INDIAN INSTITUTE OF INFORMATION TECHNOLOGY LUCKNOW

5th SEMESTER PROJECT

Clustering Algorithm for Heterogeneous Devices in Cognitive Internet of Things

---

Submitted To:

Dr. Vishal Krishna Singh

Submitted By:

Vikash Tripathi (LIT2016004)

Contents

1 Motivation

Information Technology has become a very essential part of our daily life because of automation of tasks and reduction in efforts. We have started to depend on technologies to such an extent that we cannot imagine our lives without them.

Clustering of devices is done for sending only relevant data packets that can furnish the need of all clustered devices as against sending the data packets of each device separately.

Thus the core motivation behind developing an algorithm for clustering devices in IoT is to facilitate better services provided to an end user by observing patterns that exist among data of different types. Not only does this analysis help in providing better services but also allow efficient data transmission and thus enables high resource utilization. The present is always driven by the aim to achieve something new that helps in overall improvement of human life therefore we tried to propose a methodology which if implemented in real scenarios, can help in improving the functioning of various current domains targeted by Internet of Things.

2 Abstract

With the development of information network, the popularity of Internet of Things (IoT) is an irreversible trend, and the intelligent demands for IoT is becoming more and more urgent. How to improve the cognitive ability of IoT is a new challenge and therefore has given rise to the emergence of cognitive IoT (CIoT).

In this project, a two level hierarchy of device clustering is proposed in which at the first level, devices are clustered into homogeneous clusters and the clusters so formed are heterogeneous among themselves. The clustering of devices is done using Modified Mean Shift Clustering that makes use of spatial as well as data correlation between the nodes deployed in a given environment. Then for efficient data transmission from each cluster, a cluster head selection is made using random selection allocation technique in order to facilitate data aggregation at the cluster head as all nodes within a cluster are sensing homogeneous data. Then at the second level, which is our Cognitive Processing Layer, multimodal data correlation is studied between the heterogeneous data send by cluster heads of different clusters formed at the first level of clustering.

Matlab Simulations are carried out to support the idea of clustering and cluster head selection at the first level. Results show that the proposed methodology can effectively improve the quality of data transmission and can facilitate to study heterogeneous data correlation among a larger set of deployed nodes in a given environment.

3 Introduction

Cognitive IoT is the use of cognitive computing technologies in combination with data generated by connected devices and the actions those devices can perform.

Cognition involves three key elements –

1. Understanding
2. Reasoning
3. Learning

In Internet of Things, many data sources exist that may provide related information or context for better understanding and decision making. The ability to analyze different types of data, including digital sensor data, audio, video, unstructured textual data, location data and so on, and to identify correlations and patterns across these data types is the core idea behind clustering of heterogeneous devices, in terms of data sensed, in cognitive Internet of Things.

The data in Cognitive IoT is collected from multiple heterogeneous devices and different domains, such as numerical observations, measurements from different devices, text from social media stream etc. In order to meet the social enterprise needs and extract more valuable data information, data correlation needs to be studied among such entities.

In this project, we focus on how to cluster the devices according to data correlation and device distribution.

4 Problem Definition

Efficient way of clustering devices which can sense heterogeneous data so as to facilitate better services provided to end users connected to servers or to one another through Internet of Things and also at the same time, allow efficient data transmission for better resource utilization.

5 Methodology and Implementation

A two level hierarchy of device clustering is proposed as follows :-

First level - Devices are clustered into homogeneous clusters using Modified Mean Shift Clustering that makes use of spatial as well as data correlation between the nodes deployed in a given environment.

Then for efficient data transmission from each cluster, a cluster head selection is made using random selection allocation technique.

Data transmission from clusters is improved by data aggregation at the cluster heads which can be done because all nodes within a cluster are sensing homogeneous data.

Second Level - Cognitive Processing Layer which does multimodal data correlation analysis between the heterogeneous data send to it by cluster heads of different clusters formed at the first level of clustering.

5.1 Modified Mean Shift Clustering

Mean Shift is a powerful and versatile non parametric iterative algorithm that can be used for lot of purposes like finding modes, clustering etc.

