Abstract—In this work, a real time soft fault diagnosis model is proposed for wireless sensor networks (WSNs) using particle swarm optimization (PSO) based classification approach. The proposed model follows in three phases such as initialization, fault identification, and fault classification phase to diagnose the composite faults (combination of soft permanent, intermittent, and transient fault) in the sensor network. The faulty nodes are identified in the network based on Analysis of variance (ANOVA) method. The feed forward neural network (FFNN) technique with PSO learning method is used for classification of the faulty nodes. We evaluate our model by carrying out the tested experiment in an indoor laboratory environment.

Keywords—WSN; Composite Fault; ANOVA; FFNA; PSO.

I. INTRODUCTION

Wireless sensor network (WSN) is a collection of (few tens to ten thousand) autonomous sensor nodes working together to sense the surrounding environment. The sensor node consists of low specification, low cost, and low power consumption hardware components with limited resources. WSNs have been widely used in different applications such as health monitoring, military surveillance, environmental monitoring, home automation, and other commercial applications etc [1] [2]. WSN is intended to monitor and record conditions at human inaccessible or diverse locations, which causes faults in the sensor module. Due to behavior, the faults in WSN are classified as hard fault and soft fault [3] [4]. The WSN node is usually a combination of micro-control unit, transceiver, and sensor unit. The Hard fault is usually due to the microcontroller and transceiver failure, while the soft fault is usually due to the sensor unit failure. In case of hard fault, the sensor node does not communicate with neighbor nodes in the communication range, which is also called as permanent hard fault or crash fault [5]. The soft faults are classified as soft permanent fault, intermittent fault, and transient fault. The soft permanent faulty node communicates with the neighbor nodes, but each time gives unexpected outcomes [6]. The intermittent faulty node gives unexpected outcomes for random time instances. The transient faulty node gives unexpected results for a small or spike time instance and then normal results for other time instances. The erroneous results affected the whole network computation, so it is very much necessary to diagnoses of the composite faults (soft permanent, intermittent, and transient fault) in the WSN.

Many researchers have been proposed the fault diagnosis protocols for WSNs. The protocols are briefly discussed as follows. A distributed fault detection protocol is proposed by Panda et al. [7] [8] using the neighboring co-ordination method. These protocols are used statical method such as modified three sigma edit test and hypothesis testing for faulty node detection in the network. A Comparison based distributed fault diagnosis protocol is proposed by Sahoo et al. [9] to detect the soft permanent and intermittent faulty nodes in the network. A majority voting based and distributed fault detection protocol is proposed by Chen et al. [10] to detect the soft permanent fault. Xianghua Xu et al. [11] extends this protocol for both soft permanent and intermittent fault detection. Elhadef et al. [12] proposed a fault diagnosis protocol using back propagation neural network for hard and soft permanent faults in wireless interconnected network. A radial basis function neural network (RBFNN) based fault diagnosis protocol is proposed by Zhang et al. [13] for soft permanent faults. A principle component analysis (PCA) based fault diagnosis is proposed by Zhu et al. [16] to detect the soft permanent faults in the sensor systems. Kamal et al. [17] proposed a sequence based fault detection (SBFD) in WSNs. In this protocol, the Fletcher checksum and server side network path analysis are used for node failure, link failure, and node reboot detection in the network. Nitesh et al. [18] proposed a cluster based distributed fault detection algorithm to detect the permanent and transient faulty relay nodes in two-tier WSN. Swain et al. [19] [20] proposed a fault diagnosis protocol based on neural network approach to detect the composite faults in the WSN. Swain et al. [21] proposed a simultaneous detection of crash faults and cuts using graph theory concepts. Tang et al. [22] proposed a fault diagnosis protocol using neighborhood hidden conditional random field algorithm to determining the different faulty scenarios.

The previous existing fault diagnosis protocols for WSNs are considered different types of faults independently. In best of our belief, no protocol considers the combination of different soft faults such as: soft permanent, intermittent, and transient fault together for fault diagnosis.

The main contributions of the paper are stated as follows:

1) A real time fault diagnosis protocol in WSNs is proposed to diagnoses of different soft faults such as soft
permanent, intermittent, and transient fault together in the network.

2) The protocol uses a statistical mechanism called as Analysis of Variance (ANOVA) test for faulty node detection and feed-forward neural network (FFNN) approach with Particle swarm optimization (PSO) learning technique for faulty node classification in the WSN.

3) The protocol performance is evaluated using real testbed experiments in the indoor laboratory environment using a prototype.

