CHAPTER 6

EVALUATION, RESULTS AND DISCUSSION

6.1 Introduction

This chapter presents the achieved evaluation, results and discussion based on the experimental design of the Enhancement Clustering Algorithm with 3D Euclidean Distance using Correlated Degree parameters. The four scenarios evaluated in this experimentation include cooperative, selfish, malicious and failure nodes. Lastly, this chapter is concluded with a discussion on the analyses of the results.

6.2 Experimental Set up

In order to determine the correlated degree using the 3D Euclidean Distances, four different experimental scenarios were considered. The first scenario was to test a cooperative node scenario to measure the correlated degree under the event where the initial energy level is below the 0.5 *Joules* threshold. The second scenario tested the failure node to assess the effects of the corrected degree on network connectivity. The third scenario assessed the effects of a selfish node. While the fourth scenario was a malicious node to observe the effects of the corrected degree of network connectivity nodes on the network partitioning. The experimental setup of each scenario as in Table 6.1(Page 113).

The Enhancement Clustering Algorithm (ECA) in Chapter 5 was implemented to form the network cluster. In order to form sensor node with the highest energy level to become the cluster head (CH) based on the energy threshold, the node chooses a random number between 0.0 to 1.0 where, CH is required to determine the cluster region. Based on the algorithm, the CHs are selected using three parameters which are distance D, packet β, and energy ℓ.

Table 6.1: Parameters setup for experiment

Parameters	**Value**
Number of nodes	100
Correlated degree	0.0 to1.0
Initial energy	0.5 *Joules*
Packet size	100 *bytes/packets*
Radius/distance	200 *meters*
Energy Threshold	>= 1.0
Cooperative nodes	0 to 0.69 connectivity range
Malicious nodes	> 1.0 connectivity range
Failures nodes	< 0.0 connectivity range
Selfish nodes	≥0.7 connectivity range
Broadcasting Range	15*m*
Energy Per Message	10 *aJ/bit/m*^{2}
Energy Per Step	50 *aJ/bit*
Energy Produced	5 *aj/bit*

In the selfish scenario, 20 nodes were randomly selected to become selfish in which their initial energy level is set below 0.5. The CH was selected based on the data sensed from the nodes within the radius of 200 *meters* and the packet delivery ratio from the nodes within the packet size of *100 bytes*. Next, the CH formed the clusters based on the calculated correlated degree. The selfish node did not take part in the formation because when a cluster head node initiates a route discovery to another cluster head node, the selfish neighbour node may be reluctant to broadcast the route request from a cluster. In this case, the selfish node behaves like a failure node. Although it is possible for a selfish node to forward control packets, the situation could worsen if the node selects a selfish node as the next hop and send data to it. When all neighbours of the nodes are selfish, a node is unable to establish any communication

with other nodes at a distance of more than one-hop away. Hence, a node is isolated by its selfish neighbours. Note that selfish nodes can still communicate with other cooperative neighbours, different from failure nodes.

All sensor nodes in 3D Euclidean distance is made up of $a_1, a_2, a_3 \ldots a_{100}$ representing the coordinates of the nodes in 3D Euclidean Distance to become cluster head and as anchor nodes. Distance is represented as (D), packet transfer (β), and energy (ℓ). Thus, T_{a_i} represents the parameters to become a cluster head. The calculation of correlated degree on the cluster heads is based on the 3D Euclidean distance between one-hop neighbour nodes using the 3D Euclidean Distance method. According to Shankar and Shanmugavel (2014), a cooperative node begins to change its status to a selfish node when the initial energy level is below 0.5 *J*oules. Hence, when their energy falls below the threshold value $T(a)$, the node will be automatically considered as a selfish node.

The selfish neighbour node that is reluctant to broadcast the route request from another node is assumed as a failure node. It is also possible for the node to forward control packet. Consequently, the node may discard all data to be forwarded hindering the communication between nodes. When all neighbour nodes are selfish, the nodes are unable to establish any communication with one another at a distance of more than one-hop away. Nodes are isolated by the selfish neighbours to partition the network.

The correlated degree is independent, thus, the resizing of the correlated degree does not require communication of nodes within the event areas to compute the new cluster to which they belong to Villas et al., (2014). The nodes will then select themselves as CHs based on the correlated degree threshold if it falls between 0 to 0.69. This value is referred to cooperative node. However, if the node value is less than the threshold value, the node will not be selected as a CH. Once a node is selected as a CH, it advertises its decision to the nodes in its close vicinity to form a cluster. Based on the signal strength of the CHs advertisement message, a

node which falls within the threshold value is considered as a cooperative node (Rao and Banka, 2017; Wang et al., 2018).

Unlike cooperative nodes, malicious nodes actively forward control packets. However, if the nodes are in route, they will selectively or randomly drop data packets, reorder packets, or increase jitters, which is especially harmful to Transmission Control Protocol (TCP) traffic. Thus, if the node has a neighbour as the next hop, then the node will not eventually lose communication with nodes that are at least two-hop away. If all neighbours are nodes, the nodes will be isolated by malicious neighbours, similar to the case of having selfish neighbours because malicious nodes that broadcast a routing beacon with extra high power could lead to a large number of nodes attempting to use it as their next hop in their route to the sink. As a result, those that are sufficiently far away would be simply sending their messages into oblivion. Similarly, this scenario could also be caused by a wormhole attack. A malicious node could convince other nodes that are normally multiple hops from the sink node that they are just one hop away. Hence, these nodes would try to send their packets directly to the sink node, which eventually would not be delivered.

A sensor node is grouped into disjoint sets, where each set are managed by a designated CH selected from among the sensor nodes. The cluster members will then send their collected observations (which are likely to be highly correlated) to their CH.

6.3 Evaluation on Network clustering performance

This section discusses the effects of network clustering performance towards energy consumption and network connectivity using Enhancement Clustering Algorithm (ECA). Three parameters which are distance, packets and energy were assessed in this experiment to measure network connectivity and energy consumption. Moreover, ECA which is the proposed algorithm, will be used as the main comparison standard in the discussion. The evaluation of

network connectivity will have ECA compared with Linked Clustering Algorithm (LCA), Low-Energy Adaptive Clustering Hierarchical (LEACH), Energy Efficient Clustering (EEC), Rotated Hybrid Energy-Efficient Distributed Clustering (R-HEED), and Power-Efficient Gathering in Sensor Information Systems (PEGASIS) algorithms. During the evaluation, the best trend for each scenario will be selected.

Each of the scenarios, namely the cooperative, failure, selfish, and malicious nodes for different network connectivity and the number of nodes are presented in Figure 6.1 to Figure 6.4, respectively. The effects of the sensor node on the network connectivity using four different scenarios were also discussed.

In this section, the performance of the proposed ECA was evaluated by comparing it to the LEACH (Palan, N. G., Barbadekar, B. V., & Patil, 2017), EEC (H. Lin, Wang, & Kong, 2015), R-HEED (Transactions, 2016) and PEGASIS (Hao, Chen and Huo, 2011; Somauroo and Bassoo, 2019) and LCA (Lakshmaiah, K., Krishna, S. M., & Reddy, 2020) algorithms. Hence, the correlated degree during experimental were calculated between nodes and the number of nodes based on the 3D Euclidean distance for node behaviour. Since ECA is a representative behaviour node used in 3D Euclidean distance for clustering algorithms, it was also selected for performance comparison.

6.3.1 Network Connectivity

a) Cooperative Node Scenario

Firstly, in order to clearly observe the effects of node cooperative, the recovery time was set at 0 so that the effects of node failures will be uncorrelated. The experimental also had the network connectivity set to decrease to ensure that the cooperative node varies only due to selfish node and attack. Hence, by adjusting the selfish threshold $T(a)$ and attack on the network connectivity, a series of cooperative node with network connectivity values (0 to 0.69)

according to Jr et al. (2017; Liu, Yang and Xue-song (2015), Wang et al. (2016) and Xing and Wang (2010) were obtained using ECA. Then, the node was calculated based on the correlated degree equation for a_i using Equation 4.5 with cooperative node values. The analytic results are provided in Figure 6.1(A), where the curves with markers represent the network connectivity measured from experimental data and the ones without markers are for analytical results.

