A Poisson Hidden Markov Model and Fuzzy based Chicken Swarm Optimization algorithm for efficient fault node detection in wireless sensor network

Nagarajan B
Vellore Institute of Technology

Santhosh Kumar S.V.N (✉️ santhoshkumarsvn@vit.ac.in)
Vellore Institute of Technology: VIT University

Research Article

Keywords: node fault, Poisson Hidden Markov model, optimization, fault detection

Posted Date: January 10th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2441530/v1

License: ☛ This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License
A Poisson Hidden Markov Model and Fuzzy based Chicken Swarm Optimization algorithm for efficient fault node detection in wireless sensor network

Nagarajan B¹, Santhosh Kumar SVN¹

¹School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India.

¹nagarajan.b@vit.ac.in, ¹santhoshkumar.svn@vit.ac.in

Abstract
Wireless Sensor Networks (WSN) is built with miniature sensor nodes (SN) which are deployed into the geographical location being sensed to monitor environmental condition which transfer the sensed physical information to the base station for further processing. The sensor nodes frequently experience node failure as a result of their hostile deployment and resource limitations. In WSN, node failure can cause a number of issues, namely Wireless Sensor Networks topology changes, broken communications links, disconnected portions of the network, and data transmission errors. An important concern of WSN is the detecting, diagnosing and recovering of sensor node failures. In the course of this effort, an effective strategy for sensor node failure detection algorithm using Poisson Hidden Markov Model (PHMM) and the Fuzzy based Chicken Swarm Optimization (F-CSO) is proposed for efficient detection of sensor nodes fault in the WSN. The proposed work offers optimal false alarm, false positive, energy consumption, detection accuracy, network lifetime, and least delay rates. Moreover, the F-CSO provides improved localization to locate the defective sensor nodes which are present in the WSN. The proposed work is implemented in the NS2 simulator with realistic simulation parameters and the simulation result demonstrate that the proposed work is more effective in terms of false alarm rate, false positive rate, detection accuracy, delay, energy consumption and network lifetime when it is compared with other existing state of art systems.

Keywords: node fault, Poisson Hidden Markov model, optimization, fault detection

1 Introduction
WSN is made up of a decentralized collection of numerous, sparsely spaced-out, small-sized sensors with limited computing facilities, low battery power and limited memory for data storage and processing [1]. WSN has been widely employed in many sectors, including the military, healthcare, agriculture, and environmental monitoring [2]. Wireless sensor networks, however, are the most difficult technologies to use since they employ a large number of small sensor devices
that are susceptible to failures because of inadequate battery backup, placement of sensor nodes in unattended area and other undesirable environmental conditions [3]. The sensor devices are impacted by hardware and software problems as a result of environmental threats such severe downpours, floods, strong winds, and forest fire in WSN [4].

The probability of node failure is directly proportional to the scalability of the WSN. The SNs are vulnerable to faults because of energy exhaustion, physical damage, communication link failures, and software or hardware failures. The faulty SN is impacted, unable to send data to the base station or to the receiver [5, 30]. Therefore, it's critical to accurately identify the faults in sensor nodes, to make SNs to perform unlimited services. The failure of SNs in the network results in poor network performance, partitioning of sensor network, fail to transmit accurate information to the base station. Moreover, most of the existing fault detection systems suffer from computational overhead on sensor nodes which makes vulnerable to early sensor node failure. The existing fault detection techniques, however, have high false alarm rate and has poor fault detection accuracy [6]. Additionally, they use tremendous quantity of power to identify faulty sensor nodes, which causes the earlier death of SNs [7]. Hence an efficient fault detection approach for WSNs is needed to optimize and has better fault detection accuracy.

Failures of a hardware component, physical harm, low battery, or adverse environmental factors are a few causes of sensor node failure. As a result, the failure can be managed by creating distinct fault detection for various sensor fault types. The sensor node faults are mainly categorized into hard sensor fault and soft sensor fault [8, 29]. In hardware-based concerns, faulty sensor nodes are detached from the network and unable to connect and communicate with some other network nodes. In hardware sensor faults will occur when of hardware modules failures, such as the battery depletion, sensor unit, localization unit, processing unit, microcontroller unit or transceiver unit, causes hard fault. In software-based sensor failures take place when there is fault in nodes software such as malfunction, transmit inaccurate data, and interact incorrectly with other nodes. While hard faults are permanent errors that need replacement of the sensor node, whereas software based faults can be fixed by changing the current sensor algorithms when are running in the sensor nodes.

According to fault severity, sensor node defects are further divided into permanent fault, transient fault, and intermittent fault [9]. A sensor module with a permanent defect remains inactive for the duration of its life. Since the problem is ongoing, the component needs to be changed. The sensor generates fault behaviour over a longer span of time, despite the fact that the
intermittent fault is not continuous. It is treatable and after some time may resume its flawed behaviour. The transient fault is a short-period malfunction that can be automatically corrected. It is exceedingly challenging to diagnose and control transitory errors because they can arise as a result of brief environmental changes [10]. Depending on the network components, WSN failures are even further classified into three distinct categories: network, node, and base station faults. Hardware or software malfunction led to node defect. A failure node may deviate from initial values and provide incorrect information due to energy depletion and a drop in energy level below the threshold value. Failure of a communication route or link will result in a network fault. These faults lead to issues including poor communication, data packet loss and network failure as a whole. The entire sensor network fails when a sink or base station goes down. Table 1 provides the abbreviations and short forms were used in the proposed work.

