Fault Diagnosis in Wireless Sensor Networks using Artificial Immune System

Santoshinee Mohapatra

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Fault Diagnosis in Wireless Sensor Networks using Artificial Immune System

Dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

in

Computer Science and Engineering

by

Santoshinee Mohapatra

(Roll Number: 514CS1012)

based on research carried out

under the supervision of

Prof. Pabitra Mohan Khilar

December, 2019

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Department of Computer Science and Engineering
National Institute of Technology Rourkela

December 16, 2019

Certificate of Examination

Roll Number: *514CS1012*
Name: *Santoshinee Mohapatra*
Title of Dissertation: *Fault Diagnosis in Wireless Sensor Networks using Artificial Immune System*

We the below signed, after checking the dissertation mentioned above and the official record book (s) of the student, hereby state our approval of the dissertation submitted in partial fulfillment of the requirements of the degree of *Doctor of Philosophy* in *Computer Science and Engineering* at *National Institute of Technology Rourkela*. We are satisfied with the volume, quality, correctness, and originality of the work.

Pabitra Mohan Khilar Principal Supervisor	Korra Sathya Babu Member, DSC
Judhistir Mahapatro Member, DSC	Lakshi Prosad Roy Member, DSC
External Examiner	Ashok Kumar Turuk Chairperson, DSC

Ashok Kumar Turuk
Head of the Department

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Prof. Pabitra Mohan Khilar
Associate Professor

December 16, 2019

Supervisor's Certificate

This is to certify that the work presented in the dissertation entitled *Fault Diagnosis in Wireless Sensor Networks using Artificial Immune System* submitted by *Santoshinee Mohapatra*, Roll Number 514CS1012, is a record of original research carried out by her under my supervision and guidance in partial fulfillment of the requirements of the degree of *Doctor of Philosophy* in *Computer Science and Engineering*. Neither this dissertation nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

_____
Pabitra Mohan Khilar

dedicated to my beloved parents...

Declaration of Originality

I, *Santoshinee Mohapatra*, Roll Number *514CS1012* hereby declare that this dissertation entitled *Fault Diagnosis in Wireless Sensor Networks using Artificial Immune System* presents my original work carried out as a doctoral student of NIT Rourkela and, to the best of my knowledge, contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the sections "Reference" or "Bibliography". I have also submitted my original research records to the scrutiny committee for evaluation of my dissertation.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation.

December 16, 2019
NIT Rourkela

Santoshinee Mohapatra

Acknowledgment

I am thankful to each and everyone who have helped me in the completion of this thesis.

I would like to express my gratitude to my supervisor Prof. Pabitra Mohan Khilar for his guidance, advice and constant support during my research work. His motivation encouraged me to remain focused to achieving my goal. His source of knowledge and experience truly inspired me.

I would like to thank my DSC members Prof. A. K. Turuk, Prof. K. S. Babu, Prof. J. Mahapatro of Department of CSE and Prof. L. P. Roy of Department of EC for their valuable suggestions. I am thankful to Prof. B. Majhi, Prof. S. K. Rath, Prof. D. P. Mohapatra, Prof. B. D. Sahoo and all the faculties of CSE Department for their support.

My heartfelt thanks to my best friend Priyanka whose constant motivation helped me a lot. Special thanks to my collected sisters Pallavi and Krutika for their unconditional love, support and help. I would like to thank my fellow lab mates especially Rakesh Ranjan Swain for his suggestion in the improvement of my research work. I also acknowledge all my friends, seniors, juniors and staffs of Department of CSE for their cooperation during my research work. I must acknowledge the educational resources and facilities that I have got from NIT Rourkela.

Last but not the least, I would like to thank my family members for their encouragement, love, patience and understanding. I would like to express my deepest gratitude to my in-laws for their endless love, care and support during my PhD. I could hardly have finished this without the continuous assistance and encouragement of my husband Soumyaprakash Sahoo. His understanding and patience made everything possible. I express my heartiest thanks and gratefulness to Almighty God for his divine blessing which help me to complete this research successfully.

December 16, 2019 — *Santoshinee Mohapatra*

NIT Rourkela — Roll Number: 514CS1012

Abstract

Wireless sensor network (WSN) consists of a huge number of sensor nodes, which are tiny in size, low battery powered and low cost. These sensor nodes are deployed in the environment for applications such as healthcare monitoring, environmental monitoring, and military application. Due to their physical limitations and environmental conditions, these nodes are subjected to different types of faults such as hard and soft fault. On the occurrence of hard fault a sensor node can not communicate with other sensor nodes. In case of soft faults, they are categorized into soft permanent, soft intermittent, and soft transient fault. On the occurrence of soft fault the sensor nodes can communicate with other sensor nodes but erroneously. A soft permanent faulty sensor node persists the faulty behavior by giving erroneous results. A soft intermittent faulty sensor node may behave like faulty for some period of time and fault free at other times. A soft transient faulty sensor node may occur for some duration of time and fault free for the rest of the time. In order to obtain accurate data from the deployed WSN, the sensor nodes need to be fault free. Motivated by the need of a fault free WSN in the deployed field, it is crucial to identify the faulty sensor nodes in WSN and categorize them into their respective fault types. In this thesis, four fault diagnosis algorithms have been proposed based on the concept of artificial immune system (AIS) such as clonal selection principle (CSP), negative selection algorithm (NSA), dendritic cell algorithm (DCA) and artificial immune network (AIN). To evaluate the performance of the proposed methods simulation and experimental validation are conducted using standard generic performance parameters such as fault detection accuracy (FDA), false alarm rate (FAR), false positive rate (FPR), fault classification accuracy (FCA), false classification rate (FCR), fault detection latency (FDL) and energy consumption (EC). Throughout the thesis, it is assumed that the sensor nodes can be either hard or soft faulty whereas the communication links are assumed to be fault free. To resemble the scenario in which the sensor nodes are randomly deployed in the field, the WSN follows an arbitrary network topology.

An efficient fault detection algorithm based on clonal selection principle(FDCSP) of AIS has been proposed to detect faulty sensor nodes such as hard permanent, soft permanent, soft intermittent and soft transient. This is followed by a fault classification algorithm based on the above respective fault types using the probabilistic neural network. After the actual fault status is detected, the faulty sensor nodes are isolated in the isolation phase. In fact, the proposed algorithm follows three phases such as fault detection, fault classification and fault isolation. The performance of the algorithm is evaluated by using the performance metrics

such as FDA, FAR, FPR. It is shown that, the fault detection accuracy of the proposed FDCSP algorithm is improved by 1.8%, 5.06% and 6.45% over Mohapatra et al., Panda et al. and Elhadef et al., respectively. The false alarm rate of the proposed algorithm is improved by 0.43%, 0.78% and 1.88% over Mohapatra et al., Panda et al. and Elhadef et al., respectively. The false positive rate of the proposed algorithm is improved by 1.81%, 5.07% and 6.46% over Mohapatra et al., Panda et al. and Elhadef et al., respectively. The fault classification performance is measured by fault classification accuracy and false classification rate. The simulation result also shows that the FDCSP algorithm provides less fault detection latency i.e., 4.11%, 6.19% and 10.35% over Mohapatra et al., Panda et al. and Elhadef et al., respectively and consumes less energy i.e., 11.40%, 20.95% and 49.03% over Mohapatra et al., Panda et al. and Elhadef et al., respectively.

An improved negative selection algorithm (INSA) has been proposed to diagnose the faulty sensor nodes and classified into soft permanent, soft intermittent and soft transient using the support vector machine. The performance of the algorithm is evaluated by using the performance metrics where it is shown that, the fault detection accuracy of the proposed INSA algorithm is improved by 1.55%, 4.97% and 6.49% over Mohapatra et al., Panda et al. and Elhadef et al., respectively. The false alarm rate of the proposed algorithm is improved by 0.33%, 0.85% and 1.86% over Mohapatra et al., Panda et al. and Elhadef et al., respectively. The false positive rate of the proposed algorithm is improved by 1.55%, 4.97% and 6.49% over Mohapatra et al., Panda et al. and Elhadef et al., respectively. The fault classification phase gives the average classification accuracy approximately 97% and the average misclassification rate 0.03. The simulation result also shows that the proposed algorithm provides less fault detection latency i.e., 4.26%, 8.75% and 13.09% over Mohapatra et al., Panda et al. and Elhadef et al., respectively and consumes less energy i.e., 11.64%, 21.81% and 50% over Mohapatra et al., Panda et al. and Elhadef et al., respectively.

Fault diagnosis using dendritic cell algorithm has been proposed to detect the faulty nodes. The important feature of this algorithm is that no training data is required. The performance of the algorithm is evaluated by using the performance metrics where it is shown that, the FDDCA algorithm gives better result as compared to the existing algorithms in terms of FDA, FAR, FPR, FDL and EC. The fault detection accuracy of the proposed FDDCA algorithm is improved by 1.51%, 4.7% and 6.46% over Mohapatra et al., Panda et al. and Elhadef et al., respectively. The false alarm rate of the proposed algorithm is improved by 0.25%, 0.9% and 1.85% over Mohapatra et al., Panda et al. and Elhadef et al., respectively. The false positive rate of the proposed algorithm is improved by 1.51%, 4.7% and 6.46% over Mohapatra et al., Panda et al. and Elhadef et al., respectively. The proposed algorithm provides less fault detection latency i.e., 7.48%, 12.54% and 16.56% over Mohapatra et al., Panda et al. and Elhadef et al., respectively and consumes less energy i.e., 18.49%, 26.99% and 53.51% over Mohapatra et al., Panda et al. and Elhadef et al., respectively.

An artificial immune network based fault diagnosis algorithm has been proposed to diagnose the faulty sensor nodes. In this algorithm to train and optimize the fault samples, learning, memory and suppression mechanism of immune network is used. Fault type information has been added to memory antibodies so that it can learn and memorize the same types of faults. Hence, classification accuracy can be improved. Experimental result shows that the fault detection accuracy of the proposed AINFDA algorithm is improved by 1.51%, 5.01% and 6.46% over Mohapatra et al., Panda et al. and Elhadef et al., respectively. The false alarm rate of the proposed algorithm is improved by 0.39%, 0.75% and 1.74% over Mohapatra et al., Panda et al. and Elhadef et al., respectively. The false positive rate of the proposed algorithm is improved by 1.51%, 5.01% and 6.46% over Mohapatra et al., Panda et al. and Elhadef et al., respectively. The proposed algorithm provides less fault detection latency i.e., 4.44%, 6.52% and 10.94% over Mohapatra et al., Panda et al. and Elhadef et al., respectively and consumes less energy i.e., 12.16%, 22.15% and 49.80% over Mohapatra et al., Panda et al. and Elhadef et al., respectively.

***Keywords*: *Wireless Sensor Network*; *Fault Diagnosis*; *Artificial Immune System*; *Hard and Soft Fault*; *Fault Classification*; *Arbitrary Network Topology*; *Fault Detection Accuracy*.**

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

AIN	Artificial Immune Network
AINFDA	Artificial Immune Network based Fault Diagnosis Algorithm
AIS	Artificial Immune System
APC	Antigen Presenting Cell
CSP	Clonal Selection Principle
DCA	Dendritic Cell Algorithm
EC	Energy Consumption
FAR	False Alarm Rate
FCA	Fault Classification Accuracy
FCR	False Classification Rate
FDA	Fault Detection Accuracy
FDCSP	Fault Diagnosis using Clonal Selection Principle
FDDCA	Fault Diagnosis using Dendritic Cell Algorithm
FDL	Fault Detection Latency
FPR	False Positive Rate
HIS	Human Immune System
INSA	Improved Negative Selection Algorithm
NSA	Negative Selection Algorithm
PNN	Probabilistic Neural Network
SVM	Support Vector Machine
WSN	Wireless Sensor Network

List of Symbols

af	Affinity
AB	Antibodies
AG	Antigens
$t_{bounded}$	Bounded time period
y_j	Class level
f(x)	Classification decision function
b	Classification threshold
fs	Fault status
$H_{i,j}$	Hadamard matrix
$k(x_j, x)$	Kernel function
α_j	Lagrange multiplier
c_i	Misclassification rate of class i
$Neg(sn_i)$	Neighbor of sensor node i
$(NegT_i)$	Neighbor table
Neg(.)	Neighbors list
NT_i	Node table
m	Number of support vectors
θ	Predefined threshold
pdf(.)	Probability density function
req_{msg}	Request message
rep_{msg}	Response message
sd	Sensor data
sn_i	Sensor node indexed as i
S	Set of sensor nodes
σ	Smoothing parameter
x_j	Support vector
t_{out}	Timeout period

Chapter 1

Introduction

1.1 Introduction

Wireless sensor networks (WSNs) have become popular worldwide due to their wide range of applications like forest fire monitoring, military application, health care monitoring and so on [1]. WSNs consist of large number of sensor nodes, which are usually small in size and inexpensive. Each sensor node has sensing, processing and communicating capabilities. Sensor nodes are low power devices with limited memory, storage, battery and low bandwidth. They are deployed in hostile and harsh environment and expected to run autonomously [2]. Since the sensor nodes are deployed in hostile and human inaccessible environment they are subjected to various types of faults.

The main cause of fault in a sensor node is due to the problems in the internal circuit of a sensor node, battery exhaustion, environmental deterioration or any kind of damage. Faults can be categorized into two types such as hard fault where the sensor node becomes inactive and do not respond to other sensor nodes and soft fault where the sensor node behaves arbitrarily and give erroneous data [3]. As faults are inevitable in WSNs, it is very important to determine which nodes are faulty and which are fault free. So, the main focus of this research is to identify the faulty and fault free nodes in a WSN.

If fault occurs, the consequences can be severe in terms of loss to human life, environment or any economical loss. The output from faulty sensor nodes might give erroneous results which may lead to a significant loss. The occurrence of faults can not be prevented however, the consequences of faults can be avoided or minimized by handling it properly.

The rest of this chapter is organized as follows. A brief description of fault, error and failure of sensor nodes is presented in subsection 1.1.1. Classification, causes and impacts of faults are presented in subsections 1.1.2 and 1.1.3, respectively. The fault management in WSNs is discussed in subsection 1.1.4. The performance parameters are discussed in subsection 1.1.5. Section 1.2 present the factors influencing in designing the diagnosis protocol in wireless sensor networks. The system model is described in Section 1.3. The motivation of the proposed work is described in Section 1.4. Section 1.5 presents the objective of the research work. The major contribution and organization of the thesis are presented in Sections 1.6 and 1.7, respectively. Finally, Section 1.8 discusses the summary.

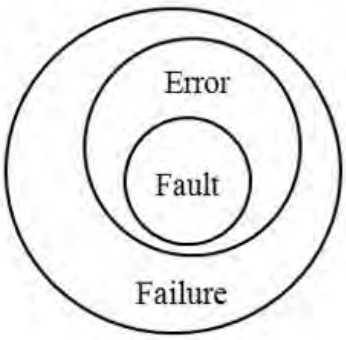

Figure 1.1: Relation between fault, error and failure

1.1.1 Faults, Errors and Failures of Sensor Nodes in WSNs

According to literature, fault, error and failure are related terms used interchangeably by different authors. However, in this work, it is assumed that a fault may cause an error and an error always causes a failure in a wireless sensor network. A fault occurs when a sensor node behaves unexpectedly [4]. When there is a fault in a sensor node, it either does not send any data to other sensor nodes, which is called as hard fault [5] or it sends an erroneous data which is called as soft fault. A soft faulty sensor node sometimes give faulty data and sometimes give faultfree data called as intermittent fault [6].

The presence of a sensor error does not mean that a sensor is hard faulty due to the fact that sometimes it produces erroneous data because of the environmental noise or malicious activities which are known as soft faulty sensor nodes. The presence of sensor fault may lead to the sensor error. A fault in the sensor node causes a sensor error which in turn causes sensor failure. In other words, the cause of the sensor failure is an error reported by sensor node and causes of error in sensor node is the occurrence of faults in sensor nodes. In fact, no physical system including WSN can experience a failure without having a fault. A hard faulty sensor node due to either physical damage or battery exhaustion is an example of a fault without any error as it does not respond to its neighboring nodes. The relation between fault, error and failure is shown in Figure 1.1. It can affect the total computation of the network.

1.1.2 Classification of Faults

Based on the behavior of a failed sensor node, faults can be classified into hard and soft fault. Hard faulty sensor nodes do not respond to other sensor nodes whereas, soft faulty sensor nodes respond with erroneous data. Similarly, based on duration the sensor faults can be classified into permanent and temporary fault [7]. As the name suggests permanent fault gives permanently faulty behavior and it can only be repaired or replaced. Temporary faults are further classified into transient, intermittent and byzantine fault. A transient fault occurs for some duration of time and fault free for the rest of the time. An intermittent faulty

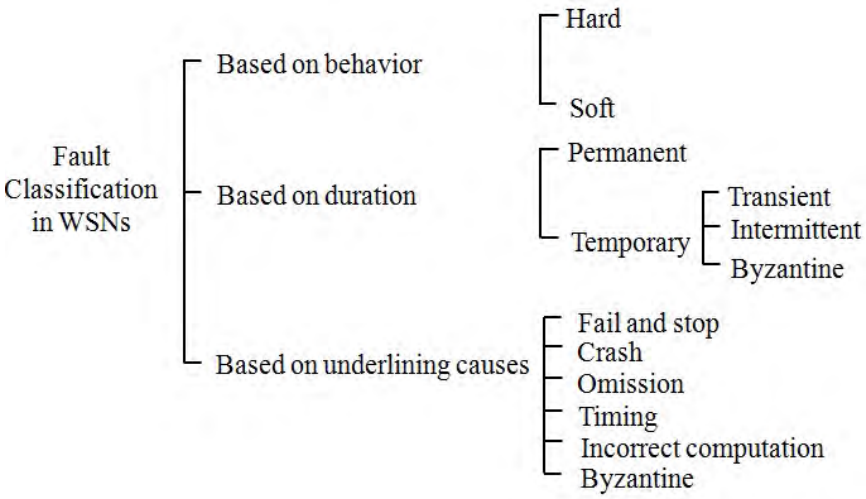

Figure 1.2: Fault classification in WSN

sensor node may behave faulty for some duration of time and fault free at other times. A byzantine fault is any type of fault which can be arbitrarily faulty hence difficult to detect. The classification of faults is clearly shown in Figure 1.2.

Based on underlying causes, faults are classified into fail stop, crash, omission, timing, incorrect computation, byzantine fault [8]. An ordered fault classification can be presented as {byzantine⊃incorrect computation⊃timing⊃omission⊃crash⊃fail stop} or {fail stop⊂crash⊂omission⊂timing⊂incorrect computation⊂byzantine}.

- *Fail Stop Fault*: This type of fault occurs when a sensor node halts its operation due to battery exhaustion and alerts its neighbors. This fault may be natural or human made fault.

- *Crash Fault*: Crash fault occurs due to complete exhaustion of the battery or due to physical damage. This type of faulty sensor nodes can not participate in the network activities. This is a natural fault caused by natural phenomena without the involvement of human being. Fail stop fault is a specific case of crash fault where computation halts, not communication.

- *Omission Fault*: This type of fault occurs when a sensor node does not respond to the sink node on time, fails to send or receive the required data on time. It can either be a mallicious fault created by a person with a malicious intent, or a normal fault. In nature this fault can either be permanent, intermittent or transient. Crash fault is a subclass of omission fault class.

- *Timing Fault*: This type of fault occurs when a sensor node completes its task correctly, but either too early or too late or never. In case, it never completes the task of sending or receiving a message, then timing fault behaves as an omission fault. So omission fault is a subclass of timing fault class

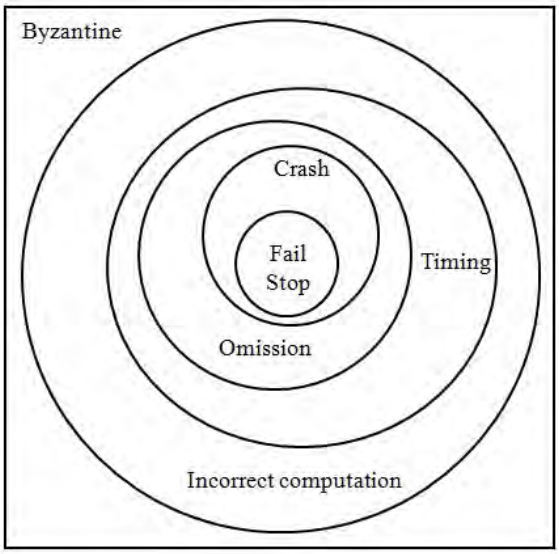

Figure 1.3: An ordered fault classification

- *Incorrect Computation Fault*: This type of fault occurs when a sensor node fails to produce correct results though the given input is correct. Like omission and timing fault this may be natural or, human made fault. This fault can be either permanent, intermittent or transient in nature.

- *Byzantine Fault*: Every possible fault is said to be byzantine fault. This can be considered as universal fault set. Each of the above mentioned fault class is a subclass of this universal fault class.

  An ordered classification of this type of fault is depicted in Figure 1.3. The taxonomy of faults presented here is to give an idea about all the types of faults available in the literature. However, in this work only hard and soft faults are considered.

1.1.3 Causes and Impact of Faults

Failure can occur due to various reasons. Sensor nodes are prone to failure and they may fail due to battery depletion, natural calamities, hardware malfunction and link failure [4]. A sensor node may fail due to any problem in hardware and the primary cause of fault in hardware is environmental condition or weather. Due to the influence of environmental noise, sensor node may collect incorrect data which may lead to an error. Old battery can give erroneous data. Low battery power not only indicates the remaining lifetime of a sensor node, but also influences the sensor readings and cause unreliable data. There are multiple links between the path from source to destination. When there is any change in environment or an external object block the link, it may fail permanently or temporarily. In this work, links are fault free in nature and taken care by underlying MAC protocol.

The presence of permanent hard fault or crash fault partitions the whole network into two or more disjoint parts due to which the routing path of the network breaks down [5]. The hard or crash fault in sensor node makes the whole network disconnected. This is because, a faulty set of sensor nodes might create a set of disjoint sets of the sensor nodes, where the member of one set might not be able to communicate with the member of another set. In the presence of soft faults such as permanent, intermittent and transient, byzantine fault and incorrect computation fault, the sensor network gives unexpected result due to which the performance of the network degrades.

1.1.4 Fault Management in WSNs

Following are the techniques used for handling the faulty sensor nodes.

- *Fault detection*: Fault detection method is used to determine the presence of faulty nodes in the network.
- *Fault identification*: This mechanism is used to identify the type and behavior of faulty nodes in the network.
- *Fault classification*: Fault classification method is used to classify the faults into different types with respect to their behavior, persistence and underlying causes.
- *Fault diagnosis*: This mechanism is used to identify the faulty and fault free sensor nodes present in the network. It follows the fault detection, identification and classification methods.
- *Fault isolation*: In this method, the faulty sensor nodes are separated from the network and the fault free nodes remain as it is. The faulty nodes does not allow to take part in the network operation.
- *Fault tolerance*: It allows the network to perform its operation without being affected by the presence of faulty nodes.
- *Fault prevention*: This is used to avoid faulty sensor nodes so that the network continues its operation without any interruption.
- *Fault recovery*: This mechanism is used to repair or recover the faulty sensor nodes during the network operation or later.

The above mechanisms are important in the fault diagnosis system to give detailed information about the faults. In this work, we have mainly focused on fault detection, identification, classification, diagnosis and isolation techniques.

1.1.5 Performance Parameters

The performance of the algorithms is measured in terms of following parameters [9].

- *Fault Detection Accuracy*: It is defined as the ratio between the number of faulty sensor nodes detected as faulty to the total number of faulty nodes present in the network.
- *False Alarm Rate*: It is defined as the ratio between the number of fault free nodes detected as faulty to the total number of fault free nodes present in the network.
- *False Positive Rate*: It is defined as the ratio between the number of faulty nodes detected as fault free to the total number of faulty nodes present in the network.
- *Fault Detection Latency*: It is defined as the total amount of time required to detect all the sensor nodes present in the network.
- *Energy Consumption*: It is defined as the total energy consumed by the network to identify the faulty sensor nodes present in the network.
- *Fault Classification Accuracy*: It is defined as the ratio between the total number of faulty nodes correctly classified as a fault type to the total number of faulty nodes present in the network.
- *False Classification Rate*: It is defined as the ratio between the total number of faulty nodes wrongly classified as a fault type to the total number of faulty nodes present in the network.

1.2 Factors Influencing in Designing the Diagnosis Protocol for WSNs

Conventional fault diagnosis algorithms designed for wired interconnected networks [10–17] and wireless networks [7, 18, 19] might not be suitable for WSNs due to the following reasons.

