IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Instrumentation Mesure Métrologie
Vol. 19, No. 2, April, 2020, pp. 111-117
Journal homepage: http://iieta.org/journals/i2m

Enhancing Reliability by Detection of Software Fault in Wireless Sensor Network Using Distributed Approach

Mufassir Syed*, Mithilesh K. Dubey

School of Computer Application (SCA), Lovely Professional University, Phagwara Jalandhar, Jalandhar 44411, India

Corresponding Author Email: yaseen.11719244@lpu.in

https://doi.org/10.18280/i2m.190205

Received: 5 December 2019
Accepted: 17 January 2020

Keywords:
wireless sensor networks, sensor nodes component, reliability, Mann-Whitney U Test, software faults

ABSTRACT

The wireless sensor network comprises of the number of wireless sensor nodes, that senses the environment for information collection and forwarding collected information to the base station. It is done by multi or single-hop communications for getting some achievement in the environment. Because of multi-functional applications, sensor modules became erroneous by different outer and inward sources that prompt to failure of the network. In wireless sensor network automated fault tolerance and diagnosis of faults are important. For software dependability, software faults are significant risks. To study the software failure in this type of the network, we analyze the consistency of mitigation processes for fault or diagnostic methods. The diagnosis of the fault approach is proposed for faulty software in the wireless sensor network, the methodology consists of a few different stages like as initializations, detection of software faults, classification of faults and fault phases. We have used the Mann-Whitney U statistical tests for the software fault (Intermittent, Transient and permanent) detection. For assessment of the proposed diagnosis methodology the parameters like false positive rate, fault classification rate, fault probability, false alarm rate, and fault detection of the accuracy are considered.

1. INTRODUCTION

Wireless Sensor Network (WSN) comprises of various autonomous sensor nodes that are connected together to form a network and actually it is simply the collection of distributed and self-governing computing sensor devices that are used for tracking and controlling of physical environments [1]. Every sensor node in a WSN has potential capabilities of sensing and processing of data. In the industry for several years, WSN technology has received a great deal of attention. The self-setup, flexibility, fast implementation, self-configuration, and easy upgrading and also low operating costs make WSN ideally best suited for industrial use. In general, Industrial Wireless Sensor Networks (IWSN) can be implemented. Furthermore, it is used in a very critical situation, including oil pumps, ball bearing motors and engines [2]. As a result, major efforts have been made over the last decade to develop a range of industrial standards and open source solutions (e.g. Wireless-HART, IEEE-802.15.4 Zigbee, ISA100, Open-WSN) for IWSN applications, providing "ready to use" solutions to enhance the technology interoperability and maturity. A fault-tolerant IWSN is configured to provide the sink node with a continuous supply of information despite disturbances. Fault tolerance is an important aspect and it is the ability for detection and diagnosis of faulty nodes in way for assuring data reliability, energy saving, high data accuracy and prolonging the lifetime of WSNs.

The fault-tolerance system of WSN is designed in a way that provides the information of delivery to the sink node. Fault diagnosis's main work is to continue for identification of faulty nodes to mitigate and to keep the effect at control on the wireless sensor network operations. Faulty sensor nodes generate the erroneous data that should be restricted to enter into the networks for data transfer capacity utilization.

When the online process of diagnosis of sensor nodes is performed, it can be effective to manage the network. Few hazards like environmental and node failures can cause network partition, frequent change in topology and failure in communication. In WSN, permanent fault detection requires a single test; in contrast, repetitive testing for intermittent fault detection requires a discrete-time. The program has to be executing for a failure to occur. Failure is coined which is related to the program behavior. Failure is not like something a "bug" or more properly "fault". It includes performance deficiency attributes and excess response time. In the program, the fault is a defect when so ever is executed under a few particular conditions is causing a failure. Therefore, we can say that the source of more than one fault is a failure. This is what we are really referring to in general when we use the term "bug". Usually, WSN experiences two main categories of faults; which affecting the performance of WSN like data loss and system faults. On one side, in data-centric view faults which comprises such as gain, offset and stuck-at. On the other hand, such as environment, low battery, calibration is categorized into system-centric faults. The WSN fault is generally divided as soft-fault and hard-fault [3]. To separate the causes of the hard and soft faults, categorically it is difficult to do so. Several few cases in which soft fault and hard fault can be caused by physical malfunction such as communication issues, energy depletion, sensor movement in odd areas and in which links suffer from

fault are known as hard faults. Both the sensor nodes as well links can be impacted. Soft faults are mainly categorized such as

(a) Intermittent-fault
(b) Permanent-fault, and
(c) Transient fault.