The paper is organized as follows: section I presents the introduction and the related works. The proposed fault diagnosis protocol with various phases is presented in section II. Section III presents the testbed experiments and results of the proposed protocol. Finally, section IV concludes the paper.

II. PROPOSED FAULT DIAGNOSIS MODELING

The proposed model is described in three (3) phases such as: initialization, fault identification, and fault classification phase. Section A presents the assumptions of the model. The initialization phase is described in section B. Section C presents the fault identification phase and the fault classification phase is described in section D.

A. Assumptions

i Each sensor node in the network is homogeneous and static in nature.

ii The coordinator nodes are tested fault free nodes with GPS enabled in the network.

iii The coordinator nodes are higher computational power and transmission range than other sensor nodes.

iv The links associated with the sensor nodes are fault free in nature.

B. Initialization Phase

Initially, N number of sensor nodes are randomly deployed in an area of \( A_1 \times A_2 \). The node \( n_i \in N \) communicate with the neighbor node \( n_j \in N \), if the distance between the nodes is less than the transmission range of the nodes. The distance between two nodes \( d_{ij} \) is calculated by Eq. 1.

\[
d_{ij} = \sqrt{(n_{ix} - n_{jx})^2 + (n_{iy} - n_{jy})^2},
\]

where \( (n_{ix}, n_{iy}) \) and \( (n_{jx}, n_{jy}) \) are the Cartesian coordinates of the sensor nodes \( n_i \) and \( n_j \) respectively.

The tested fault free nodes are uniformly deployed in the network, called as coordinator nodes. The sensor nodes communicate with the coordinator nodes by multi-hop fashion. The coordinator nodes are connected to the base station. The sensor nodes deployment and the communication with the coordinator nodes is shown in the Fig. 1. After deployment of the coordinator nodes, the nodes broadcast hello messages in its communication range. Then each sensor node in the network calculates the strength of the receiving signal. According to the signal strength, the cluster members (sensor nodes) are formed a clustering with the cluster head (coordinator node). The coordinator nodes are stored all the information of the sensor nodes in its cluster region.

C. Fault Identification Phase

After initialization phase, we follow the fault identification phase, which identify the presence of faults in the network. Each sensor node sends their sense values to its coordinator node in the particular cluster region and the coordinator node identifies the faulty nodes in the network. The statistical mechanism such as Analysis of Variance (ANOVA) [23] [24] is used to detect the faulty nodes. The ANOVA test is used to analyze the differences between the actual sensor data and faulty sensor data. ANOVA test is based on two hypotheses testing such as: (i) \( H_0 \) = Null hypothesis, i.e. No significant difference between the means of the group data and (ii) \( H_1 \) = Alternative hypothesis, i.e. At least one difference among the means of the group data. The ANOVA test algorithm step is described as follows.

i Calculate the mean within each sensor node.

\[
\bar{\mu}_j = \frac{1}{m} \sum_{i=1}^{m} s_j,
\]

where \( \bar{\mu}_j \) is mean of the sensor node \( n_i \in N \) & \( s_j \) is the sensor values of node \( n_i \) such that \( j = \{1,2,\ldots,m\} \).

ii Calculate overall mean of the sensor nodes.

\[
\bar{\mu} = \frac{\sum_{j=1}^{N} \bar{\mu}_j}{N},
\]

where \( \bar{\mu} \) is the overall mean of the \( N \) number of sensor nodes.

Fig. 1: An overview of sensor nodes and coordinator nodes deployment
iii Calculate sum of squared differences between the nodes.
\[ sdb = \sum_{i=1}^{N} m(\bar{\mu}_i - \bar{\mu})^2, \]  
where \( m \) is the number of sensor data values per node. The degrees of freedom between nodes is:
\[ df_b = N - 1, \]  
where \( N \) is the number of sensor nodes. So the mean square value between the nodes is calculated as:
\[ msb = \frac{sdb}{df_b} \]  

iv Calculate the sum of squared differences within nodes.
\[ sd_w = \sum_{i=1}^{N} \sum_{j=1}^{m} (s_j - \bar{s}_i)^2, \]  
where \( \bar{s}_i \) is the mean of node \( n_i \) such that \( i = \{1, 2, ..., N\} \) and \( s_j \) is the sensor value of node \( n_j \) such that \( j = \{1, 2, ..., m\} \). The degrees of freedom within nodes is:
\[ df_w = N(m - 1). \]  
So the mean square value within the nodes is calculated as:
\[ ms_w = \frac{sd_w}{df_w} \]  
v Calculate the F-ratio.
\[ F_{ratio} = \frac{msb}{ms_w}. \]  
Then calculate F critical value \( F_{crit}(df_b, df_w) \) at the 5% significance level using \( F \) distribution table, where \( \alpha = 0.05 \).