Based on Figure 6.1(B), the increase in network connectivity was obvious when the correlation degree increased the connectivity requirements of node a, indicating that a stronger network connectivity has more connection in terms of the cluster and topology (Bhavana and Murthy, 2014). An interesting observation is that the network connectivity of the node increases rapidly from 0.2 to 0.63 as the cooperative node increases, for example, the node for $a = 1.0$ is almost 0.2 as a_i range value from 0.2 to 0.63. This observation from random graphs indicated the existence of a critical value of cooperative node range from (0.2 to 0.63) for network connectivity (Figure 6.1(B)).

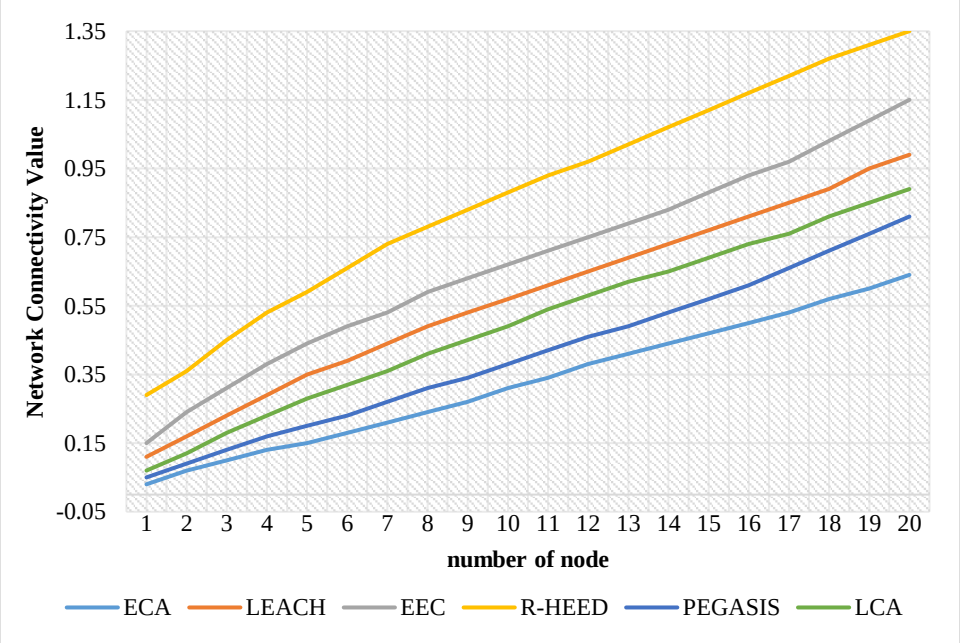

Figure 6.1(A): The effects of cooperative node on network connectivity

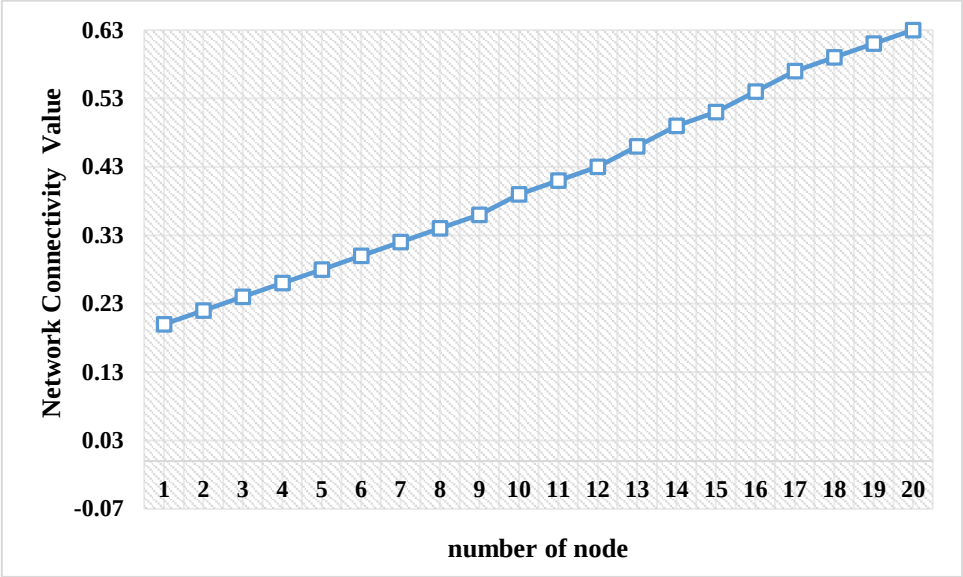

Figure 6.1(B): The effects of cooperative node on network connectivity

Figure 6.1(A) represents the network connectivity comparison for the effect node of cluster. So, when the sensor nodes are less in number, the effect node of the clusters generated is lesser. However, the effect of cooperative node is increased on the proposed ECA (0.64) as the network connectivity increase (Figure 6.1(A)). Moreover, the values of the LEACH (0.99), EEC (1.15), R-HEED (1.35) and PEGASIS (0.81) LCA (0.89) algorithms gradually increased as the network connectivity increases. The LEACH (0.99), EEC (1.15), R-HEED (1.35), PEGASIS (0.81) and LCA (0.89) algorithms still have more network connectivity value of the clusters facilitates lower network connectivity based on correlation of the clustering more nodes can be turned into sleep or failure mode. Finally, ECA is more suitable as it ends with less correlation of the clusters compared to LEACH, EEC, R-HEED, PEGASIS, and LCA. Referring to Table 6.2, ECA scored 11%, while LEACH, EEC, R-HEED, PEGASIS, and LCA scored 17%, 20%, 23%,14%, and LCA 15% respectively, indicating that ECA has better network connectivity of the cooperative nodes.

Table 6.2: The comparison of cooperative node on network connectivity

Algorithm	Network Connectivity Value	Comparison
ECA	0.64	11%
LEACH	0.99	17%
EEC	1.15	20%
R-HEED	1.35	23%
PEGASIS	0.81	14%
LCA	0.89	15%

The network connectivity value is based on the cooperative value calculated using the 3D Euclidean Algorithm. Based on Table 6.2 and Figure 6.1(A), show the values the effect node of cooperative nodes on network connectivity using each of the five algorithms were ECA 0.64, LEACH 0.99, EEC 1.15, R-HEED 1.35, PEGASIS 0.73, and LCA 0.89. The results indicated that the ECA value on network connectivity was higher for cooperative nodes when two or more eligible nodes on a cluster have the same distance or sufficient packet and energy to the sink.

The represent each round of each behaviour node in experimental scenarios of the ECA is more suitable as it ends with less correlation of the clusters compared between LEACH, EEC, R-HEED, PEGASIS, LCA algorithms ECA is great on network connectivity of the cooperative nodes. To find out the cooperative nodes on network connectivity of each algorithm. Every sensor node of the energy threshold value $(T(a)>1)$ is compared with the correlated degree of cooperative nodes on network connectivity. If the correlated degree is greater than energy threshold value $T(a)>1$, the sensor node will periodically broadcast message to its neighbouring nodes to inform that it will be the CH. Finally, the shows cooperative nodes can take part in cluster head so that the sensor node does not affected as Figure 6.1(A).

b) Scenario failure node

A scenario with failure node behaviour uses 20 sample nodes. In order to explain the effects of the failure nodes on network connectivity, the misbehaviour node was set at 0 by tuning all node as a cooperative node k, whereby, the impact of misbehaving nodes were eliminated. The initial energy of, failure node θ are set in the range value of the number of nodes was compared with the length of the network connectivity of the cluster, which was obtained by adopting the above algorithm. Therefore, in this study, initial energy values, which have different values at LEACH, EEC, R-HEED, PEGASIS, and LCA were considered, that is failure node on the network connectivity respectively. The network connectivity curve after the simulation as shown from this Figure 6.3. Then, the network connectivity for a_i = 20 nodes is calculated against each of the failure node values. The analytical results are plotted in Figure 6.2. Based on the observation, the node increased the connectivity segment as the failure node in ECA value (0.71) increases, especially after a cooperative value fall between the range of 0.0 to 0.69 to become a cluster. For example, when failure node a_i < 0.0, the connectivity for node a_i is 0.0, which implies that the network is disconnected almost surely. While an almost sure connectivity is achievable only if the node a_i > 0.0.