- *Resource constraints*: Low processor power, less memory, limited bandwidth and energy are the constraints in WSNs. Requirement of message exchanges is necessary for fault diagnosis and the number of message exchanges is directly proportional to the energy consumption [2]. So it's a challenging task to design a fault diagnosis method which consumes less energy and maintains high fault detection accuracy with low false alarm rate.
- *Random deployment*: Sensor nodes are often deployed in human inaccessible environments by human or robot. The underwater environment requires sparse

deployment whereas terrestrial environment requires dense deployment of sensor nodes. In case of threshold based diagnosis scheme [20, 21] sensor node may be wrongly detected as faulty due to the sparse deployment of the network.

- *Dynamic network topology*: It shows the variation in the densities of sensor nodes in a network. The transmission of information in the dynamic network topology is difficult because of the faultiness of sensor nodes which might disconnect the whole network. The performance also degrades due to the dynamic nature [19].

- *Attenuation and signal loss*: Channel fading affects multi hop communication in WSNs. In underwater sensor network applications, the communication between sensor nodes are performed over acoustic waves [1]. It is very challenging to design a fault diagnosis scheme for such applications due to low bandwidth and less propagation delay.

The above mentioned issues motivate the need of developing fault diagnosis algorithms with low energy consumption, less fault detection latency, high fault detection accuracy and low false alarm rate.

1.3 System Model

The system model consists of network model, fault model and energy model as follows.

1.3.1 Network Model

It is assumed that, a set of homogeneous sensor nodes S = $\{sn_1, sn_2, ..., sn_N\}$ are randomly deployed in the field. The sensor nodes form an arbitrary network topology without losing their connectivity among themselves. Each sensor node is assigned with an unique id sn_i and a common transmission range r. Each sensor node has the ability to sense various parameters such as temperature and humidity. The sensed data of a sensor node is transmitted to other sensor nodes via a wireless medium. The topology of the WSN can be illustrated as a graph G = (S, E) called the communication graph, where S is the set of sensor nodes and E is the set of edges. Two nodes sn_i and sn_j are said to be adjacent, if and only if (sn_i,sn_j) $\in$ E and sn_j is within the transmission range of sn_i. All transmissions are omni directional. Communication links are assumed to be fault free and taken care by underlying MAC layer protocol.

1.3.2 Fault Model

The fault model specifies the causes and consequences of a fault in the WSN. As the sensor network is deployed in hostile environments, sensor nodes may encounter different type of faults. On the basis of behavior, it is categorized as hard and soft fault [22]. The causes of

hard faults are battery exhaustion, moving out of the transmission range, physical damage. On the occurrence of hard fault a sensor node can not communicate with other sensor nodes. In case of soft faults, they are categorized into soft permanent, soft intermittent, and soft transient fault. On the occurrence of soft fault the sensor nodes can communicate (i.e., transmit the data as well as receive the data) with other sensor nodes but erroneously. A soft permanent faulty sensor node persists the faulty behavior by giving erroneous results. A soft intermittent faulty sensor node may behave like faulty for some period of time and fault free at other times. A soft transient faulty sensor node may occur for some duration of time and fault free for the rest of the time.

After finding the faults, the faults are classified into their respective types. Suppose, in the time duration t_1 to t_{10} if the node does not respond at all then it is said to be hard permanent fault. In the time duration t_1 to t_{10} if the node respond, but the result is erroneous continuously then it is said to be a soft permanent fault. If the fault persists for some amount of time t_0 - t_3 and fault free at other times t_3 - t_6 then it is said to be the soft intermittent fault and if the fault occurs for some instant of time t_9 and fault free for the rest amount of time then it is said to be a soft transient fault.

1.3.3 Energy Model

For the data communication, each sensor node is having a wireless transceiver which consists of transmitter and receiver. Transmitting electronics and amplifier is required by the transmitter whereas, receiving electronics is required by the receiver. Let, ET is the energy consumed to transmit the data and ER is the energy consumed to receive the data. So, the total energy is the sum of ET and ER as given in [23].

$$ET = m * (\alpha_1 + \alpha_2 * d^{\alpha}) \tag{1.1}$$

$$ER = m * \alpha_3 \tag{1.2}$$

where, m is the size of data (bytes), d is the distance between the two sensor nodes, α is the free-space-coefficient

$$\alpha = \begin{cases} 2, & d_0 \leq \text{d} \\ 4, & d_0 > \text{d} \end{cases} \tag{1.3}$$

α_1 is the energy required for the transmitting electronics, α_2 is the energy required for the amplifier, α_3 is the energy required for receiving electronics

1.4 Motivation

Fault diagnosis in WSNs is an emerging and important field in both academic as well as industry. WSNs are more prone to failure as they are deployed in human inaccessible and

hostile environment. In order to maintain the network's Quality-of-service (QOS), it is very important for WSN to be able to detect the faults and diagnosed it at the right time. Traditional diagnosis methods are test based where a test is performed between each sensor node on the basis of their measured data. These protocols mostly depend on the statistics of the observed data on sensor node vicinity.

The existing fault diagnosis algorithms for WSNs available in the literature are usually based on (I) graph theory, (II) statistical method, (III) neighboring coordination based method, (IV) comparison based method, (V) neural network based method. The types of faults, fault detection, diagnosis and recovery for an exhaustive faults such as crash, permanent, intermittent, transient, byzantine, fail stop, omission has not been found in the literature. The soft computing based fault diagnosis for WSN has potential to solve the fault diagnosis problems in addition to the above diagnosis approaches. Since the WSNs are power constraint and usually large, the efficient and effective diagnosis algorithms need to be designed and developed, to save and improve the network lifetime, detection accuracy and diagnosis latency. Artificial immune system based fault diagnosis for WSN is scarce in the literature. AIS is influenced from the principle of human immune system (HIS), which can expertly saved our bodies from bacteria and viruses.

1.5 Research Objective

Motivated by the challenges and need of efficient fault diagnosis algorithms in wireless sensor network the following objectives are undertaken.

- To design an efficient fault diagnosis algorithm using clonal selection principle (CSP) of AIS.
- To design an efficient fault diagnosis algorithm using negative selection approach (NSA) of AIS.
- To design an efficient fault diagnosis algorithm using dendritic cell algorithm (DCA) of AIS.
- To design an efficient fault diagnosis algorithm using artificial immune network (AIN) of AIS.
- To validate the proposed algorithm using NS 2.35 simulator [24].
- To evaluate the efficacy of the proposed algorithms using different performance parameters such as fault detection accuracy (FDA), false alarm rate (FAR), false positive rate (FPR), fault classification accuracy (FCA), false classification rate (FCR), fault detection latency (FDL) and energy consumption (EC).

1.6 Major Contribution

In this dissertation four algorithms are proposed based on artificial immune system. Chapters 3, 4, 5 and 6 are the contributed chapters in this thesis and are outlined as follows.

- In Chapter 3, a detection algorithm has been proposed to identify faulty sensor nodes using clonal selection principle of artificial immune system and then the faults are classified into permanent, intermittent, and transient fault using the probabilistic neural network approach. After the actual fault status is detected, the faulty nodes are isolated in the isolation phase. The performance metrics such as fault detection accuracy, false alarm rate, false positive rate, fault classification accuracy, false classification rate, fault detection latency and energy consumption are used to evaluate the performance of the proposed algorithm. The simulation results show that the proposed algorithm gives superior results as compared to the existing algorithms in terms of the performance metrics. The fault classification performance is measured by fault classification accuracy and false classification rate. It has also seen that the proposed algorithm provides less fault detection latency and consumes less energy than that of the existing algorithms proposed by Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26] for wireless sensor network. However, the major drawback of this approach is the overhead of cloning, mutation for computing affinity (fitness) value towards the solution. To reduce the computational overhead, the algorithm given in chapter 4 was proposed.

- In Chapter 4, an improved negative selection algorithm (INSA) has been proposed to identify faulty sensor nodes and then the faults are classified into soft permanent, soft intermittent, and soft transient fault using the support vector machine (SVM). The performance is evaluated using NS-2.35 simulator and the simulation result shows that the proposed method is superior to the existing protocols Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26] in terms of fault detection accuracy, false alarm rate, false positive rate, fault detection latency, energy consumption, fault classification accuracy and false classification rate. The proposed method is also validated by using real testbed experiments conducted in an indoor laboratory environment. As affinity is not required in this approach, it gives better result as compared to the approach proposed in chapter 3. The major advantage of this approach is that, it does not need any prior knowledge of fault patterns and able to adapt to the changes in the faulty and fault-free situation. However, the drawback of this approach is that it takes excessive time because of generating random numbers of detectors. It is also very difficult to know whether the number of generated detectors is large enough, to satisfy the detection fault probability. This has been solved in the next chapter which is based on the dendritic cell algorithm.

- In Chapter 5, fault diagnosis using dendritic cell algorithm (FDDCA) has been proposed to detect the faulty nodes. The important feature of this algorithm is that no training data is required. The performance of the algorithm is evaluated by using the performance metrics where it is shown that, the FDDCA algorithm gives better result as compared to the existing algorithms in terms of fault detection accuracy, false alarm rate, false positive rate, fault detection latency and energy consumption. As no training, no affinity is required in this algorithm, it gives better result as compared to the algorithms proposed in chapter 3, chapter 4 and chapter 6. This proposed algorithm requires less number of message exchanges and computational complexity than the other proposed algorithms.

- In Chapter 6, an artificial immune network based fault diagnosis algorithm (AINFDA) has been proposed to diagnose the faulty sensor nodes. In this algorithm to train and optimize the fault samples, learning, memory and suppression mechanism of immune network is used. Fault type information has been added to memory antibodies so that it can learn and memorize the same types of faults. Hence, classification accuracy can be improved. Experimental results shows that AINFDA gives better result than the existing algorithms in terms of fault detection accuracy, false alarm rate, false positive rate, fault detection latency and energy consumption. This proposed algorithm based on artificial immune network (AIN) gives better result than clonal selection principle (CSP) proposed in chapter 3. Though the proposed algorithm is similar to CSP in terms of computation of affinity, it differentiates the algorithm using CSP presented in chapter 3 with respect to consideration of different fault types and added it to memory antibodies so that it can learn and memorize the same types of faults. Hence, the classification accuracy is also improved and it performs better as compared to clonal selection principle.

1.7 Thesis Outline

The thesis is organized as follows.

- **In Chapter 1,** introduction to WSNs, an overview of the classification of faults, causes and impacts of faults, fault management in WSNs and performance parameters are presented. The motivation behind designing the fault diagnosis algorithms using artificial immune system approaches is outlined along with the research objectives. The major contributions are highlighted followed by the thesis organization.

- **In Chapter 2,** an overview of related works done by different authors in the area of fault diagnosis and artificial immune system is presented. The main focuses are given to artificial immune system related fault diagnosis.

- **In Chapter 3,** a detection algorithm has been proposed to identify faulty sensor nodes using clonal selection principle of artificial immune system and then the faults are classified into permanent, intermittent, and transient fault using the probabilistic neural network approach. The performance is analyzed by the simulation and compared with existing algorithms.

- **In Chapter 4,** an improved negative selection algorithm (INSA) has been proposed to identify faulty sensor nodes and then the faults are classified into soft permanent, soft intermittent, and soft transient fault using the support vector machine (SVM) technique. The proposed method is validated by the simulation and compared with the existing fault diagnosis protocols.

- **In Chapter 5,** the faulty nodes are detected using the proposed FDDCA algorithm. The performance of the algorithm is evaluated by using the performance metrics where it is shown that, the FDDCA algorithm gives better result as compared to the existing algorithms in terms of fault detection accuracy, false alarm rate, false positive rate, fault detection latency and energy consumption.

- **In Chapter 6,** an artificial immune network based fault diagnosis algorithm (AINFDA) has been proposed to diagnose the faulty sensor nodes. Experimental results shows that AINFDA gives better result than the existing algorithms in terms of fault detection accuracy, false alarm rate, false positive rate, fault detection latency and energy consumption.

- **In Chapter 7,** an outline of the whole work is briefly described. The comparison of the proposed algorithms is discussed with the possibilities of future scopes in this work.

1.8 Summary

This chapter briefly describes the introduction to WSNs, an overview of fault classification and its causes. Fault management, performance parameters to evaluate the proposed algorithms and factors those influences in designing the diagnosis protocols for the WSN are presented. The motivation for need for designing and developing efficient fault diagnosis algorithms for WSN are given. Based on the need of fault diagnosis, different research objectives are specified. The organization of the thesis with a brief presentation of research work and contributions are discussed in this chapter. Precisely, this chapter provides a complete overview of the whole thesis.

Chapter 2
Background and Related Works

2.1 Introduction

In this chapter, a comprehensive literature study based on traditional fault diagnosis is presented. Fault diagnosis using various artificial immune system approaches is also discussed. A large number of wireless battery constrained and tiny sensor nodes are usually deployed in human inaccessible and hostile environment in order to collect vital data for different applications [3]. Since the decision depends on the correctness and accuracy of the collected data, the diagnosis of sensor nodes becomes an indispensable task which needs to be followed before beginning of normal function of WSN. Therefore, a lot of researchers focused on designing and developing the fault diagnosis algorithm for WSN. The aim of this chapter is to survey the related works considering the traditional methods and the artificial immune system based method and present in an exhaustive way.

The rest of this chapter is organized as follows. Section 2.2 presents the fault diagnosis of WSN using traditional/generic approaches. Section 2.3 describes the fault diagnosis using AIS approaches. Finally, Section 2.4 discusses the summary.

2.2 Fault Diagnosis of WSN using Traditional/Generic Diagnosis Approaches

The popularity and applications of WSNs lead to the development of fault diagnosis methodologies. Several fault diagnosis approaches have been employed in WSNs. As illustrated in Figure 2.1 fault diagnosis techniques are broadly classified into centralized, distributed and hybrid approaches [1]. In centralized approach, a single sensor node monitors and analyzes the status of all other senor nodes in the network. The centralized approach can further classified into statistical based, sequence based, probabilistic based and soft computing based approaches. In distributed approach, each sensor node in the network monitors and analyzes the fault status of its neighbors and itself. The distributed approach can further classified into majority voting based, statistical based, probabilistic based, comparison based and soft computing based approaches. In hybrid approach, both the sensor node and the sink node participate in the fault detection and decision phase, which creates a

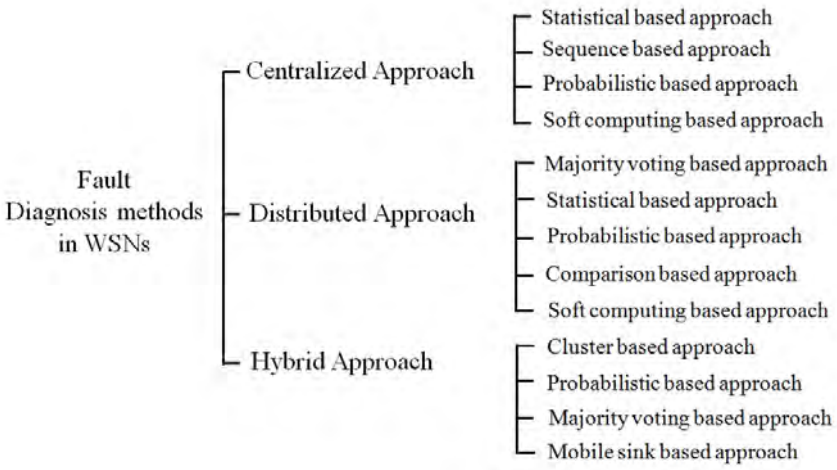

Figure 2.1: Taxonomy of fault diagnosis approaches

multi-layered architecture of WSN with the combination of both centralized and distributed approaches. The hybrid approach can further classified into cluster based, probabilistic based, majority voting based and mobile sink based approaches.

2.2.1 Centralized Approach

In centralized approaches, one central node is always responsible for fault diagnosis in the network. The central node may be a sink node or base station. The computational power, storage capacity and energy is high in the central node as compared to other sensor nodes. The central node receives the diagnostic information from other sensor nodes in the network. After receiving, it analyzes the information, and decides the fault status of the sensor nodes in the network. The centralized approach can be classified into following approaches according to some existing literature which are discussed as follows.

- **Statistical based approach**

  A centralized fault detection algorithm using statistical method was proposed by Panda et al. for WSNs [27]. The sensed data of a sensor node is represented as x(k) = A + r(k), where, A is the actual data and r(k) is the noise. The absolute difference d(k) is the absolute value of noise r(k). A node is said to be fault free if $d(k) < 3\sigma$ where σ is the standard deviation. The value of A is difficult to know in real environment so the absolute difference is recalculated. The z value is calculated using sensed data, mean and standard deviation. If the condition d(k) < z is satisfied the node is assumed to be fault free otherwise faulty [28].

  Jin et al. [29] proposed a passive diagnosis approach based on auto regressive model [30], Kuiper test [31] and Kolmogorov-Smirnov (K-S) test [32]. The Auto-regressive

model is used to pre-whiten the data before testing. The Kuiper and K-S tests are used to check the similarities between two given distributions. Here the authors mainly considered the hard fault.

- **Sequence based approach**

  Guo et al. proposed a novel method called as FIND for fault detection in WSN [33, 34]. Based on the sensor readings and physical distance from the event, FIND ranks the nodes. A node is decided as faulty if there is a large difference between the sensor data rank and distance rank. Here the authors have considered byzantine data faults.

  Kamal et al. proposed a sequence based fault detection (SBFD) in WSN [35]. The method uses in-network packet tagging to effectively deduce the route of all packets sent to the sink using the Fletcher checksum and server-side network path analysis. The sink uses control messages to check the status of affected nodes when a failure is suspected. This method mainly focused on hardware fault however, it does not consider different types of faults.

- **Probabilistic based approach**

  Lau et al. proposed a centralized Naive-Bayes approach for hardware fault detection in sensor network [36]. By using end to end transmission time analysis at the sink the fault is detected. This approach works in three phases preparatory phase, training phase and testing phase. A Maximum Likelihood Estimation (MLE) is formulated by using the normal and faulty conditional probabilities. Here only the hard fault is considered for the diagnosis.

  Tang et al. modeled the fault diagnosis into the pattern classification problem and introduces a neighborhood hidden conditional random field (NHCRF) method to detect sensor module fault and sensor path fault in different traffic conditions [37]. The NHCRF uses the signal strength, delay and frequency to detect the faulty nodes. Furthermore, Dhal et al. proposed a link failure detection mechanism using a Maximum A Posteriori Probability (MAP) method and hypothesis testing [38]. This method can not identify the numbers and types of faults in WSN.

- **Soft computing based approach**

  Abid et al. proposed a centralized fault detection technique for WSN using K nearest neighbor (KNN) and Euclidean distance method [39]. Here, the absolute data value difference between the sensor node and its neighboring nodes are calculated. Gain and offset faults are mainly targeted by the authors in this work. Yang et al. developed a telediagnostic powertracer to detect the node failure in remote WSN [40]. The proposed protocol uses external power measurements for failure detection. The authors mainly focus on battery failure, antenna failure, connectivity problem, network

disconnection, radio damage and fault due to under water. The major disadvantage of this protocol is that the number of fault scenarios increases exponentially with the number of applications.

Warriach et al. proposed a supervised machine learning approach based on Hidden Markov Model (HMM) [41] to detect the faults in WSN [42]. To detect the errors and classify the faults the authors have compared the Hidden Markov Model of faulty and fault free environment. The HMM is followed by five parameters to detect the faults and to estimate these parameters the authors have used supervised learning. The computation is carried out by the base station, which increases the computation overhead. Here, the authors construct two HMMs by considering the visible state, hidden state and faulty state. Next they analyze two HMMS with the fault model to detect the data faults in WSN.

- **Drawbacks of Centralized Approach**

  The centralized approaches are not suitable for large scale WSNs, because the multi-hop communication by each sensor node increases the energy consumption as well as message exchanges. The diagnosis latency is also increased due to multi-hop communication. The sink node is solely responsible for diagnosing all other nodes however, if it fails, the diagnosis process can not be completed. For these reasons, the fault diagnosis concerns the distributed and hybrid approaches.

2.2.2 Distributed Approach

In distributed approaches, all the sensor nodes are involved in the diagnosis process. The task is distributed among the sensor nodes of the network. Hence, the distributed approach reduces the traffic overhead and network congestion problem of centralized approach which increases the network lifetime. The distributed approach can further classified into different techniques which are discussed as follows.

- **Majority voting based approach**

  Chen et al. proposed a localized distributed fault detection method for WSNs using majority voting based approach [43]. In this method, each sensor node s_i tests its one-hop neighbor $neg(s_i)$ and generates a test result $c_{ij} \in \{0,1\}$. If the test result is 0, it indicates both the sensor nodes s_i and s_j are probably good (PG) or probably faulty (PF). If the test result is 1 it indicates one of the sensor node is probably faulty (PF). Sensors can be either probably good (PG) or probably faulty (PF), decided by performing the comparison test among its neighbors. If $\sum_{s_j \in Neg(s_i)} c_{ij} < \lceil Neg(s_i)/2 \rceil$ then s_i = PG otherwise, s_i = PF. To claim s_i is GD $\sum_{s_j \in Neg(s_i)} 1 - 2c_{ij}$ must be greater than equal to $\lceil Neg(s_i)/2 \rceil$.

Xu et al. extended this method and proposed a distributed approach for localized fault detection using local comparison based approach for soft permanent and intermittent faulty nodes in the network [44]. Saihi et al. proposed a decentralized fault detection approach based on error function [45]. This method follows the majority voting scheme for fault detection. This method solely depends on neighboring sensor nodes. Whenever the number of faulty sensor node increases in the neighbors, false alarm rate also increases which decreases the fault detection accuracy.

- **Statistical based approach**

  Panda et al. used a statistical method for fault detection [9]. The authors detected both hard and soft fault using a modified three sigma edit test. In this method each sensor node collects the data from its neighboring sensor nodes and then diagnose itself by using the modified three sigma edit test. An energy efficient soft fault detection algorithm for WSNs using z-test has proposed in [46]. Here the information from the faulty node is known as outlier. The sensor node collects neighboring data, analyzes them using z-test and decides its own fault status as well as predict the fault status of its neighboring nodes.

  Panda and Khilar introduces a novel distributed algorithm for fault detection in sparse networks to detect faulty sensor nodes [47]. Each sensor node collects the information only from its neighboring sensor nodes to reduce the communication overhead. To determine the fault status of each sensor node and neighboring sensor nodes Neyman-Pearson test method is used. To obtain the final fault status of each sensor node, a voting scheme is applied to the fault status data.

- **Probabilistic based approach**

  Yuan et al. proposed a distributed Bayesian algorithm [48]. In this method, a Bayesian network is used to calculate the prior and posterior fault probability of the sensor node. The fault detection accuracy is improved by utilizing the border nodes with a confidence factor. A flag status is generated by each sensor node by comparing the sensor reading of neighbors. Then the fault probability is calculated using the Bayesian algorithm. The fault probability can be adjusted by exploiting the border nodes. To compute the confidence level the border nodes transmit the messages to its neighbors. The fault status is obtained by comparing the fault probability with a probability of threshold. Here gain fault and stuck-at-fault is considered by the authors.

  Zhao et al. proposed a neighborhood kernel density estimation approach for fault diagnosis in WSN [49]. In this method, the authors formulate the detection of faulty nodes into a pattern classification problem and used semi supervised local kernel density estimation (SLKDE) for diagnosis of faulty nodes. This method computes

the posterior probability of different faulty and normal scenarios in the network path with various traffic situations such as normal, light and heavy congestion.

- **Comparison based approach**

  Sharma et al. proposed a reactive distributed fault detection scheme (rDFD) for soft permanent and transient fault in WSN [50]. To detect the faulty nodes, each node uses temporal correlation and spatial correlation. The proposed method consists of four phases such as self detection phase, communication phase, decision phase and reconfirmation phase. To improve the fault detection accuracy a confidence level is set for each node in the network.

  Sahoo and Khilar proposed a fault detection method based on comparison of sense data and residual energy of neighboring nodes [51]. For better fault detection the residual energy of neighboring nodes is estimated and compared. Here, permanent and transient faults are detected by the authors. An extension to the above method has been proposed by the authors in [52]. In this protocol, a spanning tree is constructed, which covers all the fault free nodes. At last, in the dissemination phase, the spanning tree is used to transmit the local diagnostic view that generates a global diagnostic decision in the sensor network. It depends on the values of neighboring sensor nodes.

