ISSN: 2229-6948(ONLINE)
DOI: 10.21917/ijct.2019.0319

SWARM OPTIMIZATION BASED IMPOSTER NODES AND RESOURCE LIMITATION AWARE NODE FAILURE DETECTION

K.B. Manikandan
Department of Computer Science, Providence College for Women, India

Abstract

This research work studies about node failures, which can be prevented pretty good by providing the necessary resources rather than by establishing a route path again. This is done by clustering the mobile nodes in accordance with the on node significance level like it is done in the earlier work and the resources among the cluster members are shared with one another to guarantee that sufficient resources are made available. The cluster is established using the Fuzzy K-means clustering technique. The cluster head is accountable for selecting those clusters members, which can share their resources with one another whereas in this technical work, the cluster head selection is carried out with the help of Cuckoo Search based Hill climbing algorithm (CS-HC). Also, to prevent the wrong information on the node failure being spread by the imposter nodes acting as credible neighbour nodes, this work presents the Imposter Node Detection algorithm. This technique guarantees the optimum detection and suppression of node failures occurring in the mobile wireless networks by presenting effective methodologies. The overall process of realization of the proposed research approach is carried out in the NS2 simulation environment, which shows that the proposed technique ensures improved performance compared to the available research approaches.

Keywords.

Wireless Sensor Network, Clustering, Node Formation, Fuzzy K-Means Clustering, Cuckoo Search Algorithm, Hill Climbing Approach

1. INTRODUCTION

The recent progress made in sensing, computing and communication technologies combined with the necessity for continuous monitoring of physical phenomena have resulted in the Wireless Sensor Networks (WSNs)' development. WSN comprises of four important elements, which include: A radio, a processor, sensors and battery. A WSN is established using sensor nodes in a densely installed application region. In most of the implementations, the sensor nodes are capable of self-organizing, to create a suitable structure so that they can carry out a specific task collaboratively. WSNs falls under the class of wireless ad hoc networks in which the sensor nodes gather, process, and communicate data obtained from the physical environment to be sent to an external Base Station (BS).The important limitations observed in developing WSNs include constrained battery power, cost, memory restraints, lesser computational capability, and the physical size of the sensor nodes. Owing to the critical energy limitations of massive number of densely implemented sensor nodes, a set of network protocols are required for the implementation of different network control and management functions including synchronization, node localization, and network security. Wireless Sensor Networks are proven to be desirable for applications like surveillance, precision agriculture, smart homes, automation, vehicular traffic management, habitat monitoring, and catastrophe detection. Routing in wireless sensor networks and conventional routing in fixed networks are different in different aspects: No infrastructure exists, wireless links are prone to failures, sensor nodes may be unreliable, and routing protocols need to satisfy stringent energy saving constraints.

In the last few years, WSN has gained the significant focus of researchers. Every sensor network comprises of several nodes, where every node comprises of processing the resources and less amount of energy. Sensor nodes can be used for sensing, measuring and collecting the information using some local decision and then transmitted to the main station [1]. A smart sensor node comprises of one or multiple sensors, one processor, and a memory unit, a kind of energy supply using less energy, transmitter, receiver, and a stimulator. Battery forms the prime source for a sensor node and is generally non-rechargeable, therefore the node will be dead, and once the energy of the batteries is depleted.

Two criteria decide the network lifespan: 1) the time in which the First Node Dies (FND) and 2) the time in which the Last Node Dies (LND). Using the various techniques to improve the lifespan of the network is regarded to be one of the most problematic topics in this fields. Several research works target at increasing the network lifespan by introducing duty cycling mechanisms [2], coverage targeted protocols [3] or new routing protocols [4]. Node clustering is one of the effective solutions to improve the network's life span [5].

Clustering holds specific significance for sensor networks which have an ample number of wireless sensors. Every network can be partitioned into smaller clusters and have a Cluster Head (CH). Sensor nodes in each cluster transmit their information to the respective CH and CH gathers information and transmits them to a primary station. In this manner, cluster nodes can save more energy by decreasing the communication range and extent. Swarm and evolutionary algorithms can be utilized in the primary station to decide the members of every cluster and also the cluster heads.

Therefore, in this research work, a new algorithm for clustering of nodes in wireless sensor networks has been suggested, which depends on Cuckoo search combined with hill climbing mechanism. The novel algorithm and various others have been utilized for the clustering process. The simulations results revealed that the network's lifespan has been improved by employing the proposed technique, in comparison with the rest.

2. RELATED WORKS

In [6], the author designed a context-awareness routing algorithm Dynamic Direction Vector (DDV) hop algorithm in MANET is proposed. The current algorithm in MANET is affected by the drawbacks of the dynamic network topology and the deficit of network's expansion capability on the node mobility. The novel algorithm carries out cluster formation for the base station employing the range of direction and threshold of

velocity. In addition, the exchange of the cluster head node probability is calculated employing the direction and velocity for maintaining cluster establishment. The DDV-hop algorithm, a probabilistic routing protocol for these networks, is studied and later compared with the previous algorithms by performing simulation experiments. The simulations are carried out on multiple clusters, network regions, transmission ranges, and the velocity of nodes in mobile networks. The obtained results show that the DDV-hop algorithm exhibits effective eventual delivery and maintains the correct number of clusters and cluster members irrespective of topology variations with a negligible communication overhead in different environments.

In [7], the authors explained about the cluster analysis, which has been identified to be a key research subject in different areas which involve improvement in energy. The primary objective of this research work is to increase the network lifetime. An Energy Efficient Clustering Protocol that employs Self Organizing Map (EECPSOM) in MANETs is developed. This application can help clustering the sensor nodes depending on added parameters like energy levels and weight of sensor nodes. Self-Organized Mapping (SOM) aids in establishing clusters such that nodes having maximum energy attract the closest nodes whose energy levels are low. The newly introduced technique facilitates in forming energy balanced clusters and helps in equal distribution of the energy usage. The results of simulation show that the novel algorithm reduces the energy usage and improves the energy efficiency such that the lifespan of the network is extended.

