BACTERIAL FORAGING OPTIMIZATION ALGORITHM IN WIRELESS SENSOR NETWORK

Pooja Raj Rana
Computer Science and Engineering
HCTM Technical Campus
Kaithal , Haryana

Er Anish Soni
Assistant Professor
Department of CSE
HCTM Technical Campus, Kaithal

ABSTRACT- *The purpose of a wireless sensor network (WSN) is to provide the sensed information from data gathered by spatially distributed sensors. For this purpose there is need of certain aggregate functions of this distributed data. This aggregate data is computed by communicating all the sensor nodes and sending the relevant data to a central collector node is a highly inefficient solution for this purpose. Thus we need an alternative technique to perform data aggregation. Now, the questions arise that: what is the optimal way to compute an aggregate function from a set of statistically correlated values stored in different nodes; what security should be applied on the data aggregation technique so that no compromised or faulty node in the network can affect the accuracy of the computed result. In this paper, we have presented Bacterial Foraging Optimization algorithm for WSNs that is secure because it eliminates malicious node by estimating its trust value and provide security against malicious insider attack by any compromised or faulty node in the network in this paper, An investigation on distributed iterative aggregation is presented in which the nodes that get localized in an iteration act as references for remaining nodes to localize. BFOA is energy and bandwidth efficient because cluster-heads prevent the transmission of redundant data from sensor nodes.*

KEYWORDS-*wireless sensor networks (WSNs), data aggregation algorithm, in-network computation, distributed estimation, security.*

I. INTRODUCTION

The intention of established data webs such as the Internet is to enable end-to-end data transfer. Data streams in such webs are grasped across point-to-point links, alongside intermediate nodes plainly forwarding data packets lacking modifying their payloads. In difference, the intention of a wireless sensor web (WSN) is to furnish the users alongside admission to the data of attention from the data gathered by spatially distributed sensors. In most requests, users need merely precise aggregate purposes of this distributed data. Examples contain the average temperature in a web of temperature sensors, a particular trigger in the case of an alarm web, or the locale of an event. Such aggregate purposes might be computed below the end-to-end data flow paradigm by conversing all relevant data to a central collector node. This, though, is a exceedingly inefficient resolution for WSNs that have harsh constraints in power, recollection and bandwidth, and whereas taut latency constraints are to be met. An alternative resolution is to present in-network computations. Though, in this case, the question that arises is how to present distributed computations above a web of nodes related by wireless links in an effectual manner. What is the optimal method to compute, for example, the average, min, or max of a set of statistically correlated benefits stored in disparate nodes? How such computations should be gave in the attendance of unreliability such as sound, packet drops, and node failures? Such inquiries join the complexities of multi-terminal data theory, distributed basis coding, contact intricacy, and distributed computation. This makes progress of an effectual in-network computing framework for WSNs extremely challenging.

In this paper, we have believed a WSN as a collective entity that performs a detecting task and have counseled a distributed estimation algorithm that can be requested to a colossal class of aggregation problems. Separately from making a tradeoff amid the level of accuracy in aggregation and the power expended in computation of the aggregate purpose, we have held in one more extremely vital and relevant factor in WSN- security. Unfortunately, even nevertheless protection has been recognized as a main trial for sensor webs, present propositions for data aggregation protocols have not been projected alongside protection in mind, and subsequently they are all vulnerable to facile attacks. Even after a solitary sensor node is seized, compromised or spoofed, an attacker can frequently impact the worth of an aggregate purpose lacking each attached, obtaining finished manipulation above the computed aggregate. In fact, each protocol that computes the average, sum, minimum, or maximum purpose is insecure opposing malicious data, no matter how these purposes are computed. Keeping in mind these menaces, we have industrialized an energy-efficient aggregation algorithm that is safeguard and robust opposing malicious aggressions in WSNs. The main menace that we have believed as arranging the counseled scheme is the inoculation of malicious data in the web by an antagonist who has compromised a sensor's detected worth by subjecting it to infrequent temperature, lighting, or supplementary spoofed environmental conditions.

In the counseled scheme, every single node in a WSN has finished data concerning the parameter being sensed. This is in difference to the snapshot aggregation; whereas the detected parameters are aggregated at the intermediate nodes till the final aggregated consequence reaches the root. Every single node, in the counseled algorithm, instead of uncasing its detected data to its parent, shows its guesstimate to all its neighbors. This makes the protocol extra fault-tolerant and increases the data potential in the network. The counseled protocol is comparable to the one counseled in. Though, it is extra safeguard and reliable even in attendance of compromised and defective nodes in a WSN.

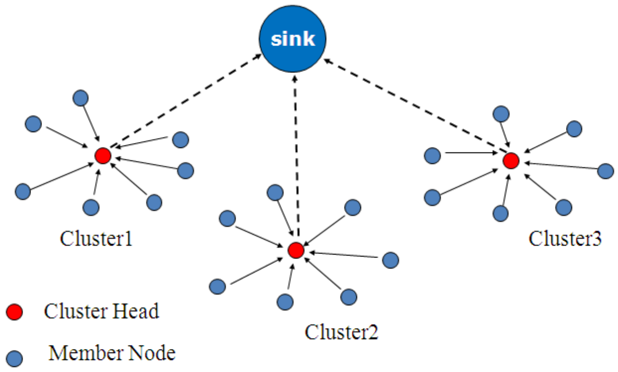


Fig 1: Data Aggregation in Wireless Sensor Networks

II. RELATED WORK

Nandini S. Patil, and P. R. Patil et al. (2010) in this paper, sensor networks are collection of sensor nodes which co-operatively send sensed data to base station. As sensor nodes are battery driven, an efficient utilization of power is essential in order to use networks for long duration hence it is needed to reduce data traffic inside sensor networks, reduce amount of data that need to send to base station. It is preferable to do in network processing inside network and reduce packet size. One such approach is data aggregation which attractive method of data gathering in distributed system architectures and dynamic access via wireless connectivity.