- **Soft computing based approach**

  To detect the faults Chanak and Banerjee proposed a fault detection and classification scheme using fuzzy rule [53]. The proposed scheme increases the fault detection accuracy by handling the uncertainties in the sensor network environment. Detections are carried out at the node as well as at the sink. This method mainly deals with the faults that affect due to sensor circuit, receiver circuit and the battery status of the node. Fault detection and fault classification are two steps used in this method. In the fault detection stage, transmitter, receiver, sensor and battery faults are detected. The transmitter faults are detected at the sink and the other faults are detected by the node. The transmitter circuit faults are detected by the heartbeat message exchange, the sensor circuit faults are detected by the comparison between neighbor sensors and represented as fuzzy linguistic variable. The receiver fault is detected by the node itself using heartbeat ok message and represented as fuzzy linguistic variable. All these fuzzy linguistic variables are represented as low, medium, high classes to decide the fault status of sensor nodes. In the next phase, to decide the hardware conditions of the sensor node fuzzy logic system is used.

  Ghorbel et al. proposed a Distributed and Efficient One-class Outlier Detection Classifier (DEOODC) based on the Mahalanobis kernel [54]. A Kernel principal component analysis based on Mahalanobis kernel is used to detect the faults in WSN.

Two phases are followed i.e., training phase and testing phase. Gain, stuck-at and out of bound faults are detected by this method.

Obst [55] proposed a spatially organized distributed echo state networks (SODESN) [56] for anomaly and fault detection in the network. The SODESN is a distributed architecture over sensor network using recurrent neural network (RNN). The SODESN is trained using echo state network (ESN) architecture and RNN learning algorithm to predict the estimated sensor value. The predicted sensor value is compared with the actual value using a preset threshold to detect the abnormal sensor nodes. The authors mainly focus on random fault and stuck at fault for detection.

Mahapatro et al. proposed a fault detection algorithm for permanent and transient faults [57]. A Timeout mechanism is used to detect the hard permanent fault. The soft permanent faults are detected by the comparison of sensor node value with its neighbors and a distinct time is set for intermittent fault detection. As the intermittent fault is not consistent and requires repeated number of testing for detection, the authors formulate it into an optimization problem based on multi-objective swarm optimization (2LB-MOPSO). This is proposed to find an optimal trade off between the fault detection accuracy and the period of fault checking.

- **Drawbacks of Distributed Approach**

  As compared to the centralized approach, the distributed approach gives better performance for large-scale WSNs. However, the distributed approaches still have some disadvantages. The sensor node in WSN has limited resources such as energy, computation power and storage capacity. So the soft computing based approaches can not be suitable due to computational overhead. In most of the distributed approach, the sensor nodes depend on the neighboring sensor nodes. If the majority of neighboring sensor nodes become faulty then the false alarm rate increases rapidly. The majority voting approach depends on the degree of sensor nodes in the network, which inevitably affects the fault detection accuracy in the sparse network.

2.2.3 Hybrid Approach

The hybrid approach is a combination of both centralized and distributed approaches. This approach was initiated to overcome the loopholes of centralized and distributed approaches. The hybrid approach performs the fault management with two decision components such as sink node and each sensor node in the network. It uses multi-tiered architecture, in which sensor nodes are arranged in cluster head and cluster members. Each cluster member sends its diagnostic information to the corresponding cluster head. After analyzing the information the cluster head decides the fault status of sensor nodes with the help of the base station. The hybrid approaches also consist of different techniques which are discussed as follows.

- **Cluster based approach**

  Wang and Chen proposed a cluster based fault detection approach using trust matrix evaluation [58]. The sensor nodes are arranged in disjoint clusters in which, the cluster head acts as a sink for each cluster region. The cluster head calculates a similarity matrix between the sensor node and its neighboring sensor nodes. Next an indirect similarity matrix is computed among the sensor nodes in the cluster region for fault detection. The trust value lies between [0,1]. Value closer to 1, higher the similarity between sensor nodes. Here, data faults are considered for diagnosis.

  Afsar et al. proposed a fault tolerant service (FTS) for clustered sensor network [59]. The FTS protocol divided into three phases such as fault detection, fault diagnosis and fault recovery. In the beginning stage, the sensor nodes formed into a group of clustered based on energy efficient distance based clustering technique (EEDC). In the fault detection phase, detection is performed by heartbeat message exchanges. In the fault diagnosis phase, the time redundancy mechanism is used for transient fault detection in cluster head and cluster members. In the fault recovery phase, the cluster head recovered by sparse cluster head using re clustering operation and the cluster member recovered by removing the faulty cluster member from the routing table.

  Zafar et al. suggested a hybrid cluster based fault diagnosis architecture for WSNs [60]. The architecture uses a diagnostic agent for periodically monitoring of sensor nodes, local cluster head within the cluster region and error specific cluster head to deal with certain types of error by error to fault mapping. The fault detection accuracy is improved by cluster based approaches, but the network cost is also increased. Nitesh and jana proposed a clustering based distributed fault detection approach for two tier WSN [61]. The protocol uses neighbors table information and time redundancy mechanism for detection of hard permanent and transient faulty relay nodes in the network.

- **Probabilistic based approach**

  Titouna et al. proposed a two level sensor fault detection in WSN using Hybrid Hierarchical fusion based outlier detection technique [62]. In the first level, detection occurs inside the sensor nodes using Bayesian classifier. In this level, each sensor node computes a temporal and spatial correlation between current and previous sensor readings using maximum posteriori (MAP) technique. In the second level, the detection is performed by the cluster heads. The cluster takes a decision on fault status of sensor, based on the sensed data and previous phase results. If the sensor data is detected as fault free in two levels of testing, then the cluster head sends it to the sink. Similar to the above, the authors have proposed another hybrid fault detection scheme (FDS) using Bayesian classifier in two successive levels [63]. In the first level, the sensor node self diagnose the fault by using conditional probability and joint

probability table. The conditional probability is computed by using the sensed data and remaining energy level of the sensed node. The joint probability is computed by all the parameters with chain rules, then a threshold checking is performed for fault detection. This first level decision is transmitted to the next level for final decision. The cluster head takes a final global decision about fault status by comparing the joint probabilities of sensor nodes in the cluster region.

- **Majority voting based approach**

  Wu et al. proposed a two layered fault detection scheme using majority voting approach [64]. This protocol divided into two stages. The first stage performs local detection of faulty sensor nodes by itself and the second stage performs global detection of faulty sensor nodes using a fusion center. Local decision of sensor nodes are stored in a record table with binary decision in the first stage. In the second stage, the fusion center computes a rate of decision using the record table. The rate values of sensor nodes are divided into some intervals and the maximum rate is decided by majority voting technique. The smallest rate group has decided as faulty in the network.

  Kaur et al. proposed an agreement based fault detection scheme to detect the failure of Cluster Heads in a clustered WSN [65]. The protocol focuses on detection of faulty cluster head using an agreement of cluster members by periodically transmission of heartbeat messages. In this method the authors considered hardware failure of cluster heads.

  Nguyen et al. proposed a fault detection and classification scheme based on neighborhood scheme and time series analysis [66]. In this protocol, the faulty nodes are detected by neighborhood voting scheme and time series analysis using ARMA (Autoregressive Moving Average) model. The faults are classified based on the frequency and pattern checking of the faulty readings. The authors considered hardware faults such as hardware failure, drift faults, and data faults such as random faults and gain faults.

- **Mobile sink based approach**

  Due to link failure the network structure can frequently change. Chanak et al. presented a mobile sink based distributed fault detection protocol for WSN [67]. The mobile sink is also known as mobile robot, which moves in the network to diagnose the hardware and software components of sensor nodes. The mobile detector starts its diagnosis from the base station. By obtaining the health status of sensor nodes, it then uploads the information in the network. By returning to the base station, it completes its operation.

Abo-Zahhad et al. proposed a mobile sink based adaptive immune energy-efficient clustering protocol (MSIEEP) to solve energy hole problem and increases the network lifetime of WSN [68]. The node which is nearer to the base station depletes more energy due to multi-hop communication, which creates energy hole problem and leads to permanent disconnection of the network. An adaptive immune algorithm is used to find the perfect location of mobile sink nodes and optimum number of cluster heads to improve the network lifetime and connectivity problem.

- **Drawbacks of Hybrid Approach**

  In the cluster based approaches, the cost of the network increases due to the formation of clusters. The network lifetime is also affected as the cluster head takes some extra burden in the diagnosis process. The cluster head selection and re clustering operation also increase the workload of the sensor network. In mobile sink based approaches, the performance depends upon the path planning traversal and hub point selection. It has also a greater detection delay.

2.3 Fault Diagnosis using AIS Approaches

There is a greater interest seen among scientists and researchers in developing biologically inspired algorithms in the past. Artificial immune system (AIS) is considered as one of the most popular approaches due to its principle [69]. AIS is influenced from the principle of human immune system (HIS), which can expertly saved our bodies from bacteria and viruses [70]. The AIS is widely used in anomaly detection [71], pattern recognition [72], computer security [73] and fault detection [74]. AIS is a sub field of biological inspired computing and is a part of computational intelligence as shown in Figure 2.2. There are four algorithms of the AIS which are typically modeled for problem solving based on the features of the immune system. The AIS field is concerned with abstracting the role of the immune system and studying the application of the system towards solving computational problems. The algorithms explain the behavioral function of HIS which is also applied to fault diagnosis of WSN.

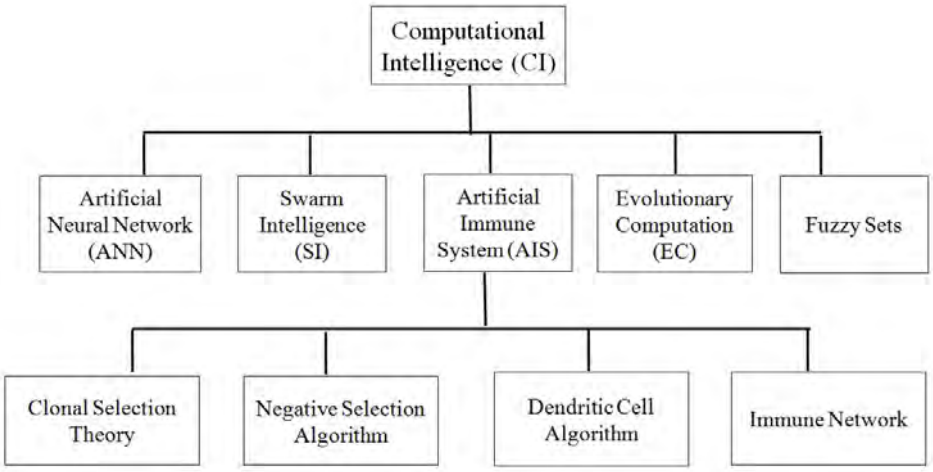

Figure 2.2: Branches of computational intelligence (CI)

2.3.1 Background of AIS

The clonal selection principle was suggested by Burnet in 1959 [75]. The principle explains how the cells of an immune system will react when a specific antigen enters the body. It establishes the idea that only those cells which are capable of recognizing an antigen will proliferate while other cells will not. Clonal selection works on both B cells and T cells. Any kind of molecule that can be identified by the adaptive immune system is known as antigens. Whenever an antibody in a B cell binds with an antigen, the B cells become activated, and it begins to proliferate. The newly produced B cells (i.e., clones) are the exact copy of the parent B cell. Then they will go for somatic hyper mutation and produce antibodies that are relatively specific to the antigen.

The immune system of a human body has a self defense mechanism, which can protect the body from being attacked by various bacteria or viruses. The main purpose of the immune system is to find out all the cells or molecules in the body and categorize them into self(fault-free) and non-self (faulty). Self usually refers to the acceptable or fault free data, and non-self refers to the measured data. There are two steps in negative selection algorithm (NSA). In step 1 there is a match between self strings and the strings which are generated randomly. The matched strings get rejected as per the principle of NSA. The strings that do not match are moved to the detector set. In step 2, protected strings are matched with those in the detector set. The strings that get matched are identified as non-self, and the rest are matched again till the faulty nodes are detected.

The self/non-self discrimination principle is a widely accepted method. It was believed that, the immune system got activated when our body recognizes an antigen or foreign entities. But later, the researchers believe that activation of immune system requires

the presence of danger signal additionally recognition of pathogens. Due to pathogenic infections, the body cells got damaged which indicates danger. This recognition is performed by dendritic cells (DCs) of the immune system [76]. The central idea of danger theory (DT) is that, the immune system responds to danger rather than non-self. Instead of responding to foreignness, the immune system reacts to danger. The danger is measured by damage to the cell i.e., when a cell dies an unnatural death [77]. The danger signal establishes a danger zone around itself.

Jerne originally proposed the concept of the immune network in 1974 [78]. In contrast to the concept of clonal selection, the theory of the immune network hypothesizes that the immune system maintains a regulated network of cells and molecules which establish connections not only between an antibody and an antigen, but also between the antibodies itself which creates a network of antibodies.

2.3.2 Clonal Selection Principle

To diagnose the hydraulic brake fault in automobile, the authors have used clonal selection classification algorithm inspired from the clonal selection theory [79]. Nine fault conditions were simulated and tested for each fault condition in a brake system. The simulation result shows that the approach reported by their work gives better classification accuracy as compared to other machine learning approaches. For the fault detection and analysis occurred in machines, Gan et al. proposed a fault detection system based on clonal selection programming [80]. By performing different tests at different conditions the above method is found to be suitable for practical industrial applications.

In the paper [25], the concept of artificial immune system is used to identify the faults in a large wireless sensor network. According to the authors their proposed approach is better and efficient because of faster diagnosis. In the paper [81], faulty sensor nodes are detected using AIS's clonal selection principle and categorized by probabilistic neural network strategy into corresponding types. The faulty sensor nodes are also isolated in the isolation phase. Performance of the proposed algorithm is compared and simulated with existing algorithms and the simulation result shows that the suggested algorithm provides better results.

The major drawback of this approach is the overhead of cloning, mutation for computing affinity (fitness) value towards the solution. This has been overcome by the next following approach negative selection algorithm (NSA) which is given as follows.

2.3.3 Negative Selection Algorithm

In the paper [82], a fault diagnosis algorithm is proposed by combining both the clonal selection principle and negative selection algorithm, which determines the fault types properly. By optimizing the mutation operator, the convergence rate of antibody generation in the detector set was also improved. The fault diagnosis model was tested by experiments.

The vibrating signals were collected and transferred by a WSN. The data were analyzed and diagnosed based on the fault diagnosis model. In the paper [83], using NSA the authors have proposed a motor fault diagnosis scheme. The motor faults can be encountered using a hierarchical structure. This structure efficiently detects the incipient motor faults as well as the fault types. In the simulation, the authors have examined the fault diagnosis method using two real-world problems.

The authors proposed a multi-operational algorithm using the negative selection principle [84]. For comparison with other algorithms, they have used the fault model of DC motor as a benchmark. In the paper [85], the authors have proposed two novel negative selection algorithm. Usually, the detectors are generated randomly, but in this work the detectors are generated in a non random ways which eliminates the training time of detectors. To examine the performances of both the experiments performed on the iris data set, ball bearing fault data set and two-dimensional synthetic data sets. The result shows that in most of the cases they give better results than the others. In the paper [86], the authors have proposed one type of data flow attack called a Sybil attack. They have implemented an improved version of NSA with learning capability, and they have also used R contiguous bit matching rule. They have compared their work with other works by taking three performance parameters such as false positive, false negative and detection rate and their work shows better results than the others.

In the paper [87], the authors examined the use of AIS to recognize the fault trends in supermarket freezer cabinet temperature information. Their primary objective is to identify the early signs of icing up in supermarket refrigerated cabinets. In addition to information encoding, the r bit matching rules provide a precise classification rate of faulty data. In the paper [88], inspired by an immune system, fault detection & isolation of a wind turbine system has been proposed. To detect and isolate both individually and simultaneously occurring faults, the authors have designed an NSA which is hierarchical in nature. To evaluate the proposed work, a non-parametric statistical comparison test has put under various fault circumstances. The simulation result shows that the NSA and support vector machine (SVM) give same performances while in certain fault circumstances NSA gives a better result than SVM.

In the paper [89], the authors have proposed a method for classification of real noise in speech sentences based on NSA. To validate the proposed method they have taken six types of real noise. This method shows better results than the classical classifiers in terms of accuracy. Aydin et al. proposed a chaotic-based NSA for anomaly detection [90]. Both clonal and negative selection approach are used to generate detectors. The detectors generated in the training stage are used to check the performances in the testing stage. The authors have also used the KNN method to generate detectors. They have analyzed their work in the broken rotor bar fault detection and Fisher Iris datasets.

The major drawback of this approach is the overhead of generating random numbers of

detectors. It is also very difficult to know whether the number of generated detectors is large enough, which can satisfy the detection fault probability. This has been solved in the next diagnosis approach based on the dendritic cell algorithm which is presented as follows.

2.3.4 Dendritic Cell Algorithm

A fault detection and isolation (FDI) method is developed using the dendritic cell algorithm in [91]. The authors have applied this method to a wind turbine test model. Their proposed method can detect as well as isolate sensor faults. A statistical comparison test is carried out to compare the performances of their proposed scheme. In the paper [92], a model was proposed which describes the mechanism of biological differentiation of dendritic cells (DC). This model abstracts the information of dendritic cell fusion process, defines the functions of external signals which are applied to WSN and defines the mathematical model of DC. A real time intrusion detection was performed and the performances were analyzed by scalability, complexity and robustness which gives better detection with less energy consumption. Usually Dendritic cell algorithm is applied for fault diagnosis in various mechanical system. In this thesis, Dendritic cell algorithm is applied for WSN to check the improvement of previous approach i.e., NSA in terms of performance parameters such as FDA, FAR, FPR, FDL and EC. The DCA performs better as compared to the algorithm proposed using clonal selection principle and negative selection algorithm.

2.3.5 Artificial Immune Network

In [93], the authors suggested a new method for the diagnosis of faults using artificial immune network. They have combined it with radial basis function (RBF) of neural network and the structure is same. Compared to RBF the suggested method has less hidden layers and the diagnosis rate is better. Wang et al. proposed an artificial immune network coupled with the fuzzy c-means clustering to detect the types of faults in transformer [94]. The experimental results indicate that, their suggested algorithm can efficiently classify the power transformer fault types .

In [95] the authors have investigated the fault diagnosis of plant systems using an immune network. The origin of failure can be detected by calculating the failure of each unit locally. The authors have carried out a simulation to validate their proposed method. To improve the capability of interpreting the result of dissolved gas analysis, the authors have proposed an artificial immune network classification (AINC) algorithm [96]. By mimicking the learning and defensive mechanism of the immune system, AINC responds to the fault samples of power transformer. The algorithm proposed by the authors gives better diagnosis accuracy and effectively classify the faults by testing many real fault samples. However, these algorithms have not been applied in fault diagnosis of WSN despite their potential for efficient fault diagnosis. The proposed algorithm based on artificial immune network (AIN)

Table 2.1: Existing set of protocols with respect to different fault diagnosis parameters

Author	Year	Network		Diagnosis method			Persistence of fault			Fault type		Approach
		Static	Dynamic	Centralized	Distributed	Hybrid	Permanent	Intermittent	Transient	Hard	Soft	
Panda et al. [27]	2014	✓		✓			✓				✓	statistical based
Panda et al. [9]	2015	✓			✓		✓			✓	✓	statistical based
Panda et al. [47]	2015	✓			✓			✓	✓		✓	statistical based
Jin et al. [29]	2015	✓		✓				✓	✓		✓	statistical based
Tang et al. [37]	2016	✓		✓				✓	✓		✓	probabilistic based
Yuan et al. [48]	2015	✓			✓			✓			✓	probabilistic based
Dhal et al. [38]	2015	✓		✓				✓	✓		✓	probabilistic based
Chen et al. [43]	2006	✓			✓		✓				✓	majority voting based
Xu et al. [44]	2014	✓			✓		✓	✓			✓	majority voting based
Saihi et al. [45]	2013	✓			✓		✓			✓		majority voting based
Sahoo et al. [52]	2014	✓			✓		✓	✓		✓	✓	comparison based
Chanak et al. [53]	2016	✓			✓		✓		✓	✓	✓	soft computing based
Mahapatro et al. [57]	2014	✓			✓		✓	✓		✓	✓	soft computing based
Zafar et al. [60]	2015	✓				✓		✓	✓		✓	cluster based
Nitesh et al. [61]	2015	✓				✓	✓		✓	✓	✓	cluster based
Guo et al. [33]	2009	✓		✓			✓				✓	sequence based
Chanak et al. [67]	2016	✓				✓	✓	✓	✓	✓	✓	mobile sink based
Mohapatra et al. [25]	2017	✓			✓		✓			✓	✓	clonal selection based
Mohapatra et al. [81]	2019	✓			✓		✓	✓	✓	✓	✓	clonal selection based
Ramsha et al. [71]	2015	✓			✓						✓	negative selection based
Gao et al. [83]	2014	✓				✓	✓	✓	✓	✓	✓	negative selection based
Chen et al. [82]	2013	✓			✓						✓	negative selection based
Alizadeh et al. [91]	2017	✓			✓					✓	✓	dendritic cell based
Xin et al. [92]	2017	✓			✓						✓	dendritic cell based
Wang et al. [94]	2014	✓			✓						✓	artificial immune network based
Hao et al. [96]	2017	✓			✓			✓	✓		✓	artificial immune network based

gives better result than clonal selection principle (CSP). Though the proposed algorithm is similar to CSP in terms of computation of affinity, it differentiates the algorithm using CSP presented in chapter 3 with respect to consideration of different fault types and added it to memory antibodies so that it can learn and memorize the same types of faults. Hence, the classification accuracy is also improved. The proposed algorithm performs better as compared to that of clonal selection principle. Table 4.3 gives an idea of the existing set of protocols with respect to different fault diagnosis parameters.

2.4 Summary

The fault diagnosis in WSN is an important field of research both for academic and industry. The WSN consisting of large sensor nodes have plenty of applications such as battle field, forest fire, environment monitoring, agricultural monitoring and so on. Since these sensor nodes are crucial for data collection from deployed environment which are hostile and human inaccessible, the diagnosis has become essential instead of necessity.

Fault diagnosis is rather important, particularly when the sensor nodes are used for safety critical applications. The traditional diagnosis algorithm for wireless sensor network are based on centralized, distributed and hybrid approach. The diagnosis algorithms are classified into different types based on different criteria such as majority voting based, neighborhood coordination based, statistical based, comparison based approaches. Apart from traditional diagnosis methods in this chapter we have highlighted the fault diagnosis in WSN based on artificial immune system which is one of the recent concept developed and serves as the method to solve the fault diagnosis problems for WSN. The various shortcomings of different existing algorithms have also been discussed in this chapter.

Chapter 3

Fault Diagnosis in WSN using Clonal Selection Principle and Probabilistic Neural Network Approach

3.1 Introduction

Due to the physical limitations and environmental conditions, sensor nodes are subjected to different types of faults. For example, when they are deployed in hostile and unattended environment, these nodes are affected by physical damage, out of range, incapable of recharging the batteries and change in behavior of nodes [81]. In order to obtain accurate data from the deployed wireless sensor network (WSN), the sensor nodes need to be fault free. Motivated by the need of a fault free WSN in the deployed field, it is crucial to identify the faulty sensor nodes in WSN and categorize them into their respective fault types so that it can be possible to isolate each of the faulty sensor nodes in the network.

In order to obtain a fault free WSN, and to prevent the network from degradation of service, the objective of this work is to design, develop and implement a fault detection algorithm and then classify the faults. To diagnose the faulty sensor nodes, there are various types of fault diagnosis algorithms available in the literature which are based on different approaches such as, statistics [9], neighboring coordination [52], comparison based [26] and neural network [22]. This chapter presents an efficient fault detection algorithm based on clonal selection principle of an artificial immune system (AIS). The way the clonal selection principle of AIS resembles to the functioning of human immune system, a resemblance of the clonal selection principle with the fault diagnosis in WSN is proposed in this work. The immune system has drawn significant attention to solve the complex computational problems [97] [98]. The primary role of the immune system is to protect the body from the foreign pathogens, called antigens such as bacteria and viruses. It is used to distinguish between the body's own cells (self) and the foreign cells (non-self or antigen). Therefore, throughout the thesis faulty and fault free sensor nodes are considered as antigens and antibodies, respectively. The major contribution of this chapter is stated as follows:

- A clonal selection based method of artificial immune system (AIS) is used to detect the soft permanent faults for WSN.

- A fault status rate with respect to threshold time is introduced to detect the soft transient and intermittent faults.
- A repeated time out mechanism with respect to response message is used to detect the hard permanent faults.
- A fault classification phase based on probabilistic neural network approach is used to classify the faults into their respective types.
- A fault isolation phase is introduced for isolation of the faulty sensor nodes.
- A validation of the proposed method is carried out using NS-2.35 simulator and comparison study is also performed with existing approaches.
- Different metrics such as fault detection accuracy (FDA), false alarm rate (FAR), false positive rate (FPR), fault classification accuracy (FCA), false classification rate (FCR), fault detection latency (FDL) and energy consumption (EC) are used to measure the performance of our proposed algorithm.