In WSN to provide quality of service (QoS), detection of wireless faulty sensors and let all fault-free sensors to receive these faulty effects as it is highly recommended. In the presence of faults, this will make the network operational but with less performance. Fault diagnosis is proposed to draw accord among the fault-free sensors about the status of all faulty sensors in the system. It is dependable on the design of systems by isolating the faulty sensors from the network. This article considers the problem of software fault detection and mitigations in wireless sensor networks.

The previous existing fault diagnosis protocols for WSNs 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. Also by doing comparision with existing techniques which suffer from the limitation of poor recognition due to the dynamic nature of faults.

On the basis of threshold it is not an easy task to work with all these types of faults in WSN. Because of large deviations in incorrect transmission of data by different faulty sensors, the statistical approach for detection of software faults by using median, mode, mean and other schemes that are based on hypothetical approaches that becomes inappropriate. This work concentrates on fault node identification by using distributed WSN in which neighboring nodes are used for identification of faults by doing comparison of sensed data with the neighbouring nodes. The main aim of this work is to design less complex fault detection algorithm which can easily detect the faults in WSN by using distributed approach.

1.1 Challenges of WSN

The reliability of the software is one of the most widely accepted features throughout the field. It is the probability of fault-free operations for a specified time in a specific environment. It implies the probability that the software operates fault free for a predefined time for which it was designed, given that it was within limits of designing and that the last failure occurred at a specific time, which was already given.

Additionally, the contribution of technology allows a worthy concern to achieve the desired full services of a wireless sensor network. The unorthodox essence of any computer system can thus not reach saturated status when the desired task is completed. Therefore, evaluation of code defects and removing of software faults associated with WSN is highly necessary. Very few code deficiencies are chartered and identified in accordance with WSN. These include the Data Assignment Error, Software Build / Package and Merge Fault, Application of Technical Fault and Software Tests.

Following are the few factors in which fault detection method in WSNs faces few difficulties:

1) There are very restricted node level means and resources that compel nodes to use classifiers because they do not require complicated computation.

2) Sensor nodes are placed in hazardous and risky settings, e.g. indoors, war areas, tropical storms, earthquakes, etc.

3) The fault detection process should be accurate and quick to prevent any loss, e.g. the method should acknowledge the distinction between unusual and normal instances, so that it may contain losses in the event of acquisition of erroneous information which may lead to false repetition.

1.2 Major contribution

The primary objective of this work is to provide non-parametric statistical testing technology of nodes in sensor networks. We start by identifying the phases of a test cycle to complete this mission. Then we define some software fault sensor types such as intermittent and permanent software faults. This work consists of a few steps digital research method performed concurrently with sensor fusion. The first point is to test data. In the next stage, data is weighted accurately and evaluation will be done. Specific steps are then used to determine ideally the value of a particular dataset, keeping environmental properties into consideration. Finally, numerical percentage techniques were used to assess the confidence interval of assessing. The major outlined contribution in this work as follows:

I. Designing an algorithm for detection software fault by utilizing the time out response for nodes with a crash fault or hard permanent, using the Mann-Whitney U test where we can set a significant threshold.

II. For detection of intermittent and permanent fault classification by using voting-based neighbor majority factor to classify faulty sensor nodes with accuracy is proposed in this article.

The presented work is very much simple so that every researcher will understand the work. We have achieved the best results so far by using this simple algorithm.

2. RELATED WORK

The fault tolerance is a field that is having numerous recommendations & studies on software fault detection. Therefore, in a single article, it is not so easy to give a full view of the state of the art. Nevertheless, WSNs are given specific characteristics, which allow current fault taxonomies to be expanded or tailored to the nature of such networks. To the best of our knowledge, this paper is the first attempt at proposing a specific work on software faults for WSNs.