Table 1 Abbreviations used in this proposed work

<table>
<thead>
<tr>
<th>Abbreviations</th>
<th>Acronyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson Hidden Markov Model</td>
<td>PHMM</td>
</tr>
<tr>
<td>Fuzzy based Chicken Swarm Optimization</td>
<td>F-CSO</td>
</tr>
<tr>
<td>Wireless Sensor Networks</td>
<td>WSN</td>
</tr>
<tr>
<td>Adaptive Neuro Fuzzy Inference System</td>
<td>ANFIS</td>
</tr>
<tr>
<td>Sensor Node</td>
<td>SN</td>
</tr>
<tr>
<td>Multi-Fault Detector</td>
<td>MFD</td>
</tr>
<tr>
<td>Cellular Learning Automata Faulty Node Detection &amp; Management</td>
<td>CLAFNMD</td>
</tr>
<tr>
<td>Faulty Node Classification &amp; Management</td>
<td>FNCM</td>
</tr>
<tr>
<td>Fault Management Framework for Markov Chain</td>
<td>FMMC</td>
</tr>
<tr>
<td>Optimal Emperor Penguin Optimization</td>
<td>OPEO</td>
</tr>
<tr>
<td>Trajectory Pattern Extraction</td>
<td>TPE</td>
</tr>
<tr>
<td>reactive Distributed Fault Detection</td>
<td>rDFD</td>
</tr>
<tr>
<td>Harmony Search Algorithm</td>
<td>HAS</td>
</tr>
<tr>
<td>Medium Access Control</td>
<td>MAC</td>
</tr>
<tr>
<td>Received Signal Strength</td>
<td>RSS</td>
</tr>
<tr>
<td>Acknowledgement</td>
<td>ACK</td>
</tr>
</tbody>
</table>
Motivating from all these observations in this work an efficient fault detection scheme is proposed which efficiently detects the sensor nodes failure. The overall contributions of this work are

- This work presented a fault detection mechanism for both the soft and hard sensor fault detection which considerably improving accuracy rate of fault detection while minimizes the false alarms.
- This proposed work employs an efficient fault detection approach namely Poisson Hidden Markov Model which is a probability based mechanism for efficient fault detection sensor nodes.
- An optimization algorithm namely Fuzzy based Chicken Optimization localization algorithm which is proposed for providing improved energy efficiency, fault detection rate, network lifespan, and minimize false positive and false alarm rates

The remaining sections of this work is divided into the following sections: Section 2 provides insight studies into the types of sensor faults and various sensor fault detection mechanisms of existing works, section 3 presents the proposed methodology, Section 4 provides the findings and discussion of the proposed work, and Section 5 concludes with recommendations for further research.

### 2 Related Works

Many researchers have proposed various methods on detecting the sensors nodes failure and provide efficient localization among them. In WSN, the authors suggested an ANFIS, a decentralized faulty node detection and categorization approach [11]. The proposed scheme categorizes sensor node faults based on the sensor nodes' crisper performance measure. The nodes in this system are classified as healthy, lifeless, dispatch, resting, or end according on their crisper performance measure. This ANFIS scheme detects the sensor node faults by intra-cluster and inter-

<table>
<thead>
<tr>
<th>Gravitational Search algorithm &amp; Particle Swarm Optimization</th>
<th>GSAPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzy one Class Support Vector Machine based Fault Detection</td>
<td>FCS-MBF</td>
</tr>
<tr>
<td>Quality of Service</td>
<td>QoS</td>
</tr>
<tr>
<td>Fault Detection Scheme</td>
<td>FDS</td>
</tr>
<tr>
<td>Neural Networks</td>
<td>NN</td>
</tr>
</tbody>
</table>
cluster failure detection estimator. The limitations are security elements in this work are not addressed in this system.

For heterogeneous WSNs, the authors [12] introduced a distributed fuzzy logic-based faulty node identification technique. This system provides weighted voting mechanism which uses a fuzzy logic in order to recognize the various sensor nodes faults and occurrences in the deployed area. This self-diagnosis system uses an approach based on spatial correlation to pinpoint the SN failure in the deployed region. Every SN in the monitoring region is assigned with a weight according to the fuzzy logic controller it has. The controller treats a node as faulty if it detects weighted values that differ from those of its surrounding nodes. Moreover, according to the sensed value, location, distance and coverage parameters, this technique is used to discover and restore the faulty sensor node. The main disadvantage of this system is that it completely depends on its neighbor sensor nodes, and it reduces sensor nodes fault detection accuracy.

For WSNs, a FNCM approach based on fuzzy rules is presented in [13] with the goal of detecting and reusing faulty nodes. The faulty sensor nodes are correctly reused during the data routing procedure using that method. The data routing technique increases the usability of malfunctioning nodes, and optimizing network lifetime. The estimation of energy gain is improved by reusing a faulty node from the proposed work. Different nodes are categorized using a fuzzy interference model in accordance with specified function and the defuzzifier system generates a function to obtain the distinct node faults. This system can be utilized for road monitoring, home automation and livestock management in order to identify physical or environmental situations. Moreover, it improves the usability of restored faulty nodes and uses fuzzy logic to get around WSN's uncertainty. The drawbacks include the difficulty for a fuzzy inference model to create prior knowledge about the node condition because of the inherent self-organization feature.

For WSN, the authors of [14] proposed a distributed CLAFNDM strategy for identifying faulty sensor nodes based on hardware condition and reusing those faulty nodes. The status of each SN is computed by cellular learning automata based on the sensor node’s hardware configuration. For efficient identification of the faults, CLA employs eight rules and a suitable machine learning methodology is employed. These eight guidelines were created so that, in the event that CLA makes incorrect decision, it can be corrected in the subsequent round and the correct outcome can be confirmed.
The author suggested FMMC technique can identify faulty sensor nodes and classify sensor faults based on their hardware sensor faults [16] in WSN. The Markov method makes it possible to assess node state and improve fault detection precision. In comparison to other methods, this model uses self-detection to evaluate node results, which consumes less energy. The FMMC model fixes all persistent and intermittent faults, restores the faulty SNs, and assigns updated states for faulty SNs depending on their current hardware state. The disadvantages of this approach include the inability to detect faults in network connectivity perception.

For dynamically determining the performance of live SNs, fixing faulty nodes, and locating the best alternate routing solution, the authors' proposed OEPO technique [20]. In their model, by analyzing the fitness function for each SN, the faulty nodes are located using cluster boundary value and the fitness value. This model enhances the performance and permanence of the network during the self-diagnosis process. Due to the fitness function that is suggested in this model, the detection accuracy produced by this model is lower.