In [8], the authors explained about MANET, is a multi-hop wireless network in which the mobile nodes exhibit dynamic behavior and offers very less bandwidth and reduced battery power. Owing to these environmental challenges, the mobile nodes can be divided into clusters to attain superior stability and scalability. Dividing the mobile nodes is known as clustering, in which a leader node is selected for the management of the whole network. In addition, the different techniques for clustering are focused on various performance metrics. Every cluster has a specific node known cluster head which, in turn, is elected as cluster head adhering to a particular metric or fusion of metrics known mobility, energy, degree, weight etc. Few clustering approaches like mobility-based clustering, energy-efficient clustering, connectivity-based clustering, and weighted-based clustering has been analyzed with their benefits and drawbacks.

In [9], the authors studied about MWSNs, which are wireless networks made up of small sensors traversing around in a particular area of coverage, transmitting their readings and data to stationary or mobile base stations. As opposed to a stationary wireless sensor network where the static node position can be utilized for authenticating a sensor node, a node's varying position in a MWSN cannot be considered for the process of authentication. This can result in intruders acting as genuine nodes anywhere in the network. An imposter can make use of the identity of a genuine node to have communication within the network and perform eavesdropping on sensitive communication or to transmit fake information. A new approach which makes use of nonce-values and blackout-time schemes for the identification of imposters in the network is also studied. The novel approach uses a quarantining technique for isolating the identified imposters and node-restoration technique to suitably authenticate and bring back the quarantined nodes into the network.

In [10], the authors designed completely distributed and entirely decentralized approaches for the detection and elimination of several imposters in MWSNs. In the case of a node replication attack, an intruder generates replicated versions of seized sensor nodes trying to have control over the information, which is being transmitted to the base station or, probably, to disrupt the operation of the network. The newly introduced approaches not just help in isolating these adversarial nodes but also offer resistance against collusion from collaborating intruders attempting to compromise the genuine nodes of the network. Therefore, the viability and efficiency of the protocols are ensured. The protocols are integrated with elaborate mathematical and experimental results, showing the practicability of the works, thus rendering them desirable for practical deployments mobile sensor network.

In [11], the authors presented new behavioral biometrics approach, known as fingerprint dynamics. The approach takes advantage of the behavioral features of the individual, found from multi-instance finger scan processes. The goal of this technical work was to evaluate the spoof resistance potential of the fingerprint dynamics-based independent identity verification system. It employs a customized hardware unit to gather the biometric samples from an overall of 50 participants, in an environment, which nearly resemble the operational environment. Data aggregation was carried out through multiple sessions per user, distributed over a time period of seven weeks. This technical work carried out an extensive analysis of different time-based features, and an ensemble set of best-performing features are chosen with the help of genetic algorithm. This technical work also carried out a systematic analysis employing support vector machine and k-nearest neighbour classifiers.

The authors conducted a set of verification experiments under three diverse and practically related attack conditions, such as: 1) combined zero-effort and active imposter; 2) only zero-effort imposter; and 3) only active imposter. They observed that the proposed approach yields potential results under all these three attack conditions.

3. PROPOSED METHODOLOGY

In a general WSN, with stationary sensor nodes, sink or different nodes can decide the credibility of sensor node by binding its character to its assured geographic region [12]; taking the help of observer nodes, location claims arriving from compromised areas in network show if replication attack exists. However, the case of MWSN, nonetheless, constantly developing nodes renders location-based detection to be a huge challenge. This the result can be identifying the genuine node and then make use of it for communicating with whatever the system has. Since sensor nodes are not tamper-resistive [13], the intruder can create replicas of nodes next to compromising the node and copying its cryptographic or other information.

As the credentials of copied are not different from those of genuine ones, it is not easy to differentiate to between the two, and imposter detection becomes a hugely challenging task. This type of attack, which is called as node replication attack in the current works, has important effects on wireless sensor systems security: by anticipating counterfeit character, the imposter can transmit misinformation, perform old packets replay, which could

change the results of aggregation or encourage various kind of attacks in the system, including selective forwarding, sinkhole attacks.

3.1 KEY MANAGEMENT

In WSNs, In order to provide secure communication between the sensor nodes, all the messages have to be ciphered and then authenticated with different secret keys. The collective number of keys made in the sensor node and system is extremely large. Therefore, it is necessary to develop strong and effective Key Management Schemes (KMS) for WSNs. In the case of unregulated conditions, it enables sensor nodes to communicate safely with the others using the crypto techniques. The objective of key management [14] for WSN is to group, disseminate and handle secret keys in sensor nodes to manage secure communication among sensor nodes. Securing the fundamental applications depend upon core management as it has to exhibit adaptability to non-crucial failure if nodes are compromised. Each and every time, new nodes are to be included or nodes have to be removed from the system, key management plans hold an important place.