ShaoJie, Tang, Jing Yuan, et al. (2010) in this paper, the benefits of using mobile sink to prolong sensor network lifetime have been well recognized. However, few provably theoretical results remain are developed due to the complexity caused by time-dependent network topology. In this work, we investigate the optimum routing strategy for the static sensor network. We further propose a number of motions stratify for the mobile sink(s) to gather real time data from static sensor network, with the objective to maximize the network lifetime. Specially, we consider a more realistic model where the moving speed and path for mobile sinks are constrained. Our extensive experiments show that our scheme can significantly prolong entire network lifetime and reduce delivery delay.

Jialiang, Lu, Fabrice Valois, et al. (2010) in this paper, wireless sensor networks (WSNs) are data centric networks to which data aggregation is a central mechanism. Nodes in such networks are known to be of low complexity and highly constrained in energy. This requires novel distributed algorithms to data aggregation, where accuracy, complexity and energy need to be optimized in the aggregation of the raw data as well as the communication process of the aggregated data. To this end, we propose in this work a distributed data aggregation scheme based on an adaptive Auto-Regression Moving Average (ARMA) model estimation using a moving window technique and running over suitable communications protocols. In our approach, we balance the complexity of the algorithm and the accuracy of the model so as to facilitate the implementation. Subsequent analysis shows that aggregation efficiency up to 60% can be achieved with a very fine accuracy of 0.03 degree. And simulation results confirm that this distributed algorithm provides significant energy savings (over 80%) for mass data collection applications.

Liu, Xiang, Jun Luo et al. (2011) in this paper, as a burgeoning technique for signal processing, compressed sensing (CS) is being increasingly applied to wireless communications. However, little work is done to apply CS to multichip networking scenarios. In this paper, we investigate the application of CS to data collection in wireless sensor networks, and we aim at minimizing the network energy consumption through joint routing and compressed aggregation. We first characterize the optimal solution to this optimization problem, and then we prove its NP-completeness. We further propose a mixed integer programming formulation along with a greedy heuristic, from which both the optimal (for small scale problems) and the near-optimal (for large scale problems) aggregation trees are obtained. Our results validate the efficacy of the greedy heuristics, as well as the great improvement in energy efficiency through our joint routing and aggregation scheme.

Dilip, Kumar, Trilok C. Aseri, et al. (2011) in this paper, in recent years, energy efficiency and data gathering is a major concern in many applications of Wireless Sensor Networks (WSNs). In this paper, we propose a novel Energy Efficient Clustering and Data Aggregation (EECDA) protocol for the heterogeneous WSNs which combines the ideas of energy efficient cluster based routing and data aggregation to achieve a better performance in terms of lifetime and stability. EECDA protocol includes a novel cluster head election technique and a path would be selected with maximum sum of energy residues for data transmission instead of the path with minimum energy consumption. Simulation results show that EECDA balances the energy consumption and prolongs the network lifetime by a factor of 51%, 35% and 10% when compared with Low-Energy Adaptive Clustering Hierarchy (LEACH), Energy Efficient Hierarchical Clustering Algorithm (EEHCA) and Effective Data Gathering Algorithm (EDGA), respectively.

Lei, Zhang, Honggang Zhang, et al. (2013) In this paper, we propose two efficient and privacy-preserving data aggregation protocols for WSNs: PASKOS (Privacy preserving based on Anonymously Shared Keys and Omniscient Sink) and PASKIS (Privacy preserving based on Anonymously Shared Keys and Ignorant Sink)—requiring low overhead. Both protocols guarantee privacy preservation and a high data-loss resilience. In particular, PASKOS effectively protects the privacy of any node against other nodes, by requiring O(logN) communication cost in the worst case and O(1) on average, and O(1) as for memory and computation. PASKIS can even protect a node's privacy against a compromised sink, requiring only O(1) overhead as for computation, communication, and memory; however, these gains in efficiency are traded off with a (slightly) decrease in the assured level of privacy.

III. PROPOSED WORK

In-network data aggregation can cut the number of contact and hence the power consumed, exceptionally in colossal WSNs. The main believed is to join partial aftermath at intermediate nodes across memo routing. One way, is to craft a spanning tree implanted at the BS, and next present in-network aggregation alongside the tree. The vital aggregates believed by the scrutiny area contain Count and Sum. It is frank to generalize these aggregates to predicate Count (e.g., the number of sensors whose reading is higher than 10 units) and Sum. In supplement, Average can be computed from Count and Sum. We can additionally facilely spread a Sum algorithm to compute Average Deviation and Statistical Moment of each order. Though, contact defeats emerging from node and transmission wrecks, that are public in WSNs, can adversely alter tree-based aggregation approaches. To address this setback, we can make use of multi-path routing methods for forwarding sub-aggregates. For duplicate insensitive aggregates such as Min and Max, this way provides a fault-tolerant solution. Unfortunately, for duplicate sensitive aggregates, such as Count and Sum, multi-path routing leads to double-counting of sensor readings

We will apply BFO (Bacterial Foraging Optimization) method employing multi objective optimization for a Safeguard Data Aggregation. BFO is an optimization method and considers the skill of resolving convoluted setbacks by cooperation. This method is additionally inspired by the communal foraging deeds like ant dominion and particle swarm optimization. It entices the researchers due to its efficiency in resolving real globe optimization setbacks and gives larger aftermath than established methods of setbacks resolving

IV. DISTRIBUTED AGGREGATION ALGORITHM

In this serving, we counsel the adjusted distributed estimation algorithm that is safeguard and resistant to associate attack by compromised and defective nodes. There are vitally two groups of aggregation functions: (i) aggregation purposes that are reliant on the benefits of a insufficient nodes (e.g., the max consequence is established on one node), and (ii) aggregation purposes whose benefits are ambitious by all the nodes (e.g., the average function). Though, computations of both these kinds of purposes are adversely altered by wrong detected consequence dispatched by even a extremely insufficient number of compromised nodes. In this paper, we ponder merely the early case, i.e., aggregation purpose that find or approximate a little kind of borders (e.g., maxima, minima), and hence the aggregation consequence is ambitious by the benefits of insufficient nodes. Though, the counseled algorithm does not accept each vision concerning the underlying physical procedure.