The TPE method has been used in this system for efficient fault detection in WSN [15]. The suggested TPE addresses the issues of fault isolation and detection as a problem of trajectory pattern extraction from various sensor node statuses. The TPE employs the Geographic constraint violation checks based on the sensing states and pattern matching to detect network's faulty sensor nodes.

The Harmony Search Algorithm (HSA) [4] is used to determine which nodes are defective in WSN. In order to detect the faulty sensor nodes in WSN, HSA bases its algorithms on metaheuristics and employs a probabilistic search focused method. The correlation and energy values of neighboring nodes are included in each memory vector of the HSA for identifying the sensor node problems. The advantages are, Better node fault detection produces.

A naive Bayesian classifier-based FDS [17] is proposed to the detection of malfunctioning sensor nodes in WSN. In proposed scheme FDS took into account both the residual energy of sensor nodes and the sensed data when looking for network issues. In their model, the network is shown as an acyclic directed graph that displays the probability distribution. To detect and identify the SN faults, this method uses a Bayesian model. Data is initially sensed and transmitted to the cluster head by the SN itself. The cluster head is then in charge of making a decision at the global level to determine whether the sensor node is functioning normally or not. Since this proposed method
uses the two layers strategies, then it lead to main drawback is the longer detection time and consumes more energy during node fault identification.

The authors [18] proposed a feed-forward neural network based approach called GSPSO. It is an automated fault node identification technique that makes use of feed-forward neural networks that were hybrid and meta-heuristic trained. This proposed model is not suitable for transition fault detection because the transition fault state was unable to be captured by Neural Networks on its own. This method analyses both sensor and communication connection issues in addition to sensor faults. The limitations are, only homogeneous nodes can be used with this method.

The authors [19] used an rDFD approach to detect both the transient faults and the permanent faults in the nodes of WSN. This design stresses the temporal and spatial correlation among the sensed information gathered by the various SNs to detect the sensor node behaviour that is suspicious. This system uses nearby nodes' confidence levels to determine if a node is faulty or normal. This approach makes advantage of using variance and mean value of most newly obtained sensed information from its neighbours to maximizes the detection accuracy. Moreover, the proposed model increases the accuracy of defect detection while lowering the number of messages sent between the base station and the SNs. The limitations of this approach, this scheme results in unstable output data and results in transient effects.

The authors [21] addressed the various classification systems that have been researched for sensor fault identification. This work utilizes LEACH, one-class SVM, SVM and fuzzy algorithms for faulty node detection. A novel FCS-MBF is suggested to maximize the lifespan of wireless network and precision of node fault detection based on its performance. In order to increase the network's energy efficiency, it chooses the supercluster head according to its energy level.

3 Proposed Sensor Node Fault Detection Scheme

Figure 1 represents the proposed PHMM-FCSO architecture. The proposed system consists of five major modules namely nods initialization and region formation phase, sensor node fault diagnosis phase, sensor node fault identification phase, faulty sensor node localization phase and faulty sensor node restoration phase. The main aim of the node initialization and region formation phase is to initialize the nodes and form the region among the nodes. The next module is sensor node fault identification phase. The main aim of this phase is to diagnosis hardware sensor node fault detection and software node fault detection. The next module is faulty sensor node detection
phase. The primary goal of this phase is to discover and identify faulty sensor nodes using the Poisson Hidden Markov Model (PHMM). The next phase is localization phase. The primary goal of this phase is to locate the faulty SNs by employing fuzzy based chicken optimization (F-CSO) algorithm. The next phase is faulty sensor node restoration phase. The primary purpose of this network restoration phase is to remove and restore the faulty sensor nodes in the given region of WSN.

![Architecture of proposed system](image)

3.1 Node Initialization and Region Formation Phase

In the proposed system, let us consider the network with sensor nodes SN such that \( M = \{n_i: 1 \leq i \leq SN\} \). Each network node is independent and deployed at random over a \( Z \times Z \) square unit area. Every sensor node \( (n_i) \) has a distinct identifier \( (SN_i) \), is deployed in a location that may be inaccessible to humans, has coordinates \( (an_i, bn_i) \), and it satisfies the equation (1).

\[
0 \leq an_i, bn_i \leq Z
\]  

(1)

Each sensor node in the 1-hop communication range, designated as \( n_i \), communicates with its neighbouring nodes \( NN(n_i) \). \( R_t \) is range of transmission, homogeneous SNs is larger than or equal to the Euclidean distance of two nodes, \( n_i \) and \( n_j \), is larger than or equal to the transmission range \( R_t \) of homogeneous sensor nodes. The equation (2) gives the representation of the Euclidean distance of between \( n_i \) and \( n_j \) nodes:

\[
d(n_i, n_j) = \sqrt{(an_i - an_j)^2 + (bn_i - bn_j)^2} \leq R_t
\]  

(2)
The MAC protocol is used by sensor nodes in the proposed IEEE 802.15.4 standard to establish connections with one-hop neighbours that are located within the transmission range. The number of SNs within the range of transmission determines a node’s degree (i.e., one-hop neighbour sensor nodes). The degree of sensor nodes is calculated by equation (3) and (4):

$$\text{deg}(n_i) = \sum_{k=1}^{\text{NN}(n_i)} n_j$$  \hspace{1cm} (3)

where \(d(n_i, n_j) \leq R_t\) \hspace{1cm} (4)

Once the sensor node initialization is completed, the next step is region formation phase. In the proposed system, the region formation plays a major role and to achieve the energy gain with fault detection accuracy. To form the network into \(R\) regions, the network is divided into several separate regions with \(n\) nodes. The average node count \(\alpha\) in the region is calculated by using equation (5)

$$\alpha = \frac{n}{R}$$  \hspace{1cm} (5)

The ratio between a residual energy and its average of sensor node determines which nodes in the network are chosen to be members of each non-overlapping region in the network. The formation of a region is depends on the energy efficiency between the region members and the region head [26]. The high energy node is elected as the region head and there are no isolated nodes in the network. The region head is elected by employing received signal strength \((P_r)\) and that is calculated by using equation (6).

$$P_r = P_t G_t G_r \left(\frac{\sigma}{4\pi d}\right)^2$$  \hspace{1cm} (6)

where \(\sigma\) is the wavelength, \(P_r\) and \(P_t\) are the received and transmitted powers, \(G_r\) and \(G_t\) are the received and transmitted gains, and distance between the receiver and sender nodes is denoted as \(d\). The notation denotes the division of the entire network into various regions based on the values of \(R\), \(\alpha\) and RSS, and is represented by the notation \(R=\{R_1, R_2, \ldots, R_x\}\).