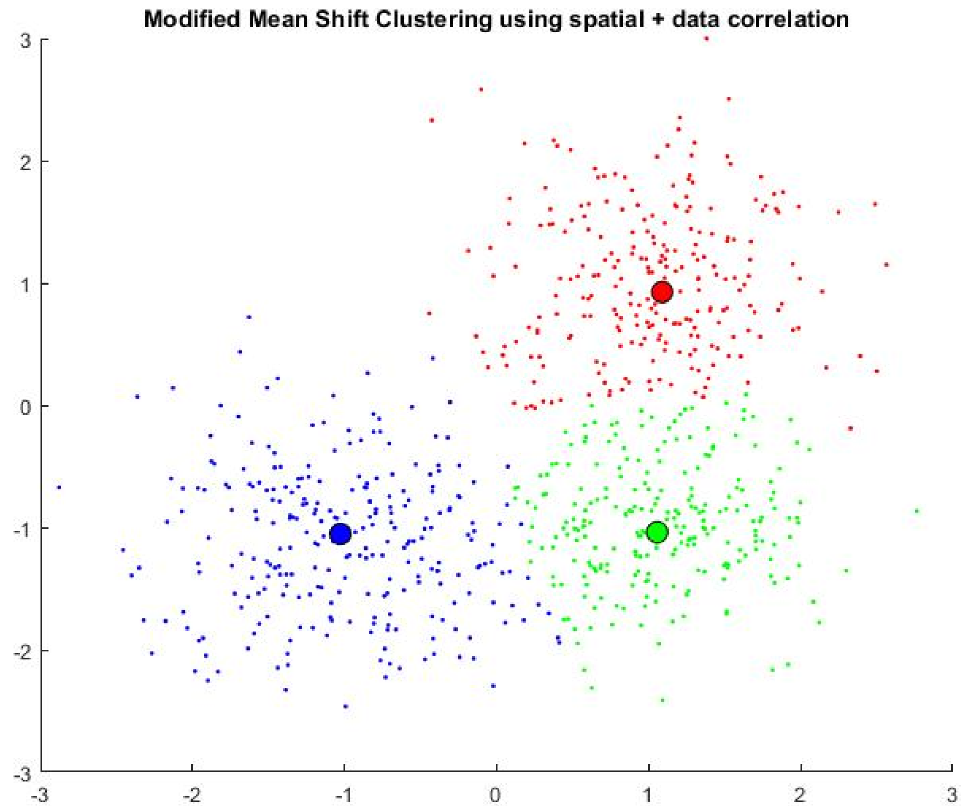

Figure 1: Cluster formation

- For each data point, Mean shift associates it with the nearby peak of the data set's probability density function. Mean shift defines a window around it and computes the mean of the data point. Then it shifts the center of the window to the mean and repeats the algorithm till it converges. In Modified Mean Shift algorithm, though the window shifts to the mean but still it includes only those data points in it which have certain percentage of similarity with the data values sensed by nodes in the previous mean cluster. Thus this facilitates spatial correlation and data correlation based clustering

5.2 The Cognitive Layer

The workload on cognitive layer reduces because instead of handling all data simultaneously from all devices in the network, it handles aggregated data sent from cluster heads. Suppose, if the cognitive layer's capability was to handle 500 distinct devices earlier, then now it can handle around 250000 devices due to the two level hierarchy.