Distributed Cooperative Approach

In the counseled distributed estimation algorithm, a sensor node instead of sending a partly aggregated consequence, maintains and if needed, transmits an estimation of the globe aggregated result. The globe aggregated description in finished will be a vector as it embodies multidimensional parameters detected by disparate nodes. A globe guesstimate will therefore be a probability density purpose of the vector that is being estimated. Though, in most of the useful situations, due to lack of adequate data, convoluted computational necessity or unavailability of urbane estimation instruments, an guesstimate is embodied

In the snapshot aggregation, a node does not have each manipulation on the rate at that it dispatch data to its parents; it has to always pursue the rate enumerated the user application. Moreover, every single node has slight data concerning the globe parameter, as it has no believed concerning what is transpiring beyond its parent. In counseled way, a node accepts estimations from all of its acquaintances, and softly gains in vision concerning the globe information. It helps a node to comprehend whether its own data is functional to its neighbors. If a node realizes that its guesstimate might be functional to its acquaintances, it transmits the new estimate. Unlike snapshot aggregation whereas the node transmits its guesstimate to its parent, in the counseled scheme, the node shows its guesstimate to all its neighbors. Moreover, there is no demand to institute and uphold a hierarchical connection amid the nodes in the network. This makes the algorithm chiefly suitable for several user, mobile users, defective nodes and transient web partition situations.

V. PURPOSED ALGORITHM

In the above-mentioned algorithm the bacteria experiences chemo taxis, whereas they like to move towards a nutrient gradient and circumvent noxious environment. Usually the bacteria move for a longer distance in a approachable environment. Figure below delineates how clockwise and counter clockwise movement of a bacterium seize locale in a nutrient solution. When they become food in adequate, they are increased in length and in attendance of suitable temperature they break in the middle to from an precise replica of itself. This phenomenon inspired by Passino to introduce an event of reproduction in BFOA. Due to the occurrence of unexpected environmental changes or attack, the chemo tactic progress could be obliterated and a cluster of bacteria could move to a little other place or a little supplementary could be gave in the swarm of concern. This constitutes the event of elimination-dispersal in the real bacterial populace, whereas all the bacteria in a span are slayed or a group is dispersed into a new portion of the nature.

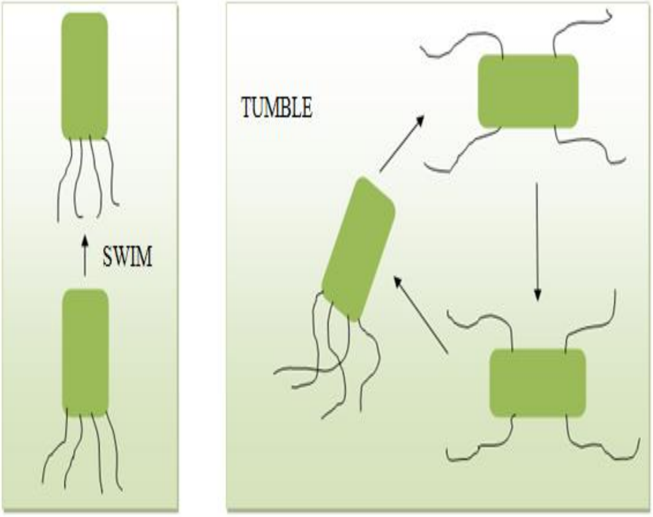


Figure 5.1: Swim and tumble of a bacterium

Now suppose that we want to find the minimum of J(θ)where‘$\theta \in \Re^p$ (i.e.θ is a p-dimensional vector of real numbers), and we do not have measurements or an analytical description of the gradient∇J(θ). BFOA mimics the four principal mechanisms observed in a real bacterial system: chemo taxis, swarming, reproduction, and elimination-dispersal to solve this non-gradient optimization problem. A virtual bacterium is actually one trial solution (may be called a search-agent) that moves on the functional surface (see Figure 3.7) to locate the global optimum agent). Let us define a chemo tactic step to be a tumble followed by a tumble or a tumble followed by a run. Let *j* be the index for the chemo tactic step. Let *k* be the index for the reproduction step. Let *l* be the index of the elimination-dispersal event. Also let

P: Dimension of the search space,
s: Total number of bacteria in the population,
N_c: The number of chemotactic steps,
N_s: The swimming length,
N_{re}: The number of reproduction steps,
N_{ed}: The number of elimination-dispersal events,
P_{ed}: Elimination-dispersal probability,
C (i): The size of the step taken in the random direction specified by the tumble.

Let P(*j*, *k*, *l*) {θ^i (*j*, *k*, *l*) | *i* 1,2,...,S} represent the position of each member in the population of the *S* bacteria at the *j*-th chemo tactic step, *k*-th reproduction step, and *l*-th
Elimination-dispersal event Here, let J(*i*, *j*, *k*, *l*) denote the cost at the location of the *i*-th bacterium θ^i(*j*, *k*, *l*)$\in \Re^p$ (sometimes we drop the indices and refer to the *i*-th bacterium position asθ^i). Note that we will interchangeably refer to J as being a “cost” (using terminology from optimization theory) and as being a nutrient surface (in reference to the biological connections). For actual bacterial populations, Scan be very large (e.g. S=109), but p= 3. In our computer simulations, we will use much smaller population sizes and will keep the population size fixed. BFOA, however, allows p> 3 so that we can apply the method to higher dimensional optimization problems. [21]

Algorithm:

[Step 1]: Initialize parameters P, s, N_c, N_s, N_{re}, N_{ed}, P_{ed},C(i)(i=1,2…S),θ^i .