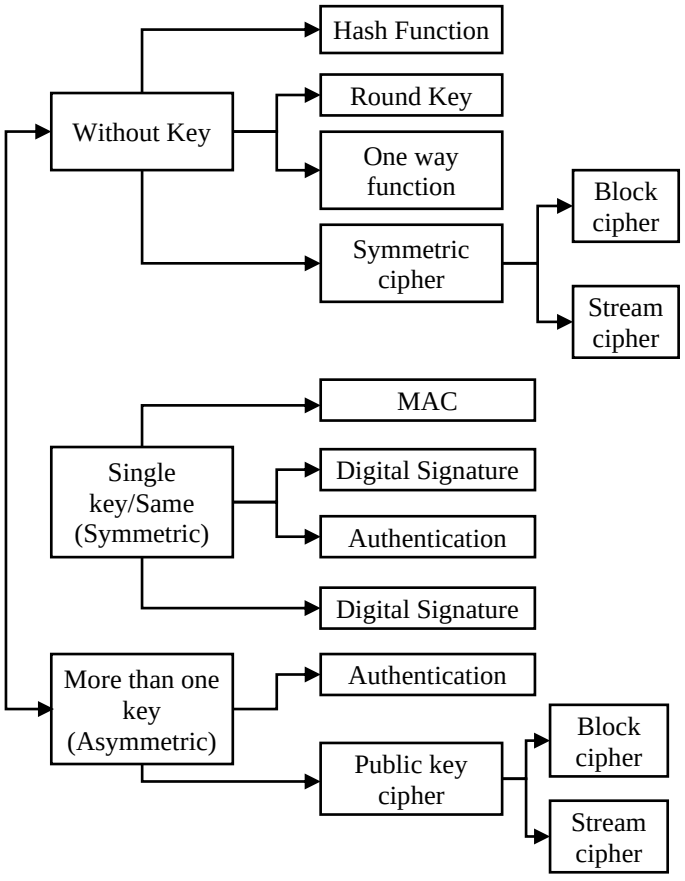

Fig.1. Taxonomy of Key Management

The Fig.1 shows the levels of Key Management. When planning the key management approaches, few important measurements to be evaluated include

- *Local/Global Connectivity*: Nodes communicates with one another in sensor field area.
- *Resilience*: Each and every time, the sensor node is affected by intruders, key management ensures the protection of the remaining communication interface against node seizure.
- *Scalability*: Strength to assist when massive numbers of nodes are added to the sensor network.
- *Efficiency*: Associated with capacity, computation and communication.

The Fig.1 shows the taxonomy of the key management. This Fig.1 gives the kinds of keys and their usage and the various applications where these keys can be used.

Efficient management of cryptographic keys [15] is a huge challenge if a huge amount of cryptographic keys are present. Every time a part is removed from or included in the group, group key has to be modified. The members in the group has to be capable of figuring out another key efficiently, and at the same time, forward and backward security has to be taken care. Forward securing indicates that any eliminated part node cannot determine any future group key, when carrying out multiple tasks. Backward security indicates that the member node included recently cannot have the authority of deciding on any previous key, when dealing with other new individuals.

3.2 IMPOSTER NODE DETECTION

In the case of MWSN, the consistent increase of nodes renders location-based detection a nearly unpredictable job. Hence, an adversary can admit the characters of genuine node and make use of it for the communication with the remaining nodes of the system. Since sensor nodes are not changing secure devices, an intruder can make replicas of nodes by endangering the node and copying its cryptographic or other information. It hints to such replications as fake when they use the characteristics of the available sensor nodes to have communication with sink or various other nodes present in the system. Mobile sensor network in which the nodes send/receive information with one another and, if basic, the detected information can reach the base station employing the right MWSN routing computations.

In order to identify an imposter the following simple component is employed: When two sensor nodes meet, each node generates an uneven nonce, which is stored in its memory, and transmits it to the next node. Again, every time these nodes meet, they verify one another by asking for the characteristics they exchanged in their earlier meeting. In case the node is not able to reply or replies with the wrong number, it is considered to be imposter and ID of nodes is regarded to be compromised.

Next, nonce esteems are used for identifying the existence of imposters and they get altered after every possible communication and the nonceList is maintained. Each node maintains the nonceList specifically and has nonce esteems expected from various nodes and also the values to be transmitted to various nodes for the authentication to be successful. For the representation of this validation process, in which two nodes u and v meet for the first time at time t_1, they their nonce's are shared and then everyone goes its own way. Before, an adversary was capable of compromising the node v and creates an imposter with an ID that is similar. Node u, with no information of this even, meets again with node bearing ID of v at time t_2, as per the procedure, expects to get the nonce it had sent to v during an

earlier experience at time t_1. Imposter *iv* cannot provide this data, and as a result, *u* understands that ID of *v* has been endangered.

3.3 IMPOSTER NODE AWARE NODE FAILURE DETECTION METHOD

Node failure detection is the most cumbersome task in the mobile wireless network scenarios having a higher mobility. A greater rate of node failure would decrease the performance level of network which has to be focused better to guarantee the node failure detection rate. This technical work highlights on these issues and presents the novel approaches that can be efficient in the detection and prevention of the node failures occurring on the networks. In this research approach, node failure can be averted substantially by providing the necessary resources rather than through the re-establishment of another route path. This is guaranteed by clustering the mobile nodes in accordance with the node importance level like it is carried out in the earlier work and the resources of the cluster members are shared with one another to make sure sufficient resources are available. The cluster head is accountable for managing the cluster members and controlling the data sharing activity between cluster members. In this technical work, optimum cluster head selection is carried out by presenting the novel Cuckoo search algorithm with hill climbing (CS-HC).