6 Matlab Code

```
%%%%%%%% Random Cluster Head selection within a given cluster and ...
    then using data aggregation to transmit aggregated data from ...
    cluster head to sink (in our case Cognitive Processing Layer)

close all;
clear;
clc;

%%%% Area of Operation %%%%

% Area Dimensions in meters %
xm = 100;
ym = 100;

% Total number of nodes in the area %
n = 100;

% Number of Dead Nodes %
dead_nodes = 0;

% Coordinates of the Sink (Cognitive Processing Layer)(location is ...
    predetermined in this simulation) %
sinkx = 50;
sinky = 190;

%%%% Energy Values %%%%

Eo = 2; % Initial Energy of a Node (in Joules) %

% Energy required to run circuity (both for transmitter and receiver...
    ) %
Eelec = 50*10^(-9); % units in Joules/bit
ETx = 50*10^(-9); % units in Joules/bit
ERx = 50*10^(-9); % units in Joules/bit

% Transmit Amplifier Types %
Eamp = 100*10^(-12); % units in Joules/bit/m^2 (amount of energy ...
    spent by the amplifier to transmit the bits)

EDA = 5*10^(-9); % Data Aggregation Energy in Joules/bit %

k = 4000; % Size of data package in bits %
p = 0.05; % a 5 percent of the total amount of nodes used in the ...
    network is proposed to give good results

No = 1; % Number of Clusters %
rnd = 0; % Rounds of Operation %
```

```
operating_nodes = n; % Current Number of operating Nodes %
transmissions = 0; % Total number of transmissions till all nodes ...
    become dead %

% Flags for telling whether all nodes are dead or not %
temp_val = 0;
flag1stdead = 0;

%%%% Use of Protocol %%%%

%% Plotting a Homogeneous IoT network cluster %%
for i = 1:n

    IOT(i).id = i;                  % Sensor's ID number
    IOT(i).x = rand(1,1) * xm;      % X-axis coordinates of a node
    IOT(i).y = rand(1,1) * ym;      % Y-axis coordinates of a node
    IOT(i).E = Eo;                  % Nodes energy level (initially ...
        set to be equal to "Eo")
    IOT(i).role = 0;                % Node acts as normal if the value...
         is "0", if elected as a cluster head then it gets the value...
         "1" (initially all nodes are normal)
    IOT(i).cond = 1;                % States the current condition of ...
        the node. When the node is operational its value is equal to...
         "1" and when dead it is equal to "0"
    IOT(i).rop = 0;                 % Number of rounds the node was ...
        operational
    IOT(i).rleft = 0;               % Rounds left for node to become ...
        available for Cluster Head selection
    IOT(i).dtch = 0;                % Node's distance from the cluster...
         head
    IOT(i).dts = 0;                 % node's distance from the sink (...
        cognitive processing layer)
    IOT(i).tel = 0;                 % States how many times the node ...
        was selected as a Cluster Head
    IOT(i).rn = 0;                  % Round in which node got selected...
         as cluster head
    IOT(i).chid = 0;                % Node ID of the cluster head

    hold on;
    figure(1)
    plot(IOT(i).x, IOT(i).y, 'or', sinkx, sinky, '*b');
    title 'An IoT Cluster';
    xlabel '(metres)';
    ylabel '(metres)';
    legend('Nodes of a given cluster', 'Sink (Cognitive Processing ...
        Layer)');
end

%% Execution of the protcol %%

while operating_nodes > 0
```

```
    rnd % Displays Current Round %
    t = (p/(1-p*(mod(rnd,1/p)))); % Threshold Value %

    tleft = mod(rnd,1/p); % Re-election Value %

    % Reseting previous Cluster Head flags in the Network %
    CLhead = 0;
    flag = 0;

    energy = 0; % Reseting Amount Of Energy Consumed In the Network ...
       in the Previous Round %

        % Cluster Heads Selection %

        for i = 1:n

            IOT(i).role = 0;       % reseting node role
            IOT(i).chid = 0;       % reseting cluster head id

            if IOT(i).rleft > 0
               IOT(i).rleft = IOT(i).rleft - 1;
            end

            if (IOT(i).E > 0) && (IOT(i).rleft == 0) && (flag == 0)
                generate = rand;

                if generate < t
                    IOT(i).role = 1;     % Assigns the node, the role...
                        of a cluster head
                    IOT(i).rn = rnd;     % Assigns the round in which...
                        the node was selected as a cluster head
                    IOT(i).tel = IOT(i).tel + 1; % Increments the ...
                       number of times the node was selected as a ...
                       cluster head
                    IOT(i).rleft = 1/p-tleft;     % Rounds for which ...
                       the node will be unable to become a cluster ...
                       head
                    IOT(i).dts = sqrt((sinkx - IOT(i).x)^2 + (sinky ...
                       - IOT(i).y)^2); % Calculates the distance ...
                       between the sink and the cluster head
                    CLhead = 1;
                    CL(CLhead).x = IOT(i).x; % X-axis coordinates of...
                        selected cluster head
                    CL(CLhead).y = IOT(i).y; % Y-axis coordinates of...
                        selected cluster head
                    CL(CLhead).id = i; % Assigns the node ID of the ...
                       newly selected cluster head to an array
                    flag = 1;
                end