vi The \( F_{crit}(df_b, df_w) \) value compares with \( F_{ratio} \) value. In this case, if \( F_{ratio} > F_{crit}(df_b, df_w) \) then the \( H_0 \) or the null hypothesis is rejected and concludes that the sensor values of the nodes are different. Otherwise, if \( F_{ratio} < F_{crit}(df_b, df_w) \), then it satisfies the null hypothesis and concludes that there is no significant difference between the sensor data of the nodes.

The cluster head (coordinator node) is performed ANOVA test between its own sense data and the other nodes sense data in its region. According to the results of ANOVA test, the coordinator node is identified that either the faulty nodes present in its cluster region or not. Then post-hoc analysis is carried out between the coordinator node mean and other sensor nodes mean in their region. In this analysis, the mean difference between coordinator node and sensor node is compared with the standard error of the node. The mean difference between two nodes \( n_i, n_j \in N \), is calculated as \( mda_{ij} = |\mu_i - \mu_j| \) and standard error of node \( n_i \) is calculated as \( sde_{i} = \frac{s_i}{\sqrt{m}} \), where \( s_i \) is the standard deviation of the node \( n_i \) and \( m \) is the number of sensor data values. The sensor node values are different from the coordinator node values, when the mean difference is more times the standard error. So it concluded that, the sensor node is faulty in nature. The condition, \( |mda_{ij} - sde_{i}| > \delta \) holds good, where \( \delta \) is the threshold value & dependsents upon the application. In this phase, coordinator node detects the faulty nodes & follow the fault classification phase.

D. Fault Classification Phase

In this phase, the faulty nodes are classified using the feedforward neural network (FFNN) with PSO learning technique [25] [26]. The faulty nodes detected in the fault identification phase passed to the fault classification phase for classification. In this phase, the FFNN is trained by some previous faulty node sensor values and then classified the faulty nodes at a given point of time. The FFNN architecture is shown in the Fig. 2. Generally, there are three layers in FFNN architecture such as: input layer, hidden layer, and output layer [27]. Initially, sensor node values are input to the input layer and the output to the input layer is input to the hidden layer. The output of the input layer is calculated by Eq. 11.
\[ o_{in} = b_1 + \sum_{j=1}^{m} s_j w_{ij}, \]  
where \( s_j \) is the sensor value of node \( n_i \), such that \( i = 1, 2, ..., N \) and \( j = 1, 2, ..., m \). \( w_{ij} \) is the random weight associated with input to hidden layer, and \( b_1 \) is the bias associated with input to hidden layer. The output to the hidden layer is \( f(o_{in}) \), where \( f \) is the sigmoid activation function is defined in Eq. 12.
\[ f(y) = \frac{1}{1 + e^{-y}} \]  
The output of the output layer is calculated by Eq. 13.
\[ out_o = f\{b_2 + \sum_{i=1}^{hidden} (f(o_{in}))w_o\}, \]  
where \( b_2 \) is the bias associated with hidden to output layer, \( w_o \) is the random weight associated with hidden to output layer, and \( f \) is the sigmoid activation function. Then the squared error is calculated by output value and target value of the corresponding input instance, which is defined in Eq. 14.
\[ error = \frac{1}{n} \sum_{i=1}^{n} (c_i - t_i), \]  
where \( c_i \) is the computed output and \( t_i \) is the target output of \( i^{th} \) instance. The error is reduced by PSO learning technique. The weights \( (w_i, w_o) \) and biases \( (b_1, b_2) \) values are updated using the learning technique.

The objective of training is to find minimum error, i.e. mean squared error (MSE). PSO is a simple algorithm for
implementation, derivative free, easily parallelized for concurrent processing, and very few algorithm parameters. PSO is a very efficient global search algorithm and search optimal weights, so that MSE gives best. Therefore, PSO is chosen as learning algorithm among wide spectrum of meta-heuristics for our proposed model.