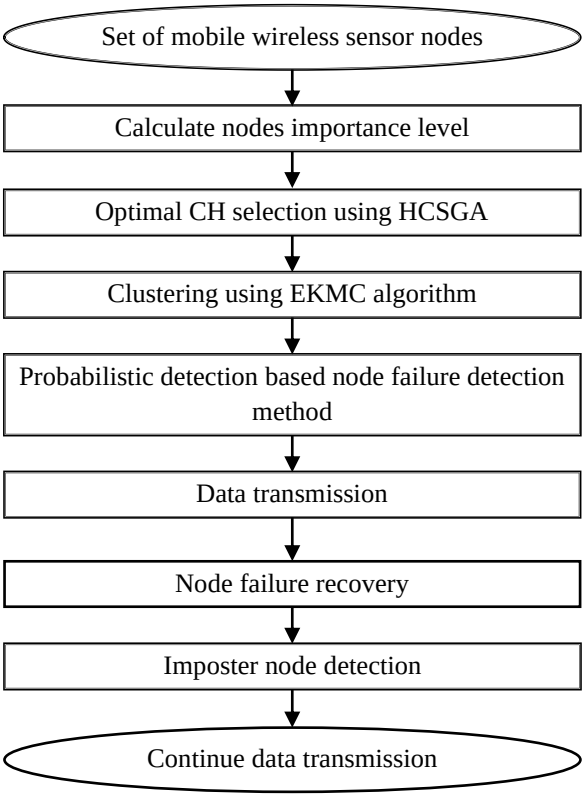

Fig.2. Processing flow of Imposter Node Detection Process

In this technical work, Imposter Node aware Node Failure Detection Method (IN-NFDM) is proposed to prevent the fake information on the node failure being transmitted from the imposter nodes acting as legitimate neighbour nodes. The Fig.2 illustrates the overall flow diagram of the novel research approach.

The Fig.2 shows the processing flow of the proposed research technique. It can be shown from this research that the proposed research mechanism can attain effective detection and prevention of node failures occurring in the mobile wireless network environments. The proposed research scheme is extensively described in the subsections that follow.

3.4 CLUSTERING NODES BASED ON NODE IMPORTANCE LEVEL

In this research work, the Fuzzy K-Means Clustering (FKMC) technique is utilized for effective cluster creation. Due to its simple implementation process and rapid convergence, k-means clustering is a suitable clustering technique, particularly in mobile wireless sensor networks. Its aim is to reduce the average squared Euclidean distance from their cluster centres.

3.4.1 K-Means:

The first step of K-means is to choose the initial cluster centers *K*. The algorithm adopts a simple means of sorting out a particular data group using a fixed number of clusters (presume *k* clusters). The fundamental concept is to decide the centroids, and every centroid is a member of one cluster. K-means algorithm is run for cluster creation with the target WSN. Suppose that the WSN of n nodes is categorized into *k* clusters. At first, *k* out of n nodes is randomly chosen to act as the CHs. Then, each one of the other nodes determines its CH closest to it based on the Euclidean distance. Once all the nodes in the network is allocated to one of *k* clusters, the centroid of every cluster is computed.

$$\text{Distance}(D) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{1}$$

$$\text{Centroid}(X, Y) = \frac{1}{s}\sum_{i=1}^{s} x_i, \frac{1}{s}\sum_{i=1}^{s} y_i \tag{2}$$

3.4.2 Fuzzy K-Means Algorithm:

Fuzzy K-Means is a modified version of K-Means algorithm which uses the degree of belonging as the criterion. It implies that an object can be a member of multiple clusters to some degree, which is not possible in K-Means. Usually, the points or objects present on the edge of cluster might have a lesser degree of belonging whereas the objects present in the center might have a greater belongingness degree. Coefficients are utilized for to providing the degree of belongingness and they are defined as below.

$$\forall X \sum_{k=1}^{num.cluster} u(X) = 1 \tag{3}$$

In fuzzy K-Means, the centroid is calculated by the mean of all the points. The objects are weighed by the degree to which they are a member of a certain cluster.

$$Center_k = \frac{\sum_x u_k(x) mx}{\sum_x u_k(x) m} \tag{4}$$

The inverse of distance to the cluster exhibits inverse association with the degree of belonging. The following formula computes this degree.

$$u_k(x) = \frac{1}{d\left(Center(k,x)\right)} \quad (5)$$

Then, the coefficients are normalized in addition to fuzzification. This fuzzification makes use of real parameter $m>1$. Therefore the sum is computed to be 1. Hence the equation below is defined.

$$u_k(x) = \frac{1}{\sum_j \left(\frac{d\left(Center(k,x)\right)}{d\left(Center(j,x)\right)} \right) \frac{2}{(m-1)}} \quad (6)$$

On the normalization of the coefficients, their sum becomes 1. When m value is 1 or nearer to 1, it implies that the point is nearest to cluster center and more weight is attained by that point. The following algorithm describes this.

1. Selecting the number of clusters
2. Assignment of the coefficients of points in random for being present in the clusters
3. Each time, a change in coefficients is observed between two iterations and the sensitivity threshold is taken into account.
4. This process goes on till the algorithm converges.

Cluster Head Selection employing Proposed Cuckoo Search Algorithm with Hill Climbing (CS-HC) Algorithm

Cluster Head (CH) refers to the process of choosing a node existing in the cluster to become a leader node. The CH maintains a record of the information associated with its cluster. This information comprises of a list made up of the nodes present in the cluster and the path to every node. The CH is responsible to communicate with each node existing within its own cluster and has the potential to communicate with the nodes that are members of other clusters also, with which it can be make direct communication, or via the respective CH or via gateways. Communication is carried out in three phases. In the first phase, all the cluster heads receives the data sent by its members, and in the next stage, it performs the data compression, and finally, it transmits the data to the base station or to the other CH. The novel technique is utilized for choosing the CH that has the highest number of cluster members and then becomes a member of that particular cluster. Compression is utilized for converting the data from an easy-to-use format to one that is optimized for lightweight. Owing to compression, the number of data packets can be minimized to the maximum degree such that the requirement of memory and bandwidth are quite low. In addition, the compressed data is similar to a scrambled message and an intruder in the middle will not be able to comprehend. Hence, data compression not just minimizes the size of the real text, but also data security is achieved through it. The CH is accountable for each of its cluster member, whose retransmissions can cover all the nodes present in that cluster. Transmission range, transmission power and transmission packets are thee factors that the cluster head considers and using the shortest distance mechanism, the data packets are either transmitted to the base station or the cluster head node. A suitable CH node can help reducing the energy usage and increase the network lifespan.