```

```
    end
end

% Caclulating the distance between nodes and cluster head %

% We calculate the distance 'd' between the node that is ...
   transmitting and the cluster head that is receiving with...
    the
% following equation : d = sqrt((x2 - x1)^2 + (y2 - y1)^2)
% where x2, y2 are the coordinates of cluster head and x1, ...
   y1 are the coordinates of the transmitting node

for i = 1:n
    if (IOT(i).role == 0) && (IOT(i).E > 0) && (CLhead > 0) ...
       % if node is normal
        d(1) = sqrt((CL(1).x - IOT(i).x)^2 + (CL(1).y - IOT(...
           i).y)^2);

        IOT(i).dtch = d(1); % Assigns the distance of node ...
           to cluster head
        IOT(i).chid = CL(1).id; % Cluster head ID
    end
end

% Energy Dissipation for normal nodes %

for i = 1:n
    if (IOT(i).cond == 1) && (IOT(i).role == 0) && (CLhead >...
        0)
        if IOT(i).E > 0
            ETx = Eelec * k + Eamp * k * IOT(i).dtch^2;
            IOT(i).E = IOT(i).E - ETx;
            energy = energy + ETx;

            % Dissipation for cluster head during reception
            if IOT(IOT(i).chid).E > 0 && IOT(IOT(i).chid)...
               .cond == 1 && IOT(IOT(i).chid).role == 1
                ERx =(Eelec)*k; %% EDA
                energy = energy + ERx;
                IOT(IOT(i).chid).E = IOT(IOT(i).chid).E - ...
                   ERx;

                if IOT(IOT(i).chid).E ≤ 0  % if cluster head...
                   's energy depletes with reception
                    IOT(IOT(i).chid).cond = 0; % Node ...
                       becomes non - operational
                    IOT(IOT(i).chid).rop = rnd; % Rounds ...
                       till it was operational
                    dead_nodes = dead_nodes +1; % Dead nodes...
                        count increases
```

```
                            operating_nodes = operating_nodes - 1; %...
                                Operational nodes count decreases
                        end
                    end
                end

                if IOT(i).E <= 0       % if nodes energy depletes with...
                    transmission
                    dead_nodes = dead_nodes +1; % Dead nodes count ...
                       increases
                    operating_nodes = operating_nodes - 1; % ...
                       Operational nodes count decreases
                    IOT(i).cond = 0; % Node becomes non - ...
                       operational
                    IOT(i).chid = 0;
                    IOT(i).rop = rnd; % Rounds till it was ...
                       operational
                end
            end
        end

        % Energy Dissipation for cluster head %

        for i = 1:n
            if (IOT(i).cond == 1)  && (IOT(i).role == 1)
                if IOT(i).E > 0
                    ETx = (Eelec + EDA) * k + Eamp * k * IOT(i).dts...
                       ^2;
                    IOT(i).E = IOT(i).E - ETx;
                    energy = energy + ETx;
                end

                if  IOT(i).E <= 0      % if cluster heads energy ...
                   depletes with transmission
                    dead_nodes = dead_nodes +1;
                    operating_nodes = operating_nodes - 1;
                    IOT(i).cond = 0; % Node becomes non - ...
                       operational
                    IOT(i).rop = rnd; % Rounds till it was ...
                       operational
                end
            end
        end

        if operating_nodes < n && temp_val == 0
            temp_val = 1;
            flag1stdead = rnd;
        end

        transmissions = transmissions + 1;
        if CLhead == 0
```

```
            transmissions = transmissions - 1;
        end

        rnd = rnd + 1; % Next Round

        trans_operating_nodes(transmissions) = operating_nodes; % ...
            Array to hold number of operating nodes per transmission
        round_operating_nodes(rnd) = operating_nodes; % Array to ...
            hold number of operating nodes per round

        if energy > 0
            trans_energy(transmissions) = energy; % Array to hold ...
                energy used per transmission
        end
end

% Plotting Simulation Results %

    % Operating Nodes per Round %
    figure(2)
    plot(1:rnd, round_operating_nodes(1:rnd), '-r', 'Linewidth', 2);
    title ('Operating Nodes per Round')
    xlabel 'Rounds';
    ylabel 'Operational Nodes';
    legend('Operational Nodes in a given Round');
    hold on;

    % Operating Nodes per Transmission %
    figure(3)
    plot(1:transmissions, trans_operating_nodes(1:transmissions), '-...
        r', 'Linewidth', 2);
    title ('Operational Nodes per Transmission')
    xlabel 'Transmissions';
    ylabel 'Operational Nodes';
    legend('Operational Nodes in a given Transmission');
    hold on;

    % Energy Consumed per Transmission %
    figure(4)

    plot(1:flag1stdead, trans_energy(1:flag1stdead), '-r', '...
        Linewidth', 2);
    title ('Energy consumed per Transmission till all nodes are ...
        alive')
    xlabel 'Transmission';
    ylabel 'Energy (Joules)';

    legend('Energy consumed in a given Transmission');
    hold on;
```

7 Simulation Results

Here we present the results generated by our algorithm.