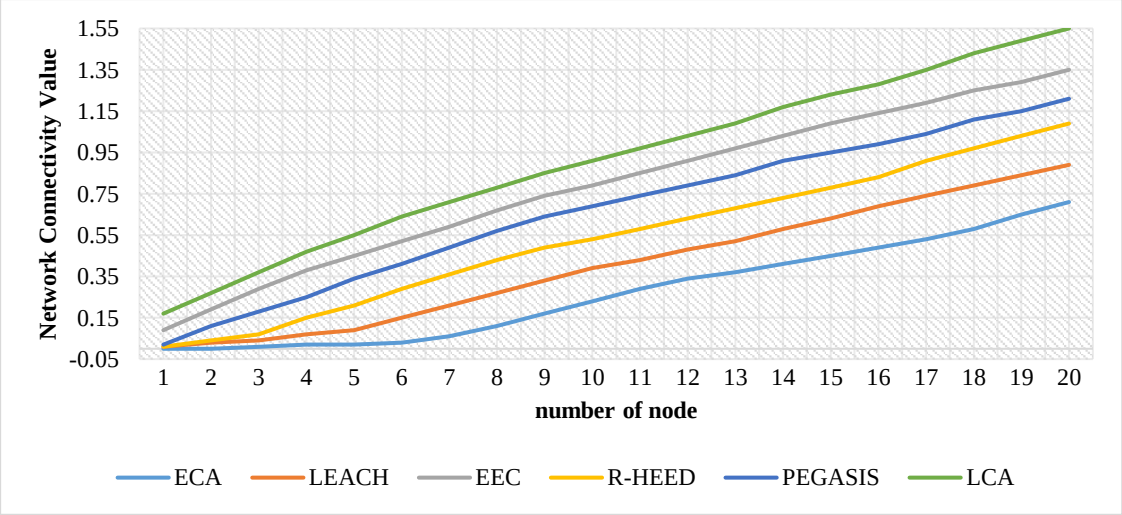

Figure 6.2: The effects of failure node on network connectivity

The second scenario discusses, the effect of the node failure and network connectivity network of the network. A network connectivity is formed whenever at least one path between any two nodes in a cluster exists. The effects of failure nodes on network connectivity illustrated in Figure 6.2 listed the respective values for ECA (0.71), LEACH (0.89), EEC (1.35), R-HEED (1.09), PEGASIS (1.21), and LCA (1.55). The ECA value for network connectivity was the better of among the other algorithms for failures node. Table 6.3 shows the comparison of the failure node scenario.

Table 6.3: The comparison of failure node on network connectivity

Algorithm	Network Connectivity Value	Comparison
ECA	0.71	10%
LEACH	0.89	13%
EEC	1.35	20%
R-HEED	1.09	16%
PEGASIS	1.21	18%
LCA	1.55	23%

The effects of failure nodes on network connectivity in a cluster are depicted in Figure 6.2. The ECA yielded great value in network connectivity. Since the network connectivity value of the effect node on the clusters segment increases in ECA (0.71), it can maintain the same cluster. Hence, there may not be significant effects on energy efficiency. The network connectivity values are very high for LEACH (0.89), EEC (1.35) R-HEED (1.09), PEGASIS (1.21), and LCA (1.55) almost indicating no network connectivity value that can be turned into a failure network. Based on Table 6.3, ECA scored 10% much better than the values from LEACH 13%, EEC 20%, R-HEED 16%, PEGASIS 18%, and LCA 23%. ECA is great on network connectivity of the failures nodes.

The present of each round of each behaviour node in experiment scenarios represents of the ECA is more suitable as it ends with less correlation of the clusters compared between LEACH, EEC, R-HEED, PEGASIS, and LCA algorithms. ECA is great on network

connectivity of the failures nodes. To find out the failure nodes on network connectivity of each algorithm, every sensor node of the energy threshold value $(T(a) > 1)$ is compared with the correlated degree of cooperative nodes on network connectivity. If the correlated degree is greater than energy threshold value $T(a) > 1$, the sensor node will periodically broadcast message to its neighbouring nodes to inform that it will be the CH. Finally, the shows failure nodes can take part in cluster head so that the sensor node does not affected. This confirms the comparison of failure node on network connectivity from the theoretical analysis.

c) Scenario selfish node

In the third scenario, a selfish node scenario was effaced from the rest of the nodes (Table 6.4). Based on Figure 6.3, ECA is value of 0.41 recorded a changes increased as the network connectivity value of 0.41 increases, LEACH 0.53, EEC 0.89, R-HEED 0.71, PEGASIS 0.97, and LCA 0.81 increases as the network connectivity. Comparatively, ECA scored a better network connectivity value for the selfish node. Therefore, the results indicated that even when a node correlated, the network connectivity of the proposed ECA performs better than the other algorithms.

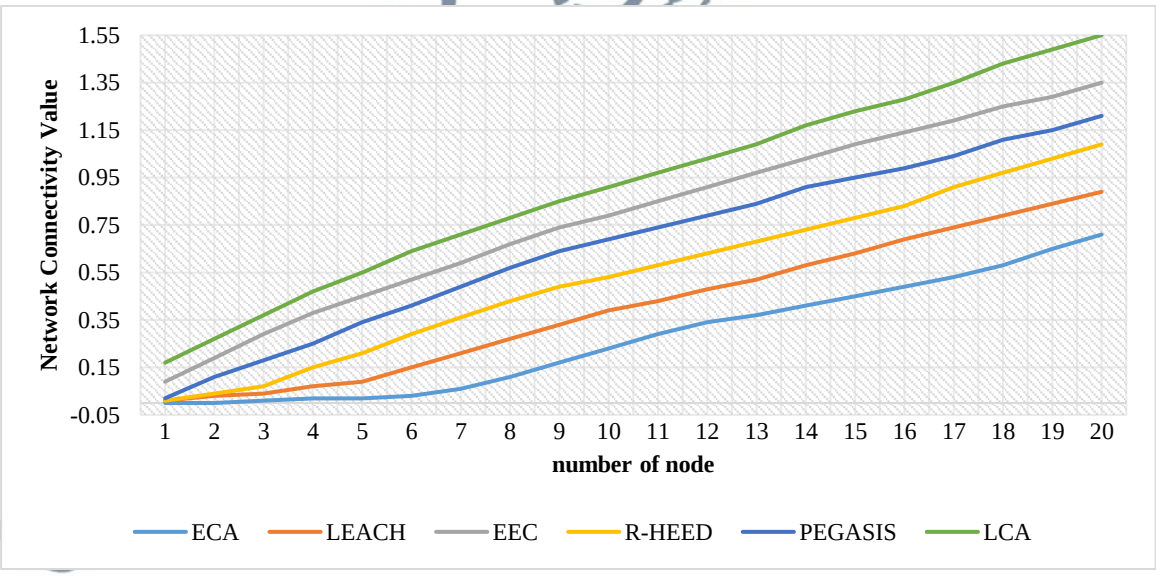

Figure 6.3: The effect of selfish node on network connectivity

Each of the selfish node scenarios used 20 sample nodes. Suppose a selfish node value ranging from $a_i \geq 0.7$ to < 1.0, and a node cannot connect to the other nodes when this node is located too far away from any other node. Thus, it can be seen by this example in the wireless packet-forwarding networks with selfish nodes according to Jr et al., (2017), Q. Liu et al., (2015), F. Wang et al., (2016) and Xing and Wang., (2010). When CH node initiates a route discovery to another CH node, the selfish neighbour node may be reluctant to broadcast the route request from the cluster. In this case, the selfish node behaves like a failure node. However, selfish nodes could also choose to forward control packets. However, if a node selects a selfish node as the next hop to send data communication between the nodes will be interrupted where all forwarded data will be discarded. So, when all neighbours of nodes are selfish, no communication will be established with other nodes at a distance of more than one-hop away. In this case, a node is isolated by its selfish neighbours. It is worth noting that selfish nodes can still communicate with other nodes that are cooperative neighbours, different from failure nodes. For example, the network connectivity for node a_{10} = 0.21 does not decrease considerably until the node becomes selfish when the node value is between $a_i \geq 0.7$ and < 1.0. Therefore, misbehaving nodes are still active in the network layer where they do not affect the distance of active nodes a_{10}, which is an important factor for network connectivity.