The existing literature on the analysis of the Wireless Sensor Network faults clearly shows that related work focuses on the attention in specific methods and fault tolerance algorithms rather than on proposing a comprehensive WSN fault detection. Most of the existing fault detection schemes for WSNs work with the assumption that sensors are either permanent faulty or fault-free. This assumption may not be true in real-time applications since in real systems more than 80% of the faults are intermittent faults. Sahoo and Khilar presented an algorithm for fault diagnosis by using a comparison of reading of neighbor's sensor for detection of permanent and intermittent faults in Wireless Sensor Network [4]. Chen et al gave a distributed localized fault detection algorithm for soft permanent faults by using majority voting from their neighbor sensor nodes in Wireless Sensor Network [5].

Swain et al. presented the model of graph theory for simultaneous detection of cut nodes in Wireless Sensor

Networks and also crash faults [6]. Panda and khilar gave an algorithm that is used for the detection of the Byzantine and software faults in WSN [7]. By this algorithm, performance is achieved and has given a good achievement in the WSN area but has difficulty in deciding the parameters. Banerjee et al gave that the parameter decision is difficult which is based on mean that is highly skewed error i.e. based on a single large.

By Wu *et al.* the first stage of fault detection scheme was given which does not provide fault detection in the early phase. This only discusses the detection in the second stage. When the majority of the nodes fail [8] use of majority-voting might fail. Mahapatro and Khilar [9] presented the online detection of fault which reduces the message overheads and energy detects both software faults and hardware.

In Wireless Sensor Networks Khan et al. [10] have utilized the TSK-Takagi Sugeno Kang fuzzy inference system (FIS) model for fault diagnosis in his protocol, in which training of each sensor node is done by FIS and sensor neighboring nodes data [11]. In a very recent study, a variant of the Fuzzy Interference System was investigated for fault diagnosis. Azzam et al. [12] developed a modified model that is a recurrent neural network (RNN) for diagnosis of WSN software faults. In such a protocol, for the learning phase, the recurrent neural network RNN is used for the neighbor's sensor data and previous output samples. This article proposed the generic software for fault detection scheme which will work for both intermittent faults and permanent in Wireless Sensor Networks.

In WSN there are various sensor nodes. Each sensor node's basic aim is to collect the data and transmit that data to the base station. Each sensor node is the collection of two types of components.

a. Hardware components.

b. Software components.

In the software components, there are various types of bugs. These software bugs may be called as the faults. These faults need to be detected because prevailing with these faults will deteriorate the performance of the network. The early detection can make the network function with its efficiency features.

These all types of software faults are right from the definition of the wrong requirements to the system and software faults. All these faults can be identified using performance matching criteria. The standard outcome is compared to the actual outcome. If the outcome matched then the software will be declared with no bug else software will be declared to have bugs. This means the test oracle has to be prepared. This test oracle is right from the requirements specification to the run time errors.

3. DETECTION PHASE OF FAULTS

At t^{th} instance time, in sensor module sm_i, sensor data $s_i(t)$ i = 1, 2, …, N is shown as in the Eq. (1) below:

$$S_i(t) = a_i(t) + \xi(t) \quad (1)$$

in which, actual data $a_i(t)$ at t^{th} time with $\xi(t)$ being the noise or error at instance time t. $S_i(t)$ is the sensed data of a module of the sensor.

$f(S_i(t)) \in N$ is representing DPF and is shown as the Eq. (2) below:

$$f(S_i(t)) = 1/\sqrt{2\pi\sigma_i} \exp(-s_i(t) - \mu_i)^{2.} /2\sigma^2 i \quad (2)$$

in which, standard deviation is σ_i & mean is μ_i.

For the fault-free sensor module data, $\sigma^2 i$ is variance expectation is the same and in case of faulty sensors, the variance is more than normal measurements of sensor. Collected sensor measurements of cluster members are sent to the cluster head in which the detection methodology executes. Mann-Whitney U statistical test is used for trying to diagnose the soft faults that are categorized such as intermittent, transient and permanent faults [13]. This test is non-parametric in nature, in which comparison is to be done by two samples that are independent. Mann-Whitney U test is performed on the basis of hypotheses (H0 and H1):

H0: Measurements of sensors are equal (samples are the same).

H1: Measurements of sensor are not equal (samples is different).

Hereby doing the assumptions fault free sensor of that cluster heads. In the cluster area, the comparisons are to be done, in which cluster head is comparing with the measurements of own sensor with every cluster member measurement.