The organization of the chapter is as follows. Section 3.2 discusses the basic principle of the clonal selection theory. The proposed fault diagnosis algorithm is given in Section 3.3. The results and discussions are given in Section 3.4. Finally, the chapter is concluded in Section 3.5.

3.2 Basic Principle of Clonal Selection Theory

The clonal selection theory was suggested by Burnet in 1959 [75]. The theory explains how the cells of an immune system will react when a specific antigen enters the body. It establishes the idea that only those cells which are capable of recognizing an antigen will proliferate while other cells will not. Clonal selection works on both B cells and T cells. Any kind of molecule that can be identified by the adaptive immune system is known as antigens. Whenever an antibody in a B cell binds with an antigen, the B cells become activated, and it begins to proliferate. The newly produced B cells (i.e., clones) are the exact copy of the parent B cell. Then they will go for somatic hyper mutation and produce antibodies that are relatively specific to the antigen. Figure 3.1 depicts the clonal selection principle.

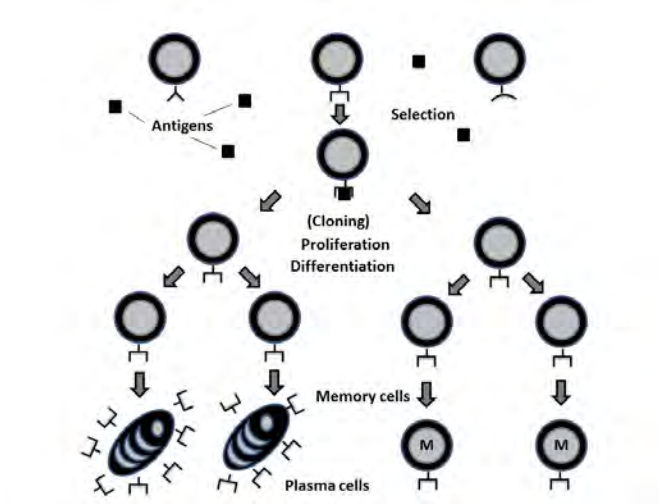

Figure 3.1: Clonal Selection Principle

3.3 Proposed Algorithm

WSN comprise of a huge number of homogeneous sensor nodes, which are initially fault free when they are deployed in the target field, but in due course of time some nodes may be faulty due to their physical limitation and environmental condition. Fault diagnosis in WSN can be designed to mimic the operations of the human immune system, where our body is protected from the foreign pathogens called as antigens. According to the immune system, the faulty nodes can be considered as antigens and the fault free nodes can be considered as antibodies.

There are three phases of the proposed algorithm such as : fault detection, fault classification and fault isolation phase.

3.3.1 Fault Detection Phase

This section is again classified into two phases, (1) detection of soft fault and (2) detection of hard fault.

Detection of Soft Fault

Sensor nodes communicate with their neighbor sensor nodes within the transmission range. Each sensor node generates a fault status, i.e, zero (0) for fault free and one (1) for faulty node. The actual fault status (fs) for N number of sensors is considered as a set of antibodies(AB), $AB = \{fs_1, fs_2, fs_3, ..., fs_N\}$. Each sensor node $sn_i \in S$, having sensor data $\{sd_1, sd_2, sd_3, ..., sd_m\}$. The sensor node $sn_i \in S$ with sensor data values $sd_i = \{sd_1, sd_2, sd_3, ..., sd_m\}$ are compared to each neighbor $sn_j \in S$ with sensor values $sd_j = \{sd_1, sd_2, sd_3, ..., sd_m\}$. If $|sd_i - sd_j| > \theta$, then fault status fs is set as 1, otherwise fs is set as 0. Here θ is a threshold value for

comparison of data sensed between sensor nodes. After comparison with each neighbor, each sensor node $sn_i \in S$ generates a fault status, which is called as antigen (ag). So, for N number of sensor nodes, N number of antigens are generated, $AG = \{(fs_1, fs_2, ..., fs_m)_1, (fs_1, fs_2, ..., fs_m)_2, (fs_1, fs_2, ..., fs_m)_3, ..., (fs_1, fs_2, ..., fs_m)_N\}$.

The node $sn_i \in S$ has compared the sensed data with its neighbor node $sn_j \in S$ for generation of fault status. Here $sn_i \in S$ is called as a comparator (cr) node and $sn_j \in S$ is called as compared (cd) node. The comparison between cr and cd will generate three (3) conditions. (i) If the cr and cd node are fault free, then the fault status fs is always zero (0). (ii) If the cr is fault free and cd is faulty or vice versa, then the fault status fs is always one (1). (iii) If both cr and cd are faulty, then the fault status is either zero (0) or one (1). For the 3^{rd} condition, the comparator node (cr) is called as suspected node. Figure 3.2 shows the different conditions of fault status (fs_{ij}) generation between two sensor nodes $sn_i, sn_j \in S$.

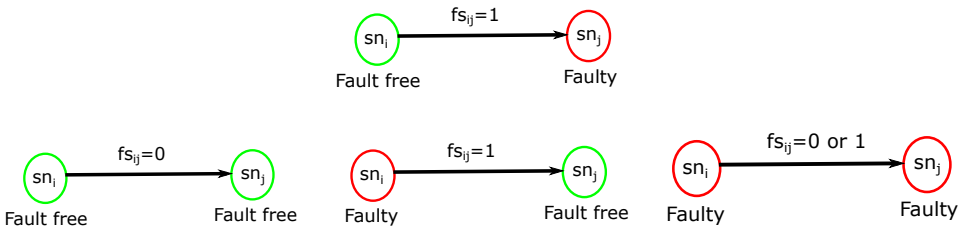

Figure 3.2: Different conditions of fault status (fs_{ij}) generation between two sensor nodes $sn_i, sn_j \in S$.

The actual fault status (initial fault status of the network), i.e., antibodies (AB) is stored in ab_i and sensor node generated fault status antigen (ag) is stored in ag_i. Initially, all the nodes are fault free, after adding the faulty nodes the probable fault set is increased rapidly. The possible antibodies set of N nodes with fault probability p_1, p_2, p_3,...,p_k is obtained as $\binom{N}{p_1} + \binom{N}{p_2} + \binom{N}{p_3} + ... + \binom{N}{p_k}$. If we compared the antigen of each sensor node with probable antibodies set, then the computation cost increases rapidly. So in this proposed work, we have compared each generated antigen of both comparator and compared node to obtain the possible fault set.

The comparison results of sensor node sn_i with its neighbor are stored in the antigen $ag_i[.]$. For each i^{th} comparator sensor and j^{th} compared sensor of antigens are compared with each other. Each node $sn_i \in S$ generated antigen $ag_i[.]$ is compared with each one-hop neighbor $sn_j \in S$ generated antigen $ag_j[.]$. Each comparison between the antigens generates a test syndrome $\sigma(i, j)$. The test syndrome $\sigma(i, j)$ is defined in Equation (3.1).

$$\sigma(i, j) = |(fs_1, fs_2, ..., fs_m)_i \cap (fs_1, fs_2, ..., fs_m)_j|, \tag{3.1}$$

where $(fs_1, fs_2, ..., fs_m)_i$ is the fault status of sensor node $sn_i \in S$ and $(fs_1, fs_2, ..., fs_m)_j$ is the fault status of sensor node $sn_j \in S$. If the fault status between antigens are matched, then the test syndrome $\sigma(i, j)$ value is the cardinality of the number of matches. Affinity (af)

usually corresponds to a metric which identifies the degree of similarity. The affinity af is the ratio between the test syndrome and the number of comparison. The affinity $af(i,j)$ is defined in Equation (3.2).

$$af(i,j) = \frac{\sigma(i,j)}{|No.of comparision|} \tag{3.2}$$

The affinity value is $af(i,j) \in [0,1]$. If the fault status between the antigens are matched, then $af(i,j)$ is calculated as one (1). For fully unmatched antigens the $af(i,j)$ is calculated as zero (0), and for partially matched the antigens $af(i,j) \in 0,1$.

After affinity calculation the probable fault set is retrieved. For affinity value one, (where the antigens are matched) the probable fault set $ag_i = fs_1, fs_2, ..., fs_m$ may be considered as an actual fault set after total affinity value calculation. When the affinity value is zero or less than one (< 1), then each antigen undergoes for cloning to detecting the probable fault set with a minimum search space. In fact, the antigens are selected for cloning according to the affinity values. The number of clones produced for each antigen is proportional to its antigenic affinity. The number of clones can be computed as presented in Equation (3.3).

$$N_c = round\left((\beta * T)/i\right) \tag{3.3}$$

Where, for each antigen β is a multiplication factor, T is the total number of antigens, i is the current antigen and round (.) is the operator that round its variable towards the closest integer. Then a mutation occurs which enhances the diversity as well as expands the search space for finding the solution. After the mutation, the affinity of maturated clones are calculated until the stopping criteria is reached. The cloning is basically performed to match the antigens for retrieving the probable fault set. Then for each probable fault set (pf), the summation of affinity $\sum af$ is computed.

The affinity $af(i,j)$ of fault set $pf(i,j)$ is the hamming distance between antigens ag_i and ag_j, respectively. The summation of all affinity $\sum af(i,j)$ for each fault set $pf(i,j)$ is provided the actual fault status. The highest sum affinity of the probable fault set $pf(i,j)$ is considered as the actual fault set or antibody of the sensor network. So, the faulty nodes are detected successfully by using the actual fault set. The location of the suspected nodes is found out by these following Algorithm 3.1.

Algorithm 3.1 FDCSP Algorithm

Initialize multiplication factor (β), initial antibodies (ab), initial antigen (ag), and threshold (θ);
for each sensor node i = 1 to N **do**
  for each neighbor sensor j = 1 to m **do**
    Senor node $sn_i \in S$ compared the sensed value with each one-hop neighbor $sn_j \in S$;
    if $|sd_i - sd_j| > \theta$ **then**
      $fs_{ij} = 1$
    else
      $fs_{ij} = 0$
    end if
    antigen (ag_i) $\leftarrow fs_{ij}$
  end for
end for
for i = 1 to N **do**
  for j = 1 to m **do**
    Compared antigens $ag_i[.]$ with $ag_j[.]$;
    Calculate test syndrome $\sigma(i,j) \leftarrow |ag_i \cap ag_j|$;
    Calculate affinity $af(i,j) \leftarrow \frac{\sigma(i,j)}{|No.of comparison|}$;
  end for
end for
for each affinity $af(i,j)$ **do**
  if $af(i,j) \neq 1$ **then**
    Produce clones
    Number of clones (N_c) = round $((\beta * T)/i)$
    Perform mutation
    performed up to $af(i,j) == 1$
  end if
end for
for each affinity $af(i,j)$ **do**
  Generate the probable fault set $pf(i,j)$;
end for
for each unique fault set $pf(i,j)$ **do**
  Calculate the respective affinity summation $\sum af(i,j)$;
end for
Sort the unique fault set $pf(i,j)$ according to the total $af(i,j)$ value in ascending order;
$pf_k \leftarrow pf(i,j)_1, pf(i,j)_2, ..., pf(i,j)_k$;
The highest $\sum af(i,j)$ value for $pf(i,j)$ is considered as faulty node set.
In the fault set $pf(i,j)$ the faulty nodes are detected successfully;

The fault detection using clonal selection principle, i.e., (FDCSP) algorithm detects the soft faulty nodes in the network, but the soft faulty nodes are classified into their respective types depending upon the behavior with respect to the time units. The description of these faults is explained in Section 1.3. These different soft faults are identified by a sequence of fault status from time units T=1 to ΔT, where ΔT is a maximum observation time unit. So for the diagnosis of different types of faulty nodes, we have to examine each sensor node $sn_i \in S$ for time unit 1 to ΔT. The FDCSP algorithm runs from 1 to ΔT time units and generates the fault status fs for each time unit. So for sensor node sn_i fault status fs_i is checked for each time unit 1 to ΔT. According to soft faulty behavior for permanent fault, the fault status is fixed as one for every time unit of 1 to ΔT. In case of intermittent and transient, the fault status is fixed as one for some arbitrary time unit of 1 to ΔT and set as zero for remaining time unit of 1 to ΔT. Then we calculate the summation of fault status rate fs_{rate} by using Equation (3.4)

$$\frac{1}{\Delta T} \times \sum_{i=1}^{\Delta T} fs_i \tag{3.4}$$

After calculating the fs_{rate}, then it compares with a minimum time unit δT for which the

node is determined as faulty nodes. The condition $fs_{rate} \geq \delta T$ holds good for different types of soft fault identification in the WSN. Here, δT is a minimum threshold time unit and depending upon the network applications. For our protocol implementation the δT value is set as 0.2. In this fault detection phase, the different faulty nodes are detected with respect to the time units. But the actual class of the fault is not determined precisely with respect to the time units. So the different patterns of fault behaviors are classified in the fault classification phase, which is described in the next Section 3.3.2. The soft fault detection is presented in Algorithm 3.2.

Algorithm 3.2 Soft Fault Detection Algorithm

Input: Sensor node data values $sd_i = \{sd_1, sd_2, sd_3, ..., sd_m\}$, Neighbors list ($Neg(.)$);
Output: List of different soft faulty nodes (permanent, intermittent, & transient);
Initialize: T, ΔT, δT, $fs_i \in \{0, 1\}$, & $count_i = 0$;
for sensor node sn_i, $i = 1$ to N **do**
  for time unit $T = 1$ to ΔT **do**
    Sensor node $sn_i \in S$ having one hop neighbor set $Neg(sn_i) \in S$;
    FDCSP(.) algorithm (ref. Algo. 3.1) runs with sensor sn_i & neighbors $Neg(sn_i)$ sensed data values;
    $fs_i^{(T)}$ = FDCSP($sn_i, Neg(sn_i)$);
    if ($fs_i^{(T)} == 1$) **then**
      $count_i = count_i + 1$;
    end if
  end for
  $fs_{rate}(sn_i) = \frac{1}{\Delta T} \times count_i$;
  if ($fs_{rate}(sn_i) \geq \delta T$) **then**
    Sensor sn_i is soft faulty (permanent, intermittent, & transient);
  else
    Sensor sn_i is fault free;
  end if
end for
The fault information is broadcast in the WSN;

Detection of Hard Fault

For the hard faulty node detection, we have used a repeated timeout mechanism. Every sensor node broadcasts a request message req_{msg} to its one-hop neighbor sensors within the transmission range for a repeated time frame sequence 1 to δT and wait timeout (t_{out}) period for response message rep_{msg}. The t_{out} period is decided by the initial stage of the network having fault free sensors. If the node receives the rep_{msg} of its neighbor sensor within t_{out} time period, then the corresponding counter variable is increased with respect to the neighbor sensor. If the node does not receive the rep_{msg} of its neighbor sensor within t_{out} period, then the corresponding counter variable is unchanged. After δT time frame, for each neighbor sensor the summation of counter variable is computed. If the summation of counter variable is greater than or equal to $\lceil \frac{\delta T}{2} \rceil$, then the sensor node is examined as fault free in nature. If the summation of counter variable is less than $\lceil \frac{\delta T}{2} \rceil$, then the sensor node is examined as faulty in nature. After examination of the hard fault status, each node immediately sends the fault status to the controller (base-station). The hard permanent fault detection is described in Algorithm 3.3.

An overview of the time frame sequence with t_{out} period is shown in Figure 3.3. For each time instance, the value of the counter variable is updated after t_{out} time period. The

fault status of a node sn_j is decided by a condition checking. The condition for hard fault detection is defined in Equation (3.5).

$$sn_j = \begin{cases} \sum_{T=1}^{\delta T} count_j^{(T)} \geq \lceil \frac{\delta T}{2} \rceil & \text{, fault free} \\ \text{Otherwise} & \text{, faulty} \end{cases} \tag{3.5}$$

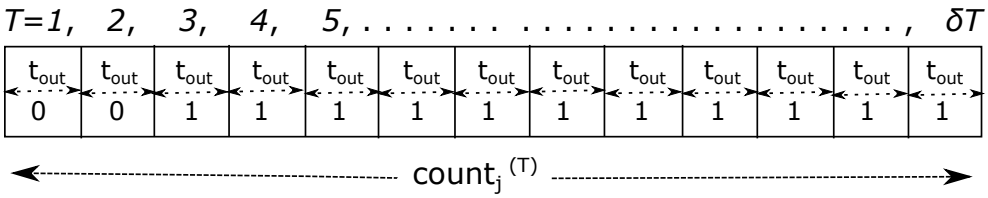

Figure 3.3: An overview of time frame updation over each t_{out} period for hard fault detection

Algorithm 3.3 Hard Fault Detection Algorithm

Input: Neighbors list ($Neg(.)$), timeout period (t_{out}), $count = 0$;
Output: List of hard permanent faulty nodes;
for sensor node $i = 1$ to N **do**
  for time frame 1 to δT **do**
    Sensor node $sn_i \in S$ sends a request message req_{msg} to its one hop neighbors $Neg(sn_i)$;
    $sn_i \in S$ waits for t_{out} unit of times for the response message rep_{msg};
    for each neighbor $sn_j \in Neg(sn_i)$ **do**
      if sn_i received the rep_{msg} && response time $\leq t_{out}$ **then**
        $count_j = count_j + 1$;
      else
        $count_j = count_j$;
      end if
    end for
  end for
  for each neighbor $sn_j \in Neg(sn_i)$ **do**
    if $count_j \geq \left\lceil \frac{\delta T}{2} \right\rceil$ **then**
      the sensor node sn_i is fault free;
    else
      the sensor node sn_i is hard faulty;
    end if
    The fault information is broadcast in the WSN;
  end for
end for

3.3.2 Fault Classification Phase

After the faulty nodes are detected, they are classified into different types using probabilistic neural network (PNN) approach [99], [100]. PNN is an optimal classifier with faster order of magnitude than back propagation learning. PNN classifies an unknown pattern into a particular class and also has no local minima problem.

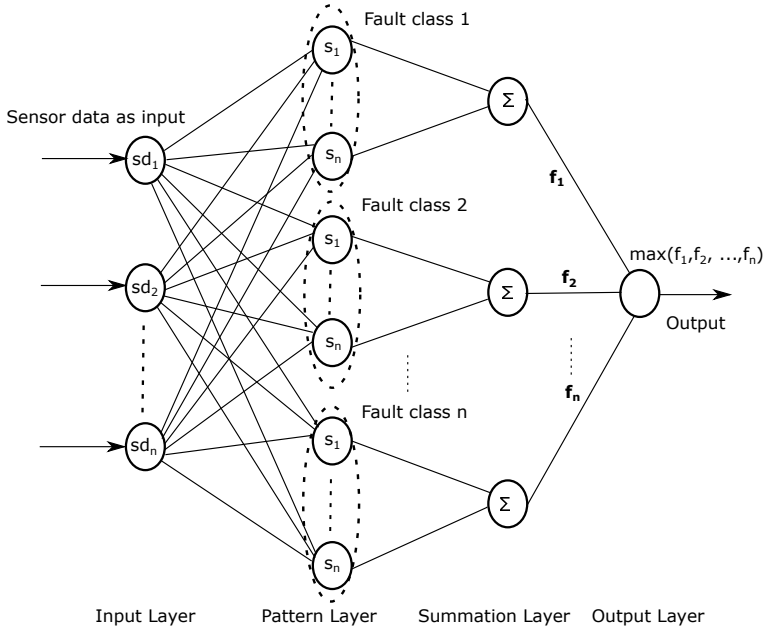

Figure 3.4: Probabilistic Neural Network (PNN) Architecture

In general, PNN has four (4) layers such as the input layer, pattern layer, summation layer and output layer[101]. The basic architecture of PNN is shown in Figure 3.4. The input layer has the number of feature vectors. The pattern layer has collection of groups or classes of representative patterns. Each group in the pattern layer receives the complete input feature vectors and performed a computation using Gaussian function. The summation layer has a simple average operation of each pattern layer group or class. The output layer has performed a maximum operation with the values obtained by the summation layer. So the largest value associated with the particular group or class is decided by the characterization of unknown feature vector.

In this protocol, the fault classification phase is performed in the base station. After fault identification, the base station collects the faulty node sensor data for fault classification. For each faulty node $sn_i \in S$ the sensor data values $sd_i \in \{sd_1, sd_2, sd_3, ..., sd_n\}$ are the input feature vectors and input to the input layer of PNN. Each node in the pattern layer is performing a Gaussian value computation using input feature vector with selected representative feature vector. The probability density function $pdf(.)$ for each pattern layer node using a Gaussian function is defined in the Equation (3.6).

$$pdf(sd_i) = \frac{1}{(2\pi)^{\frac{1}{2}}\sigma} \times exp(-\frac{(sd_i - s_i)^2}{2\sigma^2}), \tag{3.6}$$

where, sd_i is the i^{th} sensor values or input vector, s_i is the representative vector of respective classes, and σ is the smoothing parameter. The smoothing parameter σ is represented as standard deviation. The value of σ is selected at the time of representative pattern selection. PNN gives efficient accuracy when the σ value is in between 1 to 3. So here σ value is set

as one (1).

The selection of representative patterns is performed in the training phase. The pattern layer groups are dependent upon the number of fault types or classes. In this classification phase, we have considered three types of soft faults, so the pattern layer is divided into three (3) groups or classes. The class 1 is represented as permanent fault, class 2 is represented as intermittent fault, and class 3 is represented as transient fault. The value of each fault class is represented as $s_i = \{s_1, s_2, s_3, ..., s_n\}$. In general, we have selected such samples into the historical faulty sensor data values, which are moderately distributed with respect to fault classes and act as representative vector. Initially, we have collected the sensor data values and then applying k-means clustering algorithm to partition the n observations (sensor data values) into k sets (the value of k is dependent upon the fault set). The k-means clustering algorithm is used to partition the n patterns or observation into k clusters or groups, in which each pattern belongs to the fault class with the closest mean value. So the selected feature vector by k-means clustering algorithm is called as representative pattern or feature vector. The representative pattern also acts as the data points center for the Gaussian function in each pattern layer node computation.

After pattern layer computation by the Equation (3.6), then the output of the pattern layer is the input for the summation layer. The number of summation layer node is dependent upon the number of group or class in the pattern layer. In this case, we have taken three (3) summation layer nodes for three (3) fault classes. Each node in the summation layer performs an average summation operation with each group or class output of pattern layer. The output of the summation layer calculates the average probability density function of the pattern layer computations using Equation (3.7).

$$f_i(sd_i) = \frac{1}{n} \times \frac{1}{(2\pi)^{\frac{n}{2}}\sigma^n} \times \sum_{i=1}^{n} exp(-\frac{(sd_i - s_i)^2}{2\sigma^2}), \tag{3.7}$$

where n is the number of observations in the input set and also same as the number of patterns in each group of pattern layer, sd_i is the i^{th} sensor node value, s_i is the i^{th} representative pattern of respective fault class, and σ is the smoothing parameter. The output of summation layer is passed as input for the output layer. In the output layer a maximization operation is performed with pdf values of each summation layer node. The final output classifies a unknown input set $sd_i = \{s_1, s_2, ..., s_n\}$ by comparing the functional values in the output layer, i.e., $f_1(sd_i)$, $f_2(sd_i)$, $f_3(sd_i)$,..., $f_n(sd_i)$. The maximum value, i.e., $Max(f_1(sd_i), f_2(sd_i), f_3(sd_i))$ is the final output of the output layer, which is assigned to one of the fault classes. The $f_1(.)$, $f_2(.)$, and $f_3(.)$ functional values are related to permanent fault (class 1), intermittent fault (class 2), and transient fault (class 3), respectively. In other words, the unknown patterns nearer to the representative patterns having higher probability functional value and farther to the representative patterns having lower probability functional

value. The classification conclusion is decided by using the Equation (3.8).

$$c_i.f_i(x) > c_j.f_j(x), \text{all } i \neq j, \tag{3.8}$$

where $f_i(.)$ & $f_j(.)$ are the probability density functions (pdf) for class i & j, respectively, c_i & c_j are the cost of misclassification rate, and x is the unknown sample. If the Equation (3.8) is satisfied, then the unknown sample x belongs to the class i, otherwise the unknown sample x belongs to the class j.

The c_i & c_j are the misclassification rate of class i and j, respectively, where $i \neq j$. In general, the misclassification rate is the percentage of patterns that are misclassified with respect to the representative classes. The misclassification parameters are included in the probability density function of each of the samples to know the belongingness of the sample more accurately. In the training phase, the misclassification cost was derived, which is the cost of incorrectly classifying the unknown samples. The misclassification rate value depends upon the training phase data samples. In the implementation of the fault classification phase, the misclassification rate value for permanent, intermittent, and transient faults are computed as 0.010, 0.016, and 0.023, respectively.