Table 6.4: The comparison of selfish node on network connectivity

Algorithm	Network Connectivity Value	Comparison
ECA	0.41	10%
LEACH	0.53	12%
EEC	0.89	21%
R-HEED	0.71	16%
PEGASIS	0.97	22%
LCA	0.81	19%

Table 6.4 illustrates the comparison network connectivity of the number of nodes on a cluster indicated that when the sensor nodes are less in number, the number of nodes in a cluster generated is less. However, as the sensor nodes decrease the network connectivity in the number of nodes, the network connectivity of the proposed ECA (0.41) is a better value for network connectivity almost rapidly showed as in Figure. 6.3. The LEACH (0.53), EEC (0.89), R-HEED (0.71), PEGASIS (0.97), and LCA (1.81) algorithms gradually increased as network connectivity increases. These are algorithms still a high value of the network clustering on network connectivity value as in correlation based on clustering more nodes can be turned into the selfish node.

For network connectivity with more correlated nodes, ECA is suitable as it ends with a number of nodes in clusters compared to LEACH, EEC, R-HEED, PEGASIS, and LCA. Table 6.4 further proved that ECA scored a better value (10%) for network connectivity of the selfish node compared to LEACH at 12%, EEC 21%, R-HEED 16%, PEGASIS 22%, and LCA 19%.

The present of each round of each behaviour node in experimental scenarios represents of the ECA is more suitable as it ends with less correlation of the clusters compared between LEACH, EEC, R-HEED,PEGASIS, and LCA algorithms. To find out the selfish nodes on network connectivity of each algorithm, every sensor node of the energy threshold value $(T(a)>1)$ is compared with the correlated degree of cooperative nodes on network connectivity. If the correlated degree is greater than energy threshold value $T(a)>1$, the sensor node will periodically broadcast message to its neighbouring nodes to inform that it will be the CH. Finally, the shows network connectivity of selfish nodes can take part in cluster head so that the sensor node does not affected.

d) Scenario malicious node

As for the fourth scenario, the effect malicious node on network connectivity. In this scenario, the network connectivity for the cooperative node failure due to various reasons, such

as energy exhaustion and misconfiguration (Table 6.5). Besides, Figure 6.4 demonstrated that ECA scored a value of 1.47, relatively better than the values recorded by LEACH (2.87), EEC (3.41), R-HEED (3.89), PEGASIS (1.99), and LCA (2.43) respectively. Although this malicious node is prone to be configured on purpose as a selfish node for the sake of power saving or to become as a malicious node, the proposed ECA possessed better connectivity compared to the other algorithms (Table 6.5).

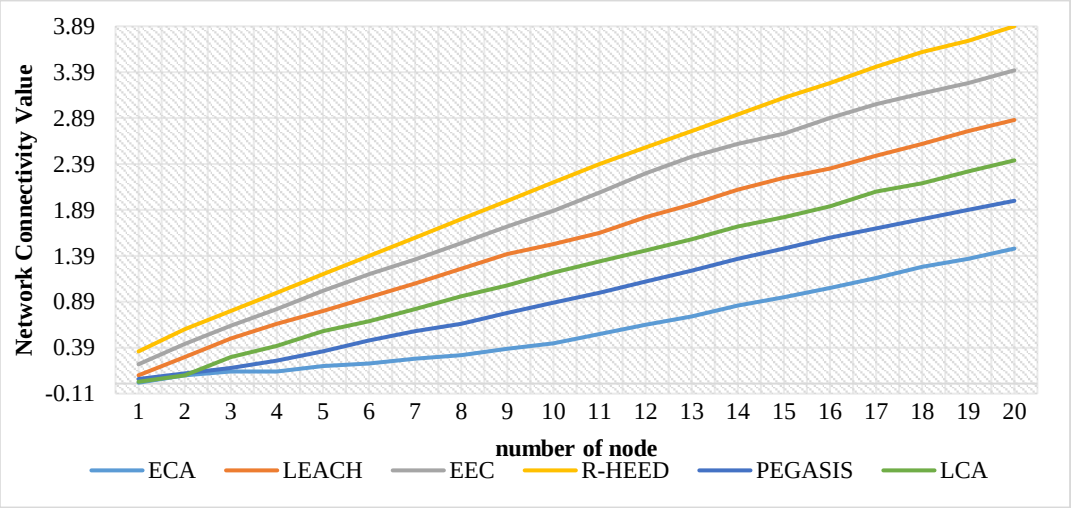

Figure 6.4: The effects of malicious node on network connectivity

A scenario of malicious node uses 20 sample nodes, that consider the scenario where there is one or more malicious nodes value δ ranging $a_i > 1.0$ in the neighbourhood of node. The scenario for malicious nodes considered here is quite general, and allows a node to update its value in a completely arbitrary manner appropriate choices of at each time-step. As such, this scenario encapsulates a wide variety of malicious behaviour including a conspiracy by a set of malicious nodes. However, we do not treat the case where malicious nodes try to influence the result of the computation by modifying their initial values. Jr et al., (2017), Q. Liu et al., (2015), F. Wang et al., (2016). Compared to selfish nodes, malicious nodes are always active in forwarding control packets, however, nodes in a route that could selectively

or randomly drop data packets, reorder packets or increase jitters, are harmful to TCP traffic. Thus, if a node has a neighbour as the next hop, the adversarial nodes are allowed to update their nodes however they wish. That the above dynamics are purely local in the sense that they do not require the regular nodes to know anything about the network topology other than their own in-neighbours, then the node will eventually lose communications with nodes that are at least two-hop away.

If all neighbours are malicious nodes, the specific nodes are isolated by malicious neighbours, similar to the case of having selfish neighbours. As mentioned in section 2.2.1, malicious nodes are active in terms of route discovery and launching attacks. These nodes launch DoS attacks on the cooperative network layer in the routing stage, forwarding data packets or disrupting legitimate path selections through the broadcast of fake route replies. Consider (R. Singh, Singh, & Singh, 2017) malicious nodes that degrade the streaming quality of their neighbours attacks in this research due to their severe impact on correlated degree.

For example, as suppose Ad hoc On-Demand Distance Vector Routing AODV is used as the cluster when a node discovers the route to the next node by broadcasting, a neighbour, say, a node can respond to the node with a fake message immediately claiming that it is in the optimal path or only one-hop away to the node. Consequently, a node selects another node as the next hop and sends data to it, but the node will dump all packets. Hence, without proper counter-measures, a single node may trap all traffic initiated from a node whenever the destination is beyond its one-hop neighbourhood. Moreover, the node may be able to trap all traffic of its neighbours, implying multiple node isolations that could worsen the case. For example, the malicious node for $a_i = 1$ is almost as $a_i > 1.0$, while the network connectivity value 0.05 of node 1 jumps to almost as $a_i > 1$.

A malicious node can become a failure node, but it will not be considered to be cooperative or selfish any more even if its disruptive behaviours are intermittent only. Based

on the analyses, we can be concluded that malicious node attacks make node isolation more complicated and may affect the connectedness of every node.

Table 6.5: The comparison of malicious node on network connectivity

Algorithm	Network Connectivity Value	Comparison
ECA	1.47	9%
LEACH	2.87	18%
EEC	3.41	21%
R-HEED	3.89	24%
PEGASIS	1.99	13%
LCA	2.43	15%

The effects of malicious nodes on network connectivity in a cluster are depicted in Figure 6.4. The ECA scored a great value (1.47) in network connectivity. Since the network connectivity value effect node on the clusters segment increases in ECA (1.47), it can maintain the same cluster. Hence there may not be significant effects on energy efficiency. Besides, the network connectivity was found to be higher in LEACH (2.87), EEC (3.41) and R-HEED (3.89) PEGASIS (1.99), and LCA (2.43), as charge segment increases. Lower network connectivity values explain that the nodes can be randomly turned into sleep mode or activated. Based on Table 6.5, the network connectivity for malicious node using ECA was 9%, LEACH 18%, EEC 21%, R-HEED 24%, PEGASIS 13%, and LCA 15%.

The present of each round of each behaviour node in experimental scenarios represents of the ECA is more suitable as it ends with less correlation of the clusters compared between LEACH, EEC, R-HEED, PEGASIS, and LCA algorithms. To find out the malicious nodes on network connectivity of each algorithm. Every sensor node of the energy threshold value $(T(a)>1)$ is compared with the correlated degree of cooperative nodes on network connectivity. If the correlated degree is greater than energy threshold value $T(a)>1$, the sensor node will periodically broadcast message to its neighbouring nodes to inform that it will be the CH.

Finally, the shows malicious nodes can take part in cluster head so that the sensor node does not affected.