Here we have discussed the Mann-Whitney U Test as:

In the cluster region, for each sensor modules $sm_i \in N$ having measurements of sensor {s1, s2, s3, ..., sτ} and the cluster head ch_j, j=1, 2, ..., m having sensor measurement {c1, c2, ..., ck}. The two group samples, such as {s1, s2, s3, ..., sk} and {c1, c2, ..., ck} are put into one set {s1, s2, s3, ..., sk, c1, c2, ..., ck}. By sorting the group in an ascending order & after assigning the rank that is numeric for each set of value, as the ranking starts with minimum values which is 1 and for unadjusted rank midpoint is assigned.

By using the neighbour cordination approach based software detection technique, ManWhitney Test is performed on the cordinator node between its own sensed data and other sensor nodes in a specific transmision range. According to the results of Manwhitney test, the coordinator node is identified that either the faulty nodes are present in its cluster region or not.

3.1 Soft fault diagnosis

Here $n_i \in N$ for every sensor node sends its sensed data sd_{ij} to one-hop communication for r rounds of time. In this scenario, every node n_i has to store the node ID & for sensing neighboring sensor nodes the value is maintained in a table NT_i. Every sensor node n_i is also maintaining a register variable status S_i & $Ci \in \{1, 0\}$, that is computed at every time instance i.e. Boolean flag variable.

For every node $n_i \in N$ and instances $\tau \in [1, r]$, for neighbors $n_j \in NEG(ni)$, after doing a comparison of sensed data values. After doing the comparison of the results, i.e. $sd_{ij} - sd_{ji}$, if it is exceeding the value of threshold θ after then for each testing by the neighbor node the status register, S_i is incremented. The threshold θ is actually dependent on application criticality intended of the sensor networks and additionally, it is user-specific. S_i Swain et al. is compared with $\lceil NEG(n_i)2 \rceil$, once the testing has finished. After doing

the comparison, for the majority of neighbors if the values exceed the threshold test, then instance time τ of Boolean is set as one (1),

$C_i(\tau)$=one(1), otherwise Ci(τ)=zero(0).

For each node n_i then we defined fault classification factor and is then computed.

3.2 Software fault algorithm

Initialization: Boolean Variable Ci E {0,1} Status register variable S_i= 0, Neighboring Table NT_i.
For node N_i belongs N **do**
For every time-instance = 1 to r **do**
Node ni sends data sd_{ij} to its neighboring set where $n_j \in$ NEG (ni)
End for
For every time period τ =1 to r **do**
For each neighbor $n_i \in$ neg (n, i)
do
If sd_{ij}-sd_{ji} > θ then increment status register variable s_i
End if
If $S_{i\tau}$ > ⌈NEG(n_i)/2⌉
$C^{\tau}{}_i$ =1;
Else
$C^{\tau}{}_i$ = 0;
Endif
End For
Calculate $Cf_i = \sum_{r-1}$
If δ P1 ≤ $Cf_i \leq \delta P_2$ Then
We can say n_i has soft permanent fault
else if $\delta_i 1 \leq CF_i \leq \delta i_2$ Then
n_i has intermittent fault
else-if $\delta_{i1} \leq CF_i \leq \delta i_2$ Then
n_i is transient faulty node
else
n_i is fault free node;
end-if
end for

Design

```
clear all
%Optimal node Probability oo
%to become cluster head
TDMA=1;
ETX=50*0.000000001;
ERX=50*0.000000001;
%Transmit Amplifier types
Efs = 10*0.000000000001;
Emp = 0.0013*0.000000000001;
%Data Aggregation Energy
EDA = 5*0.000000001;
sv = 0;
%temprature range
tempi = 50;
tempf = 200;
m = 0.0;
%\alpha
a = 1;
%maximum number of rounds
rmax = 100;
N = 100; %total number of nodes in the sensor field
M = 100; %width of the sensor field
Xm = M/3;
Ym = M/3; %height of the sensor field
N1= 10*N/100; %10% of total nodes N are deployed
uniformly in each grid
N5 = 20*N/100; %20% SNs are deployed in central grid i.e.
5thgw = 0.5; gw1=0.3; years = 100; sum = 0;
for i = 1:100
sum = sum+y(i); end
average = sum/i; if(average>36.23)
for i=1:100
average=average+gw;
end
disp('temperature after 100 years in the area=',average);
else
for i=1:100
average=average+gw1;
end
disp('temperature after 100 years in the area=',average);
end
'disp(average); 'disp(sum);
```