Algorithm 3.4 Fault Classification Algorithm

Input: Faulty sensor node's data values;
Output: Faulty sensor nodes with actual fault information (i.e., permanent, intermittent, & transient fault);
Initialize: The input feature vector $sd_i = \{sd_1, sd_2, sd_3, ..., sd_n\}$;
Set the representative patterns using k-means clustering algorithm;
Sort the representative patterns into k sets (k fault classes);
for each $k = 1$ to 3 **do**
    for each sample $i = 1$ to n **do**
        Define the probability density function (pdf) by a Gaussian function using Equation (3.6);
    end for
    Performed the average summation operation of all Gaussian values using Equation (3.7);
end for
Performed the maximization operation with functional values of summation layer output (i.e., $f_1(.)$, $f_2(.)$, & $f_3(.)$);
Calculate, $max \leftarrow Maximum(f_1(sd_i), f_2(sd_i), f_3(sd_i))$;
if $(max == f_1(sd_i))$ **then**
    Sample is associated with class 1 (permanent fault);
else if $((max == f_2(sd_i)))$ **then**
    Sample is associated with class 2 (intermittent fault);
else if $((max == f_3(sd_i)))$ **then**
    Sample is associated with class 3 (transient fault);
else
    Sample associates with byzantine fault;
end if
The fault information message is broadcast in the WSN;

3.3.3 Fault Isolation Phase

The decision making process will be affected if faulty nodes are present in the WSN. So, after the actual fault status is detected, it is necessary to take an action against the faulty nodes in the WSN. The fault isolation step is used to isolate the faulty sensors in the WSN and the fault free sensors resides as in the network. Our proposed fault isolation phase is used to isolate the hard, soft permanent, and soft intermittent faulty nodes from the WSN. As the transient fault appears for δt small time interval, so the presence of transient fault may

not affect the decision making process of the end-user. Hence, the fault isolation phase does not consider the transient faulty nodes in the isolation process. In this isolation phase, four (4) scenarios are generated, which is described in Figures 3.5 to 3.8. In these figures, a WSN consists of 9 sensor nodes and the sensor node sn_1 sends the data packets to the destination node sn_d through their consecutive neighbors. Initially, a routing path is established by the routing protocol and routing table stores the path information. After fault diagnosis, each node sn_i in the WSN maintains a neighboring list table $Neg(.)$, which contains the actual fault class of the one hop neighbor sensors $Neg(sn_i)$.

- **Case 1:** In Figure 3.5, the initial path for data transmission is set up as $sn_1 - sn_2 - sn_3 - sn_d$. After some time units, sensor node sn_3 reacts as like hard faulty node. At that time, by proposing a fault detection method, the one-hop neighbors $Neg(sn_3)$ knows the fault status of sn_3. Then immediately the transmission path changes at the point sn_2 and the new path is set up as $sn_1 - sn_2 - sn_4 - sn_5 - sn_d$. The node in the path of transmission, in which divergence occurs (or path changes) is called as diverted node (sn_2).

- **Case 2:** In Figure 3.6, the initial path for data transmission is set up as $sn_1 - sn_2 - sn_3 - sn_d$. After some time units, sensor node sn_3 reacts as like permanent or intermittent faulty node. In this case, the path of transmission is not redirected at diverted point, but it follows the same path $sn_1 - sn_2 - sn_3 - sn_d$ by converting the node sn_3 as a relay node. Here the node sn_3 does not transmits its own sensed data to the respective one-hop neighbors, otherwise the respective neighbor sn_d is discarded the faulty node sensed data by the node sn_3. As each data packet is associated with the respective node id, so the destination sn_d discards the data packet of node sn_3.

- **Case 3:** In Figure 3.7, the initial path for data transmission is set up by routing as $sn_1 - sn_2 - sn_3 - sn_d$. After some time intervals, the sensor node sn_3 reacts as like hard faulty node and sn_4 reacts as like permanent or intermittent faulty node. In this case, the data transmission path is redirected at the divergence point sn_2 and the new path is set as $sn_1 - sn_2 - sn_4 - sn_5 - sn_d$ by including the node sn_4 as a relay node. In the new path $sn_1 - sn_2 - sn_4 - sn_5 - sn_d$, the sn_4 act as relay node and the node next to relay node sn_5 discards the data packets of sn_4. In conclusion of case 3, if diverted node is discovered two paths, one path carries hard faulty node and another path carries soft faulty node (permanent, intermittent, and transient), then diverted node select the path having soft faulty node towards the destination node. Similarly, if both the path carries permanent or intermittent faulty node, then the diverted node follows any one of these paths towards the destination node in the WSN.

- **Case 4:** In Figure 3.8, the initial path for data transmission is set up by routing as $sn_1 - sn_2 - sn_3 - sn_d$. After some time intervals, the sensor node sn_3 and sn_4 react

as like hard faulty node, and do not respond to their neighbors. The diverted node looks for another path towards destination node, i.e., $sn_1 - sn_2 - sn_4 - sn_5 - sn_d$. But, here both the path carries hard faulty node between diverted node to the destination node. In this case, the data packet is unable to find the transmission path due to hard fault and destination node sn_d is not received the data packets of node sn_1. In this situation, the network is partitioned into different parts due to cut node [102]. Hence, in this case 4, the isolation phase is failed to establish the transmission path of data packets between sensor nodes.

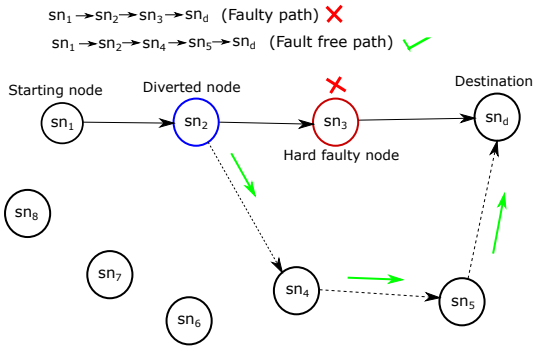

Figure 3.5: Overview of fault isolation: Case 1

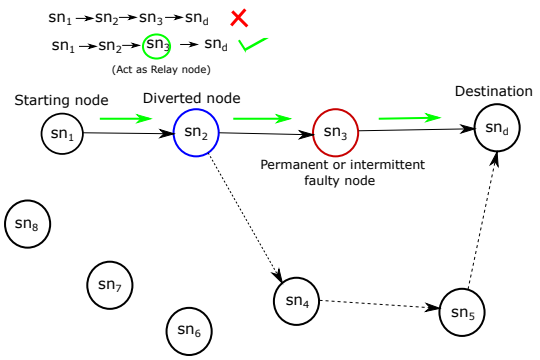

Figure 3.6: Overview of fault isolation: Case 2

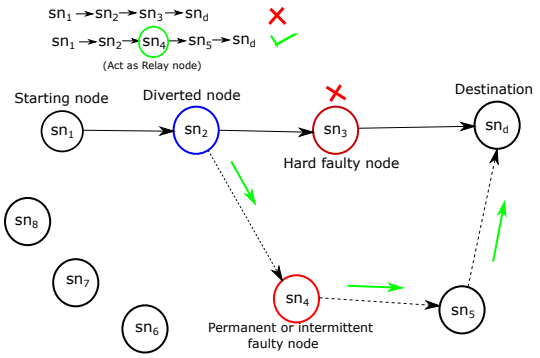

Figure 3.7: Overview of fault isolation: Case 3

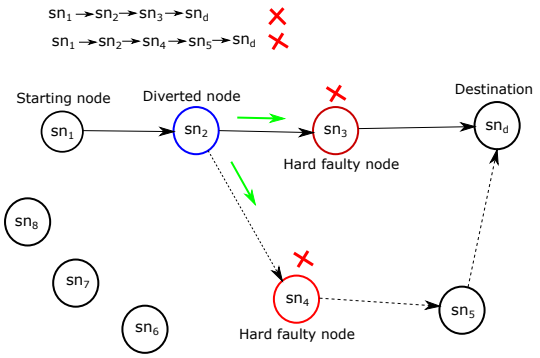

Figure 3.8: Overview of fault isolation: Case 4

3.4 Simulation Results and Discussions

The performance of the FDCSP algorithm is measured by calculating the performance parameters such as fault detection accuracy (FDA), false alarm rate (FAR), false positive rate (FPR), Fault Classification Accuracy (FCA), False Classification Rate (FCR), fault detection latency (FDL) and energy consumption (EC). The FDCSP algorithm is simulated using NS-2.35 simulator [24] and the performances are compared with the existing algorithms such as Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26]. The parameters and their values that are used in simulation are given in Table 3.1.

Table 3.1: Simulations environment variables

Parameter	Value
No. of nodes	1000
Packet rate	1 pkt/s
Simulation time	150 s
Initial energy	10 J
Transmission range	150 m
Channel rate	250 kbps
α_1	50 nJ/bit
α_2	10 pJ/bit/m^2
α_3	50 nJ/bit
Traffic	CBR
Grid size	1000×1000 m^2
MAC layer	IEEE 802.15.4
Propagation model	TwoRayGround
Packet size	512 bytes
Antenna model	Omni directional Antenna
Routing layer	CTP
No. of iteration	20
Smoothing parameter (σ)	1-3
No. of fault class	3
Topology	Arbitrary network

3.4.1 Performance Analysis with respect to Faulty Sensor Nodes

Fault detection accuracy with respect to the percentage of hard faulty sensor nodes is shown in Figure 3.9 and with respect to soft faulty sensor nodes is shown in Figure 3.10. We have compared our proposed FDCSP algorithm with that reported by Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26]. Figure 3.11 shows the fault detection accuracy with respect to the percentage of soft intermittent faulty sensor nodes and Figure 3.12 shows fault detection accuracy with respect to the percentage of soft transient faulty sensor nodes. In Figures 3.11 and 3.12 the proposed FDCSP algorithm can detect the soft intermittent fault and soft transient fault, respectively whereas, the existing Mohapatra et al. [25] detect hard and soft faults, Panda et al. [9] detect hard permanent and soft permanent faults and Elhadef et al. [26] algorithm can detect only permanent fault.

Fault detection accuracy decreases while the percentage of faulty sensor node increases. Here we have taken 1000 sensor nodes and gradually increase the percentage of faulty sensor nodes. We have injected the faults gradually from 5% to 30% and here we have considered the hard, soft, intermittent and transient faulty sensor nodes.

Figure 3.9 shows the fault detection accuracy with respect to hard faulty sensor nodes of four protocols such as Mohapatra et al. [25], Panda et al. [9], Elhadef et al. [26] and proposed method. The existing protocol such as Mohapatra et al. [25], Panda et al. [9]

and Elhadef et al. [26] used the time-out mechanism for hard fault detection. Sometimes due to packet loss and traffic congestion, the sensor node could not respond for a temporary period in the network. If a sensor node gives 90% faulty results and 10% fault free results, then the existing fault detection techniques are unable to identify the faulty node. However, the proposed protocol uses the repeated time-out mechanism for hard fault detection. The proposed protocol repeatedly checks the response message within the time frame sequence and then the fault status is decided after the condition checking. So, the irregular behavior of a hard faulty node could identify by using the proposed scheme.

Figure 3.10 shows the fault detection accuracy with respect to soft permanent faulty sensor node of four protocols such as Mohapatra et al. [25], Panda et al. [9], Elhadef et al. [26] and proposed method. The existing protocol Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26] have used an artificial immune system based principle, statistical method and neural network approach for soft fault detection at a instant time. If a sensor node gives 90% faulty results and 10% fault free results, then the existing fault detection techniques are unable to identify the faulty node. However, the proposed protocol has used the comparison based clonal selection principle in a time sequence to identify the soft faulty nodes. So, the proposed protocol could identify the different faulty behavior of the sensor nodes, which outperforms the existing schemes. The performance analysis using FDA, FAR and FPR are presented in the next following subsection.

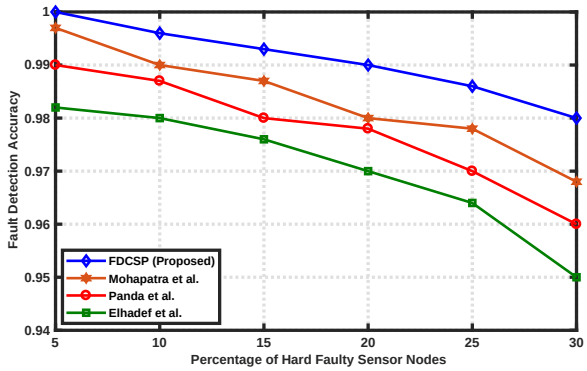

Figure 3.9: FDA vs. Percentage of Hard Faulty Sensor Nodes

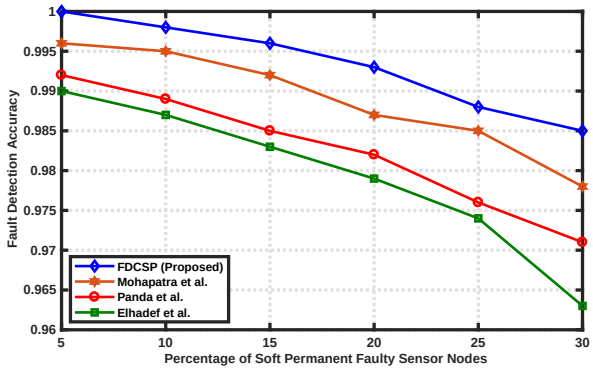

Figure 3.10: FDA vs. Percentage of Soft Permanent Faulty Sensor Nodes

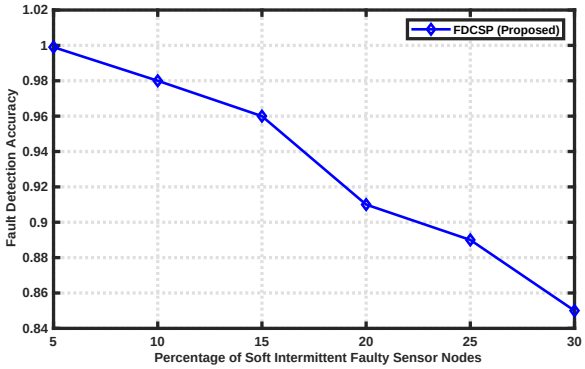

Figure 3.11: FDA vs. Percentage of Soft Intermittent Faulty Sensor Nodes

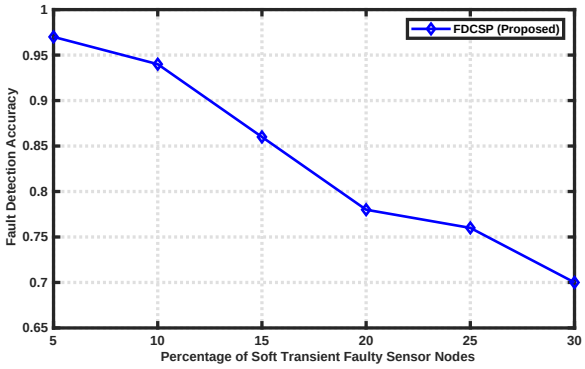

Figure 3.12: FDA vs. Percentage of Soft Transient Faulty Sensor Nodes

3.4.2 Performance Analysis using FDA, FAR and FPR

The FDA, FAR and FPR with respect to percentage of faulty sensor nodes are plotted in Figures 3.13, 3.14 and 3.15, respectively. We can see that the proposed FDCSP algorithm outperforms the existing algorithms as the existing algorithm can not detect the intermittent and transient fault. The proposed FDCSP algorithm gives higher FDA and lower FAR and

FPR than the existing algorithms. The FDA decreases and the FAR and FPR increases when the percentage of faulty sensor node increases. Here the percentage of faulty sensor node includes hard permanent, soft permanent, soft intermittent and soft transient faulty sensor nodes. The proposed FDCSP algorithm gives an average of 99.11% FDA, 3.55% of FAR and 0.88% of FPR whereas, the existing algorithm Mohapatra et al. [25] gives 97.31%, 3.98% and 2.69%, Panda et al. [9] gives 94.05%, 4.33% and 5.95% and Elhadef et al. [26] gives 92.66%, 5.43% and 7.34% of FDA, FAR and FPR, respectively.

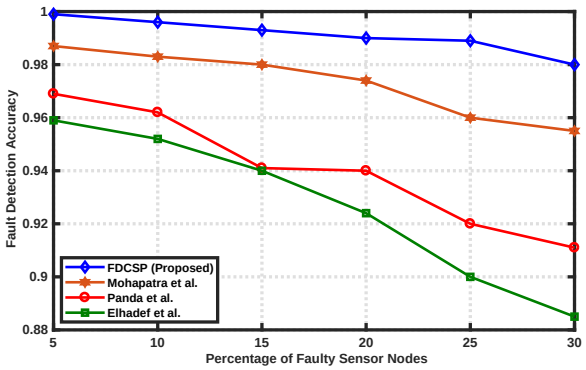

Figure 3.13: FDA vs. Percentage of Faulty Sensor Nodes

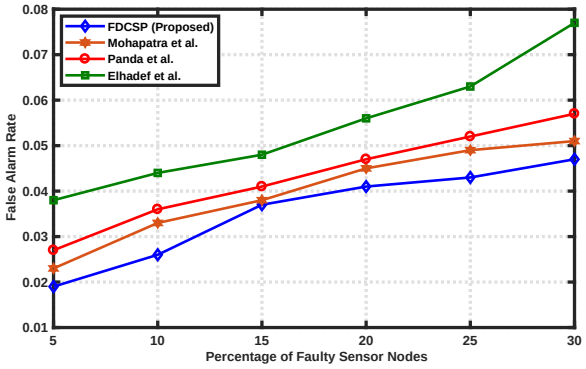

Figure 3.14: FAR vs. Percentage of Faulty Sensor Nodes

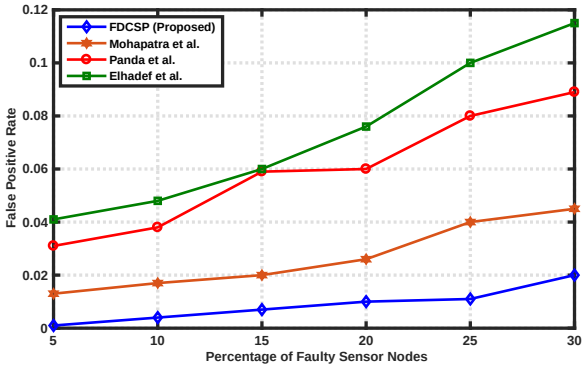

Figure 3.15: FPR vs. Percentage of Faulty Sensor Nodes

3.4.3 Performance Analysis of Fault Classification

The fault classification phase is implemented using MATLAB R2017a. The classification phase performance is measured by the performance metrics such as fault classification accuracy (FCA) and false classification rate (FCR). FCA is the correct classification rate and FCR is the misclassification rate. 1000 sensor data values are collected, in which approximately 70% (700) sensor data values are randomly selected for training purposes and 30% (300) sensor data values are randomly selected for testing purposes. After training, the testing data values are unknown for the PNN model and the performance results (FCA and FCR) are calculated by testing this sensor data values. The sensor data values are the collection of three (3) different types of soft fault (permanent, intermittent, and transient). From the training phase, we have set the representative pattern vectors (3 fault classes) by using k-means clustering algorithm and also selected the smoothing parameter value as 1 for peak accuracy results. In the testing phase, the faulty sensor node increases consistently from 50 to 300 for performance analysis. The faulty nodes contain different types of soft faults. It is observed that, for increasing the faulty nodes the fault classification accuracy (FCA) decreases and false classification rate (FCR) increases. The results are shown in the Table 3.2. Sometimes, intermittent and transient faults behaved arbitrarily (like byzantine fault), as the fault patterns are not fixed and changes with respect to the time interval. Therefore, the correct classification rate (FCA) decreases from 0.9802 to 0.9331 and the misclassification rate (FCR) increases from 0.0198 to 0.0669 by increasing the faulty nodes from 50 to 300 consistently. The fault classification phase gives the average classification accuracy approximately 95.66% and the average misclassification rate 0.04, which examined that the performance is consistent for the excessive faulty environment.

Table 3.2: Fault Classification Results

Faulty nodes	Fault classification accuracy	False classification rate
50	0.9802	0.0198
100	0.9711	0.0289
150	0.9619	0.0381
200	0.9523	0.0477
250	0.9414	0.0586
300	0.9331	0.0669

3.4.4 Fault Detection Latency

Fault detection latency with respect to the percentage of faulty sensor nodes is shown in Figure 3.16. By taking 1000 sensor nodes we are increasing the percentage of faulty sensor nodes from 5% - 30%. When the percentage of faulty sensor node increases the average fault detection latency for FDCSP algorithm is 3.03 second whereas, 3.16, 3.23 and 3.38

second for the existing algorithm Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. There is less fault detection latency of our proposed FDCSP algorithm as compared to the existing algorithms.

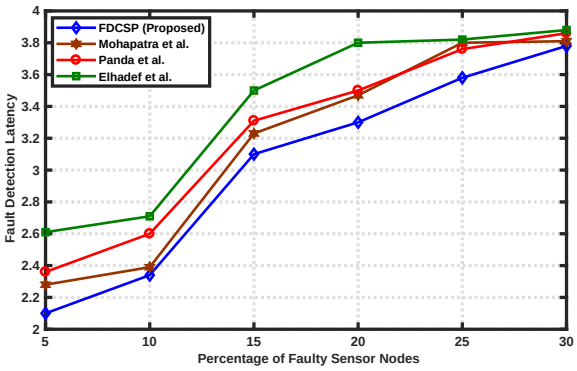

Figure 3.16: FDL vs. Percentage of Faulty Sensor Nodes

3.4.5 Energy Consumption

Energy consumption with respect to the percentage of faulty sensor nodes is shown in Figure 3.17. The total energy consumption depends upon the total amount of energy required to transmit the data and to receive the data for the diagnosis. Here the percentage of faulty sensor node gradually increases from 5% to 30% whereas the total number of sensor nodes are constant. The percentage of energy consumption is 1.32 joule for the proposed FDCSP algorithm and 1.49 joule, 1.67 joule and 2.59 joule for Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. Hence, the proposed algorithm consumes less energy than the existing algorithms.

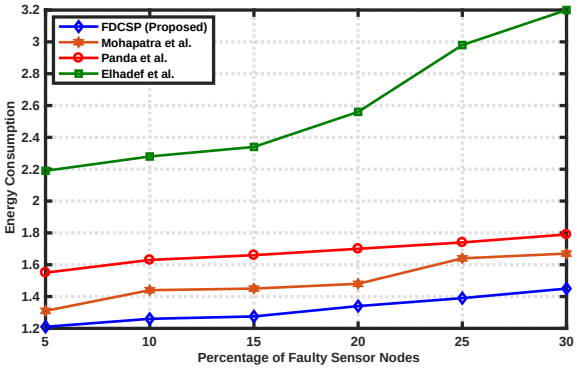

Figure 3.17: EC vs. Percentage of Faulty Sensor Nodes

3.5 Summary

In this chapter, the faulty nodes are detected using the proposed FDCSP algorithm and then the faults are classified into respective types using the probabilistic neural network

approach. After the actual fault status is detected, the faulty sensor nodes are isolated in the isolation phase. The proposed algorithm does not depend on the neighboring sensor nodes for computation so, increase in faulty neighboring nodes could not affect the accuracy. The existing algorithms handled only hard permanent and soft permanent faults however, in addition to that the proposed algorithm considered intermittent and transient faults. The performance of the algorithm is evaluated by using the performance metrics where it is found that the fault detection accuracy of the proposed FDCSP algorithm is improved by 1.8% over Mohapatra et al. [25], 5.06% over Panda et al. [9] and 6.45% over Elhadef et al. [26] algorithm. The false alarm rate of the proposed algorithm is improved by 0.43%, 0.78% and 1.88% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. The false positive rate of the proposed algorithm is improved by 1.81%, 5.07% and 6.46% over Mohapatra et al.[25], Panda et al. [9], Elhadef et al. [26], respectively. The fault classification performance is measured by fault classification accuracy and false classification rate. The simulation result also shows that the FDCSP algorithm provides less fault detection latency i.e., 4.11%, 6.19% and 10.35% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively and consumes less energy i.e., 11.40%, 20.95% and 49.03% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively.

The proposed FDCSP algorithm uses cloning and mutation for the convergence of the algorithm. The major drawback of this approach is the overhead of cloning, mutation for computing affinity (fitness) value towards the solution. This has been overcome in the next chapter, in which a negative selection based algorithm for fault detection and support vector machine for fault classification is proposed.