Electing a specific node to act as a cluster head holds huge importance even though this task is challenging. Various factors have to be considered for choosing the best node as a cluster head. Some of these factors include the location of the node with respect to other nodes, mobility, energy, trust, and the throughput of node. The WSN nodes have limited battery and resources. The election process results in an overall increase in the processing overhead in the network. Therefore, the election process must also consider the processing and energy constraints of the nodes. One CH node per cluster must to be selected during the election procedure, as multiple cluster heads existing within one single cluster can lead to re-establishment of cluster, QoS, and routing management issues.

In this technical work, CH node election is carried out employing the novel Cuckoo Search algorithm with Hill Climbing (CS-HC) depending on the least energy usage, distance and higher packet delivery ratio values to the objective functions. It takes the election of the optimum cluster head for the resultant clusters as its objective.

3.5 CUCKOO SEARCH ALGORITHM

CS algorithm is a Meta-heuristic search process which is based on the ideal rules given as below:

- Each cuckoo lays one egg at a time, and throws it in a nest that is randomly chosen.
- The best nests with superior quality of eggs (solutions) will be taken over to the next generations.
- The number of gettable host nests is constant, and with a probability function, the host can decide about a foreign egg.

Here, the host bird can either throw away or abandon nest to build another nest somewhere else. The important phases of cuckoo search algorithm are reduced in cuckoo search algorithm. During the generation of new solutions for i^{th} cuckoo, the next Levy flight procedure is conducted:

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Levy(\lambda) \quad (7)$$

where α refers to the step size, which needs to be correlated with the problem of interest scale. Product implies multiplications done entry-wise. Levy flight is measured in this process in which the step-lengths are distributed depending on the next subsequent probability distribution

$$Levy\, u = t^{-\lambda},\ 1 < \lambda \le 3 \quad (8)$$

This results in a variance that is infinite. In this step, the sequential jumps/steps of cuckoo basically create the random walk process, which helps in deriving the power-law step length distribution with significant tail.

Algorithm 1: Cuckoo Search

Decide the Objective function $f(x)$, $x=(x_1, x_2,\ldots,x_d)$

Initialize population of n host nests $x_i(i=1,2,\ldots,d)$

While ($t <$ *MaxGeneration*) or (stop criterion);

Get cuckoo (say i) stochastically and generate new solution using levy flights

Estimate its quality/ fitness F_i

Choose nest between n (say j) in random

If ($F_i > F_j$)

Replace j by a new solution

End

Discard a fraction (P_a) of the worse nests and build new ones at fresh locations using levy flights

Keep the best solutions or nests with high quality solutions

Sort the solutions and get the current best

End while

Post procedure results and visualization

End

3.6 HILL CLIMBING (HC) ALGORITHM

Hill Climbing (HC) is a mathematical optimization approach that falls under the category of local search. It looks out for a better solution in the neighborhood by assessing the present state. In case, it is a goal state also, then go back to it and stop. Else, go on updating the current state, if it is possible. After this, iterate through a loop until a solution is got or until no new operators remain to be used in the present state. In addition, there are two steps within this loop. In the first step, an operator is selected, which has not yet been used on the present state and it is used for generating the new state. In the second step, the new state is evaluated. The Fig.3 depicts the pseudo-code of HC algorithm, which corroborates how much simple hill climbing is.

In accordance with the above, in HC the fundamental concept is to always continue towards a state that is better compared to the present one. Therefore, it always helps in improving the solution quality.

i = initial solution

While $f(s) \leq f(i)$; $s \in$ Neighbours(i) do

Generate an $s \in$ Neighbours(i)

If fitness(s) > fitness(i) then

Replace s with i

End If

End While

HC offers few benefits, such as it can be adapted quite easily as per the current problem. Nearly any characteristic of the algorithm may be modified and tailored. For instance, it can be utilized in transformations and also discrete domains.

3.7 CUCKOO SEARCH ALGORITHM WITH HILL CLIMBING ALGORITHM

The novel CS-HC algorithms are collaborative fusions of the CS and HC approaches. CSA depends on the obligate brood parasitic behavior observed in few species of cuckoo, combined with the Levy flight, which is a kind of random walk having a power law step length distribution and a strong tail. It draws its inspiration from the behavior of few birds and fruit flies [17]. Levy flight is utilized for global exploration process and its resourcefulness has been proven by the fine results that it achieves [16]. Therefore, the CSA is regarded to be an effective metaheuristic swarm-based algorithm, which provides a good balance between the local neighborhood exploitation and global wide exploration conducted in the search space problem. But, at times, its solution exploitation is quite poor with the convergence rate being slow. Due to this, the novel algorithm increases the search capability of the fundamental CSA by merging it with HC technique for intensifying the exploitation; and the so-known CS-HC algorithm is helpful in optimizing the benchmark functions:

Pseudo Code: CS_GA

Start

Specify Objective function $f(x)$, $x=(x_1, x_2,\ldots,x_d)$

Initialize a population of n host nests $x_i(i=1,2,\ldots,d)$

Specify the cuckoo search parameters P_a

Start CS

Assess its quality/ fitness F_i

Obtain a cuckoo (say i) in random and produce a new solution by levy flights

Select a nest among n (say j) in random

If ($F_i > F_j$)

Substitute j with the new solution

End

Then

Compute the neighbouring nest

Find the maximum of nearby nest

If (bigger than the present one)