[Step-2] Elimination-dispersal loop: $l = l + 1$

[Step-3] Reproduction loop: $k = k + 1$

[Step-4] Chemo taxis loop: $j = j + 1$

(1) For $i = 1, 2,\dots, S$, calculate cost function value and efficiency- for each bacterium i as follows:

(a) N_{is} Signal samples are passed through the model.
(b) The output is then compared with the corresponding desired signal to calculate the error.
(c) The same of the squared error averaged over N_{is} is finally stored in J(i, j, k, l). The cost function is calculated for number of input samples.
(d) End of for loop.

(2) For $i = 1, 2,\dots, S$ take the tumbling/swimming decision
Tumble: Generate a random vector $\Delta(i)$ with each element
$\Delta m(i)$ $m = 1, 2, p$, a random number.
Move: Let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

Fixed step size in the direction of tumble for bacterium i is considered.
Compute $J(i, j + 1, k, l)$ and then
Let
$J_{sw}(i,j+1,k,l)$ = J(i, j+1,k,l) + $J_{cc}\theta^i(j+1,k,l),\ P(j+1,k,l)$
Swim:

(i) Let $m = 0$; (counter for swim length)

(ii) While $m < N_s$(have not climbed down too long)

- Let $m = m + 1$
- If J_{sw} $(i, j + 1, k, l) < J_{last}$ (if doing better), let $J_{last} = J_{sw}$ $(i, j + 1, k, l)$ and Let $\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$and use this θ^i $(j + 1, k, l)$ to compute the new $J(i, j + 1, k, l)$
- Else, let $m = N_s$. This is the end of the while statement.

(3) Go to next bacterium $(i + 1)$ if $i \neq S$ (*i.e.* go to b) to process the next bacterium.

[Step-5] If $j < N_c$, go to step 4. In this case, continue chemotaxis since the life of the bacteria is not over.

[Step-6] Reproduction:

(a) For the given k and l, and for each $i = 1, 2,\dots, s$, Let $J_{healt\,h}$ = min J_{sw} (i, j, k, l) be the health of the bacterium i (a measure of how many nutrients it got over its life time and how successful it was at avoiding noxious substance). Sort bacteria in order of ascending cost $J_{healt\,h}$(higher cost means lower health).
(b) The $S_r = s/2$ bacteria with highest $J_{healt\,h}$values die and other S_r bacteria with the best value split (and thecopies that are made are placed at the same location as their parent).

[Step-7] If $k < N_{re}$ go to 3. In this case, the number of specified reproduction steps has not been reached, so the next generation of the chemo tactic loop is started.
[Step-8] Elimination-dispersal: For $i = 1, 2,\dots, s$, with probability P_{ed}, eliminates and disperses each bacterium (this keeps the number of bacteria in the population constant).
To do this, if a bacterium is eliminated, simply disperse another one to a random location on the optimization domain. If $l < N_{ed}$, then go to step 2; otherwise, print the results and stop.[21]

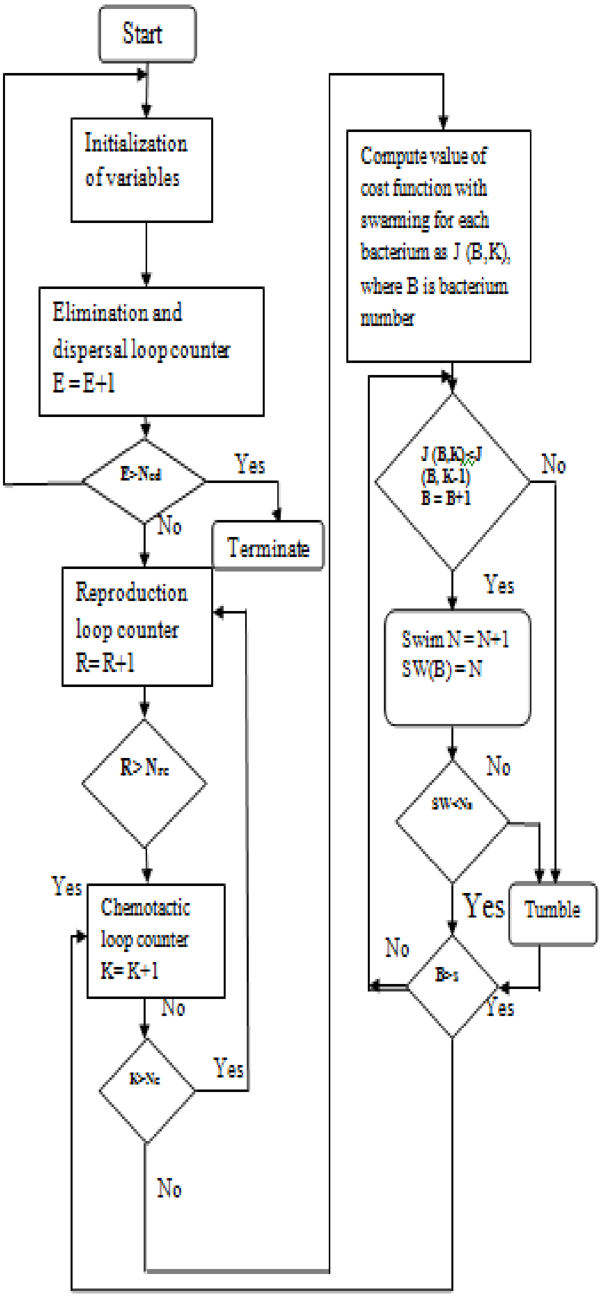
Below we briefly describe the four prime steps in BFOA:

CHEMOTAXIS: This process simulates the movement of an E.colicell through swimming and tumbling via flagella. Biologically an E.colibacterium can move in two different ways. It can swim for a period of time in the same direction or it may tumble, and alternate between these two modes of operation for the entire lifetime. Suppose θ^i (*j*, *k*, *l*) represents *i*-th bacterium at *j*-th chemo tactic, *k*-th reproductive and *l*-th elimination-dispersal step. *C* is the size of the step taken in the random direction specified by the tumble (run length unit). Then in computational chemo taxis the movement of the chemo taxis the movement of the bacterium may be represented by

$$\theta^i(j+1,k,l)=\theta^i(j,k,l)+C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

Where Δ indicates a vector in the random direction whose elements lie in [-1, 1].