Chapter 4

Fault Diagnosis in Wireless Sensor Network using Negative Selection Algorithm and Support Vector Machine

4.1 Introduction

There is a greater interest seen among scientists and researchers in developing biologically inspired algorithms in the past. Due to the principle of artificial immune system (AIS) it is considered as one of the most popular approach among genetic algorithm (GA), ant colony optimization (ACO), particle swarm optimization (PSO) [69]. AIS is influenced from the principle of human immune system (HIS), which can expertly saved our bodies from bacteria and viruses [70]. Negative selection algorithm (NSA) is one of the most important method in AIS, which is based on the principle of self/non-self discrimination theory. It was first proposed by Forrest et al. [103] and is widely used in anomaly detection [71], pattern recognition [72], computer security [73] and fault detection [74].

Classical NSA (CNSA) takes unnecessary computational time by randomly generating the detectors. In addition, it is extremely hard to tell whether the quantity of detectors is huge enough to satisfy the fault probability. To tackle these drawbacks, an improved NSA (INSA) has been proposed for the detection of faults in wireless sensor network (WSN). The proposed approach follows two phases. They are (i) fault detection and (ii) fault classification. The fault detection phase is used to detect the faults using INSA for reducing computational overhead. To classify the faults, the well-known machine learning technique support vector machine (SVM) is used.

The fault-free and faulty nodes of a WSN are considered as self and non-self of AIS, respectively. In NSA the detectors are generated randomly from the fault-free nodes. We have generated the detector set randomly and added three non-self-pattern containing soft permanent, soft intermittent, and soft transient fault data to make it more efficient. This is considered as the improved version of NSA. Here, we have used AIS for detecting the nodes that causes a negative impact on the working condition of the wireless sensor network. We have classified faults in the classification phase using SVM. The result shows that the improved NSA of AIS can work effectively for the detection of faults in WSNs and it makes

the network more stable. The proposed INSA has compared with a clonal selection algorithm proposed by Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26]. The result shows that INSA outperforms than Mohapatra et al., [25], Panda et al. [9] and Elhadef et al. [26]. The major advantage of this approach is that, it does not need any prior knowledge of fault patterns and able to adapt to the changes in the faulty and fault-free situation. The major contribution of this chapter is stated as follows:

- An improved negative selection algorithm (INSA) is used to identify the soft faults in WSN.
- A bounded time period with acknowledgment message and majority voting scheme is used to detect hard permanent fault.
- A fault classification phase based on support vector machine is used to classify the faults into their respective types.
- Validation of the proposed method is carried out using NS-2.35 simulator and comparison study is also performed with existing approaches.
- Different metrics such as fault detection accuracy (FDA), false alarm rate (FAR), false positive rate (FPR), fault classification accuracy (FCA), false classification rate (FCR), fault detection latency (FDL) and energy consumption (EC) are used to measure the performance of our proposed algorithm.

The organization of the chapter is as follows. Section 4.2 discusses the basic principle of NSA. The proposed algorithm is given in Section 4.3. The simulation results are discussed in Section 4.4. Finally, the conclusion is given in Section 4.5.

4.2 Basic Principles of NSA

The immune system of a human body has a self defense mechanism, which can protect the body from being attacked by various bacteria or viruses. The main purpose of the immune system is to find out all the cells or molecules in the body and categorize them into self (fault-free) and non-self (faulty). Self usually refers to the acceptable or fault free data, and non-self refers to the measured data. There are two steps in NSA which are shown in Figures 4.1 and 4.2. The self strings are nothing but the set of real-valued data collected from the environment which are sensed by the sensor nodes in WSN. Since the WSN is initially fault free, the collected data are supposed to be accurate. Another set of real-valued data strings is generated randomly within a bounded range — for example, the temperature value recorded from a temperature sensor for a fixed duration of time. In Figure 4.1 there is a match between self strings and the strings which are generated randomly. The matched strings get rejected as per the principle of NSA. The strings that do not match are moved to the detector set. In

Figure 4.2, protected strings which are nothing but self strings are matched with those in the detector set. The strings that get matched are identified as non-self, and the rest are matched again till the faulty nodes are detected. It is noted that the strings which were not matched in the censoring phase and matched in the detection phase will lead to detect the set of faulty nodes.

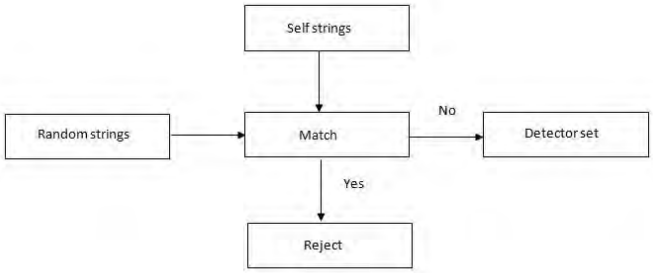

Figure 4.1: Censoring

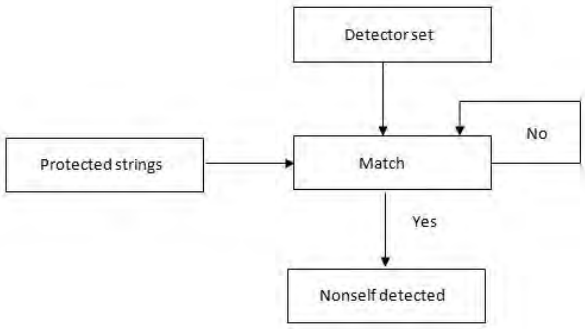

Figure 4.2: Detection

4.3 Proposed Algorithm

The proposed algorithm consists of two phases such as (1) fault detection and (2) fault classification phase. In the fault detection phase, an algorithm based on negative selection principle is proposed to identify the faulty nodes in WSN. Then in the classification phase, the faulty nodes are classified into different types such as permanent, intermittent and transient fault using multi-class support vector machine.

4.3.1 Fault Detection Phase

This section is further classified into two phases, i.e., (1) detection of hard fault and (2) detection of soft fault.

Detection of Hard Fault

The data sensed by each sensor node sn_i is sent to its neighboring sensor nodes $sn_j \in Neg(sn_i)$. Upon receiving the message, node sn_j sends an acknowledgment message (ACK message) to the node sn_i. A node table NT_i is maintained by every sensor node sn_i where, node id with information of ACK message of neighboring sensor nodes has stored. Each sensor node sn_i maintains a status register $SR_{ij} \in \{0, 1\}$ for all of its neighboring sensor nodes. Within a bounded time period ($T_{bounded}$), if the node sn_i receives an ACK message from its neighboring sensor nodes $sn_j \in Neg(sn_i)$ then the SR_{ij} is set to 1 otherwise 0. For each sensor node $sn_i \in S$, if the summation $\sum_{sn_j}^{Neg(sn_i)} SR_{ij}$ is less than $\lceil \frac{Neg(sn_i)}{2} \rceil$ then the node sn_i is identified as hard permanent faulty node. Similarly, the status of all other nodes can be computed. The detection of hard fault is described in Algorithm 4.1.

Algorithm 4.1 Algorithm for Detection of Hard Fault

Initialize: Node table (NT_i), bounded time period ($t_{bounded}$), and status register SR_{ij};
for each sensor node $sn_i \in S$ **do**
    Node sn_i sends its sensed data to its neighboring nodes $Neg(sn_i)$;
    Upon receiving the data, the neighboring nodes $Neg(sn_i)$ sends an ACK message to sn_i;
    for each neighbor $sn_j \in Neg(sn_i)$ **do**
        if the ACK message received within the bounded time ($t_{bounded}$) **then**
            SR_{ij} = 1;
        else
            SR_{ij} = 0;
        end if
    end for
    After the bounded time period expires;
    if $\sum_{sn_j}^{Neg(sn_i)} SR_{ij} < \lceil \frac{Neg(sn_i)}{2} \rceil$ **then**
        the node sn_i is identified as hard permanent faulty node;
    else
        the node sn_i is fault free;
    end if
end for
Broadcast the status of the faulty nodes in the network;

Detection of Soft Fault

An algorithm based on negative selection principle has been proposed to identify the soft faulty nodes. There are two phases (I.) detector generation phase and (II.) matching phase.

I. **Detector generation phase**

In this phase, a detector set is generated randomly which is used to detect faulty nodes. It is necessary to ensure that the detector set does not match with the self set. The detectors generated in this phase cover the non self space with the minimum number of detectors. The detectors can distinguish the faulty nodes from all the nodes. However, the types of faults are detected during the classification phase. In this work, the detector set is updated after its generation, which makes it more efficient. We have added three non-self pattern containing soft permanent fault, soft intermittent fault, and soft transient fault data.

Algorithm 4.2 Pseudo code for generation of detectors

```
Input: Self data;
Output: Detector set;
Initialize: Detector set = ϕ ;
while (¬ Stop condition()) do
    Detectors ← Generate Random Detectors();
    for (Detector_i ϵ Detector set) do
        if (¬ Matches (Detector_i, Self data)) then
            Detector set ← Detector_i;
        end if
    end for
end while
Return (Detector set);
```

II. **Matching phase**

Different matching rules such as binary matching, r-contiguous matching, r-chunk matching, hamming distance matching, and Euclidean distance matching rules are available in the literature. For binary data r-contiguous matching, r-chunk matching, hamming distance matching rules are used, and for real-valued data Euclidean distance, Manhattan distance, Minkowski distance matching rule are used [83], [104].

The Euclidean distance matching rule is the most widely used method to calculate the matching. In this work as the data is real-valued, Euclidean distance matching rule is used to measure the distance between the detector and sample data as given in Equation (4.1).

$$d(x_i, D_i) = \sqrt{\sum_{i=1}^{N}(x_i - D_i)^2} \tag{4.1}$$

If d(x_i, D_i) $\leq \theta$ there is a match. Where, d(x_i, D_i) is the distance, θ is a predefined threshold and N is the total number of nodes in WSN.

4.3.2 Fault Classification Phase

In this phase, SVM is used to classify the faulty nodes. A hyperplane is constructed by SVM which can classify all the input data into two classes, i.e., positive class or negative class. The hyperplane should chosen in such a way that, there should be maximum distance between the hyperplane and the nearest element from the hyperplane. The hyperplane is defined by the following Equation

$$g(x) = w^t x + b = 0 \tag{4.2}$$

Where, x is the input feature vector, w is the weight vector and b is the bias.

g(x) = w^tx+b > 0 x$\in$ class1

g(x) = w^tx+b < 0 x$\in$ class2

The total margin is computed by $\frac{2}{||w||}$. By minimizing the weight vector we will have the biggest margin to split the two classes. Minimizing w is a nonlinear optimization task which is solved by using Lagrange multiplier α_j.

$$f(x) = sgn\sum_{j=1}^{m} \alpha_j y_j k(x_j, x) + b \tag{4.3}$$

where, x_j is the support vector, y_j is the class level, m is the number of support vectors, b is the classification threshold, α_j is Lagrange multiplier and $k(x_j, x)$ is the kernel function.

SVM was first designed to perform two class classification. However, if we need to classify more than two classes, we have to extend the SVM to multi class classification. Some methods are there to generate multi-class SVM from binary class SVM [105]. In this chapter, we have considered three fault classes. To classify these faults we need to extend the SVM to multi-class classification. Three approaches are there for multi-class SVM such as One-Against-All (OAA), One-Against-One (OAO) and Error-Correcting Output Codes (ECOC).

One-Against-All (OAA): This method is also called as winner-take-all classification. It constructs m binary classifiers where each classifier is trained to distinguish one class from the remaining m-1 classes. During the testing phase, the class level is determined by the binary classifier which gives maximum output value. There are some drawbacks in this method. First, the memory requirement is high. If the training data set is large then it takes more memory. Second, the size of the training sample is unbalanced.

One-Against-One (OAO): This method is also called as pairwise classification. It constructs m(m-1)/2 classifiers. Comparing to OAA approach, this method is more symmetric, and the size of classifiers is much larger which makes the training speed faster but, the number of classifier increases when the number of class increases.

Error-Correcting-Output-Code (ECOC): The erroneous data which is transmitted is checked and corrected by this method. The errors may occur during the transmission of data which leads to erroneous outcomes due to various reasons. The ECOC algorithm improves the performance by encoding into various categories and then converting into corresponding codes.

The essential part of ECOC algorithm is, a coding matrix should be prepared which needs to satisfy two conditions.

I) The rows of the coding matrix are not correlated.

II) The column of the coding matrix is not correlated and not complementary.

The above two conditions ensure the proper separation between the pair of rows and columns. Therefore, each row refers to a code word which corresponds to one of the fault class.

Though various coding strategies such as One-Against-All [106], One-Against-One [107], dense and sparse random [108] are available, in this thesis, we have adopted Hadamard matrix, and the rules of this matrix are as follows [109].

i) A Hadamard matrix was generated assuming that there are total c classes. If $2^{j-1} < c \leq 2^j$ then dimension of the matrix is 2^j, using recursion high order matrices can be obtained

from low order matrices.

$$H_{2^j} = \begin{bmatrix} H_{2^{j-1}} & H_{2^{j-1}} \\ H_{2^{j-1}} & -H_{2^{j-1}} \end{bmatrix} \tag{4.4}$$

$-H_{2^{j-1}}$ is the complementary of $H_{2^{j-1}}$. The two order matrix is $H_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

ii) First column of the matrix can be deleted because of all the zeros and we got the matrix as $2^j \times (2^j - 1)$.

iii) From step 2, select the first k rows of the matrix of $c \times (2^j - 1)$.

$(2^j - 1)$ is the code length of the coding matrix, and the hamming distance between the rows is $(2^j - 1)$. As the rows are not correlated and the columns are neither correlated nor complementary it meets the requirement of ECOC coding.

In this thesis, we have considered three fault classes such as soft_permanent (SP), soft_intermittent (SI) and soft_transient (ST). Let, c=3. In our case the coding matrix will be

$$H = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{4.5}$$

The output of the sensor node, i.e., the sensed data can reflect the state of the network. When there is a fault in a sensor node, the sensed data from this node changes abnormally and therefore considered as faulty. The characteristics of a sensor node data can be reflected as a condition for a faulty state. In this chapter, we have collected the data of sensor nodes in normal condition as well as faulty condition. The input data is taken as an ECOC matrix for training. The classification decision function is given in Equation (4.3).

ECOC-SVM for multi-class classification: By applying ECOC-SVM for multi-class classification we have to transform c class to l class classification.

- Training Phase

  Given m distinct classes, create an m $\times$ n binary matrix of M. Each class is assigned one row of M. Each row represents a dichotomizer. Each column of the matrix divides the class into two groups. Train all the base classifiers to learn the n binary problems.

- Testing Phase

  To test a new data apply each of the n classifiers to the new data. Combine the predictions to obtain a binary string (code word) for the new point. Classify to the class with the nearest code word, i.e., the hamming distance.

$$k_i = \sum_{j=1}^{l} |f(x) - H_{i,j}| \qquad i = 1, 2, ..., c \tag{4.6}$$

  where, $H_{i,j}$ is the Hadamard matrix and f(x) is the classification decision function.

To show the effectiveness of ECOC-SVM, we make a comparison of accuracy shown in Table 4.1. The same data set has been applied to all i.e., OAA, OAO and ECOC-SVM. From the table we can see that, ECOC-SVM gives an average of 94.15% whereas, the OAO and OAA gives 79.57% and 93.67% classification accuracy, respectively.

Table 4.1: Accuracy Comparison

No.	OAO	OAA	ECOC-SVM
1	80.65 %	90.19%	91.23%
2	79.30%	96.17%	96.05%
3	78.77%	94.65%	95.18%

4.4 Simulation Results and Discussions

The performance of the existing Mohapatra et al. [25], Panda et al. [9], Elhadef et al. [26] and the proposed INSA algorithm is implemented using the NS-2.35 simulator [24]. DHT22 sensors are considered for experiment which sense temperature and humidity shown in Figures 4.3 and 4.4. To evaluate the performance of the proposed algorithm we have conducted an extensive experiment in our laboratory. The data i.e., temperature and humidity was collected inside our laboratory environment from 10:00 AM to 04:00 PM. Performances of the algorithms are compared by using different performance metrics such as fault detection accuracy (FDA), false alarm rate (FAR), false positive rate (FPR), fault classification accuracy (FCA), false classification rate (FCR), fault detection latency (FDL) and energy consumption (EC). In this simulation, 1000 sensor nodes are randomly deployed in an area of 1000*1000 m^2. The parameters and their values that are used for simulation are given in Table 4.2.

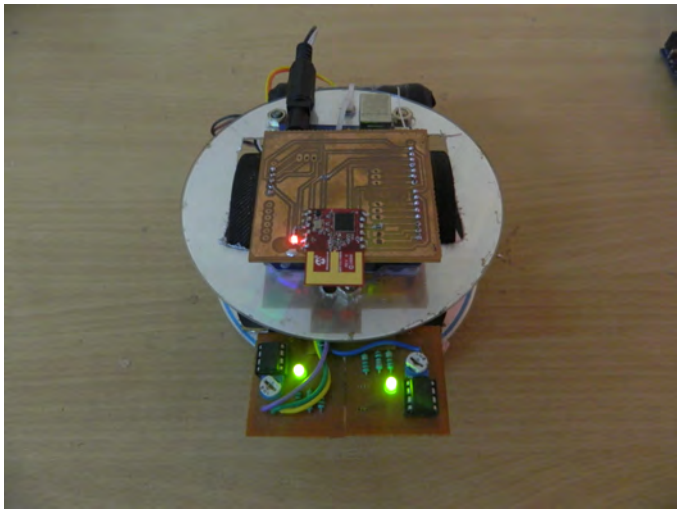

Figure 4.3: DHT22 Sensor Node

Table 4.2: Simulation Parameters

Parameter	Value
No. of nodes	1000
Simulation time	150 s
Transmission range	150 m
Channel rate	250 kbps
Traffic	CBR
MAC protocol	IEEE 802.15.4
Propagation model	TwoRayGround
Initial energy	10 J
α_1	50 nJ/bit
α_2	10 pJ/bit/m^2
α_3	50 nJ/bit
Packet size	512 bytes
Packet rate	1 pkt/s
Antenna model	Omni Antenna
Grid size	1000×1000 m^2
No. of fault class	3
Topology	Arbitrary network

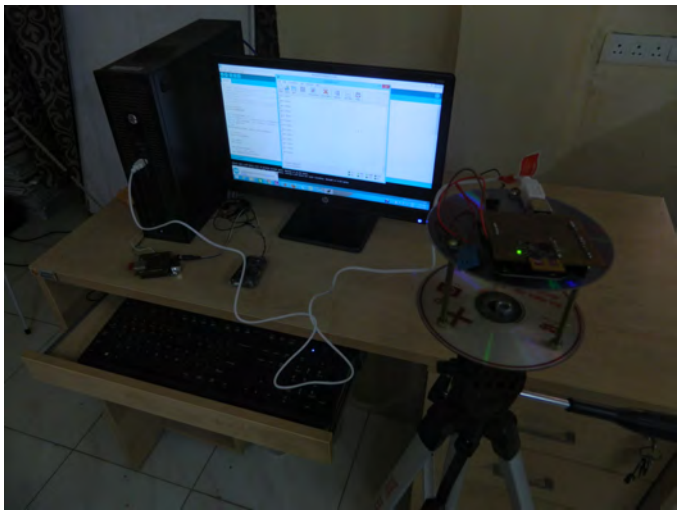

Figure 4.4: Sensor Node for Experiment

4.4.1 Performance Analysis with respect to Faulty Sensor Nodes

Initially, the nodes in the network is assumed to be fault free. Gradually faulty nodes are added to the network with varied percentage of 5, 10, 15, 20, 25 and 30. The proposed INSA algorithm is evaluated using different performance metrics such as FDA, FAR, FPR, FCA, FCR, FDL and EC. The performance metrics are defined as follows.

FDA is defined as the ratio between the number of faulty sensor nodes detected as faulty to the total number of faulty nodes present in the network. FAR is defined as the ratio between

the number of fault free nodes detected as faulty to the total number of fault free nodes present in the network. FPR is defined as the ratio between the number of faulty nodes detected as fault free to the total number of faulty nodes present in the network. FCA is defined as the ratio between the total number of faulty nodes correctly classified as a fault type to the total number of faulty nodes present in the network. FCR is defined as the ratio between the total number of faulty nodes wrongly classified as a fault type to the total number of faulty nodes present in the network. FDL is defined as the total amount of time required to detect all the sensor nodes present in the network. EC is defined as the total energy consumed by the network to identify the faulty sensor nodes present in the network.

4.4.2 Performance Analysis using FDA, FAR and FPR

The FDA, FAR and FPR with respect to the percentage of faulty sensor nodes are plotted in Figures 4.5, 4.6 and 4.7, respectively. It is observed that the proposed improved negative selection algorithm (INSA) outperforms the existing algorithms as the existing algorithms cannot detect the intermittent and transient fault. The proposed INSA gives higher FDA and lower FAR and FPR than the existing algorithms as it does not depends upon the neighboring sensor node for computation. Increasing in the faulty neighboring sensor nodes could not affect the performance of the proposed algorithm. The FDA decreases, and the FAR and FPR increases when the percentage of faulty sensor node increases. Here the percentage of faulty sensor node includes hard permanent, soft permanent, soft intermittent and soft transient faulty sensor nodes. The INSA gives an average of 99.25% FDA, 3.3% of FAR and 0.75% of FPR whereas, the existing Mohapatra et al. [25] gives 97.7%, 3.63% and 2.3%, Panda et al. [9] gives 94.28%, 4.15% and 5.72% and Elhadef et al. [26] gives 92.76%, 5.16% and 7.24% of FDA, FAR and FPR, respectively.

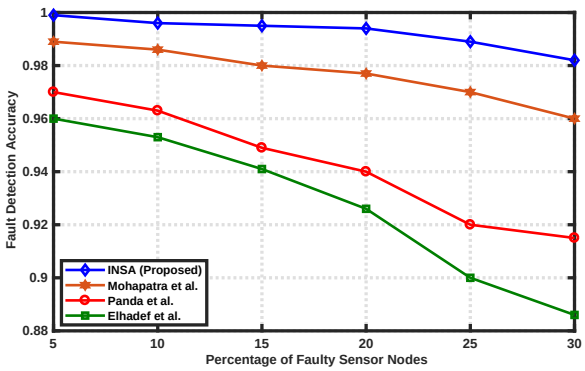

Figure 4.5: FDA vs. Percentage of Faulty Sensor Nodes

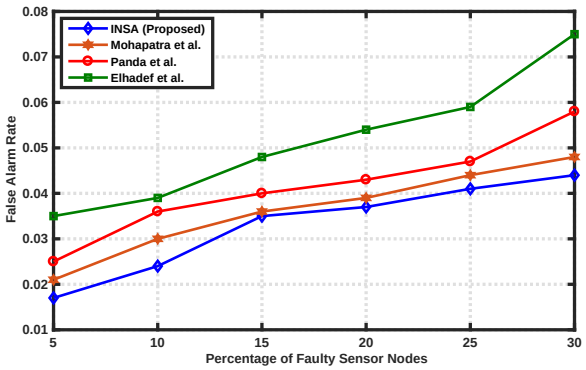

Figure 4.6: FAR vs. Percentage of Faulty Sensor Nodes

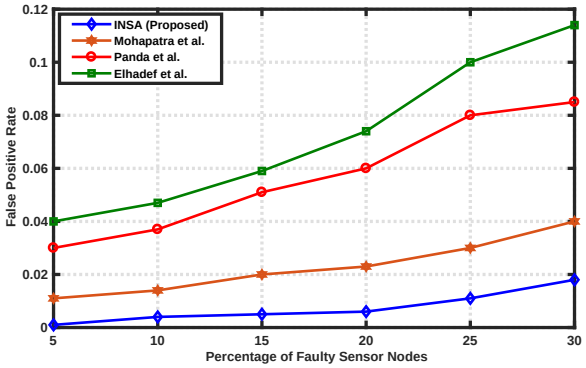

Figure 4.7: FPR vs. Percentage of Faulty Sensor Nodes

4.4.3 Performance Analysis of Fault Classification

The fault classification phase is implemented using MATLAB R2017a. Fault classification accuracy (FCA) and false classification rate (FCR) are the two parameters to measure the performance of the classification phase. The fault classification accuracy (FCA) is the correct classification rate, and false classification rate (FCR) is the misclassification rate. One thousand sensor data values are collected, in which approximately 70% (700) sensor data values are randomly selected for training purposes, and 30% (300) sensor data values are randomly selected for testing purposes. We have taken three distinct classes, and each class is assigned to one row of the matrix M in the training phase. Each column of the matrix divides the class into two groups.