### 3.2 Sensor Node Fault Diagnosis Phase

The next step in the proposed system is SN fault diagnosis phase. The main objective of this phase is to diagnosis and identifies the faults of SNs deployed in the respective regions. In the proposed system, the sensor node fault diagnosis is subdivided into two major phases namely, hard and soft sensor node fault detection phase [27, 28]. In hard fault detection phase the faults related to
hardware components of the sensor are identified. In the soft fault detection phase, the faults related to the software components of the sensor nodes are identified.

3.2.1 Hard Sensor Fault Detection Phase

This algorithm identifies the occurrence of hardware based faults in the nodes of network. The status of every SN in the region is maintained by every sink or base station. The variable $Stat_{ij}$ is used to represent the status of sensor nodes. The base station transmits a fresh message in the entire region at each $t$ time duration. Each member node sends an acknowledgement (ACK) to the base station after receipt a message, the ACK must be delivered within the specified time frame, $t_{out}$. The SN is announced as faulty and its status is set to 1 if the ACK is not received for the predetermined amount of time. When a node experiences a transient fault, its status is changed to 1 if the time period $t_{out}$ exceeds the time $T_t$. When a node experiences an intermittent fault, its status is set to 1 if the time period $t_{out}$ exceeds the time $T_i$. When a node experiences a permanent fault, its status is changed to 1 if the time period $t_{out}$ exceeds the time $T_p$. The SN is announced as hard sensor fault, when

$$\sum_{i=1}^{NN(nk)} S_{ik} \geq \left[ \frac{NN(nk)}{2} \right]$$

is true. Then the status of each node can also be determined similarly. Algorithm 1 describes the hard fault detection process.

**Algorithm 1**

Number of deployed nodes $N=\{n_1, n_2, \ldots, n_k\}$

Input: neighboring node table $(NN_i)$, Timeout period $(T_{out})$, faulty node $FN_i$, threshold values $(TH_p, TH_i, TH_t)$, status of node $Stat_{ij}$

for every sensor node $n_i \in SN$ do

    for time period 1 to time period $t$ do

        each node $(n_i)$ broadcast the sensed information $m$ to its neighbor node $n_j \in NN(n_i)$

        then neighbor $n_j$ send ACK to $n_i$

        for each neighbor $n_j \in n_i$ do

            for each time period $=1, 2, \ldots, t$ do

                $N_i$ sends data and wait for ACK an $2xT_{out}$ period;

                if ACK received && within the time period then

                    failure count = count+1;

                else count$_j$ = count$_i$
end if  //ACK
end for
end for
for each sensor node $n_i$ calculates $T_{out}$
if $T_{out} > TH_t$ then
    Set $n_{ij} = 1$ and node is transient fault
else if $T_{out} > TH_i$ then
    Set $n_{ij} = 1$ and node is intermittent fault
else if $T_{out} > TH_p$ then
    Set node is permanent fault
else
    set $n_{ij} = 0$;
end if
end for
if $(TH_p \leq TH_i)$ then
    upon receive ACK from $n_i$
    set $n_j$ is non-faulty (microcontroller and transceiver is not faulty)
else if $(TH_p \geq T_1)$ then
    upon no ACK from $n_j$
    set residual battery fault in $n_j$
else if $(TH_p \geq T_2)$
    upon no ACK from $n_j$
    set microcontroller & transceiver is fault in $n_j$
else
    non faulty
end for
for each node $n_k \in N$ do
    compute status of $S_{ik}$
    if $\sum_{i=1}^{NN(nk)} S_{ik} \geq \left\lfloor \frac{NN(nk)}{2} \right\rfloor$ then
        the node $n_k$ treated as hard fault
else
    node \( n_k \) is treated as fault free
end if
end for

By using algorithm 1, the various kinds of hard faults are detected and it is informed to the base station for the future action.

3.2.2 Soft Sensor Fault Detection phase

Once hard fault phase is completed, and then the next step is to detect the fault based software. The soft faults in the WSN are grouped as permanent, intermittent and transient soft fault [9,22]. Initially, each SN \( n_i \) broadcasts its sensed sensor information to the sink or base station in the communication range of a region. Then the base station detects and identifies the presence of faulty sensor nodes in the region by comparing sensed data with the predefined threshold values \( \Theta_p, \Theta_i, \Theta_t \) to identify the type of faults. The fault status of each node can also be determined similarly. Algorithm 2 gives the steps involved soft fault detection phase.

**Algorithm 2**

Input : \( N=\{n_1, n_2, \ldots n_k\} \) neighboring node table \( NN_i \), time period \( T \), status of node \( \text{Stat}_{i}=0 \), predefined threshold values \( \Theta_p, \Theta_i, \Theta_t \) and count Boolean

for each \( n_i \in N \) do
    for each \( n_i \) broadcast sensed data \( m \) to its neighbor nodes \( n_j \in NN(n_i) \)
    end for
    for each time \( T=1 \) to \( t \) do
        for every neighbor node \( n_j \in NN(n_i) \) do
            if \( (\text{Stat}_{ij}-\text{Stat}_{ji})> \Theta \) then
                increment the status of node \( S_i' \)
            end if
        end for
    end for
    compute mean \( c_{fi} = \sum_{r=1}^{r} \frac{c_i'}{r} \)
if (cfi > Θp)
    then set Stat_{ij} = 1
    declare n_i is declared as permanent soft fault
if (cfi > Θt)
    then set Stat_{ij} = 1
    declare n_i is declared as transient soft fault
if (cfi > Θi)
    then set Stat_{ij} = 1
    declare n_i is declared as intermittent soft fault
else
    declare node n_i is node without fault
end if
end for

3.3 Sensor Node Fault Identification Phase

The next phase of the proposed system is sensor node fault identification stage. The primary purpose of this stage is to detect and identify the sensor hard fault and sensor soft fault present in WSN. The network’s nodes are denoted as \{n_1, n_2, \ldots, n_k\} where k indicates the maximum number of SNs in the network. The regions formed from that network is written as R = \{R_1, R_2, \ldots, R_x\}, where x is number of regions formed. The damaged region of the network is determined by using the Poisson distribution. Each region and its nodes can denoted as R_x = \{n_{ij}, \ldots, n_{qm}\}, where qm indicates the region number and total number of nodes in that particular region.