(i) SWARMING: An interesting cluster deeds has been noted for countless motile species of bacteria encompassing E.coliand S. typhimurium, whereas intricate and stable spatio-temporal outlines (swarms) are industrialized in semisolid nutrient medium. A cluster of E.colicell organizes them in a voyaging ring by advancing up the nutrient gradient after allocated amidst a semisolid matrix alongside a solitary nutrient chemo-effecter. The cells when stimulated by an elevated level of succinate, discharge an attractant aspertate, that helps them to aggregate into clusters and therefore move as concentric outlines


Start
Initialization of variables
Elimination and dispersal loop counter E=E+1
E>Ned
Yes
No
Terminate
Reproduction loop counter R=R+1
R> Nre
Yes
Chemotactic loop counter K=K+1
No
K>Nc
Yes
No
Compute value of cost function with swarming for each bacterium as J (B,K), where B is bacterium number
J (B,K)<J (B, K-1) B=B+1
No
Yes
Swim N=N+1 SW(B)=N
No
SW<Ns
Yes
Tumble
No
B>s
Yes

Flow chart Steps of BFO Algorithm

Of swarms alongside elevated bacterial density the cell-to-cell indicating in E. coli swarm could be embodied by the pursuing purpose.

$$J_{cc}\big(\theta, P(j,k,l)\big) = \sum_{i=1}^{S} J_{cc}\left(\theta, \theta^{i}(j,k,l)\right) =$$

$$\sum_{i=1}^{S}\left[-d_{attractant}\ exp\left(-w_{attractant}\ \sum_{m=1}^{P}\left(\theta_m - \theta_m^i\right)\ ^{2}\right)\right]$$

$$+\sum_{i=1}^{S}\left[h_{repellent}\ exp\left(-w_{reppellent}\ \sum_{m=1}^{P}\left(\theta_m - \theta_m^i\right)\frac{1}{2}\right)\right]$$

where $J_{cc}\big(\theta, P(j,k,l)\big)$ is the objective function value to be added to the actual objective function (to be minimized) to present a time varying objective function, *S* is the total number of bacteria, *p* is the number of variables to be optimized, which are present in each bacterium and$\theta = \left[\theta_1, \theta_2, \ldots. \theta_p\right]\frac{1}{T}$ is a point in the *p*-dimensional search domain. $d_{attractant}$,$w_{attractant}$,$h_{repellent}$,$w_{repellent}$ Are different coefficients that should be chosen properly?

(ii) REPRODUCTION: The least healthy bacteria eventually die while each of the healthier bacteria (those yielding lower value of the objective function) asexually split into two bacteria, which are then placed in the same location. This keeps the swarm size constant.

(iii) ELIMINATION AND DISPERSAL: Gradual or unexpected adjustments in the innate nature whereas a bacterium populace lifetimes could transpire due to assorted reasons e.g. a momentous innate development of temperature could slaughter a cluster of bacteria that are presently in a span alongside a elevated compression of nutrient gradients. Events can seize locale in such a style that all the bacteria in a span are slayed or a cluster is dispersed into a new location. To simulate this phenomenon in BFOA a little bacteria are liquidated at random alongside a extremely tiny probability as the new replacements are randomly initialized above the find space

VI. RESULTS AND ANALYSIS

In this serving, we delineate the simulations that have been given on the counseled scheme. As the counseled algorithm is an expansion of the algorithm gave in, we present here the aftermath that are extra relevant to our contribution, i.e., the presentation of the protection module. The aftermath connected to the power consumption of nodes and aggregation accuracy for disparate threshold benefits are gave in detail in and consequently these are not inside the scope of this work. In the simulated nature, the requested request accomplishes temperature monitoring, established on MATLAB. The nodes sense the temperature unceasingly and dispatch the maximum detected temperature merely after it differs from the last data dispatched by extra than 2%.In order to simulate the temperature deeds of the nature, random numbers are generated pursuing a Gaussian allocation, seizing into thought average deviation of 1°C from an average temperature of 25°C. The simulation parameters are gave in Tab.5.1

Parameter	**Value**
No. of nodes	160
Simulation time	200
Coverage area	120m*120m
Initial energy in each node	5 Joules
MAC protocol	IEEE 802.11
Routing algorithm	None
Node Distribution	Uniform random
Transmission power of each node	12 mW
Transmission range	15 m

Node capacity	5 buffers
Energy spent in transmission	0.75 W
Energy spent in reception	0.25 W
Energy spent in sensing	10 mV
Sampling Period	0.5 s
Node mobility	Stationary

TABLE 6.1 SIMULATION PARAMETERS

IMPLEMENTATION RESULTS

After making the Messaging scheme adaptive nowadays we apply BFO algorithm for optimization. Flow chart 4.1 displays the steps encompassed in BFOA optimization. Goal purposes i.e. nodes belief worth, stay worth and power is allocated to BFO as its parameters.