Local minimum is get

If not again compute the nest in the neighborhood

If the local minimum is greater than the present local minimum

Discard a fraction (P_a) of the worse nests and construct new ones at another place using levy flights

Maintain the best solutions or nests having quality solutions

Sort the solutions and get the current best depending on the max iteration

End

End

End if

Keep the final best population of nests

End begin CS

Therefore, it can be stated that, CS-HC begins the search by exploiting the standard cuckoo search algorithm for all the iterations. Then the best of the acquired solution is given the HC to speed up the search and get over the slow convergence problem that the basic cuckoo search algorithm faces. HC is, in fact, an iterative algorithm, which begins with a random solution to a problem and then in the next step, tries to decide on a better solution by consecutively varying a single element of the solution. If the change yields a better solution, then an incremental transformation is carried out on the new solution, and this is repeated till there are no further improvements. Thereafter, it retrieves the solution to the CSA for checking it using the fraction probability P_a.

3.8 IMPOSTER NODE DETECTION

In this technical work, the mobile sensor network environment is taken into consideration where the nodes pass information to one another and, if required, the sensed data can get to the base

station by running the suitable MWSN routing algorithms. For the detection of an intruder, the simple strategy given below is employed: "On the first meeting of two sensor nodes, a random nonce is generated by each node, which is stored in its memory, and relays it to the other node. When these nodes meet for the next time, they ask each other for the values that they had shared in their earlier meeting. In case, a node is not able to reply or replies with the incorrect number then it is considered to be an imposter and the ID of the node is taken to be endangered".

Therefore, the nonce values are utilized for detecting the presence of intruders and they are modified every time communication is successful and this is maintained in a nonceList. There is also nonceList maintained separately by every node and it has the nonce values required from other nodes and also the values to be transmitted to other nodes for making the authentication successful. To elaborate this authentication procedure, let Fig.3 be taken in which the two nodes u and v are meeting up for the first time at time t_1, and they shared their nonces and later they go their own random path. Earlier, node *v* was compromised by an intruder and generates an intruder having the same ID (this is shown by node i_v in the Fig.3). Node *u*, which has no knowledge of this occurrence, has another meeting with a node having the ID of *v* at time t_2, and therefore it anticipates to gets the nonce it has given to v during the earlier meeting at time t_1. The imposter i_v is now not able to given this information, and therefore u gets to known that there has been a compromise with the ID of *v*.

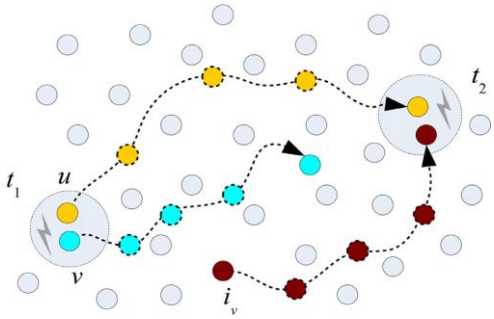

Fig.3. Sample of Imposter Node Detection technique

It must be evident from this definition, that for imposter detection to occur a sensor node has to meet both the genuine node and then its intruder since the node met secondly will not be capable of replying with the right nonce that is shared with first node. When this occurs, it tags the node's ID to be compromised and then adds it to its quarantine List. Now, the information regarding the imposter can be passed to other nodes or BS as provided.

Table.1. Operations of Sensor node entities

Node *u*	**Node *w/BS***
Node *u* detects imposter *v* $u, v\{(u,v)\}_{K_u^{-1}}$	Receive and authenticate claim

A claim is a pair <$detector_{id}$, $imposter_{id}$>, which has the signature of the node that makes the claim. With the help of an identity based signature approach, in case the node u makes a claim that the node with ID_v is being taken by an intruder, the claim is signed with *u*'s private key K_u^{-1} and contains the form $\{\langle u,v\rangle\}K_u^{-1}$. Then the receiver authenticates the claim by generating *u*'s ID-based public key $K_u = F(u)$ and then the signature is verified. When radio communication takes up much of the energy in sensor network protocols, and therefore protocols with needless communication overhead has to be averted, an authentication alternate, which depends just on symmetric cryptography primitives like hash functions is described.

When this technique needs much lesser computational effort for the generation and verification of signatures, there is a substantial growth in the signature size, thus involving increased communication overhead compared to the ID-based solution. In order to use the asymmetry in signing rendered by public key cryptography, *r*-times signatures can be brought into use. An *r*-time signature approach is identical to a public-key approach in that it can be utilized for signing messages, which can be validated with the help of openly available information. These signatures drastically reduce the signing and verification time in comparison with public-key signatures, but, an individual can just sign up to *r* messages using a key pair given. After this, one has to create a new signing key for signing the next messages, else there is a lapse in security. The new public key, which is required for verifying new messages, can be chained to the earlier one, and therefore the reliability of public keys is still guaranteed.

The secret key comprises of *t* random l -bit values organized in a pre-determined number of Merkle trees whose leaves correlate with these secret values when intermediate nodes have the hashes of their children values. Next, the roots of these trees are considered to be the public key of the mechanism. For signing a message m, a subset of the secret values is published in addition to their respective authentication paths in the trees. For the verification of a signature, a node just performs a re-evaluation of the authentication values given with the signature and checks to verify if they are matching with the roots of the trees present in the public key.

Creation and verification of signatures is very resourceful since it needs just hash and comparison operations. Making use of the parameters recommended in this technical work, a 1200 bytes long signature, can be verified through less than 20 hash computations or nearly 200ms in simple sensor nodes. But, when signature generation and verification time is quite less, the time for transmitting this signature is prohibitory. Therefore the usage of ID-based cryptography is considered to be the best alternate, in spite of its increased verification cost. Since the message transmission consumes a substantial amount of energy in sensor networks, the energy can be preserved by transmitting smaller signed messages but incurs increased computation overhead for the task of verification.