The faulty nodes contain different types of soft faults. It is observed that while increasing the faulty nodes the fault classification accuracy (FCA) decreases and false classification rate (FCR) increases as shown in table 4.3. As the intermittent and transient fault patterns are not fixed and changes with respect to the time interval. Sometimes, intermittent and transient faults are behaving arbitrarily according to the time interval. Therefore, the correct classification rate (FCA) decreases from 0.9902 to 0.9479 and the misclassification rate (FCR) increases from 0.00980 to 0.0521 by increasing the number of faulty nodes. The

fault classification phase gives the average classification accuracy approximately 97% and the average misclassification rate of 0.03, which examined that the performance is consistent for the excessive faulty environment.

Table 4.3: Fault Classification Results

Faulty nodes	Fault classification accuracy	False classification rate
50	0.9902	0.00980
100	0.9824	0.0176
150	0.9765	0.0235
200	0.9638	0.0362
250	0.9522	0.0478
300	0.9479	0.0521

4.4.4 Fault Detection Latency

Fault detection latency with respect to the percentage of faulty sensor nodes is shown in Figure 4.8. By taking 1000 sensor nodes we are increasing the percentage of faulty sensor nodes from 5% - 30%. The existing algorithms follows distributed based fault diagnosis where, each sensor node sends its own sensed data to its neighboring sensor nodes and then collects the data for processing. So, computation of each node takes more delay. When the percentage of faulty sensor node increases the average fault detection latency for INSA algorithm is 2.92 second whereas, 3.05, 3.2 and 3.36 second for the existing algorithms Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. There is less fault detection latency of our proposed INSA algorithm as compared to the existing algorithms.

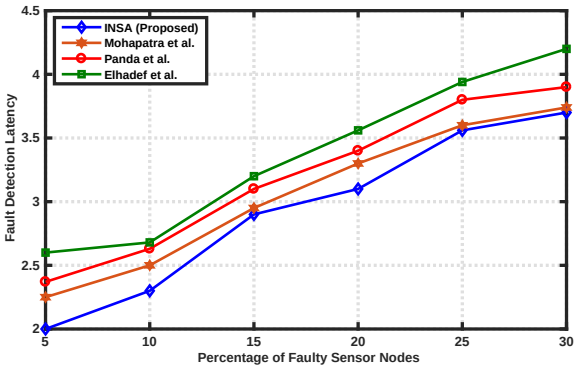

Figure 4.8: FDL vs. Percentage of Faulty Sensor Nodes

4.4.5 Energy Consumption

Energy consumption with respect to the percentage of faulty sensor nodes is shown in Figure 4.9. The total energy consumption depends upon the total amount of energy required to transmit the data and to receive the data for the diagnosis. Here the percentage of faulty

sensor node gradually increases from 5% to 30% whereas the total number of sensor nodes are constant. The percentage of energy consumption is 1.29 joule for the proposed INSA algorithm and 1.46 joule, 1.65 joule and 2.58 joule for Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. Hence, the proposed algorithm consumes less energy than the existing algorithms.

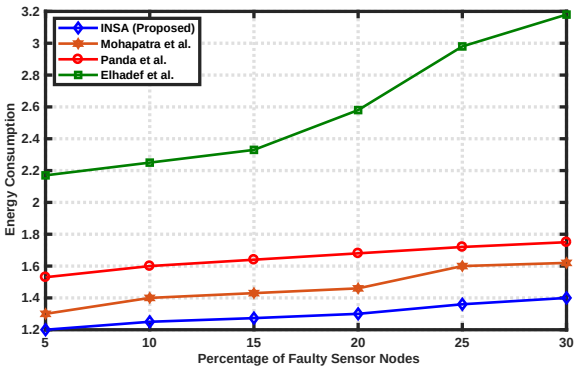

Figure 4.9: EC vs. Percentage of Faulty Sensor Nodes

4.5 Summary

In this chapter, an improved version of the negative selection algorithm is proposed. Here, the faulty sensor nodes are diagnosed as well as classified into soft permanent, soft intermittent and soft transient using the support vector machine. The performance of the algorithm is evaluated by using the performance metrics where it is shown that, the fault detection accuracy of the proposed INSA algorithm is improved by 1.55%, 4.97% and 6.49% over Mohapatra et al.[25], Panda et al. [9], Elhadef et al. [26], respectively. The false alarm rate of the proposed algorithm is improved by 0.33%, 0.85% and 1.86% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. The false positive rate of the proposed algorithm is improved by 1.55%, 4.97% and 6.49% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. The fault classification phase gives the average classification accuracy approximately 97% and the average misclassification rate 0.03. The simulation result also shows that the proposed algorithm provides less fault detection latency i.e., 4.26%, 8.75% and 13.09% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively and consumes less energy i.e., 11.64%, 21.81% and 50% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively.

The proposed method is based on the negative selection principle of AIS. The major advantage of this approach is that, it does not need any prior knowledge of fault patterns and able to adopt the changes in the faulty and fault free situations but it takes excessive computing time because of random generation of detectors. It is also very difficult to know whether the number of generated detectors is large enough, that can satisfy the detection

fault probability. This has been solved in the next chapter which is based on dendritic cell algorithm.

Chapter 5

Fault Diagnosis in Wireless Sensor Network using Dendritic Cell Algorithm

5.1 Introduction

For decades immunologist has been thinking that immune system differentiate between self and non-self. This was based on the idea of self/non-self discrimination theory. It states that the immune system has the capacity to distinguish between body's own cells (self) from foreign cells (non-self). Based on this theory, foreignness is the reason that leads to stimulating the immune response. However, Polly Matzinger explained an alternative way of working principle of immune system [77]. Based on the principle of danger theory (DT), any damage to the body triggers the immune system by sending danger signals, and in the case of absence of a danger in a tissue, the innate immune system can suppress the immune response. More specifically, the DT is based on the actions of special immune cells called dendritic cells (DCs) and their behavioral influence has led to the development of an immune classification algorithm called the dendritic cell algorithm (DCA).

Due to the characteristics and features of DCA, it can be applied to different problems in the real world. This inspired many researchers interest in further exploring the algorithm and analyzing its behavior. The major contribution of this chapter is stated as follows.

- A dendritic cell algorithm is used to detect the faults in WSN.
- A validation of the proposed method is carried out using NS-2.35 simulator and comparison study is also performed with existing approaches such as Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26].
- Different metrics such as fault detection accuracy (FDA), false alarm rate (FAR), false positive rate (FPR), fault detection latency (FDL) and energy consumption (EC) are used to measure the performance of our proposed algorithm.

The organization of the chapter is as follows. Section 5.2 discusses the background of danger theory. The proposed algorithm is given in Section 5.3. The simulation results and discussions are given in Section 5.4. Finally, the chapter is concluded in section 5.5.

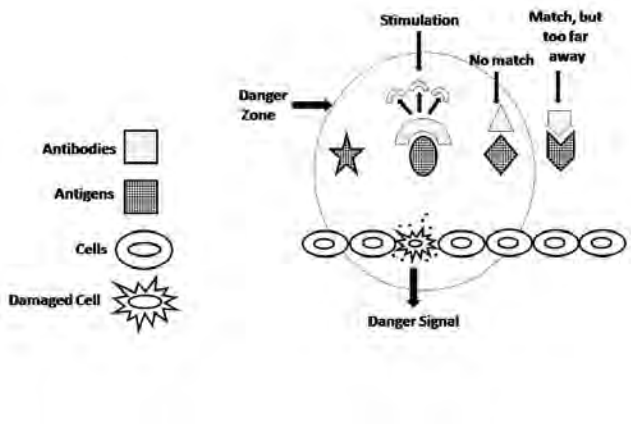

Figure 5.1: Illustration of Danger Theory

5.2 The Danger Theory

5.2.1 Biological Background

The self/non-self discrimination principle is a widely accepted method. It was believed that, the immune system got activated when our body recognizes an antigen or foreign entities. But later, the researchers believe that activation of immune system requires the presence of danger signal additionally recognition of pathogens. Due to pathogenic infections, the body cells got damaged which indicates danger. This recognition is performed by dendritic cells (DCs) of the immune system [76].

The central idea of danger theory (DT) is that, the immune system responds to danger rather than non-self. Instead of responding to foreignness, the immune system reacts to danger. The danger is measured by damage to the cell, i.e., when a cell dies an unnatural death [77]. The danger signal establishes a danger zone around itself. Figure 5.1 depict the illustration of danger theory.

5.2.2 Dendritic Cells

Dendritic cells (DCs) are antigen presenting cells (APCs) whose task is to gather, process and present antigens to the T-cells. Under healthy conditions, the body cells which are no longer required, may commit suicide to manage the development and growth of cells. This is called as programmed cell death or apoptosis. Because of apoptosis, immuno suppressive molecules (safe signals) released which indicates everything is normal in the tissue and consequently promoting immune tolerance. In contrast, when there is a damage in the tissue, they die during a process. This is called as necrosis, which is caused by external factors. Because of necrosis, a cell bursts and releases danger signals [110].

In addition, DCs have the capacity of sensing the danger signals (released because of

necrosis), safe signal (released because of apoptosis) and PAMPS (pathogenic associated molecular pattern) which are associated with pathogens such as bacteria, viruses, fungi etc. DCs are also sensitive to inflammatory signals. Inflammatory signals have not any effect if it is used alone, but amplify the effects of other three signals [111]. There are four (4) types of immunological signals namely, (1) pathogenic associated molecular pattern (PAMP), (2) safe signal(SS), (3) danger signal (DS) and (4) Inflammatory cytokines (IC).

1. **PAMP**: Released because of pathogens such as bacteria, viruses or fungi.
2. **Safe Signal (SS)**: Released because of apoptosis.
3. **Danger Signal (DS)**: Released because of necrosis.
4. **Inflammatory Cytokines (IC)**: Released because of inflammation in a tissue.

The behavior of DCs depends on the concentration of above immunological signals. DC always exist in three maturity states [112].

- **Immature DC (IDC)**: In tissue DCs are found immature. This is the initial maturation state. The signals collected by IDCs are PAMPs, SSs, DSs and ICs. Based on the proportions of these signals, IDC change their state to either partially matured state (semi mature DC) or fully matured state (mature DC).
- **Semi mature DC (SDC)**: Immature DC becomes semi mature DC if there is higher amount of safe signal than PAMPs and danger signals.
- **Mature DC (MDC)**: Immature DC becomes mature DC if there is higher amount of PAMPs and danger signals than the safe signals.

5.3 Proposed Algorithm

The characteristics of dendritic cells of the danger theory are used in the proposed scheme. The proposed work consists of four phases such as, (1) initialization phase, (2) detection phase, (3) context evaluation phase and (4) classification phase.

In DCA, each agent is represented as cell. Each cell is capable of collecting data items called antigens that represent the data to be classified. Signal and antigens are two types of input data to the DCA. Signals are represented as real valued number vectors, while antigens are the IDs of the data item. The binary dendritic cell classifier classifies each antigen either as fault free (context of semi-mature cells) or, faulty (context of mature cells). Therefore, the output of the algorithm is the antigen context value, i.e., either 0 (fault free) or, 1 (faulty).

Algorithm 5.1 Dendritic Cell Algorithm

```
Input : signals and antigens;
Output : antigen context value (0/1);
for each DC do
    Initialize DC();
end for
while CSM < Th_m do
    get antigen();
    get signal();
    calculate inter();
     update cumulative();
end while
if smDC > mDC then
    cell context = 0;
else
    cell context = 1;
end if
for each antigen do
    if cell context == 1 then
        nb mature ++;
    end if
end for
for each antigen do
    MCAV = nb mature / nb antigen;
end for
```

- Initialization Phase : DCA implementation also requires a pre-processing step of data to properly map a particular problem domain to the algorithm's input space. The pre-processing phase involves two main steps: reducing the feature and categorizing the signal. More precisely, DCA selects from the input training data set the most important features (attributes) and assigns each selected attribute to its specific signal category; either as SS, DS or PAMP signal. Each attribute is mapped as a category of signal based on the previously stated immunological definitions.

- Detection Phase : DCA creates a signal database throughout the detection process by integrating the input signals with the antigens. This is achieved through using both get-antigens and get-signals functions. The signal list represents the antigens to be identified and the three signals represent the attributes: SS, PAMP, and DS. For each antigen, the attribute values are calculated on the basis of specific processes as discussed in the following

  Calculation of PAMP and SS : A suitable attribute is selected and then the median of all the selected attribute values across both classes of data is calculated. For each attribute value, determine if it is a PAMP or a safe signal. If the attribute value is greater than the median, then this value is used to form a safe signal. The absolute distance from the mean is calculated and attached to the safe signal value and the PAMP signal value takes 0 and vice versa.

  Calculation of DS : Compute mean values using the values of faulty class. By taking each attribute value, calculate the absolute distance between the attribute values and the calculated mean. Use the calculated distance values in a further calculation to form the single value for the DS. This value is the mean value of the absolute distances

calculated, with the derivation shown in Equation 5.1. Repeat this process for all entries of the selected attributes.

$$DS = \frac{\sum absolutedistance}{numberofattributes} \tag{5.1}$$

Upon generation of these signals, the result is a set of feature vectors representing a set of signal data. The algorithm processes its input signals on the basis of the induced signal database to obtain three cumulative output signal values known as the co stimulatory molecule signal value (CSM), semi-mature signal value (smDC) and the mature signal value (mDC). This task is performed by using the calculate inter () function where a signal processing equation and a set of weights are used to calculate these cumulative output signal values. DCA uses the following weighted sum equation to calculate the interim output signals.

$$C = \frac{(W_{PAMP} * \sum_i PAMP_i) + (W_{SS} * \sum_i SS_i) + (W_{DS} * \sum_i DS_i)}{(W_{PAMP} + W_{SS} + W_{DS})} * \frac{1 + IC}{2} \tag{5.2}$$

If there are several signals per category, $PAMP_i, DS_i and SS_i$ are the PAMP, danger and safe output signal values for all signals (i) of that category. The weights used for PAMP, SS and DS are W_{PAMP}, W_{SS} and W_{DS}, respectively shown in Table 5.1. IC is the inflammatory cytokines. Three times, once per output signal, this equation is repeated. This is to measure the values of the interim output signal for the output of CSM, smDC and mDC. Over time, these values are summed up cumulatively. A migration threshold Th_m is assigned to each DC upon its creation. So, if the CSM value reaches Th_m, the DC stops sampling antigens and signals otherwise the algorithm continues to sample and also continues to measure and update the CSM, smDC, and mDC values via the update cumulative() function.

Table 5.1: Weights used for signal processing

	PAMP	SS	DS
CSM	2	1	2
smDC	0	0	1
mDC	2	1	-1.5

- Context Evaluation Phase : Once the cell has migrated, each DC has the ability to process and collect signals and antigens through the context evaluation phase. The DC generates a cell context by producing cumulative output signals that is used to perform fault detection in antigens assessment. The accumulated output signals are evaluated after migration and the greater semi mature or mature output signal becomes

the context of the cell. This cell context is used to label the derived context value of 1 or 0 for all antigens collected by the DC.

- Classification Phase : The calculated value for the cell context is used to derive the nature of the response by measuring the number of DCs that are fully mature. This generated number is represented by the mature context antigen value (MCAV). The closer the MCAV is to 1, the higher the likelihood of antigen being faulty. The MCAV is calculated by dividing the number of times an antigen appears in the mature context, nb mature by the total number of presentation of that antigen, nb antigen. Once the MCAV is calculated for each antigen, the algorithm can perform its classification task. This is done by comparing the MCAV of each antigen to an anomalous threshold. The anomaly threshold can be either a user-defined parameter or can be generated automatically from the data.

$$Th_a = \frac{AN}{TN} \tag{5.3}$$

where, Th_a is anomalous threshold, AN is the no. of anomalous data or faulty data items and TN is the total no. of data items. Those antigens whose MCAVs are greater than the Th_a are classified into the faulty set while the others are classified into the fault free set.

5.4 Simulation Results and Discussion

The proposed algorithm is evaluated using standard simulator NS-2.35 using generic parameters such as fault detection accuracy, false alarm rate, false positive rate, fault detection latency and energy consumption. Sensor nodes are randomly deployed in an area of $1000 \times 1000\ m^2$. To measure the performance of the proposed algorithm with existing algorithms proposed by authors Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], various performance parameters are calculated. The network parameters and their values used in the simulation are shown in Table 5.2.

5.4.1 Performance Analysis using FDA, FAR and FPR

Initially, the nodes in the network is assumed to be fault free. Gradually faulty nodes are added to the network from 5% to 30% in increment of 5%. The proposed algorithm fault diagnosis using dendritic cell algorithm (FDDCA) is evaluated using different performance metrics such as FDA, FAR, FPR, FDL and EC. The FDA, FAR and FPR with respect to the percentage of faulty sensor nodes are plotted in Figures 5.2, 5.3 and 5.4, respectively. It is observed that the proposed fault diagnosis using dendritic cell algorithm (FDDCA) algorithm outperforms the existing algorithms. The proposed FDDCA gives higher FDA and lowers FAR and FPR than the existing algorithms. The FDA decreases, and the FAR

Table 5.2: Network Parameters

Parameter	Value
No. of nodes	1000
Simulation time	150 s
Transmission range	150 m
Channel rate	250 kbps
Traffic	CBR
MAC protocol	IEEE 802.15.4
Propagation model	TwoRayGround
Initial energy	10 J
Packet size	512 bytes
Packet rate	1 pkt/s
Antenna model	Omni Antenna
Grid size	$1000 \times 1000\ m^2$
Topology	Arbitrary network

and FPR increases when the percentage of faulty sensor node increases. In the simulation, the percentage of faulty sensor nodes include hard permanent, soft permanent, soft intermittent and soft transient faulty sensor nodes. The FDDCA gives an average of 99.31% FDA, 3.21% of FAR and 0.69% of FPR whereas, the existing Mohapatra et al. [25] gives 97.8%, 3.46% and 2.2%, Panda et al. [9] gives 94.61%, 4.11% and 5.39% and Elhadef et al. [26] gives 92.85%, 5.06% and 7.15% of FDA, FAR and FPR, respectively. The improvement of the proposed algorithm over the existing algorithms are presented in summary Section 5.5.

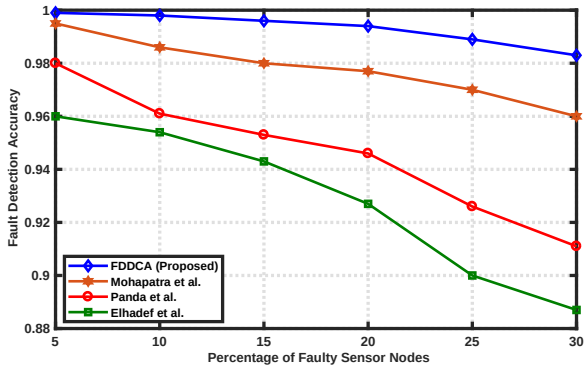

Figure 5.2: FDA vs. Percentage of Faulty Sensor Nodes

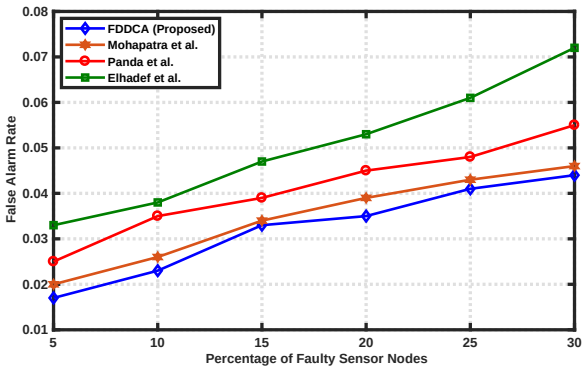

Figure 5.3: FAR vs. Percentage of Faulty Sensor Nodes

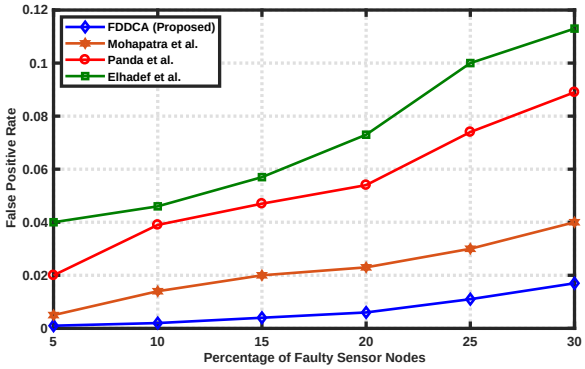

Figure 5.4: FPR vs. Percentage of Faulty Sensor Nodes

5.4.2 Fault Detection Latency

Fault detection latency with respect to the percentage of faulty sensor nodes is shown in Figure 5.5. By taking 1000 sensor nodes we are increasing the percentage of faulty sensor nodes from 5% - 30% in the increment of 5% at every step. When the percentage of faulty sensor node increases the average fault detection latency for FDDCA algorithm is 2.72 second whereas, 2.94, 3.11 and 3.26 second for the existing algorithm Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. There is less fault detection latency of our proposed FDDCA algorithm as compared to the existing algorithms.

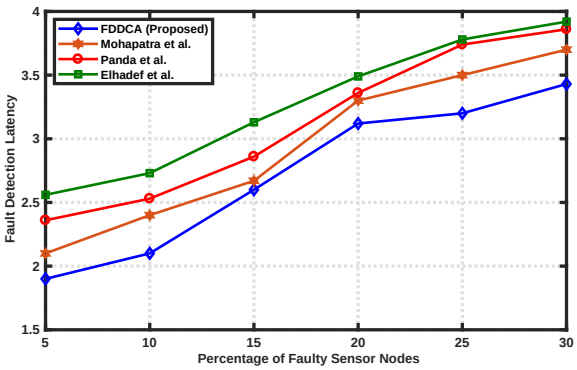

Figure 5.5: FDL vs. Percentage of Faulty Sensor Nodes

5.4.3 Energy Consumption

Energy consumption with respect to the percentage of faulty sensor nodes is shown in Figure 5.6. The total energy consumption depends upon the total amount of energy required to transmit the data and to receive the data for the diagnosis. Here the percentage of faulty sensor node gradually increases from 5% to 30% whereas the total number of sensor nodes are constant. The percentage of energy consumption is 1.19 joule for the proposed FDDCA algorithm and 1.46 joule, 1.63 joule and 2.56 joule for Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. Hence, the proposed algorithm consumes less energy than the existing algorithms. The percentage of improvement in energy consumption by the proposed algorithm with respect to existing algorithm is given in the next section.

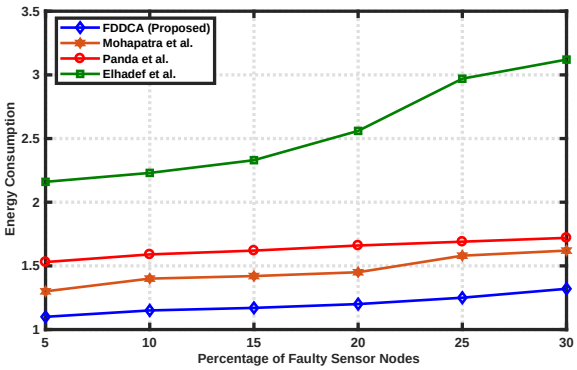

Figure 5.6: EC vs. Percentage of Faulty Sensor Nodes

5.5 Summary

In this chapter, the faulty nodes are detected using the proposed FDDCA algorithm. The important feature of this algorithm is that no training data is required. The performance of the algorithm is evaluated by using the performance metrics where it is shown that, the FDDCA algorithm gives better result as compared to the existing algorithms in terms of fault detection

accuracy, false alarm rate, false positive rate, fault detection latency and energy consumption. The fault detection accuracy of the proposed FDDCA algorithm is improved by 1.51%, 4.7% and 6.46% over Mohapatra et al.[25], Panda et al. [9], Elhadef et al. [26], respectively. The false alarm rate of the proposed algorithm is improved by 0.25%, 0.9% and 1.85% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. The false positive rate of the proposed algorithm is improved by 1.51%, 4.7% and 6.46% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. The proposed algorithm provides less fault detection latency i.e., 7.48%, 12.54% and 16.56% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively and consumes less energy i.e., 18.49%, 26.99% and 53.51% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively.

Chapter 6

Fault Diagnosis in Wireless Sensor Network using Artificial Immune Network

6.1 Introduction

Immune system inspires many researchers and scientists to solve various computational problems. This is because of the advantages of the immune system i.e., immune recognition, reinforcement learning, immune memory, feature extraction, diversity and robustness. Such attributes are combined by the immune system to efficiently construct pattern classifier. The way human body uses adaptive learning and defense mechanism to protect our body, it can also be used as fault diagnosis in wireless sensor network. An artificial immune network uses the concept of immune network theory. It receives antigens as input and returns an immune network composed of set of B cells and the connections between them [113].