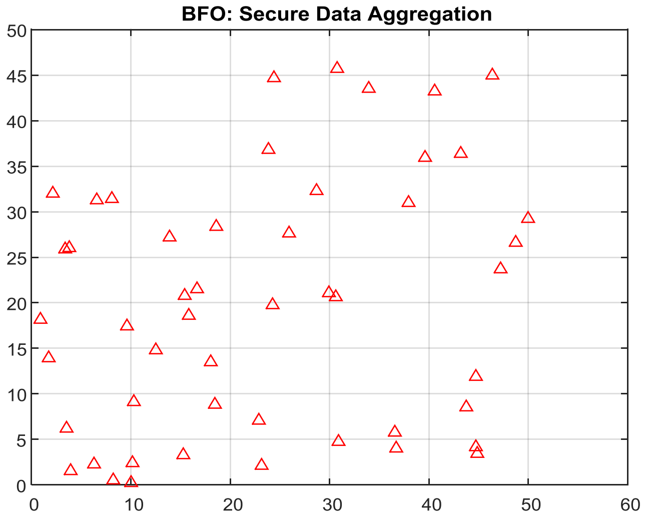


Fig 6.1: BFO WSN Initial Deployment

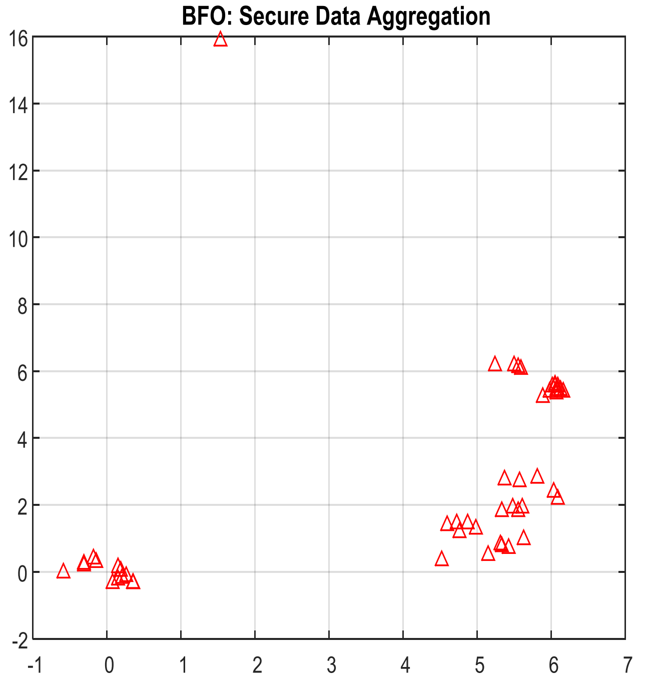


Fig 6.2: Cluster head formulation during secure aggregation

All the output aftermath of adaptive scheme and optimization outputs are calculated. So, nowadays we do analogy of all the aftermath to find out that is better. After we difference the output of Normal WSN and BFO-WSN we finished that power consumed by Normal WSN is extra than BFO-WSN. Additionally after number of nodes increases power consumption in Normal WSN is increased quickly as it stays concerning stable in BFO-WSN. Here power given to nodes is 160joules**.**

Table 6.2 Comparison of remaining energy in Normal WSN&BFO-WSN

	Remaining Energy	
No. of nodes	**Normal WSN**	**BFO-WSN**
5	170	187
10	155	167
15	155	166
20	163	187
25	152	165