4. EXPERIMENTAL RESULTS

This section presents the quantitative performance analysis. In all these simulations, the node movement happens in a 500×500 m^2. The overall number of nodes, *N*, varies between 20 and 150. A 2D Poisson distribution is followed for describing the initial positions of the nodes. The transmission range of a node is a circle with the radius, *r*, varying between 30m and 130m.

The above mentioned parameter combination results in an extensive range of neighbourhood density for assessing this technique. The performance measures taken into consideration in this research work for the its performance evaluation over the available technique, probabilistic NFDM-CT are given in terms of detection rate, false positive rate, communication overhead, energy consumption and end to end delay.

The comparison proposed and existing research methodologies is given in terms of node failure detection rate. The results of simulation is given in Fig.4.

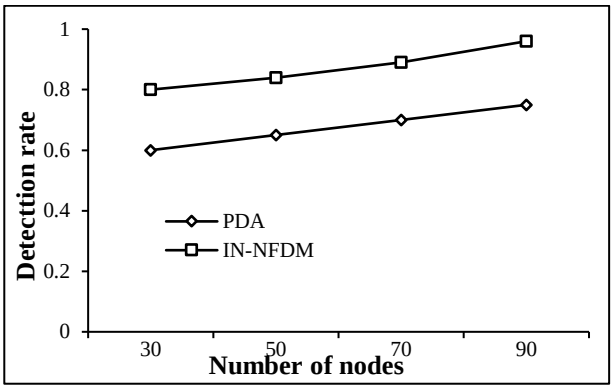

Fig.4. Detection Rate Comparison

The Fig.4 illustrates the comparison analysis of detection rate between the available and the novel algorithm for node failure detection process. The x-axis has the number of nodes and the y-axis is plotted along the detection rate. The number of nodes differs between 30 and 90 for the available and proposed techniques. The PDA technique yields much lesser detection rates while the proposed technique IN-NFDM yields greater detection rates. IN-NFDM yields much better detection rate performance owing to the sensor nodes being covered all the over transmission range. So it can be concluded from the result that the novel IN-NFDM exhibits fine location tracking performance compared to the available algorithms where the detection rate is 36% higher compared to PDA.

The comparison of the newly introduced and the available research approach is carried out in terms of false positive rate whose simulation values are given in Fig.5.

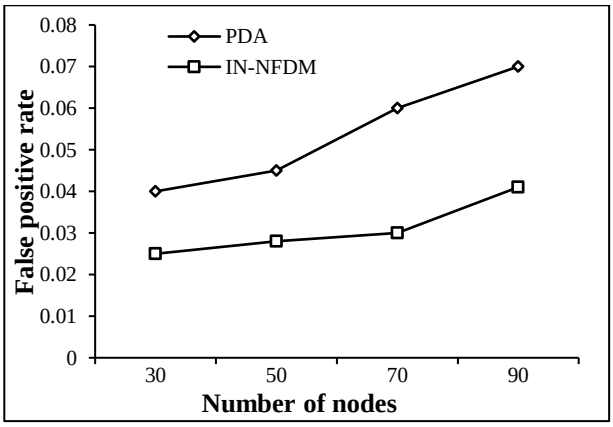

Fig.5. False Positive Rate

The Fig.5 illustrates the comparison analysis of false positive rate between the available and the novel algorithm for node failure detection. The x-axis has the number of nodes and along the y-axis, the false positive rate is plotted. The number of nodes differs between 30 and 90 for the available and proposed techniques. The novel IN-NFDM yields much lesser false positive rate while PDA yields much better false positive rate. IN-NFDM yields much better false positive rate compared to the available techniques owing to the capability of differentiate between a node failure and the node getting out of the transmission range.IN-NFDM achieves a comparatively good CH selection performance. IN-NFDM exhibits 44% lesser rate compared to PDA.

The comparison analysis of the novel and available research techniques are carried out in terms of energy usage and graphical comparison is given in the Fig.6. The Fig.6 illustrates the comparison analysis of energy usage between the available and the newly introduced algorithm for node failure detection. The number of packets are plotted along the x-axis and the energy consumption is plotted along the y-axis.

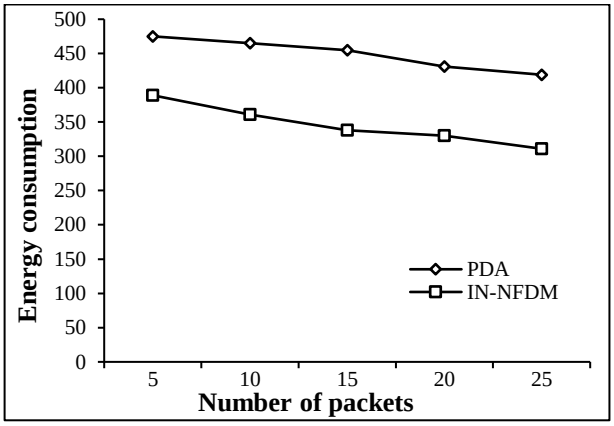

Fig.6. Energy Consumption

PDA needs more energy in comparison with the energy needed in the novel IN-NFDM. It yields relatively lesser energy compared to the available techniques due to the location information being available with source nodes. The less energy consumed in IN-NFDM improves the network efficiency. IN-NFDM takes about 24% lesser energy compared to PDA.