The better value recorded for network connectivity of malicious nodes confirms the theoretical analysis of network connectivity. As a conclusion, the network connectivity value based on the correlated degree of the behaviour nodes in a cluster indicated that the cooperative node increases based on the correlated degree of the cluster. The network connectivity value is higher value as showed in based on comparisons of cooperative node or correlated degrees that observations are tightly coupled with each other. Thus, the failure node, malicious node and selfish node increased its network connectivity. However, such that each node forms its own local neighborhood each node selects some set of important neighbors and maintains state about each of them. In order to obtain an accurate estimation of the effects of the neighbour node behaviour on network connectivity, another additional scenario was also performed in which a cooperative node was connected to the three parameters from the initial position respectively in four separate scenarios.

After the comparison of ECA algorithm with LEACH, EEC, R-HEED, PEGASIS, and LCA Algorithm with the same scenario is done, it was also observed that the proposed ECA algorithm always chose the same neighbouring nodes and formed the same sensor node. However, the ECA acted differently based on 3D Euclidean distance using correlated degree measurement. According to (Liu, J., & Kato, 2016) a general network node was defined which corresponds to a snapshot of the complicated message propagation process, and identify all transient nodes and absorbing states. Then, that define the transitions for each transient node and calculate the corresponding transition nodes between neighbouring nodes. However, the transition time from one state to another is totally based on the random behaviours of a node as it is very difficult for parameters to transition through exponential distributions. For instance, a node is more inclined to fail due to energy consumption as time passes, and the less residual

energy left, the more likely a node changes its behaviour to selfish. This implies that the future action of a node may depend on how long it has been in the current state and transition intervals may have arbitrary distributions.

The most important interpretation of the result is that the proposed Enhancement Clustering Algorithm (ECA) provides a more dependable and stable network. The justification based on selfish nodes will then select themselves as CHs based on the correlated degree threshold, if it falls between $a_i > 0.7 < 1.0$. However, if node value is less than the threshold value, the node will not be selected as a cluster head (CH), cooperative nodes will then select themselves as CHs based on the correlated degree threshold, if it falls between range values 0.0 to 0.69. However, if node value is less than the threshold value, the node will not be selected as a CH. A cooperative node which uses a sample of the cooperative node range value from 0.0 to 0.69 will become cluster hear. The node will not isolated from the cluster and it will not partition. Malicious nodes will not select themselves as CHs based on the correlated degree threshold, because, the malicious node is range values $a_i > 1.0$, the node is declared as malicious and is exempted from the CH selection. A malicious node which uses sample of $a_i > 1.0$ malicious node will not become cluster head. Once a node is selected as CH, it advertises its decision to the nodes in its close vicinity to form a cluster. A Failure node will not become cluster head. The node will be isolated from the cluster and it will partitioned as depicted. As the correlated degree increases, the effect of nodes of the cluster radius did not shrink. For higher correlated degrees, nearby observations are tightly coupled with each other, while for the LEACH, EEC, R-HEED, PEGASIS, and LCA algorithms, the sensor nodes are disconnected from the CH of a wireless sensor network either because the nodes possess faulty connectivity or they have actually left the monitoring region or neighbours (Kim, Jin, & Jin, 2016). Therefore, the results indicated that the proposed ECA formed a more dependable

network with less fluctuation. Compared to the different scenarios that have been deployed, the effects of network performance based on network connectivity as illustrated in Figures 6.1, and 6.3.

On the other hand, Figure 6.4 presents the network connectivity for a node behaviour with parameters that control the range of random correlation based on selfish nodes (0.7 > a_i < 1.0), cooperative node (0 > a_i < 0.69), malicious nodes (a_i > 1.0) and failure nodes (a_i < 0.0). It is worth noting that the performance depends on the parameters' network connectivity and energy consumption. Besides, the correlation between the nodes in a network - a parameter on which further decisions are based on computed intuitively.

Based on four scenarios discussed on the impact of connectivity, it can be concluded that the complexity of computing network connectivity increases significantly due to the boundary segmentation process. The threshold distance varies according to field size from the initial position. Figure 6.1(A) demonstrated that the proposed ECA yielded better connectivity, based on the three parameters (distance, packet and energy) compared to the LEACH, EEC, R-HEED, PEGASIS, and LCA algorithms. The considered network connectivity value ECA value (0.64) rapidly changes, value of the LEACH (0.99), EEC (1.15), R-HEED (1.35), PEGASIS (0.73), ad LCA (0.89). Finally, ECA is more suitable as it ends with less correlation of the clusters compared to LEACH, EEC, R-HEED, PEGASIS, and LCA. Referring to Table 6.2, ECA scored 11%, while LEACH, EEC, R-HEED, PEGASIS, and LCA scored 17%, 20%, 23%, 14%, and 15% respectively, indicating that ECA has better network connectivity of the cooperative nodes.

Based on these data, the effect of nodes in a cluster was calculated using the 3D Euclidean on network connectivity for each clustering pattern on the four different scenarios namely cooperatives nodes, selfish nodes, malicious nodes and failure nodes. Furthermore, as for the correlated degree, the dependency between sample readings exponentially decreased

with the fixed distance. The ECA possesses a higher value of cooperative node, in the network connectivity.

Based on the results presented in Table 6.2 to Table 6.5, the proposed ECA provided better network connectivity, although ECA restricted the correlated degree based on the three parameters which are distance, packet and energy. Close paragraphs in these scenarios, such as a cooperatives node, selfish nodes, malicious nodes, and failure nodes. The showed different value of network connectivity as shown in Figure 6.1 until Figure 6.4 indicated as selfish nodes, cooperative node, malicious nodes, and failure nodes respectively.

6.3.2 Energy Consumption

This study extensively tested the performance of the proposed ECA on the energy consumption of the network cooperative, selfish, malicious and failure nodes scenarios of the behaviour nodes. The proposed ECA outperformed the other algorithms with the behaviour of sensor nodes of the cluster. According to previous literature by Bhavana and Murthy., (2014); Baghouri, Hajraoui, and Chakkor., (2015); Rao and Banka., (2017); Guo et al., (2015); Zhao et al., (2019), the justification is that the behaviours of node clusters in the proposed method considers the energy of the sensor nodes. CH actually consumes more energy than the cooperative sensor nodes. The energy of selfish nodes, malicious nodes and failure nodes depleted more quickly than the cooperative sensor nodes. If a sensor node is selected with low energy, it can die quickly and hamper the network connectivity

The comparison based on cooperative, selfish, malicious and failure nodes on energy consumption using four different scenarios will also be discussed. In order to obtain an accurate estimation of the effects of neighbour nodes behaviour on the energy consumption of network, another scenario was performed in which a cooperative node was connected to three parameters from the initial position in four separate scenarios.

The primary goals of any wireless sensor networks are to route the data assembled by sensors and forward it towards the sink. The simplest method to communicate data is a direct transmission, where the nodes have to direct their data to the sink node. However, if the distance between the sink and network is considerable, the node will die out quickly due to unnecessary energy consumption.

Enhancement clustering algorithm reduces the unwanted energy consumption in delivering data to sink by grouping the network into clusters. Each cluster is assigned a Cluster Head that sends data to t sink. An important stage in the enhancement the clustering algorithm is the Cluster Head election process that should guarantee uniform energy distribution among the sensor nodes.

a) Scenario cooperative node

The energy consumption for CH to transmit data to a sink node is calculated and illustrated in Figure 6.5. Based on the figure, parameters that are required for the transmission of data from one node to another are given as $\ell p = 5aJ/bit$, ℓps = 50aJ/bit, ℓpm = 10aJ/bit/m2, when the sensor nodes are less in number compared to the number of the node the clusters have generated. And as the effect nodes show ECA reduce the energy consumption compared to LEACH, EEC, R-HEED, PEGASIS and LCA Algorithm. The proposed enhancement clustering algorithm (ECA) value (0.95) segment changes in Figure 6.5. Then the LEACH (1.59), EEC (1.41), R-HEED value (1.25), PEGASIS value (1.53) and LCA value (1.81) Algorithm gradually increases as the energy consumption decreases. Table 6.6 presents that ECA score of 11%, LEACH 19%, EEC 16%, R-HEED 15%, PEGASIS 18%, and LCA 21% where the value of ECA on energy consumption was reduced for the cooperative node. However, LEACH, EEC, R-HEED, PEGASIS and LCA algorithm there is still less network connectivity between the cooperative nodes on the clusters facilitating lower network

connectivity as in correlation-based clustering more nodes can be turned into sleep mode. Therefore, based on the energy consumption network reduced with correlated nodes on ECA, ECA is found to be suitable as it has better correlated cooperative node of the clusters compared to LEACH, EEC, R-HEED PEGASIS and LCA.

