination via multihop data transmission. Traditionally, all nodes were active and transferred data directly to the destination, consuming more energy. The simulation showed that the proposed method saves energy, and the accuracy of transferred data is also acceptable.

REFERENCES

Alamri, A., Ansari, W. S., Hassan, M. M., Hossain, M. S., Alelaiwi, A., and Hossain, M. A. (2013). A survey on sensor-cloud: architecture, applications, and approaches. *International Journal of Distributed Sensor Networks,* 9(2), 917923.

Bodik, P., Hong, W., Guestrin, C., Madden, S., Paskin, M., and Thibaux, R. (2004). Intel lab data. *Intel Berkeley Research lab,* [Online URL: http://db.csail.mit.edu/labdata/labdata.html] accessed on November 11, 2020.

Booranawong, A., Sopajarn, J., Sittiruk, T., and Jindapetch, N. (2018). Reduction of RSSI variations for indoor position estimation in wireless sensor networks. *Engineering and Applied Science Research*, 45(3), 212-220.

Das, K., Das, S., Darji, R. K., and Mishra, A. (2018). Survey of energy-efficient techniques for the cloud-integrated sensor network. *Journal of Sensors,* 2018, 1597089.

Das, K., Das, S., and Mohapatra, A. (2020). A novel energy-efficient sensor cloud model using data prediction and forecasting techniques. *Karbala International Journal of Modern Science*, 6(3), 267-274.

Guha, R. K., Gunter, C. A., and Sarkar, S. (2007). Fair coalitions for power-aware routing in wireless networks. *IEEE Transactions on Mobile Computing*, 6(2), 206220.

Ighravwe, D. E., Oke, S. A., and Adebiyi, K. A. (2018). Selection of an optimal neural network architecture for maintenance workforce size prediction using grey relational analysis. *Engineering and Applied Science Research*, 45(1), 1-7.

Kirtsaeng, S., Sukthawee, P., Khosuk, B., Masthawee, F., Pantong, N., and Taorat, K. (2016). Development of daily temperature prediction model for northeastern thailand using artificial neural networks. *Engineering and Applied Science Research*, 43(S3), 487-490.

Kumar, K., and Kumar, S. (2018). Energy efficient link stable routing in internet of things. *International Journal of Information Technology,* 10(4), 465-479.

Kurumbanshi, S., and Rathkanthiwar, S. (2018). Increasing the lifespan of wireless adhoc network using probabilistic approaches: a survey. *International Journal of Information Technology*, 10(4), 537-542.

Lemos, M., Rabelo, R., Mendes, D., Carvalho, C., and Holanda, R. (2019). An approach for provisioning virtual sensors in sensor clouds. *International Journal of Network Management,* 29(2), e2062.

Mishra, J., Sheetlani, J., and Reddy, K. H. K. (2022). Data center network energy consumption minimization: a hierarchical FAT-tree approach. *International Journal of Information Technology*, 14(1), 507-519.

Misra, S., Chatterjee, S., and Obaidat, S. M. (2017). On theoretical modeling of sensor cloud: a paradigm shift from wireless sensor network. *IEEE Systems Journal*, 11(2), 1084-1093.

Panedpojaman, P., and Intarit, P. (2016). Maximum temperature prediction for concrete sections during cooling phase. *Engineering and Applied Science Research*, 43(S2), 294-298.

Rao, M., and Kamila, N. K. (2018). Bayesian network based energy efficient ship motion monitoring. *Karbala International Journal of Modern Science*, 4(1), 69-85.

Samarah, S. (2015). A data predication model for integrating wireless sensor networks and cloud computing. *Procedia Computer Science,* 52, 1141-1146.

Sethi, D., and Anand, J. (2019). Big data and WBAN: prediction and analysis of the patient health condition in a remote area. *Engineering and Applied Science Research*, 46(3), 248-255.

Shi, H. L., Li, D., Qiu, J. F., Hou, C. D., and Cui, L. (2014). A task execution framework for cloud-assisted sensor networks. *Journal of Computer Science and Technology*, 29(2), 216-226.

Yuriyama, M., and Kushida, T. (2010). Sensor-Cloud infrastructure - Physical sensor management with virtualized sensors on cloud computing. In *Proceeding of the 13th IEEE International Conference on Network-Based Information Systems (NBiS)*, pp. 1-8. Takayama, Japan.