In the proposed system, Poisson distribution is used to determine the faulty nodes in the region by using the standard deviation and mean of each node. The mean of particular node in the given region is calculated from the sensed data of each neighborhood sensor nodes and the standard deviation of sensed data. The mean of particular node in the given region is calculated by using equation (7) and

\[
\text{Mean}_i = \frac{1}{R_i} \sum_{j \in R_i/N_i} M_j
\]  

(7)

Standard Deviation of the nodes in the particular region is calculated by using equation (8).

\[
\text{SD}_i = \frac{1}{R_i - 1} \sum_{j \in R_i/N_i} M_j - \text{Mean}_i
\]  

(8)
The each region’s mean is computed by using equation (9).

$$\text{Mean} = \frac{1}{q} \sum_{j \in R/N} \text{Mean}_i$$  \hspace{1cm} (9)

### 3.3.1 Faulty Node Detection using Poisson Hidden Markov Model (PHMM)

The proposed fault detection system Poisson Hidden Markov Model (PHMM) [25] is employed for detecting the faulty sensor nodes by computing the active of nodes using the Poisson distribution. The model equation for Poisson distribution for the nodes is defined by the equation (10)

$$y_i = \text{Mean}_i + \xi_i$$  \hspace{1cm} (10)

where mean$_t$ is the sum of mean calculated by the observed value of each sensor and $\xi_t$ is the residual error and it is constant variance which is normally distributed as random variables. The probability mass function of $y$ is computed by using equation (11)

$$P(y = y_i | \text{Mean}_i) = \frac{e^{\text{Mean}_i} \text{Mean}_i^{y_i}}{y_i !}$$  \hspace{1cm} (11)

where probability of Poisson distributed $y$ with Mean$_t$.

Considering k state Markov process that is to be assumed in some state $j \in [1,2,3,\ldots k]$ which is shown in equation (12)

$$y_i = \text{mean}_y + \epsilon_i \text{ when } s_t = j$$  \hspace{1cm} (12)

The exponentiated mean of Poisson Hidden Markov model is computed by using equation (14)

$$\text{Mean}_{ij} = e^{\text{Mean}_i}b_t$$  \hspace{1cm} (13)

The probability predicted Hidden Markov model of observing $y_i$ at time $t$ is computed by using equation

$$P(y = y_i | \text{Mean}_i) = \sum_{j=1}^{k} \left[ \left( \frac{e^{\text{Mean}_i} \text{Mean}_{ij}^{y_i}}{y_i !} \right) P(s_t = j) \right]$$  \hspace{1cm} (14)

The Poisson Hidden Markov model has been used to detect the both software and hardware sensor faults in the cluster region of WSN. The proposed technique accepts input from various original datasets, returns the fault-injected dataset, and then passes the dataset to determine the accuracy of the fault node detection. The proposed Poisson Hidden Markov Model fault detection algorithm is described in algorithm 3.
Algorithm 3: Fault detection algorithm using Poisson Hidden Markov Model

Input: number of deployed nodes $N = \{n_1, n_2, \ldots, n_k\}$, number of regions in the network $R = \{r_1, r_2, \ldots, r_x\}$, number of nodes in the region $R_i = \{n_{ij}, \ldots, n_{qm}\}$

Output: Return faulty sensor node $N_{ij} \in R$

1. for all $r_1$ in region $r$ do
2. compute $SD_R$ for each region using equation 8
   - if ($SD_R(R_i)$ == large) then
     - $R_i$ = damaged region;
   - else
     - $R_i$ = fault-free region;
   - end if
3. end for
4. while ($R_i$ = damaged region)
5. for all $N_{ij}$ in $R_i$ do
6. compute $P$ for each node in region $R_i$ using equation 14
   - if ($P(N_{ij}) \approx 0$)
     - return $N_{ij}$=faulty node;
     - broadcast fault node information to rest of nodes in region $R_i$;
   - else
     - return $N_{ij}$=non-faulty node;
     - proceed normal operation;
   - endif
7. end for
8. end while

3.4 Faulty Sensor Node Localization Phase

The next phase is faulty sensor node localization phase. The main objective of faulty sensor node localization phase is to identify and localize [31, 32] the faulty sensor node from the given region. To identify and locate the faulty sensor node from the given region, the proposed system employs fuzzy based chicken swarm optimization algorithm. The location of the faulty sensor node is determined by Geographical representation of the network. The Fuzzy based Chicken Swarm
Optimization (F-CSO) provides a precise and efficient operation to identify the exact location of the faults.

### 3.4.1 Fuzzy based Chicken Swarm Optimization (F-CSO)

The bio-inspired chicken swarm optimization (CSO) algorithm [23, 24] is an advanced intelligent algorithm which represents the different behaviors related to chickens, hens and cocks in their process of food search. It is a random search algorithm that resembles the hierarchical structure and behaviour of the chicken swarm. The CSO approach has many subgroups, each subgroup in CSO, consists of several chicks, few hens and a rooster. The rooster has the highest fitness value and rooster act as leaders of each subgroups. The chicks have worst fitness values and it is randomly grouped into subgroup. The positions of each rooster, hens, and chicks represent solution to a problem.