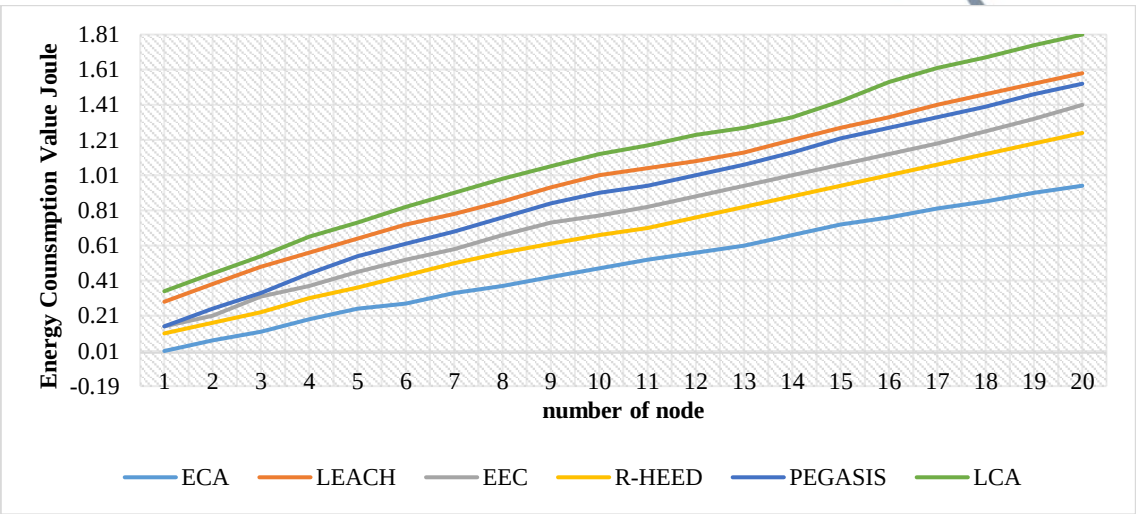

Figure 6.5: The effects cooperative node on energy consumption

Table 6.6: The comparison of cooperative node on energy consumption

Algorithm	Energy Consumption Value	Comparison
ECA	0.95	11%
LEACH	1.59	19%
EEC	1.41	16%
R-HEED	1.25	15%
PEGASIS	1.53	18%
LCA	1.81	21%

b) Scenario failure node

The energy consumption for CH to transmit data to sink node is calculated and results are in the Figure 6.6 shown below is the comparison energy consumption of number node of the clusters. This shows that parameters which are assumed for transmission of data from one node to another are given as ℓp = *5aJ/bit*, ℓps = 50aJ/bit, ℓpm = 10aJ/bit/m2 when the sensor

nodes are less in number the number node of clusters generated. And as the sensor nodes reduce the energy consumption of the proposed enhancement clustering algorithm (ECA) value (1.07) segment changes in Figure 6.6. Then the LEACH (2.39), EEC (1.67), R-HEED value (1.97), PEGASIS value (1.46), and LCA value (2.17) gradually increases as the energy consumption decreases. From Table 6.7 shows ECA value is 10%, LEACH value of 22%, EEC value 16%, R-HEED value 18%, PEGASIS value 14% and LCA value 20% which shows a ECA value is reduce on energy consumption network of failures of node. Less correlation network on correlated node of the clusters facilitates lower connectivity as based on correlation of the clustering more nodes can be turned into sleep mode. For the reduce energy consumption network with Lowe correlated of nodes, ECA is suitable as it ends with less correlation of the failure node on the clusters when compared to LEACH, EEC, R-HEED, PEGASIS, and LCA. This confirms the energy consumption found derived from our theoretical analysis.

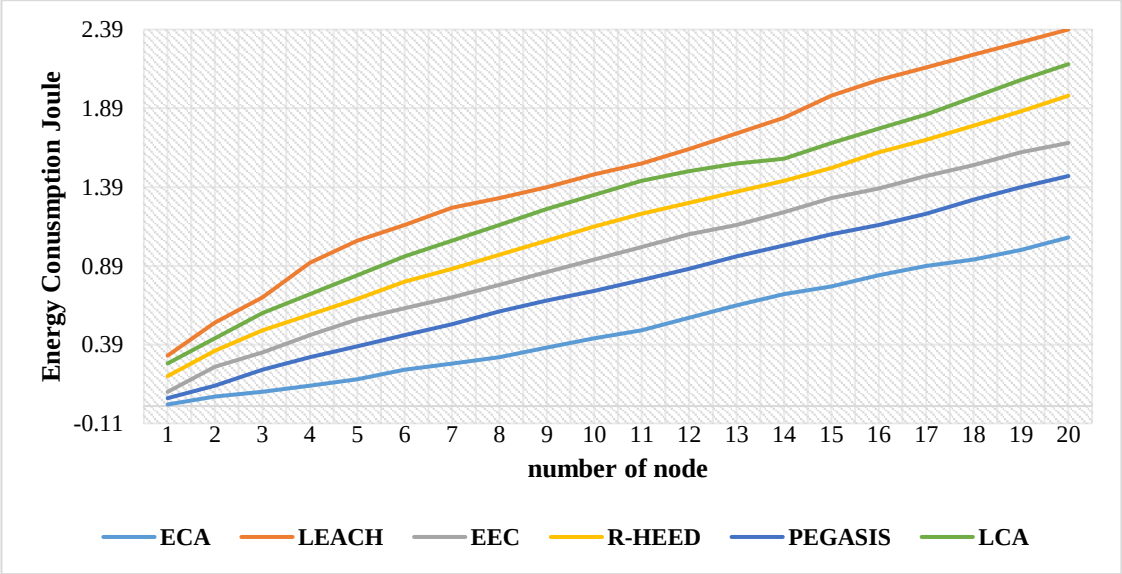

Figure 6.6: The effects of failure nodes on energy consumption

Table 6.7: The comparison of failure nodes on energy consumption

Algorithm	Energy Consumption Value	Comparison
ECA	1.07	10%
LEACH	2.39	22%
EEC	1.67	16%
R-HEED	1.97	18%
PEGASIS	1.46	14%
LCA	2.17	20%

The present of each round of each behaviour node in experimental scenarios represents of the ECA is more suitable as it ends with less correlation of the clusters compared between LEACH, EEC, R-HEED, PEGASIS, and LCA algorithms. To find out the failure nodes on energy consumption of each algorithm in Figure 6.6 shows ECA, LEACH, EEC, R-HEED, PEGASIS, and LCA which shows a ECA value is reduce on energy consumption network of failures of node.

c) Scenario selfish node

The node's selfish behaviour is monitored depending on the send and receive data from the cluster heads' nodes. The selfish node is calculated depending upon the request distance data send within the communication in cluster head network clustering. For a selfish node in the cluster heads, the node is calculated depending upon the difference between the number of a request distance data received to a particular node and the number of distance reply data to the number of request distance data received. The energy consumption for CH to transmit data to sink node is calculated and results are in the Figure 6.7 shown below is the comparison energy consumption of number node of the clusters. And as the sensor nodes reduce the energy consumption of the proposed enhancement clustering algorithm (ECA) value (1.21) segment changes in Figure 6.7. Then the LEACH (1.48), EEC (1.58), R-HEED value (1.77), PEGASIS

value (2.01), and LCA (2.32) Algorithm gradually increases as the energy consumption decreases.

This shows that parameters which are assumed for transmission of data from one node to another are given as ℓp = *5aJ/bit*, ℓps = 50aJ/bit, ℓpm = 10aJ/bit/m2 when the sensor nodes are less in number the number node of clusters generated. The sensor node is calculated based on the behaviour of the nodes. If the value of the sensor node is found to be greater than 0.7, it is a normal node which is able to send and receive data. If the value of the sensor node is found to be lesser than 0.7, then the node is a selfish node. If the value of the sensor node is found to between range 0 to 0.69, it is a cooperative node which is able to send and receive information. If the value of the sensor node is found to be greater than 0.7, then the node is a selfish node. Depending upon the nodes in the network within its communication range the nodes are combined to form clusters in the network. The cluster head is selected for every cluster group with the node having high energy and highest sensor node. If the node is found to be the cluster it sends the data to the cluster head and only through cluster head, the data is sent to the sink node. As showed in Table 6.8 shows ECA value is 12%, LEACH value of 14%, EEC value 15%, R-HEED value 17% , PEGASIS value 20%, and LCA value 22% which shows a ECA value is reduce on energy consumption network of selfish node. Less energy consumption network for selfish node of the clusters facilitates lower connectivity as based on correlation node of the clustering more nodes can turn into sleep or activate mode. For the network with correlation of nodes, ECA is suitable as it ends with less correlation node of clusters when compared to LEACH, EEC, R-HEED, PEGASIS, and LCA. This confirms the energy consumption found derived from our theoretical analysis.