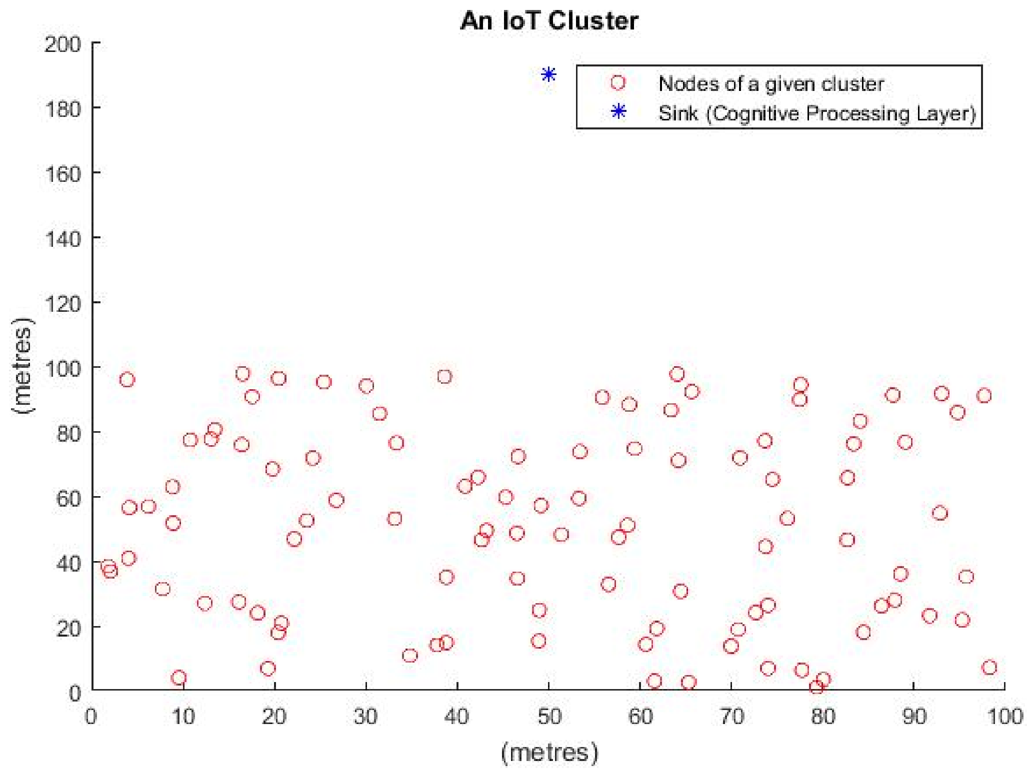

Figure 2: A Cluster out of the various clusters formed in Modified Mean Shift Clustering