The identities of the rooster, hens and chicks are determined by its fitness value (energy) and it is calculated by equation 16. The nodes with highest fitness value and called as roosters. The rooster numbers indicate the maximum number of subgroups in the network. The remaining nodes are referred to as hens, while the nodes with the least fitness value are referred to as chicks. In CSO few numbers of hens and several chicks are randomly formed as a subgroup. To establish relationships with the chicks, the predetermined number of hens is picked at random intervals. The Fuzzy based CSO (F-CSO) algorithm is proposed to identify the faulty sensor nodes location. The random factors rand and F are completely random, at the initial stage of iteration (the first few iterations) would have a large searching space in order to identify the global optimum as much as feasible. The general steps of F-CSO shown in the figure 2. Based on equations 22 and 23 the random factors rand and naggr have large searching range, which map the fuzzy values high, very high, and medium accordingly. When the Fuzzy based Chicken Swarm Optimization algorithm would have narrowing random factors medium, very low and low while running on its final stage of iterations (large iteration). When the optimization speed and iteration time is medium, we have mapped the fuzzy values as high, medium and low.
The position of each chicken is denoted by $X_{ijt}$ and it is obtained by calculating at the with $t^{th}$ iteration, $i^{th}$ chicken and the $j^{th}$ dimension, then $i \in \{1, \ldots, CN\}$, $j \in \{1, \ldots, DN\}$, $t \in \{1, \ldots, IN\}$, where
CN indicates the number of chickens (nodes), DN stands for the dimension number and IN represents iteration.

Rooster position
The location of the rooster is evaluated by using the equation (15)

\[ y_{i,j}^{t+1} = y_{i,j}^t * (1 + \Phi(0, \sigma^2)) \]  \hspace{1cm} (15)

where \( \sigma^2 = \begin{cases} 1, & \text{if } k \in [1, RN], k \neq i \\ \exp \left( \frac{f_k - f_i}{|f_i| + \epsilon} \right), & \text{if } k \in [1, RN], k \neq i \end{cases} \)  \hspace{1cm} (16)

where \( k \) is the maximum of roosters chosen at random, \( f_i \) and \( f_k \) are the ith and kth roosters fitness values of the, \( \epsilon \) is a constant and \( \Phi(0, \sigma^2) \) is a random number derived by Gaussian distribution function with an expectation of zero and variance (\( \sigma^2 \)).

Hen position
The position of hen is determined by using the equation (17)

\[ y_{i,j} = y_{i,j}(t) + H_1 \cdot \text{rand.}(y_{r1,j}(t) - y_{i,j}(t)) + H_2 \cdot \text{rand.}(y_{r2,j}(t) - y_{i,j}(t)) \]  \hspace{1cm} (17)

Here \( H_1 \) and \( H_2 \) are the learning factors, \( r_1 \) is the rooster index, \( r_2 \) is number of a rooster or hen that is chosen at random, and \( r_1 \neq r_2 \). Here \( H_1, H_2 \) is computed by using the equations (18) and (19)

\[ H_1 = \exp \left( \frac{f_{ih} - f_{r1}}{\text{abs}(f_i + \epsilon)} \right) \]  \hspace{1cm} (18)

\[ H_2 = \exp (f_{r2} - f_i) \]  \hspace{1cm} (19)

Chicken position
The position of chick is calculated by using the equation (20)

\[ y_{i,j}(t + 1) = y_{i,j}(t) + F \cdot (y_{m,j}(t) - y_{i,j}(t)) \]  \hspace{1cm} (20)

Here \( F \) is a random factor with a range between 0 and 2 and \( y_{m,j}(t) \) is the mother hen of chicks.

IF-THEN fuzzy rules are formulated using the fuzzy proposition. The random factors values can be adjusted by using the following set of rules

\[ R_n^{(r)} = \text{if } < F_1^{(r)} > \text{then } < F_2^{(r)} >, r = 1, 2, ..., 9 \]  \hspace{1cm} (21)

The random rand, naggr value are computed by using the equation
The general steps of the proposed system Fuzzy based Chicken Swarm Optimization algorithm is provided in Algorithm 4.

**General Steps of the FCSO Algorithm**

**Algorithm 4**

Input: CN, T, RN, MG, IN, and DN

Output: Localization of faulty node and optimization

1. Initialize number of number of chickens CN and its position;
2. Determine the each chicken’s fitness value and initialize iteration t=1;
3. while (t < MG) // MG is maximum generation
4. if (t % MG = = 0), then run the fuzzy systems to compute the number of chicken and random factors naggr and rand
   - i. Calculate rand and naggr by using equations (22) and (23) // fuzzification operation
   - ii. Compute fuzzy rule bases $R(r)rf$ in equation (20) and $R(r)n$ in equation (21) // fuzzy inference rule
   - iii. Calculate FL, Rand and CN and take the results as a new parameter to compute next generation; // defuzzification
5. Establish the hierarchies according to the fitness values of chickens, the node with the maximum value is elected as a rooster, node with the least value is elected as a chick, and rest are hens.
6. Establish subgroups equals to the number of roosters and establish the relationship between the hens and chicks of each subgroups.
7. end if;
8. Calculate the fitness values of chicks, hens and rooster and update the location of chicks, hens and rooster by using the equations (15), (17) and (20)
9. Update the best local and global position of each node in the network;

\[
\text{rand} = \begin{cases} 
\frac{1}{2\Pi^*\alpha} \exp \left( -\frac{\text{rand}^2}{2\sigma^2} \right), & \text{rand} \leq 1 \\
0, & \text{otherwise}
\end{cases} 
\]  

(22)

\[
\text{naggr} = \begin{cases} 
\frac{1}{2\Pi^*\alpha} \exp \left( -\frac{\text{naggr}^2}{2\sigma^2} \right), & \text{naggr} \leq 1 \\
0, & \text{otherwise}
\end{cases} 
\]  

(23)
10. if \( t = \text{IN} \), then obtained better accuracy criteria, FCSO gives the final result; else repeat step 4 to 9.
11. end while;

### 3.4.2 Fuzzy rule mapping for adjusting chickens

Table 2 displays the fuzzy rules mapping for modifying the number of chickens (nodes) used by the fuzzy inference system. The fuzzy inference systems take two parameters optimization speed and chicken aggregation to adjust the number of subgroup member nodes (or "chicks") in the network.