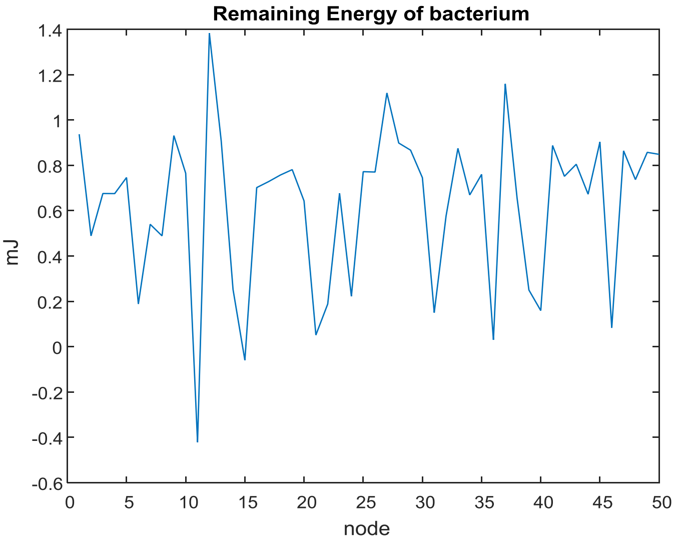


Fig 6.3: remaining energy of bacterium after data aggregation

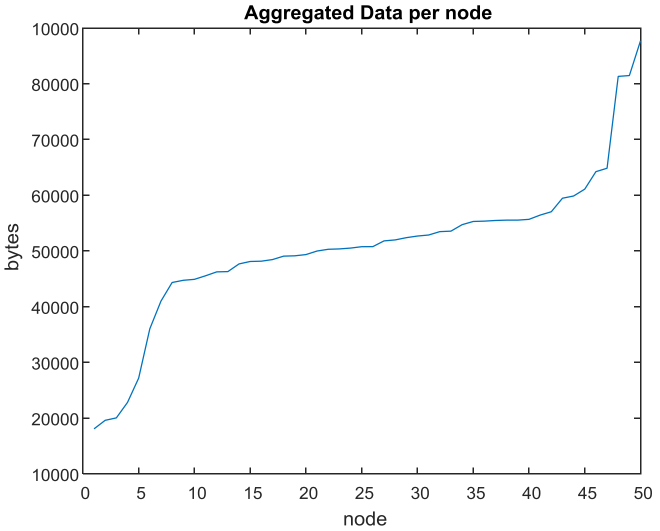


Fig 6.4 : Total data aggregation per bacterium

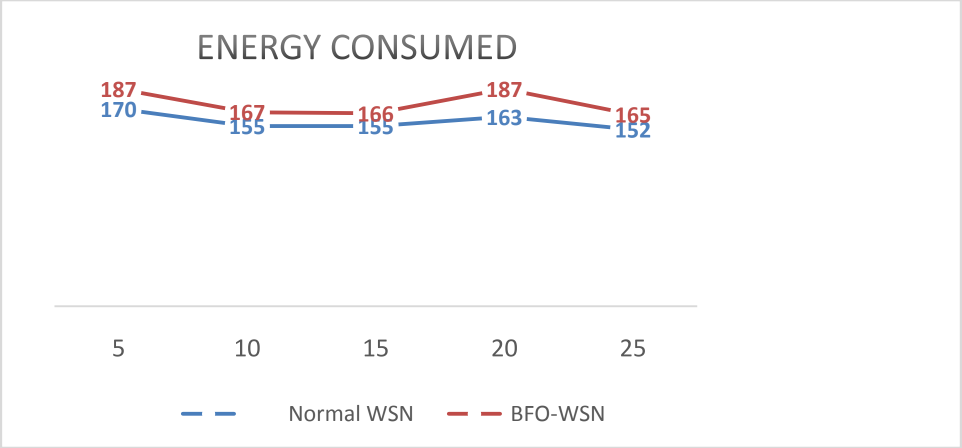


Fig 6.5 Energy consumed by Normal WSN&BFO-WSN w.r.t No. of event occurred

Now, analogy is completed on the basis of event occurrence. Each node has primarily 1 joule of energy. When we contrasted Normal WSN and BFO-WSN we finished that alongside rising no. of events power consumption is decreased extra in BFO-WSN as it goes down in Normal WSN quickly.

Table 6.3 Energy consumed by Normal WSN&BFO-WSN w.r.t No. of event occurred

Energy consumed		
No. of event occurred	**Normal WSN**	**BFO-WSN**
10	0.92	0.95
20	0.85	0.89
30	0.75	0.83
40	0.67	0.78
50	0.58	0.71
60	0.49	0.66
70	0.40	0.60
80	0.31	0.54

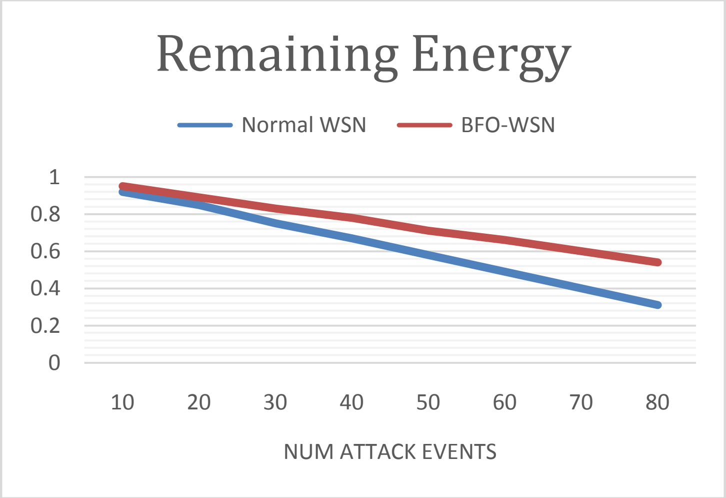


Fig 6.6 Remaining energy in NORMAL WSN and BFO WSN

Also to evaluate the performance of the security module of the proposed algorithm, two disparate scenarios are simulated. In the early case, the aggregation algorithm is executed in the nodes without invoking the security module to estimate the power consumption of the aggregation algorithm. In the second case, the security module is invoked in the nodes and some of the nodes in the network are intentionally compromised. This examination allows us to estimate the overhead associated alongside the security module of the algorithm and its detection effectiveness

It is observed that transport ratio (ratio of the packets dispatched to the packets received by the nodes) is not affected by invocation of the security module. This is expected, since the packets are transmitted in the same wireless environment, invocation of the security module should not have any influence on the transport ratio.