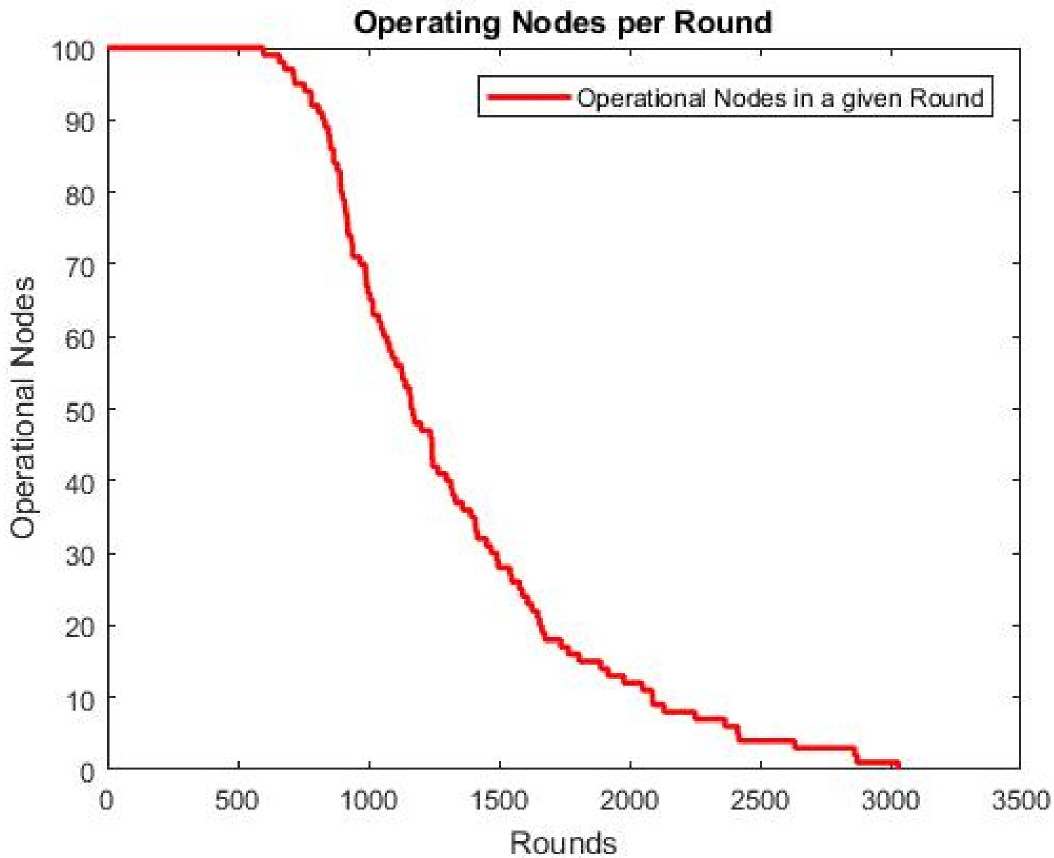

Figure 3: Operational nodes w.r.t Rounds.

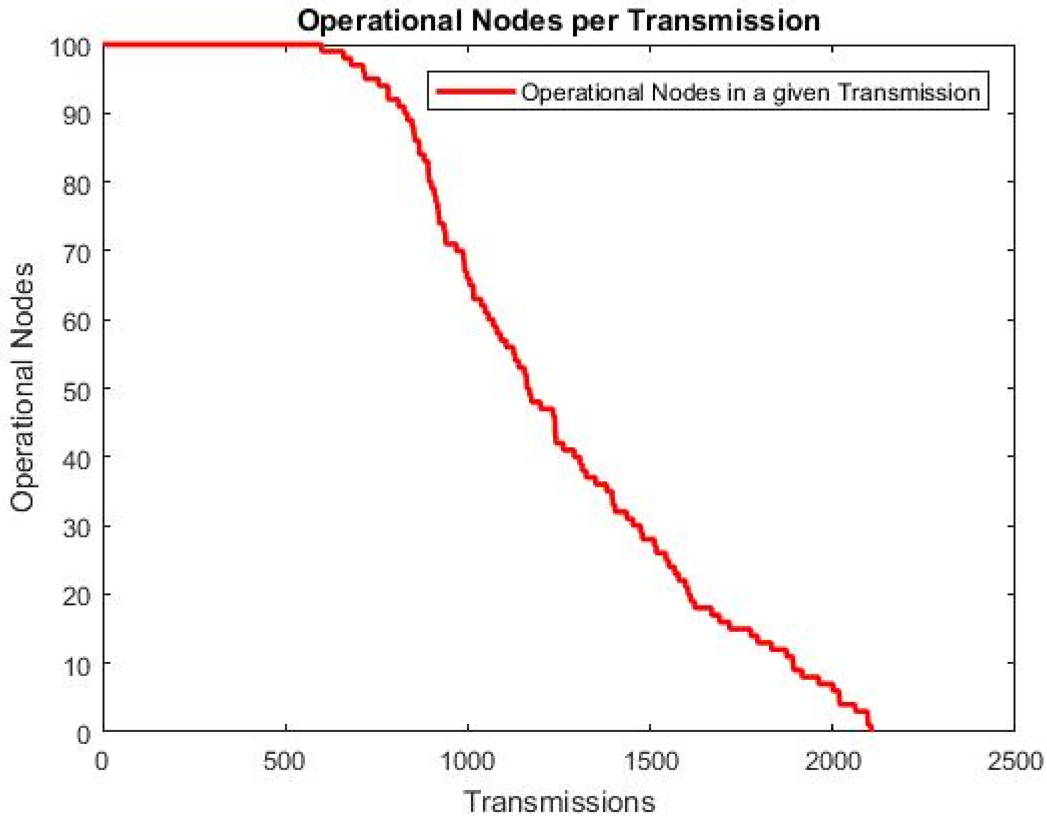

Figure 4: Operational nodes per transmission.