#### Table 2 Fuzzy rule mapping for adjusting chickens

<table>
<thead>
<tr>
<th>S.no</th>
<th>Fuzzy rules for adjusting chickens</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IF (Optimization speed is Low) and (Chicken Aggregation is Low) THEN (number of chicken is Medium possible)</td>
</tr>
<tr>
<td>2</td>
<td>IF (Optimization speed is Low) and (Chicken Aggregation is Medium) THEN (number of chicken is High possible)</td>
</tr>
<tr>
<td>3</td>
<td>IF (Optimization speed is Low) and (Chicken Aggregation is High) THEN (number of chicken is Very High possible)</td>
</tr>
<tr>
<td>4</td>
<td>IF (Optimization speed is Medium) and (Chicken Aggregation is Low) THEN (number of chicken is Low possible)</td>
</tr>
<tr>
<td>5</td>
<td>IF (Optimization speed is Medium) and (Chicken Aggregation is Medium) THEN (number of chicken is Medium possible)</td>
</tr>
<tr>
<td>6</td>
<td>IF (Optimization speed is Medium) and (Chicken Aggregation is High) THEN (number of chicken is High possible)</td>
</tr>
<tr>
<td>7</td>
<td>IF (Optimization speed is High) and (Chicken Aggregation is Low) THEN (number of chicken is Very Low possible)</td>
</tr>
<tr>
<td>8</td>
<td>IF (Optimization speed is High) and (Chicken Aggregation is Medium) THEN (number of chicken is Low possible)</td>
</tr>
<tr>
<td>9</td>
<td>IF (Optimization speed is High) and (Chicken Aggregation is High) THEN (number of chicken is Very Medium possible)</td>
</tr>
</tbody>
</table>

### 3.5 Faulty Sensor Node Restoration
The next phase is the faulty sensor node restoration approach demonstrates the node relocation procedures. Each node in the network will broadcast hello messages with a transmission range to inform other nodes of its position. In order to acknowledge one another, each node exchanges data such as its ID and current position (ACK). Moreover, every node in the network transmits the sensed data within the broadcasting range on occasion. This broadcast message is used to identify the failing nodes in that region. If there is no broadcast message, the presence of the indication of failure nodes is assumed. By doing so, each node refreshes its list of neighbours and initiates the network's mobility process. Algorithm 5 gives the steps involved in faulty sensor restoration phase.

**Algorithm 5**

Input : N set of Sensor Nodes $n_i \in N$, NN neighbor nodes $NN_i$, $d$ distance $s_i -$ sink node

Output : Faulty node restoration

begin:
check if $(node, energy < TH)$
transmit (node ID of a “Rooster”)
check each “Chicken Nodes” energy level
designate the maximum energy nodes as “recovery node”
send message ("recovery node")
generate (restore plan based on energy level)
for each $n_i \in N$ do
    $Ds = |s_i - n_i|$
    if $Ds \geq d/2$ 
        while $Ds > d/2$ do
            $s_i = s_i + Ds$ // Keep moving towards sink node(Rooster)
            $Ds = |s_i - n_i|$ //update the new distance to sink
        //notify position of neighbors via HeartBeat message
        for each $n_i \in NN$
            $nn_i = neighbor(N, n_i)$
            if $n_i$ detects that $nn_i$ has failed then
                $n_i$ broadcast HeartBeat message with a range equals to $d$
if \( n_i \) receives ACK(\( n_m \)) then
    update the position of neighbor node
else //No ACK
    take recovery action
broadcast(restore plan)
end if

4. Results and Discussion

The network simulator NS2 is used to implement the proposed system Poisson Hidden Markov Model (PHMM) and fuzzy rule based chicken swarm optimization (F-CSO) algorithm under windows environment. On a wireless sensor network, the proposed PHMM-FCSO model is evaluated using the parameters enlisted in Table 3. The effectiveness of proposed system (PHMM-FCSO) is compared with four existing models HAS [4], OEPO [20], FCS-MBF [21] and GSPSO [18].

Table 3: Parameters used for simulation

<table>
<thead>
<tr>
<th>Simulation Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor nodes</td>
<td>100 – 500 nodes</td>
</tr>
<tr>
<td>Sensing range of the nodes</td>
<td>100 m - 200m</td>
</tr>
<tr>
<td>Sensor node’s initial energy</td>
<td>0.5J</td>
</tr>
<tr>
<td>Sensing range</td>
<td>30 m</td>
</tr>
<tr>
<td>Data rate</td>
<td>250 kpbs</td>
</tr>
<tr>
<td>Size of packets</td>
<td>1024 bytes</td>
</tr>
<tr>
<td>Individual node’s sensing range</td>
<td>36 m</td>
</tr>
<tr>
<td>Maximum iterations</td>
<td>100</td>
</tr>
</tbody>
</table>

The effectiveness evaluation of the PHMM-FCSO is compared with existing models by using three important metrics namely
- Fault node detection evaluation
- QoS parameters of the network
- Localization improvement

4.1 Fault node Detection Evaluation
In this section the performance evaluation of fault node detection in the network using the throughput, fault detection accuracy, false positive and false alarm rates.

**4.1.1 Fault Detection Accuracy**

Figure 3 show the graphical representation of the fault detection accuracy. From the comparative analysis graph, increasing number of sensor nodes involved in simulation results in minimization of detection accuracy. However, for nodes 300, comparative analysis shows the resulting accuracy of 89.5% for the proposed system and 86.46% (OEPO), 83.03% (HAS), 79.79% (GSPSO) and 76.36% (FCS-MBF) for existing system respectively. From this result, the accuracy of fault detection of PHMM-FCSO was observed to be comparatively better than the state of the art methods. The proposed work significantly detects faulty SNs in the network. The use of a suitable fault detection method, like PHMM in fault detection, is the primary factor in achieving this improved accuracy. The proposed model could identify both hardware and software sensor faults, hence the performance of detection accuracy is greater to the existing works. This in turn results in the improvement of fault detection accuracy using the proposed system by 3 % compared to OEPO, 6 % compared to HAS, 10% compared to GSPSO and 13 % compared to FCS-MBF.