The cluster is formed based on the location of the node. The nodes that are communicated within transmission range are grouped. Then the cluster head is selected based on energy level and sensor node. The energy level is compared with each sensor node and the

node with the higher energy as well as a sensor node with 12% is selected cluster heads. Finally, data is transmitted from source to sink successfully.

It also reduces the overall energy consumption, which improves the network connectivity. The energy needed for entire operations for one round using the ECA algorithm is lesser than that of LEACH, EEC, R-HEED, PEGASIS, and LCA a network clustering. This method is used effectively to increase packet delivery ratio. It reduces packet loss in network connectivity and energy consumption in the clustering algorithm. Then CH is selected based on energy and sensor node. The energy is higher on LEACH (1.48), EEC (1.58), R-HEED value (1.77), PEGASIS value (2.01), and LCA (2.32) algorithm gradually increases as the energy consumption and compared to ECA decreases as the selfish nodes on energy consumption. Then ECA decreases as the selfish nodes on energy consumption, and ECA algorithm with sensor node 12 % are selected as cluster heads. Finally, data is transmitted from source to sink successfully. This proposed method has increased packet delivery rate, energy and distance in network connectivity and energy consumption in the clustering algorithm.

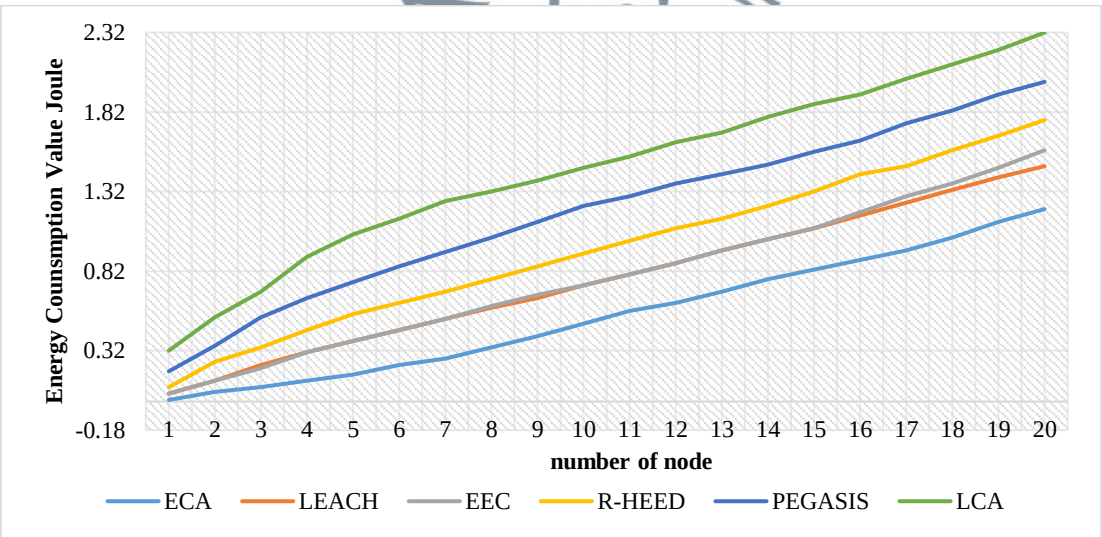

Figure 6.7: The effects of selfish nodes on energy consumption

Table 6.8: The comparison of selfish nodes on energy consumption

Algorithm	Energy Consumption Value	Comparison
ECA	1.21	12%
LEACH	1.48	14%
EEC	1.58	15%
R-HEED	1.77	17%
PEGASIS	2.01	20%
LCA	2.32	22%

ECA is more suitable to reduce energy in data transmission set of the node that can move from one node to another node within a network clustering. In Figure 6.7 shows ECA, compared between LEACH, EEC, R-HEED, PEGASIS, and LCA, which shows ECA value is reduced in a selfish node on energy consumption network. Selfish node in enhancement clustering network is an approach implemented in which uses a set of the node that can move from one node to another node within a network. That reduced network energy consumption on selfish node by moving the computation for data analysis to the location of the intrusion.

In fact, sensor nodes might reduce overload communications. With this purpose, it was tried to deliver an algorithm which prevents the direct connection between all cluster heads and sink; instead, cluster heads should use other cluster heads near to the sink in order to transfer data to the sink; they were selected based on the factors such as cluster head distance with sink and distance among cluster heads. Also, in this ECA algorithm, sensor nodes are selected. The selected sensor nodes length is almost equal and finally results in work load balance among cluster heads and decrease energy consumptions and increase network connectivity. In addition, time delay for data transfer to sink becomes level off in all nodes. A conclusion ECA is more suitable to reduce energy when data transmission in the network clustering, a selfish node on energy consumption of each algorithm in Figure 6.7 shows ECA, compared to LEACH, EEC, R-HEED, PEGASIS, and LCA, which shows ECA value is reduced in a selfish node on energy consumption network.

d) Scenario malicious node

The energy consumption for CH to transmit data to sink node is calculated and results are in the Figure 6.8 shown below is the comparison energy consumption of number node of the clusters. This shows that parameters which are assumed for transmission of data from one node to another are given as ℓ_p = *5aJ/bit*, ℓ_{ps} = 50aJ/bit, ℓ_{pm} = 10aJ/bit/m2 when the sensor nodes are less in number the number node of clusters generated. The threshold distance is varied according to network connectivity. The sensor nodes reduce the energy consumption of the proposed enhancement clustering algorithm (ECA) value (1.13) segment changes in Figure 6.8. Then the LEACH (2.17), EEC (1.95), R-HEED value (1.25), PEGASIS value (1.48), and LCA value (1.71) Algorithm gradually increases as the energy consumption decreases. Table 6.9 shows that the ECA value is 12%, LEACH value of 22%, EEC value 20%, R-HEED value 13%, PEGASIS value 15%, and LCA value 18% which shows in. And in ECA value is reduced on energy consumption of malicious node. To achieve connect the network clustering lower energy consumption as based on correlation node of the clustering more nodes may turn into sleep or activity mode. For the network with less correlation of nodes, ECA is suitable as it ends with less correlation node of the clusters when compared to LEACH, EEC, R-HEED, PEGASIS, and LCA. This confirms the energy consumption found derived from our theoretical analysis.

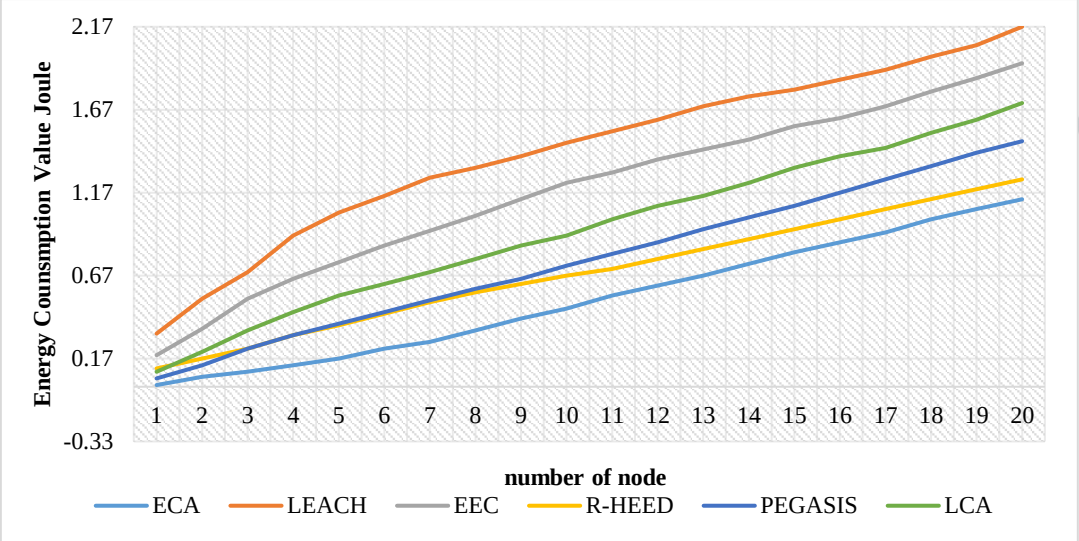

Figure 6.8: The effects of malicious nodes on energy consumption

Table 6.9: The comparison of malicious nodes on energy consumption

Algorithm	Energy Consumption Value	Comparison
ECA	1.13	12%
LEACH	2.17	22%
EEC	1.95	20%
R-HEED	1.25	13%
PEGASIS	1.48	15%
LCA	1.71	18%

CH that each node energy consumption and cluster-heads are unevenly distributed, it was advocated to increase the probability of the energy node that becomes cluster-heads. It then set the minimum distance between cluster-heads to reduce networks energy consumption. The cluster around consumption determined it, and the irrational consumption cluster would be booted. That can choose the optimum consumption values, ensure the networks with minimal energy consumption, reduce energy consumption, and prolong network connectivity.