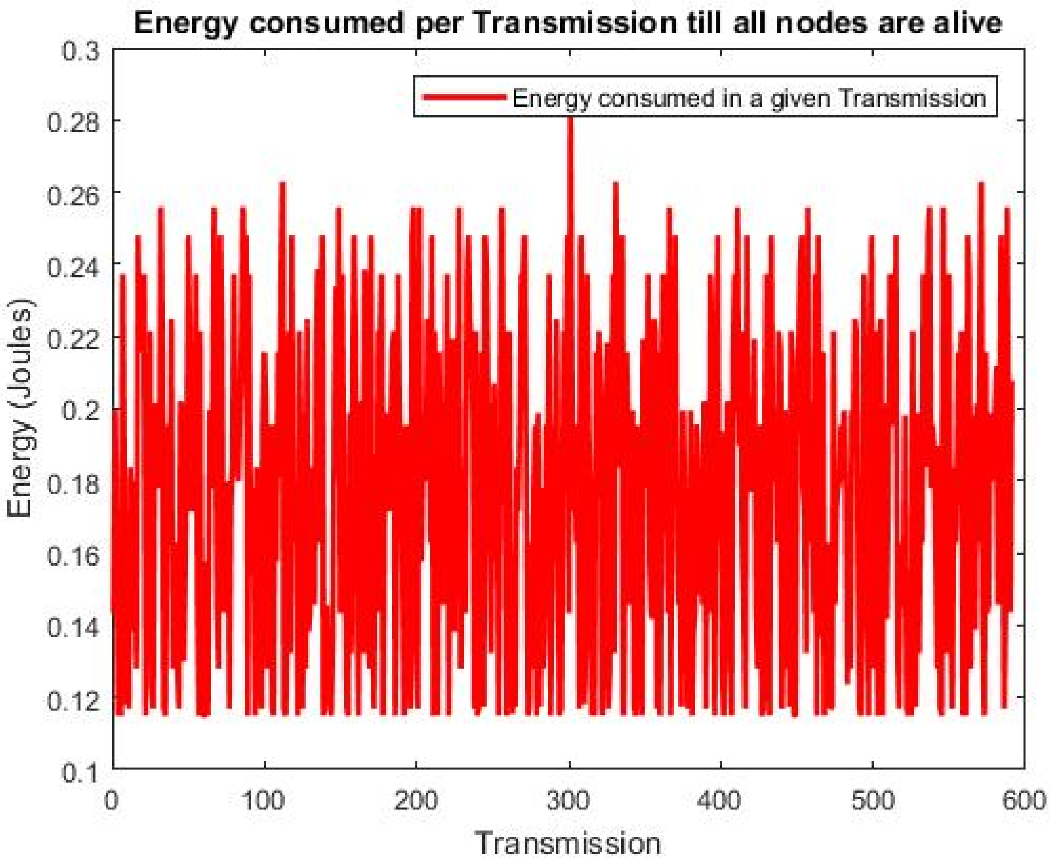

Figure 5: Energy Consumed per transmission till nodes are alive

8 Conclusion

In our project we devised an alogrithm that formulates clustering based on two levels which is capable of classifying the device according to the data correlation and device distribution.
The algorithm clusters the heterogeneous devices in CIoT according to their correlation by using the result of the data correlation mining model.

Extensive simulations are performed to evaluate the proposed algorithm. The results show that the designed algorithm has the potential to transform into a practical technique in CIoT.

9 References

1. Kai Lin, Di Wang, Fuzhen Xia, and Hongwei Ge, "Device Clustering Algorithm Based on Multimodal Data Correlation in Cognitive Internet of Things", IEEE Internet of Things Journal, vol. 5, no. 4, August 2018,

2. Zhikui Chen, Song Yang, Liang Li and Zhijiang Xie, "A Clustering Approximation Mechanism based on Data Spatial Correlation in Wireless Sensor Networks"
`https://ieeexplore.ieee.org/document/5479626?reload=true`

3. Top 5 Clustering Algorithms ,
`https://towardsdatascience.com/the-5-clustering-algorithms-data-sci`

4. Mean Shift Clustering,
`https://spin.atomicobject.com/2015/05/26/mean-shift-clustering/`

5. Correlation Clustering,
`https://en.wikipedia.org/wiki/Correlation_clustering`

6. Clustering high-dimensional data,
`https://en.wikipedia.org/wiki/Clustering_high-dimensional_data`