In this work, the sensor node ni ∈ N sends m number of messages to the coordinator nodes. So the complexity is O(N ×m) ~O(N), where m is a constant value. The algorithm depends upon the number of sensor nodes N and the number of sensor values per node. So the run-time complexity is O(N × m) ~O(N), where m is constant. The total complexity of the proposed model is calculated by doing the summation. The message complexity of this detection algorithm is O(n) and the number of bits exchanged to diagnose the WSN are O(n $\log_2$ n).

4. SIMULATION AND PERFORMANCE ANALYSIS

The NS-2.35 is used for simulations. Performance and evaluation of this proposed methodology is done. Here we have used 1000 nodes which are deployed in 1000 × 1000 m^2 area randomly in a range r of communication, nodes are communicating with their neighbors and during the simulation time the degree and number of sensor nodes vary from time to time. Table 1 shows the default settings of similar nodes.

Table 1. Parameter settings

Parameters	Values
Packet Size	32 bytes
No. of nodes	1000
Area of Network	1000x1000 m^2
Total simulation time	500 seconds
MAC-Protocol	IEEE-802-15.4
Source-rate	1 Pac./sec
Time of Carrier	350 seconds
Range of Communication	150 m
Channel-rate	250 ps

The proposed algorithm is very much effective in terms of performance measures such as false positive rate, false alarm rate and detection accuracy that are better than conventional methods already used for detection of faults in WSNs. Moreover, the software fault algorithm did not have a complex operation which makes it more energy efficient as well.