In the malicious node on energy consumption network in ECA is more suitable than the LEACH, EEC, R-HEED, PEGASIS, and LCA algorithms. ECA algorithm with low power consumption, which uses methods packet delivery rate, energy and distance for clustering algorithm to organize nodes, and the cluster region is divided and fixed by the sink, according

to the dynamic selection of the node energy value, the cluster-heads has achieved the purpose of prolonging the network connectivity. Figure.6.8 shows that ECA is more suitable for reduced energy than the other algorithms in a malicious node on energy, and distance parameters are introduced to modify the cluster-heads improved network performance, which can reduce energy consumption and prolong the network connectivity.

Moreover, to reduce the energy consumption, the data is transformed between nodes on the clusters with multi-hop routing approach. A conclusion ECA is more suitable to reduce energy when data transmission in the network clustering, a malicious node on energy consumption of each algorithm in Figure 6.8 shows ECA, compared between LEACH, EEC, R-HEED, PEGASIS, and LCA, which shows ECA value is reduced in a malicious node on energy consumption network.

The first scenario of the cooperative node in Table 6.6 demonstrated that ECA scored a value of 11%, LEACH 19%, EEC 16%, R-HEED 15%, PEGASIS 18% and LCA, 21% whereby the value of energy consumption was reduced in cooperative node using ECA. Table 6.7 representing the second scenario, indicated that ECA failure node value was 10%, LEACH 22%, EEC 16%, R-HEED 18%, PEGASIS 14% and LCA 20%. Yet again, the ECA value for energy consumption was the lowest for failure nodes. As for the third scenario of selfish nodes in Table 6.8, ECA scored the lowest value of 12% compared to the other algorithms indicating a low energy consumption by the selfish node. A similar trend was also observed in the fourth scenario of a malicious node (Table 6.9), where ECA scored the lowest value of 12% compared to other algorithms.

For instance, the proposed algorithm was assessed to compare the energy consumption of the network with 20 sensor nodes and number of CHs. Based on the tables and figures discussed, obviously that the proposed ECA algorithm shows value (0.95) segment changes in Table 6.6, than the LEACH (1.59), EEC (1.41), R-HEED value (1.25), PEGASIS value (1.53),

and LCA value(1.81) Algorithm gradually increases as the energy consumption decreases compare to LEACH, EEC, R-HEED, PEGASIS and LCA algorithms in energy consumption. This is due to the fact that the derived 3D Euclidean algorithm took care of the energy consumption of the cooperative sensor nodes by reducing the distance between the sensor nodes and their clusters. Whereby, the nodes will select themselves as CHs based on the correlated degree threshold that falls between based on the selfish nodes ($a_i \geq 0.7$), cooperative node (0 to 6.9), malicious nodes ($a_i > 1$) and failure nodes ($a_i < 0$) the node value is less than the threshold value, the node is not selected as a CH. However, once a node is selected as a CH, it advertises its decision to the nodes in its close vicinity to form clusters.

In the ECA, LEACH, EEC, R-HEED, PEGASIS and LCA clustering algorithms, the CHs are selected randomly which may cause imbalance of the clusters. Also, there is no correlation of CHs causing severe energy inefficiency of the network. For example, when the essential parameters like distance and energy were not considered, energy consumption may be caused.

The present of each round of each behaviour node in experimental scenarios represents of the ECA is more suitable as it ends with less correlation of the clusters compared between LEACH, EEC, R-HEED, PEGASIS, and LCA algorithms. To find out the cooperative nodes on energy consumption of each algorithm in Figure 6.5 presents that LEACH (1.59), EEC (1.41), R-HEED value (1.25), PEGASIS value (1.53), LCA value (1.81), and ECA value (0.95), on energy consumption was reduced for the cooperative node.

Table 6.6 to Table 6.9 represent the energy consumption comparison of the effects of correlated node behaviours such as cooperative, selfish, malicious and failure nodes, respectively. The energy consumption of LEACH, EEC, R-HEED, PEGASIS and LCA algorithms were higher than that of the proposed ECA algorithm due to excessive involvement of the correlated nodes in the forwarding of a data packet and redundant packets' transmissions

as mentioned earlier. Based on the figures, the energy consumption of the parameters increased with the increase in network connectivity. The increase is due to more correlation nodes becoming eligible to forward the data packet with the increase in energy consumption network. However, LEACH, EEC, R-HEED, PEGASIS and LCA there have more correlation nodes the basis of the parameters of the sensor nodes there were higher than that of the proposed Enhancement Clustering Algorithm ECA. The utilisation of the parameter of the correlation nodes can reduce the effect node of the clusters since sensor nodes have similar values. In contrast, ECA on energy consumption value is reduced with correlated nodes based on the three parameters, distance, packet and energy. Furthermore, in ECA, due to the 3D Euclidean distance, nodes have enough difference in their holding times. Therefore, the nodes are able to hold a packet to suppress their transmission upon overhearing the transmission of the same packet form a high priority sensor node.

Data correlation is an important factor in reducing network energy consumption, as their analysis is fundamental. The calculation of the CH is based on the 3D Euclidean Distance method. The calculation 3D Euclidean Distance the correlated degree threshold, if it falls between Based on the selfish nodes (a_i > 0.7), cooperative node (0.0 to 6.9), malicious nodes (a_i > 1.0) and failure nodes (a_i < 0.0) respectively. The findings in this study indicated that ECA had its energy consumption reduced the effect with respect to the cluster distance to the sink and of the correlated degree. Therefore, there is no identical and globally energy consumption for the whole clusters compared to LEACH, EEC, R-HEED, PEGASIS and LCA that minimize the entire network energy consumption.

As a conclusion, the cooperative node is suitable to reduce the energy consumption for the correlation degree of the cluster. The energy consumption is reduced as both cooperative nodes and correlated degrees are linked with each other. The observations are globally identical. Therefore, it would suffice if each sensor node reports its data to its immediate

neighbour. Since both observations are identical, the neighbouring node need only forward one instance of the observation. Thus, the failure node, malicious node and selfish node were decrease the effects of nodes on energy consumption. However, such that each node forms its own local neighbourhood.

6.4 Summary

This chapter presents the results and discussion achieved from experimental analyses of the ECA with 3D Euclidean Distance using Correlated Degree parameters. The parameters employed were the energy of nodes, packet, the number of neighbours a node acts as their relay, the distance/ radius and the distance between nodes. All previous literature did not assess all parameters together in one particular work as they only focused on one parameter at a time. More importantly, this study was significant in exposing the interplays between dominant parameters which affected the overall energy consumption, opening doors for new energy optimisation methods.

The correlated degree is also able to detect the same group of node behaviour in correlated regions, as the ECA searches for nodes with the least energy consumption in their clusters. Compared to the LEACH, EEC, R-HEED, PEGASIS, and LCA algorithms used in most networking communication algorithms and extensive experimental, ECA demonstrated a superior improvement in terms of the number of successfully delivered packets, number of packets lost in the cluster and network, and energy consumption of the network connectivity. In this chapter, ECA evaluation was applied to a typical topology of WSN. Results based on the different behaviour node scenarios indicated that the ECA built by 3D Euclidean Distance using correlated degree provided better network performance. Moreover, findings on the behaviour nodes achieved by ECA can provide better environmental information compared to LEACH, EEC, R-HEED, PEGASIS, and LCA algorithms.