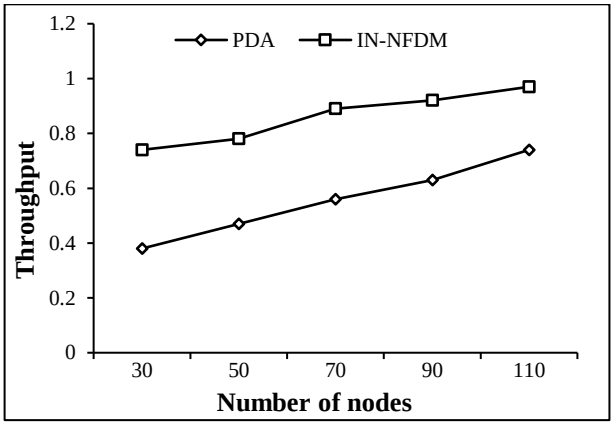

Fig.7. Throughput Consumption

The comparison of the novel and the available research techniques is carried out in terms of throughput. The Fig.7 provides the graphical comparison of the throughput.

The Fig.7 illustrates the comparison of throughput between the available and the novel algorithm. The number of nodes is plotted along the x-axis and throughput is plotted along the y-axis. IN-NFDM yields a greater throughput in comparison with PDA technique. The overall performance of the sensor network sees a gradual increase in the novel system compared to the existing ones as the location information and optimum CH node selection which is close to the destination. IN-NFDM yields 50% improved throughput compared to PDA.

5. CONCLUSIONS

In this research, the node failure is substantially reduced by providing the necessary resources rather than by re-creating another route path. The cluster head is accountable for selecting which of the clusters members want to partake with their resources and in this technical work, the cluster head selection is carried out employing Cuckoo Search with Hill Climbing (CS-HC). In this research approach, to prevent the fake information on the node failure being propagated from the malicious nodes acting as credible neighbour nodes, this technical work introduces the Imposter Node Detection algorithm. The simulation comparison analysis shows that the proposed research approach guarantees the optimum detection and prevention of node failures occurring in the mobile wireless networks.

REFERENCES

[1] J. Yick, B. Mukherjee and D. Ghosal, "Wireless Sensor Network Survey", *Computer Networks*, Vol. 52, No. 12, pp. 2292-2330, 2008.

[2] Z. Rezaei and S. Mobininejad, "Energy Saving in Wireless Sensor Networks", *International Journal of Computer Science and Engineering Survey*, Vol. 3, No. 1, pp. 23-37, 2012.

[3] M. Parvin, E. Jafari and R. Azizi, "A Multi-Armed Bandit Problem-Based Target Coverage Protocol for Wireless Sensor Network, Computing", *Proceedings of International Conference on Communication and Networking Technologies*, pp. 1-5, 2014

[4] S. Getsy S.R. Kalaiarasi, S. Neelavathy Pari and D. Sridharan, "Energy Efficient Clustering and Routing in Mobile Wireless Sensor Network", *International Journal of Wireless and Mobile Networks*, Vol. 2, No. 1, pp. 106-114, 2010.

[5] A.A. Abbasi and M. Younis, "A Survey on Clustering Algorithms for Wireless Sensor Networks", *Computer Communication*, Vol. 30, pp. 2826-2841, 2007.

[6] Y.J. Oh and K.W. Lee, "A Clustering Algorithm based on Mobility Properties in Mobile Ad Hoc Networks", *International Journal of Distributed Sensor Networks*, Vol. 11, No. 6, pp. 1-12, 2015.

[7] X.A. Shiny and R.J. Kannan, "Energy Efficient Clustering Protocol using Self Organizing Map in MANET", *Indian Journal of Science and Technology*, Vol. 8, No. 28, pp. 23-36, 2015.

[8] M. Alinci, E. Spaho, A. Lala and V. Kolici, "Clustering Algorithms in MANETs: A Review", *Proceedings of IEEE International Conference on Complex, Intelligent, and Software Intensive Systems*, pp. 330-335, 2015.

[9] E.A. Alrashed and M.H. Karaata, "Imposter Detection in Mobile Wireless Sensor Networks", *International Journal of Computer and Communication Engineering*, Vol. 3, No. 6, pp. 1-14, 2014.

[10] T. Dimitriou, E.A. Alrashed, M.H. Karaata and A. Hamdan, "Imposter Detection for Replication Attacks in Mobile Sensor Networks", *Computer Networks*, Vol. 108, pp. 210-222, 2016.

[11] I. Bhardwaj, N.D. Londhe and S.K. Kopparapu, "Study of Imposter Attacks on Novel Fingerprint Dynamics based Verification System", *IEEE Access*, Vol. 5, pp. 595-606, 2017.

[12] W.T. Zhu, J. Zhou, R.H. Deng and F. Bao, "Detecting Node Replication Attacks in Wireless Sensor Networks: A Survey", *Journal of Network and Computer Applications*, Vol. 35, No. 3, pp. 1022-1034, 2012.

[13] A. Becher, Z. Benenson and M. Dornseif, "Tampering with Motes: Real-World Physical Attacks on Wireless Sensor Networks", *Proceedings of International Conference on Security in Pervasive* Computing, pp. 104-118, 2006.

[14] M.A. Simplicio, P.S. Barreto, C.B. Margi and T.C. Carvalho, "A Survey on Key Management Mechanisms for Distributed Wireless Sensor Networks", *Computer Networks*, Vol. 54, No. 15, pp. 2591-2612, 2010.

[15] D. Wallner, E. Harder and R. Agee, "*Key Management for Multicast: Issues and Architectures*", RFC Editor, 1999.

[16] X.S. Yang and S. Deb, "Cuckoo Search: Recent Advances and Applications", *Neural Computing and Applications*, Vol. 24, No. 1, pp. 169-174, 2014.

[17] X.S. Yang and S. Deb, "Cuckoo Search Via Levy Flights", *Proceedings of World Congress on Nature and Biologically Inspired Computing*, pp. 210-214, 2009.