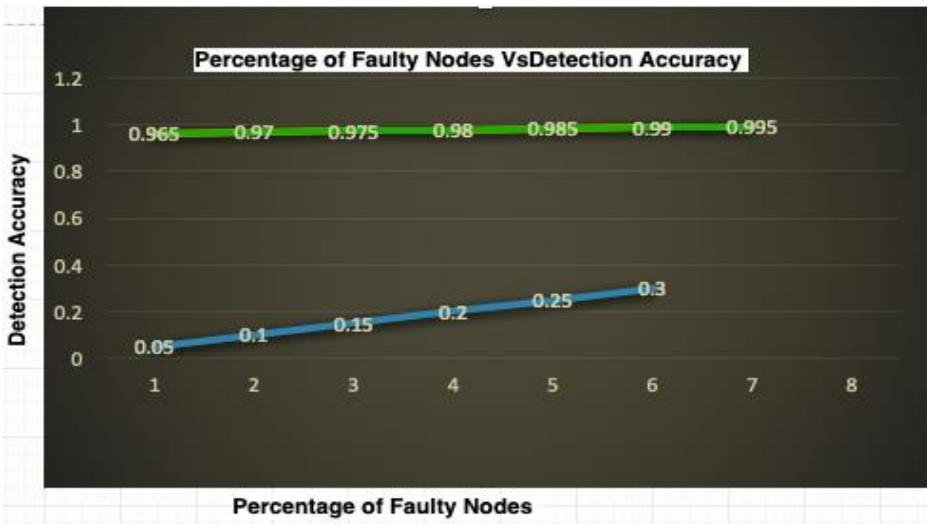

Figure 1. Percentage of faulty nodes vs detection accuracy

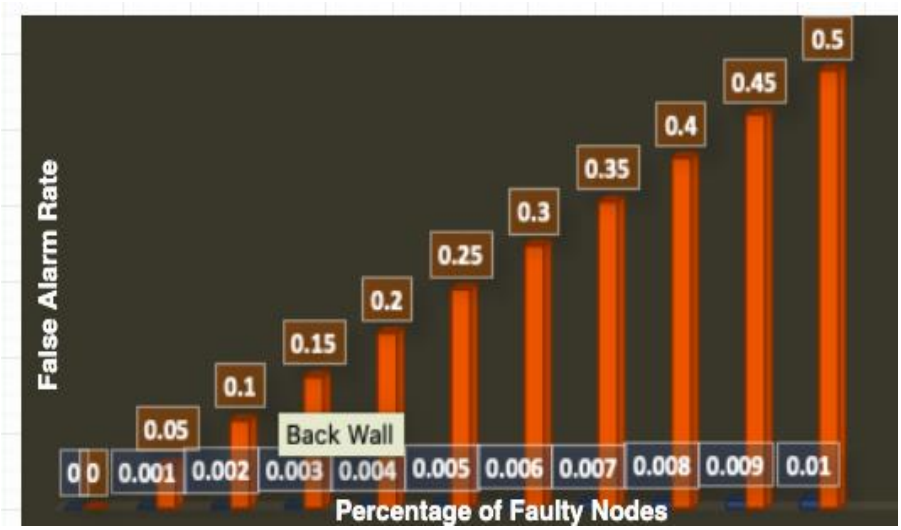

Figure 2. Performance analysis of fault detection

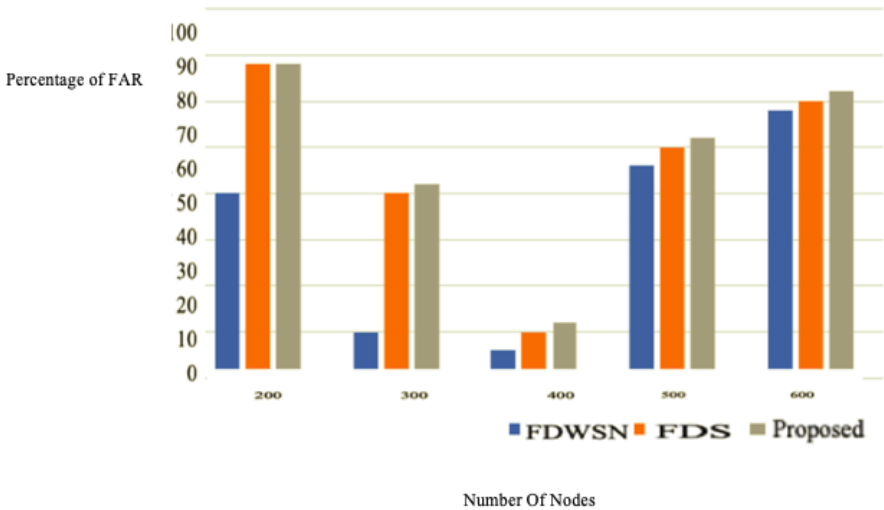

Figure 3. Comparison of proposed work with FDWSN and FDS [14, 15]

The performance is dependent on the wireless nodes deployed in the region. In the simulations process, we are assuming that the software faults are independent of each other. It is clear here that the proposed work is more efficient as compared with the previous schemes in terms of detection rate and packet delivery false alarm rate.

Figure 1 shows the graph between the percentage of the faulty nodes and the detection accuracy of these faulty nodes.

Figure 2 shows the performance analysis of fault detection.

By doing the comparison analysis with the data of the other researchers we got better results. In Figure 1, the accuracy increases after we have mitigated the faults in our scenario by using our methodology. Here we got the conclusion that our proposed method performance is much better than other researchers. In Figure 2, we have done the performance analysis of fault detection in which performance increases after the mitigating the software fault and removing faulty nodes in our proposed work.

Performance detection of faulty nodes is pictorially shown in Figure 3. As clearly shown, in the devised model the rate of detection is higher than FDWSN and FDS [14, 15]. Also, this scenario shows that the proposed model is more scalable with respect to the nodes in the whole network.

5. CONCLUSION

In this paper, we have proposed an algorithm that can detect various types of faults in wireless sensor networks. It was found that the proposed model was more efficient than the previous schemes in terms of detection rate, false alarm rate, packet delivery rate and packet removal rate. Also, these models were tested and evaluated with respect to different number of sensor nodes. Simulation results show that the performance of proposed scheme is better than FDS and FDWSN method in terms of DR, FAR, PDR and energy. As a direction for further research, a model cab is designed in future studies which can smartly control sensor nodes via communication with sink and supervise them through messages.

ACKNOWLEDGMENT

I am very much thankful to Dr. Mithilesh Kumar Dubey, Associate Professor, LPU and Dr. Gholam Reza, Assistant Professor, Islamic Azad University, Yazad Iran and I am very thankful to Er. Syed Farah Naz Shri Mata Vashinov Devi University Katra India for her valuable feedback and sugggestions.