![Fault Detection Accuracy](image)

**Figure 3. Comparative analysis of FDA**

**4.1.2 False Alarm Rate (FAR)**
Figure 4 show the graphical representation of the fault alarm rate. From the comparative analysis graph, increasing number of sensor nodes involved in simulation results in maximization of false alarm rate. However, for nodes 300, comparative analysis presents the resulting false alarm rate of 17% for the PHMM-FCSO and 20.35% (OEPO), 23.95% (HAS), 29.01% (GSPSO) and 32.96% (FCS-MBF) for existing system respectively. From this result, the false alarm rate of proposed system was observed to be comparatively better than the existing methods. The key reason for attaining this minimum False Alarm Rate is the proper implementation of an optimization algorithm like F-CSO. The Fuzzy based Chicken Swarm Optimization algorithm yields minimum false alarm rate and it predicts the faulty sensor nodes very accurately. Moreover, F-CSO balances the exploration and exploitation in creating new hierarchy and it enables the searching entire region to attain the optimal solutions. This in turn results in the improvement of false alarm using the proposed system by 3 % compared to OEPO, 6 % compared to HAS, 12% compared to GSPSO and 16 % compared to FCS-MBF.

4.1.3 False Positive Rate

Figure 5 show the graphical representation of the fault positive rate. From the comparative analysis graph, increasing number of faults percentage in sensor nodes involved in simulation results in better false positive rate. However, the comparative analysis presents the resulting false positive rate, the fault probability for PHMM-FCSO system is very low by comparing with all other existing works OEPO, HAS, GSPSO and FCS-MBF respectively. From this result, the false
positive rate of proposed system was observed to be comparatively optimum than the existing methods. The proposed work achieves very low rate of false positive than the other four existing works. The key reason behind for attaining this minimum false positive rate is, the proposed work identifies both the hard and soft sensor faults.

![Fault Positive Rate Graph](image)

**Figure 5.** Comparative analysis of False Positive Rate

### 4.1.4 Throughput

Figure 6 show the graphical representation of the through. From the comparative analysis graph, increasing number of sensor nodes involved in simulation results in maximization throughput. However, for nodes 300, comparative analysis shows the resulting better throughput for the proposed system comparing with existing works OEPO, HAS, GSPSO and FCS-MBF for respectively. As a result, the proposed system's throughput was shown to be significantly superior to the existing approaches. This is because of the implementation of F-CSO model which maintain the fault free region of nodes during the data transmission.
4.2 QoS Parameters of the network

In this section the evaluation of fault node detection in the network using the Quality of Service parameters such as energy consumption, network lifetime and delay.

4.2.1 Energy Consumption

Figure 7 show the graphical representation of the average residual energy. From the comparative analysis graph, increasing number of sensor nodes involved in simulation results in minimization of residual energy. However, for nodes 300, comparative analysis shows the resulting accuracy of 70.05% for the proposed system and 62.96 % (OEPO), 60.03% (HAS), 55.99% (GSPSO) and 52.06 % (FCS-MBF) for existing system respectively. From this result, the remaining residual energy of proposed system was observed to be comparatively optimum than the existing methods. Because the proposed work owing the robust fault management strategy and can able to detect the premature death of a sensor node. The proposed work avoids such premature death of sensor nodes at the earliest and battery unit failure. Moreover, the proposed work prolong the lifetime of network. This in turn results in the improvement of energy efficiency using the proposed system by 7 % compared to OEPO, 10 % compared to HAS, 14% compared to GSPSO and 18 % compared to FCS-MBF.
Figure 7 Comparative analysis of Energy

4.2.2 Delay

Figure 8 show the graphical representation of comparative analysis of delay. From the comparative analysis graph, increasing number of sensor nodes involved in simulation results in maximization of delay. However, for nodes 300, comparative analysis shows the resulting minimum delay is observed comparing existing system. In the proposed work, the base station handles the fault detection phase while each sensor node provides its corresponding data to the region head for procession. In comparison to the other four existing works, this proposed work uses less energy and has a lower delay for fault detection. As a result, as nodes increased along with the probability of a fault, the performance of the proposed work was better than the existing works.
4.2.3 Network Lifetime

Figure 9 Comparative analysis of Network Lifetime

Figure 9 show the graphical representation of comparative analysis of network lifespan. From the comparative analysis graph, increasing number of sensor nodes involved in simulation results in minimization of network lifetime. However, for nodes 300, comparative analysis shows the resulting of proposed system was observed to be comparatively optimum than the existing...
methods. The proposed work considerably increases the network lifetime by maintaining one-hop neighbour node information to identify the faulty sensor nodes.

5. Conclusions and Future Work

An effective sensor node failure detection technique using the Poisson Hidden Markov Model (PHMM) and the Fuzzy based Chicken Swarm Optimization (F-CSO) is proposed in this work to handle these issues and achieve the best optimization in terms of QoS. Improved sensor node failure detection is provided by the suggested system, and as a result, better quality of service is achieved in terms of better fault detection accuracy, false positive rate, throughput, false alarm rate, energy consumption, network lifetime, and least latency. Additionally, the F-CSO offers improved localization following a network fault brought on by a sensor node. NS2 simulator is used to implement the suggested task while using a variety of simulation parameters. According to the simulation results, the suggested work is more effective than the current state-of-the-art systems in terms of fault detection accuracy, fault positive rate, throughput, false alarm rate, delay, network lifetime, and energy consumption.

Compliance with Ethical Standards

Disclosure of potential conflicts of interest: There is no potential conflicts of interest among the authors.

Research involving human participants and/or animals: This article does not contain any studies with human participants or animals performed by any of the authors.

Funding: There is no funding available.

Competing Interests: There is no competing interests among the authors.

Author Contributions: Dr Santhosh Kumar SVN has deigned the algorithms and Mr Nagarajan Bhas carried out literature survey and perform simulation for our proposed system.

Data Availability: Data is available based on personal request.

References