Particle swarm optimization (PSO) is a stochastic search method introduced by Kennedy and Eberhart [28] [29]. PSO is initialized with a random population called as swarm and assigned randomized velocity to each potential solution called as particles. In PSO, each particle considered two values such as: (i) The positions or coordinates of the best fitness value \( p_{bst} \) and (ii) The location of the overall best fitness value considering the whole swarm \( g_{bst} \) [30]. The fitness function is followed the Eq. 14. Let the position and velocity of particle \( i \) is represented by following vectors as: \( x_i = \{x_{i1}, x_{i2}, ..., x_{in}\} \) and \( v_i = \{v_{i1}, v_{i2}, ..., v_{im}\} \) respectively. The position and velocity of the particle \( i \) at time instance \( \tau \) denoted as \( x_i(\tau) \) and \( v_i(\tau) \) respectively. The position of the individual particle is updated by Eq. 15.

\[
x_i(\tau + 1) = x_i(\tau) + \delta \times v_i(\tau + 1) \tag{15}
\]

where \( \delta \) is a random number. The velocity of the particle is updated by Eq. 16.

\[
v_i(\tau + 1) = \omega \times v_i(\tau) + c_1 \times (p_{bst} - x_i(\tau)) + c_2 \times (g_{bst} - x_i(\tau)) \tag{16}
\]

where \( \omega \) is the inertia weight, \( c_1 \) and \( c_2 \) are two positive constant, and \( \theta_1 \) and \( \theta_2 \) are random numbers between interval 0 to 1. The PSO based learning procedure described as follows.

i. Initialize the population of particles with random positions and velocities.

ii. Calculate fitness value for each particle.

iii. Compare the current fitness with previous best fitness value of particle’s. If the current fitness better than the best fitness then sets \( p_{bst} \) value as current fitness value and \( g_{bst} \) location equal to the current location.

iv. Compare the current fitness with overall previous best fitness. If current value is better than \( g_{bst} \), then set \( g_{bst} \) value and location.

v. Update the inertial weight \( \omega \).

vi. Update the velocity \( v \) and position \( x \) for each particle using Eq. 15 and 16.

vii. The procedure is repeated until for sufficient good fitness value.

The training phase is conducted by taking the historical data of sensor nodes with different faulty behavior. In the training phase, no message is transferred between sensor nodes in the network, so the overhead of the network is less. After the training phase, the real time sensor values are processed and the trained neural net will give the behavior of the faulty nodes. In this way, the classification of different faulty nodes was performed by neural network model.

order of complexity of the proposed model: In the initialization phase, the of sensor node \( n_i \in N \) sends \( m \) number of messages to the coordinator nodes. So the complexity is \( O(N \times m) \simeq O(N) \), where \( m \) is a constant value. In the fault identification phase, the algorithm depends upon the number of sensor nodes \( N \) and the number of sensor values per node \( m \). So the runtime complexity is \( O(N \times m) \simeq O(N) \), where \( m \) is constant. In the fault classification phase, the FFNN computation depends upon the number of input neuron \( N_i \) (same as number of sensor nodes), number of hidden neurons \( H_i \), and number of output neurons \( O_i \). The forward computation is required \( O(N_i \times H_i) \) times and next computation is required \( O(H_i \times O_i) \) times. So the total computation is required as \( O(N_i \times H_i)), because N_i \times H_i \gg H_i \times O_i \). The \( O(N_i \times H_i) \) computation will carry out for each agent. So the overall complexity multiply by the population size \( n_{pop} \), which is represented as: \( O(n_{pop} \times N_i \times H_i) \). The total complexity of the proposed model is calculated by summation of each phase complexity. Therefore, the total complexity of the proposed model is \( O(N \times m) + O(N \times m) + O(n_{pop} \times N_i \times H_i) \simeq O(n_{pop} \times N_i \times H_i) \).

III. TESTBED EXPERIMENT

The testbed experiment has been performed in the indoor laboratory setup. So, the proposed model is validated in the real time environmental setup. The sensor node is designed by Arduino uno embedded board, DHT11 temperature sensor, ATMega 8-bit micro controller, MRF24J40MA transceiver, and 7.4 volt 2200mAh Li-ion battery. Fig. 3 shows the sensor node module.

In this experiment, 5 sensor nodes are deployed in a region of \( 3 \times 3 \ m^2 \) and one coordinator sensor node is attached to the base station within 5m. The 5 sensor nodes and the coordinator node are presented in the communication range of each other. Fig. 4 shows the sensor node deployment setup.