REFERENCES

[1] Yick, J., Mukherjee, B., Ghosal, D. (2008). Wireless sensor network survey. Computer Networks, 52(12): 2292-330. https://doi.org/10.1016/j.comnet.2008.04.002

[2] Watteyne, T., Vilajosana, X., Kerkez, B., Chraim, F., Weekly, K., Wang, Q., Glaser, S., Pister, K. (2012). OpenWSN: a standards‐based low‐power wireless development environment. Transactions on Emerging Telecommunications Technologies, 23(5): 480-493. https://doi.org/10.1002/ett.2558

[3] You, Z.Y., Zhao, X.B., Wan, H., Hung, W.N.N., Wang, Y.K., Gu, M. (2011). A novel fault diagnosis mechanism for wireless sensor networks. Mathematical and Computer Modelling, 54(1-2): 330-43. https://doi.org/10.1016/j.mcm.2011.02.018

[4] Sahoo, M.N., Khilar, P.M. (2014). Diagnosis of wireless sensor networks in presence of permanent and intermittent faults. Wireless Personal Communications, 78(2): 1571-1591 https://doi.org/10.1007/s11277-014-1836-6

[5] Chen, J., Kher, S., Somani, A. (2006). Distributed fault detection of wireless sensor networks. In Proceedings of the 2006 Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks, pp. 65-72. https://doi.org/10.1145/1160972.1160985

[6] Swain, R.R., Dash, T., Khilar, P.M. (2017). An effective graph−theoretic approach towards simultaneous detection of fault (s) and cut (s) in wireless sensor networks. International Journal of Communication Systems, 30(13): e3273. https://doi.org/10.1002/dac.3273

[7] Panda, M., Khilar, P.M. (2014). Energy efficient distributed fault identification algorithm in wireless sensor networks. Journal of Computer Networks and Communications. https://doi.org/10.1155/2014/323754

[8] Kuo, T.W., Sha, E., Guo, M., Yang, L.T., Shao, Z. (2007). Embedded and Ubiquitous Computing. IFIP International Conference, EUC 2007, Taipei, Taiwan, https://doi.org/10.1007/978-3-540-77092-3

[9] Mahapatro, A., Khilar, P. (2014). Online fault diagnosis of wireless sensor networks. Central European Journal of Computer Science, 4(1): 30-44. https://doi.org/10.2478/s13537-014-0203-8

[10] Khan, S.A., Daachi, B., Djouani, K. (2012). Application of fuzzy inference systems to detection of faults in wireless sensor networks. Neurocomputing, 94: 111-20 https://doi.org/10.1016/j.neucom.2012.04.002

[11] Swain R.R., Dash T., Khilar P.M. (2019). Investigation of RBF Kernelized ANFIS for Fault Diagnosis in Wireless Sensor Networks. In: Verma N., Ghosh A. (eds) Computational Intelligence: Theories, Applications and Future Directions - Volume II. Advances in Intelligent Systems and Computing, vol 799. Springer, Singapore.

[12] Moustapha, A.I., Selmic, R.R. (2008). Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection. IEEE Transactions on Instrumentation and Measurement, 57(5): 981-988. https://doi.org/10.1109/TIM.2007.913803

[13] Fay, M.P., Proschan, M.A. (2010). Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules. Statistics Surveys, 4: 1-39. https://doi.org/10.1214/09-SS051

[14] MacFarland, T.W., Yates, J.M. (2016). Mann–whitney u test. In Introduction to Nonparametric Statistics for the Biological Sciences Using R, Springer, Cham, pp. 103-132. https://doi.org/10.1007/978-3-319-30634-6

[15] Titouna, C., Aliouat, M., & Gueroui, M. (2016). FDS: Fault detection scheme for wireless sensor networks. Wireless Personal Communications, 86: 549-562. https://doi.org/10.1007/s11277-015-2944-7

NOMENCLATURE

sm	sensor module
n	sensor node
N	Number of nodes
ch	cluster head
j	range of cluster heads
k	range of samples
s	group sample 1

c	group sample 2
sd	sensed data
r	time period range

Greek symbols

π	dimensionless constant, 3.14
σ	standard deviation
σ^2	variance
μ	mean
τ	time instance
θ	threshold
ξ	error

Subscripts

i	instance