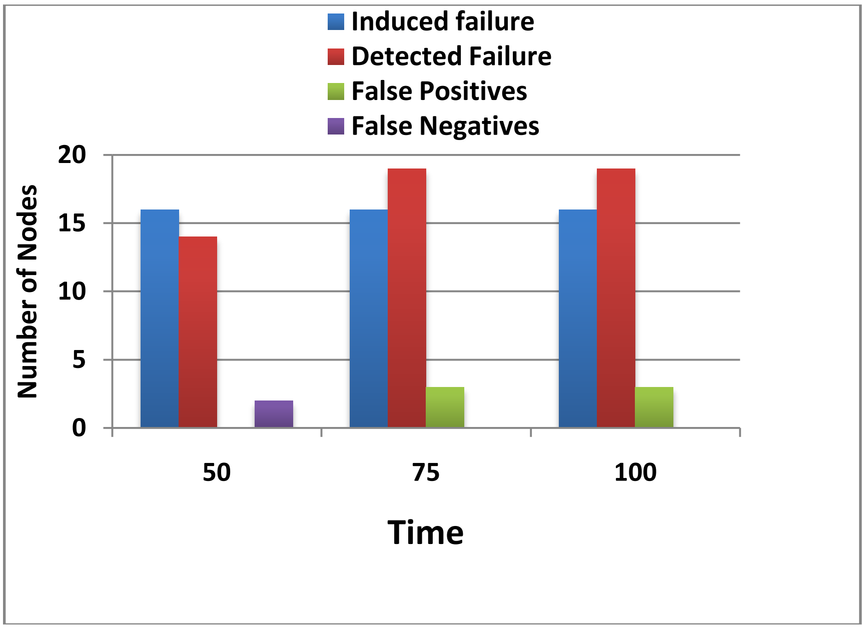


Figure 6.7 Detection effectiveness with 10% nodes in the network faulty

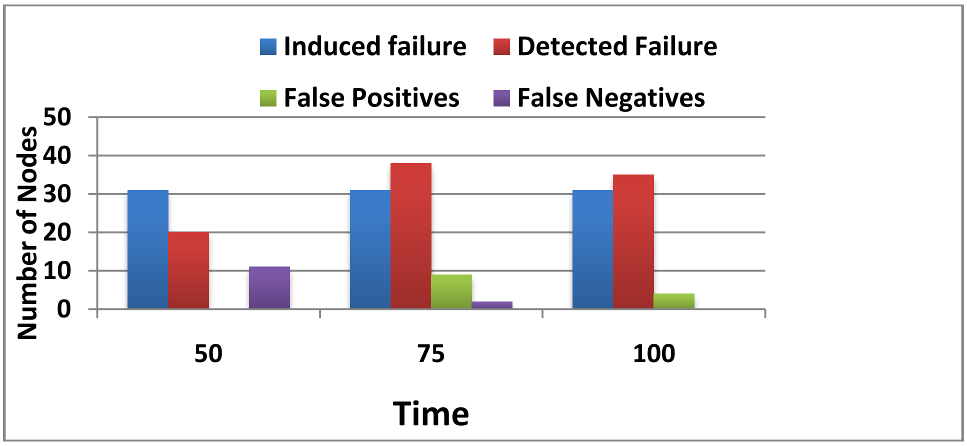


Figure 6.8 Detection effectiveness with 20% nodes in the network faulty

From the perspective of power consumption, it is observed that the invocation of the security module causes an average increase of 30% power consumption in the nodes in the network. This increase is observed after 20% of the nodes chosen randomly are compromised intentionally after the aggregation algorithm was executed. This increase in power consumption is due to additional transmission and reception of messages after the security module is invoked.

To evaluate the detection effectiveness of the security scheme, more examinations are conducted. For this purpose, disparate percentage of nodes in the network is compromised and the detection effectiveness of the security scheme is evaluated. Fig5.7 and Fig5.8 present the aftermath for 10% and 20% compromised node in the network respectively. In these diagrams, the false positives denote to the cases whereas the security scheme wrongly identifies a sensor node as faulty as it is actually not so. False negatives, on the other hand, are the cases whereas the detection scheme fails to recognize a sensor node which is actually faulty. It is observed that even after there are 20% compromised nodes in the network the scheme has a extremely elevated detection rate alongside extremely low false positive and false negative rate. The aftermath show that the proposed mechanism is quite effective in detection of filed and compromised nodes in the network

VII. CONCLUSION AND FUTURE WORKS

Aggregation reduces the number of web traffic that helps to cut power consumption on sensor nodes. It though perplexes the by now continuing protection trials for wireless sensor webs and needs new protection methods tailored specifically for this scenario. Bestowing protection to aggregate data in wireless sensor webs is recognized as safeguard data aggregation in WSN. Were the early insufficient works debating methods for safeguard data aggregation in wireless sensor networks? Two main protection trials in safeguard data aggregation are confidentiality and integrity of data. As encryption is conventionally utilized to furnish conclude to conclude confidentiality in wireless sensor web, the aggregators in a safeguard data aggregation scenario demand to decrypt the encrypted data to present aggregation. This exposes the plaintext at the aggregators, making the data vulnerable to aggressions from an adversary. Comparably an aggregator can inoculate fake data into the aggregate and make the center station accord fake data. Thus, as data aggregation enhances power efficiency of a web, it perplexes the continuing protection challenges. In this paper we have debated the protection vulnerabilities of data aggregation arrangements, and present a survey of robust and safeguard aggregation protocols that are resilient to fake data inoculation attacks. In upcoming we will counsel a novel scheme for cutting aggressions in WSN as data aggregation.

VIII. REFERENCES

1. Erdal Cayirci, “Data Aggregation and Dilution By Modulus Addressing In Wireless Sensor Networks “IEEE Communications Letters 7, NO. 8 (2003): 355-357.
2. Jamal N., Al-Karaki, Raza Ul-Mustafa, And Ahmed E. Kamal. "Data Aggregation In Wireless Sensor Networks-Exact And Approximate Algorithms" In High Performance Switching And Routing, 2004 Hpsr. 2004 Workshop On, Pp. 241-245. Ieee, 2004
3. Marc, Lee, And Vincent Ws Wong "An Energy-Aware Spanning Tree Algorithm For Data Aggregation In Wireless Sensor Networks" In Communications, Computers And Signal Processing, 2005 Pacrim. 2005 Ieee Pacific Rim Conference On, P p. 300-303. IEEE, 2005
4. Claude, Castelluccia, Einar Mykletun, And Gene Tsudik. "Efficient Aggregation Of Encrypted Data In Wireless Sensor Networks." In Mobile And Ubiquitous Systems: Networking And Services, 2005. Mobiquitous 2005. The Second Annual International Conference On, Pp. 109-117. IEEE, 2005
5. Yingpeng, Sang, Hong Shen, Yasushi Inoguchi, Yasuo Tan, And Naixue Xiong. "Secure Data Aggregation In Wireless Sensor Networks: A Survey." In Parallel And Distributed Computing, Applications And Technologies, 2006. Pdcat'06. Seventh International Conference On, Pp. 315-320. IEEE, 2006
6. Hongwei,Du, Xiaodong Hu, And Xiaohua Jia. "Energy Efficient Routing And Scheduling For Real-Time Data Aggregation In Wsns." Computer Communications 29, No. 17 (2006): 3527-3535.
7. Ossama, Younis, , Marwan Krunz, And Srinivasan Ramasubramanian. "Node Clustering In Wireless Sensor Networks: Recent Developments And Deployment Challenges. ." Network, IEEE 20, No. 3 (2006): 20-25.
8. Einar, Mykletun, Joao Girao, And Dirk West Off. "Public Key Based Cryptoschemes For Data Concealment In Wireless Sensor Networks." In Communications, 2006. Icc'06. Ieee International Conference On, Vol. 5, Pp. 2288-2295. Ieee, 2006.