The transmission power is set to -16.5 dBm for approximately communication range 5m and receiver within the
distance of 5m. The receiving threshold value is set to -85 dBm. The experimental parameters and their values are shown in Table 1. The sensor data are collected from the nodes in the normal daytime between 9:00 AM to 4:00 PM. It observed that the temperature values varies in between 26°C to 30°C. So, we set the minimum and maximum temperature value as 26°C and 30°C respectively. The temperature value exceeds the minimum and maximum value is considered as a faulty value. Initially, the sensor nodes are fault free, then we added random value with sensor node sensing value to make it faulty node. In this experiment, for soft fault 95% to 100% values are considered as wrong, intermittent fault 30% to 50% values are wrong in regular time instances, and transient fault 10% to 20% values are wrong in random time instances.

The proposed model runs on base station having Matlab 2013a, which is connected to the coordinator node. All the sensor nodes send the data to the coordinator node to identify the faulty node and then classify the faulty behavior. Fig. 5 represents the neural network computation of sensor nodes by the coordinator node. The performance of this protocol is observed by the metrics such as: detection accuracy (DA), false alarm rate (FAR), false positive rate (FPR), and false classification rate (FCR). The DA is the ratio between a total number of faulty nodes detected as faulty to the total number of faulty nodes present in the network. The FAR is the ratio between fault free nodes detected as faulty to total fault free nodes present in the network. The FPR is the ratio between faulty nodes detected as fault free to total faulty nodes present in the network. The FCR is the ratio between a number of faulty nodes wrongly classified as faulty to total faulty nodes present in the network. Fig. 6 shows the impact on DA with increasing the faulty nodes. In this figure, increasing in the faulty node the DA decreases. It observed that, transient fault gives lowest DA than other faults because, the transient fault is not occurring periodically. Fig. 7 shows the graph between FAR and faulty node. In this figure, increasing the faulty node the FAR also increases and at last gives 0 because there is no fault free node present in that case. Fig. 8 shows the graph between FPR and faulty node. In this figure, increasing the faulty node the FPR also increases. In all these cases, the soft fault gives better performance than other two faults because, soft fault gives continuously faulty results to the environment each time interval. In case of transient fault occurs in small time interval and then disappears suddenly, which causes lower performance than other two faults. Fig. 9 shows the graph between FCR and faulty node. In this figure, increasing the faulty node the FCR also increases. In this figure, three types of faults are added gradually in the network, then the proposed protocol identified the faults and also classified their behavior. This figure is shown that increasing the faulty node the FCR also increases.

IV. CONCLUSION

In this work, a real time soft fault diagnosis protocol has been proposed for wireless sensor networks using PSO based classification approach. The proposed model diagnoses the different type of soft faults such as soft permanent, intermittent, and transient fault in the WSN. The proposed model works in three phases such as initialization phase, fault identification phase, and fault classification phase. The fault identification

---

### TABLE I: Experimental parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transceiver</td>
<td>MRF24J40MA</td>
</tr>
<tr>
<td>Range</td>
<td>55000m</td>
</tr>
<tr>
<td>Transmitter power</td>
<td>-31.5 dBm</td>
</tr>
<tr>
<td>Operating Frequency</td>
<td>2.405 GHz</td>
</tr>
<tr>
<td>Selected channel frequency</td>
<td>2,400 MHz</td>
</tr>
<tr>
<td>Modulation bit rate</td>
<td>Radio BWC 1.5 k</td>
</tr>
<tr>
<td>Receiver sensitivity</td>
<td>-90 dBm</td>
</tr>
<tr>
<td>Network size</td>
<td>3 × 3 m²</td>
</tr>
<tr>
<td>Number of sensor nodes</td>
<td>4</td>
</tr>
<tr>
<td>Number of coordinator node</td>
<td>1</td>
</tr>
<tr>
<td>Communicator range</td>
<td>2.4 GHz</td>
</tr>
<tr>
<td>Packet size</td>
<td>4 bytes</td>
</tr>
<tr>
<td>Packet receiving threshold</td>
<td>-85 dBm</td>
</tr>
<tr>
<td>Packet sending rate</td>
<td>1 packet/0.1 sec</td>
</tr>
<tr>
<td>Number of time tested</td>
<td>30</td>
</tr>
</tbody>
</table>

---

Fig. 5: Neural network computation of sensor nodes

Fig. 6: DA vs Faulty Node

Fig. 7: FAR vs Faulty Node
phase is modeled by statistical mechanism, i.e. ANOVA test to identify the faulty nodes. The fault classification phase is modeled based on FFNN based architecture with PSO learning technique to classify the faulty nodes in the network. The performance of the proposed protocol is measured by the performance metrics DA, FAR, FPR, and FCR. Experimental implementation and results show that, it is feasible to implement in the real life application of WSNs.

REFERENCES


https://en.wikipedia.org/wiki/F-test#One-way_ANOVA_example