Once a sensor node is faulty in a wireless sensor network, it often triggers a fatal situation as the impact of failed sensor node propagates throughout the system. To prevent this type of situation, fault diagnosis techniques have become more important. The major contribution of this chapter is stated as follows.

- An artificial immune network based fault diagnosis algorithm is used to detect the faults in WSN.
- A total time period with status register and neighbor table is used to detect hard permanent faults.
- K nearest neighbor (KNN) approach is used to classify the faults into their respective types.
- Validation of the proposed method is carried out using NS-2.35 simulator and comparison study is also performed with existing approaches such as Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26].
- Different metrics such as fault detection accuracy (FDA), false alarm rate (FAR), false positive rate (FPR), fault classification accuracy (FCA), false classification rate (FCR),

fault detection latency (FDL) and energy consumption (EC) are used to measure the performance of our proposed algorithm.

The organization of the chapter is as follows. A brief description of the immune system and the mechanism of the artificial immune network is described in Section 6.2. The proposed algorithm is given in Section 6.3. The results are discussed in Section 6.4. Finally, the chapter is concluded in Section 6.5.

6.2 Artificial Immune Network

6.2.1 Brief Description of Immune System

The function of the immune system is to protect our body from being attacked by various pathogens such as viruses and bacteria. Pathogens have different types of antigens and once our body sees those antigens they will start producing specific antibodies against those antigens to specifically destroy those cells. The antibody response will very specific though it will take longer time to develop that is why it is also called as delayed response. This delayed response produces some cells known as lymphocytes. There are two types of cells that are produced in lymphocytes, they are T cells and B cells. T cells are helping other cells, for example T cells activates B cells. Once B cells are active they will produce antibodies. T cells can divide into two types one is a helper T cell which help other cells to develop and the other is a killer T cell which kills after finding specific target.

The main features of the immune system are learning and memory. The immune system will learn the antigen structures and remember those structures so that by generating a large initial high-affinity clone, subsequent responses to the similar antigens will be quick and effective. This is an inherent scheme of a human immune system reinforcement learning technique.

The mechanism of the immune system is shown in Figure 6.1. I-II shows pathogens are entered into the body and activation of T cell occurs in III, which activates B cells in IV, antigens are matched in V, antibodies are produced in VI and in VII antigen destroy it [114].

6.2.2 Mechanism of Artificial Immune Network

The theory of clonal selection and maturation of affinity defines the basic characteristics of an adaptive immune response to an antigenic stimulus. This develops the principle that only those cells of high affinity are chosen for proliferation (clone) and mutation, whereas those of low affinity are not. The selected cells undergo a maturation process of affinity, which increases in concentration and affinity, while those not selected undergo a process of clonal deletion, clonal anergy, or editing of receptors. The immune system learns to know the structures of this antigen when repeatedly exposed to a given antigen to increase its recognizing capability [115].

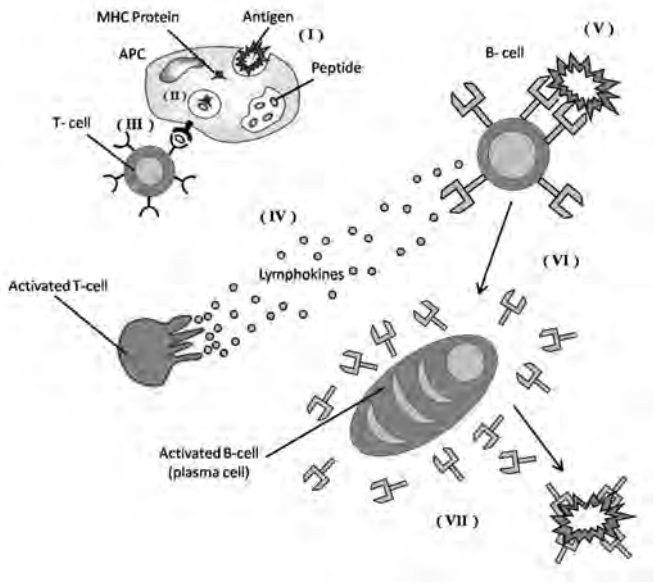

Figure 6.1: Mechanism of immune system

Jerne [78] originally proposed the concept of the immune network in 1974. In contrast to the concept of clonal selection, the theory of the immune network hypothesizes that the immune system maintains a regulated network of cells and molecules which establish connections not only between an antibody and an antigen, but also between the antibodies itself. If an antigen is recognized by an antibody ab_1, then ab_1 may be recognized by ab_2 and ab_2 may be recognized by ab_3, which creates a network of antibodies. Antibodies identification will result in a negative response, resulting in antibodies tolerance and suppression. This will suppress the similar type of antibodies. The immune system will eventually reach the stabilization. This ensures that antibodies to memory can be distributed uniformly in antigen space. Therefore, although the number of antibodies in the immune system is relatively small, they cover the entire space of antigens and identify all antigens.

6.3 Proposed Algorithm

The proposed algorithm consists of two phases such as (1) fault detection and (2) fault classification phase. In the fault detection phase, an algorithm was proposed to identify the faulty nodes in WSN based on artificial immune network. Then in the classification phase, using the k nearest neighbor (KNN) approach the faulty nodes are classified into different types.

6.3.1 Fault Detection Phase

This section is further classified into two phases, i.e., (1) detection of hard fault and (2) detection of soft fault. The details of the fault detection mechanism are given as follows.

Detection of Hard Fault

Each sensor node sn_i sends a request message req_{msg} to its neighboring sensor nodes $sn_j \in Neg(sn_i)$. Upon receiving the message, node sn_j sends a response message rep_{msg} to the node sn_i. To complete the message transmission the total time period required is defined in Equation (6.1)

$$T_t = T_{req} + T_{rep} + T_{qd} + T_{td} + T_{pd} \tag{6.1}$$

where, T_{req} is the propagation time of the request message from sn_i to sn_j, T_{rep} is the propagation time of response message from sn_j to sn_i, T_{qd} is the queuing delay, T_{td} is the transmission delay and T_{pd} is the processing delay.

Each sensor node sn_i sends a request message to its neighboring sensor nodes $sn_j \in Neg(sn_i)$ and wait for a response message. Each sensor node sn_i maintains a status register $SR_{ij} \in \{0, 1\}$ for all of its neighboring sensor nodes. A time period 2 $\times T_t$ time is set for each node in the network (T_t is defined in Equation (6.1)), if the node sn_i receives a reply (rep) message from its neighboring sensor nodes $sn_j \in Neg(sn_i)$ then the SR_{ij} is set to 1 otherwise 0. For each sensor node $sn_i \in S$, if the summation $\sum_{sn_j}^{Neg(sn_i)} SR_{ij}$ is less than to $\lceil \frac{Neg(sn_i)}{2} \rceil$ then the node sn_i is identified as hard permanent faulty node. Similarly, the status of all other nodes can be computed. The detection of hard fault is described in Algorithm 6.1.

Algorithm 6.1 Hard Fault Detection Algorithm

Initialize: Neighbor table ($NegT_i$), time period ($T_{req}, T_{rep}, T_{qd}, T_{td}, T_{pd}$), and status register $SR_{ij} \in \{0, 1\}$;
for each sensor node $sn_i \in S$ **do**
    Node sn_i communicate with its neighboring nodes $Neg(sn_i)$ by sending request message req_{msg};
    Upon receiving the data, the neighboring nodes $Neg(sn_i)$ sends a response message rep_{msg} to sn_i;
    Calculate the total time period T_t;
    $T_t = T_{req} + T_{rep} + T_{qd} + T_{td} + T_{pd}$;
    Node sn_i sends a req message to its neighboring nodes $Neg(sn_i)$ and wait for rep message;
    for each neighbor $sn_j \in Neg(sn_i)$ **do**
        if the rep message is received within the 2 $\times T_t time$ **then**
            SR_{ij} = 1;
        else
            SR_{ij} = 0;
        end if
    end for
    if $\sum_{sn_j}^{Neg(sn_i)} SR_{ij} < \lceil \frac{Neg(sn_i)}{2} \rceil$ **then**
        the node sn_i is identified as hard permanent faulty node;
    else
        the node sn_i is fault free;
    end if
end for
Broadcast the status of the faulty nodes in the network;

Detection of Soft Fault

Initially, the fault samples of the WSN are mapped into the set of antigens (ag). Then n number of antibodies are generated randomly and some antigens are chosen as memory antibodies (ab_m). In this chapter, the antigens are represented as $AG = \{ag_1, ag_2, ag_3, ..., ag_n\}$. Affinity ($af$) usually corresponds to a metric which identifies the

degree of similarity. The affinity af is the ratio between the test syndrome and the number of time comparison. Affinity af(i,j) between the antigen and antibody is defined in Equation (6.2).

$$af(i,j) = \frac{\sigma(i,j)}{|No.of comparision|} \tag{6.2}$$

The test syndrome $\sigma(i,j)$ is defined in Equation (6.3).

$$\sigma(i,j) = |(fs_1, fs_2, ..., fs_m)_i \cap (fs_1, fs_2, ..., fs_m)_j|\,, \tag{6.3}$$

where $(fs_1, fs_2, ..., fs_m)_i$ is the fault status of sensor node $sn_i \in S$ and $(fs_1, fs_2, ..., fs_m)_j$ is the fault status of sensor node $sn_j \in S$. If the fault status between antigens are matched, then the test syndrome $\sigma(i,j)$ value is the cardinality of the number of matches.

The information of fault types were added in the antigen set (ag) and memory antibodies (ab_m) to enhance the learning capability of antibodies (ab). Hence, the improved antigen set represented as $AG = \{ag_1, ag_2, ag_3, ..., ag_n, F\}$. Here, F is the type of fault and the memory antibodies set represented as same with the improved antigen set. The role of F in ab_m lies in (i) the interaction of ab and ag not only depends on affinity but also on its type. Antigen is identified by antibodies of same type of fault. (ii) the information of types of fault is taken care by ab_m for both suppression and recognition. (iii) the stopping criteria depends on both the affinity and types of fault of ab_m. For the convergence of the algorithm, the similar types of antigens are eliminated directly. The antibodies recognizing antigens with high affinity value are activated to proliferate and mutate and thus the affinity value will increase. After immune suppression these antibodies became memory antibodies. The number of clones (N_c) can be computed as given in Equation (6.4).

$$N_c = round\,((\beta * T)/i) \tag{6.4}$$

Where, for each antigen β is a multiplication factor, T is the total number of antigens, i is the current antigen and round (.) is the operator that round its variable towards the closest integer. The algorithm is presented in Algorithm 6.2.

Algorithm 6.2 AINFDA Algorithm

Initialize multiplication factor (β), initial antibodies (ab), initial antigen (ag), suppression threshold θ_s, network threshold θ_n;
Randomly generate n number of antibodies;
for each antigen ag $\in$ AG **do**
    Calculate the affinity af(i,j);
    Select n antibodies with highest affinity for cloning and put it in a clone set CS;
    for each antibody in CS **do**
        Perform mutation which is inversely proportional to affinity;
        Recompute affinity;
    end for
    Select the antibodies with highest affinity and place in M_j;
    Remove all elements of M_j whose affinity is less than θ_s and generate M_j^*;
    $ab_m = ab_m^* \cup M_j^*$
end for
Determine affinities between each pair of antibodies in ab_m;
Remove all antibodies whose affinity with another antibody is less than θ_n;
A new randomly generated antibody is introduced to ab_m;
Repeat the process until the stopping criteria has been met;

6.3.2 Fault Classification Phase

After the faulty nodes are detected successfully in the detection phase, they are classified into different types in this phase using k nearest neighbor (KNN) approach [116]. The KNN algorithm is simple and easy to implement.

This assigns the majority class an input pattern according to Euclidean distances in the training data set between the pattern and its k nearest neighbors. The developed ab_m are eligible for classification after the artificial immune network training has been completed. By measuring the Euclidean distance between test antigen and ab_m, a majority vote of the outputs of the most activated ab_m is used to determine the type of fault of the test antigen [96].

Algorithm 6.3 Fault Classification Algorithm

Input: Load the sensor node's data values;
Output: Predicted class (i.e., permanent, intermittent, & transient fault);
Initialize and define k;
for each $i = 1$ to n **do**
    Calculate the Euclidean distance between input data sample and training data sample;
end for
Sort the calculated distance in ascending order based on their distance values;
Get top k rows from the sorted array;
Get the most frequent class of these rows;
Return the predicted class;

6.4 Simulation Results and Discussion

The artificial immune network based fault diagnosis algorithm (AINFDA) has been compared with three existing algorithms such as Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26] and evaluated using the network simulator NS-2.35 [24]. The performance is measured by calculating the performance parameters such as fault detection accuracy (FDA), false alarm rate (FAR), false positive rate (FPR), Fault Classification Accuracy (FCA), False Classification Rate (FCR), fault detection latency (FDL) and energy

Table 6.1: Simulation Parameters

Parameter	Value
No. of nodes	1000
Simulation time	150 s
Transmission range	150 m
Channel rate	250 kbps
Traffic	CBR
MAC protocol	IEEE 802.15.4
Propagation model	TwoRayGround
Initial energy	10 J
k	3
Packet size	512 bytes
Packet rate	1 pkt/s
Antenna model	Omni Antenna
Grid size	1000×1000 m^2
Topology	Arbitrary network

consumption (EC). 1000 sensor nodes are deployed randomly in an area of 1000*1000 m^2. The parameters which are used in the simulation are provided in Table 6.1.

6.4.1 Performance Analysis using FDA, FAR and FPR

Initially, the nodes in the network is assumed to be fault free. Gradually faulty nodes are added to the network from 5% to 30% in increment of 5%. The proposed method is evaluated using different performance metrics such as FDA, FAR, FPR, FCA, FCR, FDL and EC. The performance metrics are defined as follows.

- FDA = $\frac{\sum \text{No. of faulty nodes detected as faulty}}{\text{Total no. of faulty nodes}}$
- FAR = $\frac{\sum \text{No. of fault free nodes detected as faulty}}{\text{Total no. of fault free nodes}}$
- FPR = $\frac{\sum \text{No. of faulty nodes detected as fault free}}{\text{Total no. of faulty nodes}}$
- FCA = $\frac{\sum \text{No. of faulty nodes classified correctly}}{\text{Total no. of faulty nodes}}$
- FCR = $\frac{\sum \text{No. of faulty nodes classified wrongly}}{\text{Total no. of faulty nodes}}$
- FDL = Amount of time required to detect all the sensor nodes
- EC = Total energy consumed by the network to identify the faulty sensor nodes

The FDA, FAR and FPR with respect to the percentage of faulty sensor nodes are plotted in Figures 6.2, 6.3 and 6.4, respectively. We can see that the proposed artificial immune network based fault diagnosis algorithm (AINFDA) outperforms the existing algorithms as the existing algorithms cannot detect the intermittent and transient fault. The proposed

AINFDA gives higher FDA and lowers FAR and FPR than the existing algorithms. The FDA decreases, and the FAR and FPR increases when the percentage of faulty sensor node increases. Here the percentage of faulty sensor node includes hard permanent, soft permanent, soft intermittent and soft transient faulty sensor nodes. The AINFDA gives an average of 99.16% FDA, 3.46% of FAR and 0.84% of FPR whereas, the existing Mohapatra et al. [25] gives 97.65%, 3.85% and 2.35%, Panda et al. [9] gives 94.15%, 4.21% and 5.85% and Elhadef et al. [26] gives 92.7%, 5.2% and 7.3% of FDA, FAR and FPR, respectively.

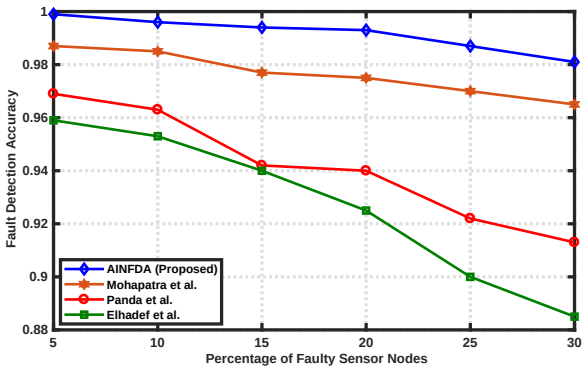

Figure 6.2: FDA vs. Percentage of Faulty Sensor Nodes

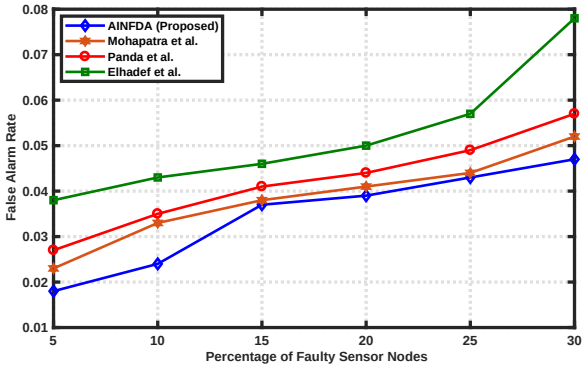

Figure 6.3: FAR vs. Percentage of Faulty Sensor Nodes

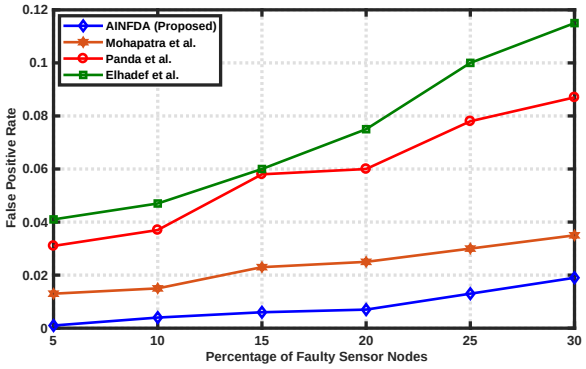

Figure 6.4: FPR vs. Percentage of Faulty Sensor Nodes

6.4.2 Performance Analysis of Fault Classification

The fault classification phase is implemented using MATLAB R2017a. Fault classification accuracy (FCA) is the correct classification rate and false classification rate (FCR) is the misclassification rate are the two performance parameters which is used to check the performance of the classification phase. Data of 1000 sensor nodes are collected, in which approximately 70% of data are randomly selected for training purposes and 30% of data are selected for testing purposes. After training, the testing data values are unknown for the KNN model and the performance results (FCA and FCR) are calculated by testing this sensor data values. The collected sensor data values are of three different types of soft fault (permanent, intermittent, and transient). In the testing phase, the faulty sensor node increases consistently from 5% to 30% for performance analysis. The faulty nodes contain different types of soft faults. It is observed that, for increasing the faulty nodes the fault classification accuracy (FCA) decreases and false classification rate (FCR) increases. The results are shown in the Table 6.2. As the intermittent and transient fault patterns are not fixed and changes with respect to the time interval it behaves arbitrarily (like byzantine fault). Therefore, the correct classification rate (FCA) decreases from 0.9805 to 0.9381 and the misclassification rate (FCR) increases from 0.0195 to 0.0619 by increasing the faulty nodes from 5% to 30% consistently. The fault classification phase gives the average classification accuracy approximately 95.91% and the average misclassification rate 0.04.

Table 6.2: Fault Classification Results

Percentage of faulty node	Fault classification accuracy	False classification rate
5	0.9805	0.0195
10	0.9727	0.0273
15	0.9668	0.0332
20	0.9543	0.0457
25	0.9424	0.0576
30	0.9381	0.0619

6.4.3 Fault Detection Latency

Fault detection latency with respect to the percentage of faulty sensor nodes is shown in Figure 6.5. By taking 1000 sensor nodes we are increasing the percentage of faulty sensor nodes from 5% - 30%. When the percentage of faulty sensor node increases the average fault detection latency for AINFDA algorithm is 3.01 second whereas, 3.15, 3.22 and 3.38 second for the existing algorithm Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. There is less fault detection latency of our proposed AINFDA algorithm as compared to the existing algorithms.

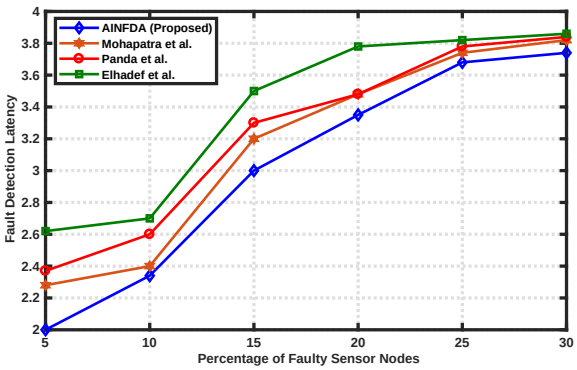

Figure 6.5: FDL vs. Percentage of Faulty Sensor Nodes

6.4.4 Energy Consumption

Energy consumption with respect to the percentage of faulty sensor nodes is shown in Figure 6.6. The total energy consumption depends upon the total amount of energy required to transmit the data and to receive the data for the diagnosis. Here the percentage of faulty sensor node gradually increases from 5% to 30% whereas the total number of sensor nodes are constant. The percentage of energy consumption is 1.30 joule for the proposed AINFDA algorithm and 1.48 joule, 1.67 joule and 2.59 joule for Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. Hence, the proposed algorithm consumes less energy than the existing algorithms.

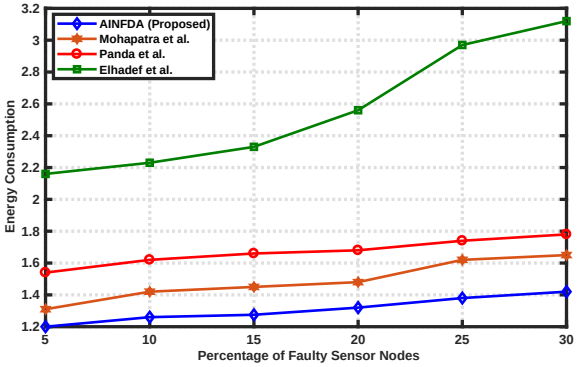

Figure 6.6: EC vs. Percentage of Faulty Sensor Nodes

6.5 Summary

In this chapter, an artificial immune network based fault diagnosis algorithm has been proposed to diagnose the faulty sensor nodes. In this algorithm to train and optimize the fault samples, learning, memory and suppression mechanism of immune network is used. Fault type information has been added to memory antibodies so that it can learn and memorize the same types of faults. Hence, classification accuracy is improved. To classify the faults

KNN algorithm is used. Experimental result shows that AINFDA gives better result than the existing algorithms in terms of fault detection accuracy, false alarm rate, false positive rate, fault detection latency and energy consumption. The fault classification performance is measured by fault classification accuracy and false classification rate. The fault detection accuracy of the proposed AINFDA algorithm is improved by 1.51%, 5.01% and 6.46% over Mohapatra et al.[25], Panda et al. [9], Elhadef et al. [26], respectively. The false alarm rate of the proposed algorithm is improved by 0.39%, 0.75% and 1.74% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. The false positive rate of the proposed algorithm is improved by 1.51%, 5.01% and 6.46% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. The proposed algorithm provides less fault detection latency i.e., 4.44%, 6.52% and 10.94% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively and consumes less energy i.e., 12.16%, 22.15% and 49.80% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively.

Chapter 7

Conclusion

The research in this thesis is focuses on the artificial immune system approaches for fault diagnosis of wireless sensor network. The comparison outcome of contributions is presented. Simulation result shows that the proposed algorithms perform better than the existing algorithms. For the extension of this work, future problems are also outlined.

7.1 Conclusion

In this thesis, fault diagnosis protocols have been proposed using different artificial immune system approaches such as clonal selection principle (CSP), negative selection algorithm (NSA), dendritic cell algorithm (DCA) and artificial immune network (AIN). Simulations and testbed experiments in the indoor laboratory environment have been conducted to evaluate the performance of the protocols using standard generic parameters such as fault detection accuracy (FDA), false alarm rate (FAR), false positive rate (FPR), fault classification accuracy (FCA), false classification rate (FCR), fault detection latency (FDL) and energy consumption (EC). The result shows that the FDA, FAR, FPR, FCA, FCR, FDL and EC of the proposed protocols such as FDCSP, INSA, FDDCA and AINFDA are better as compared to the existing algorithms such as Mohapatra et al.[25], Panda et al. [9] and Elhadef et al. [26]. The comparison outcome of proposed algorithms is also discussed. Throughout the thesis the sensor nodes are assumed to be faulty while links are fault free which are taken care by underlying MAC layer protocols. All the proposed algorithms consider both hard and soft faults in sensor nodes. An arbitrary network topology has been considered to represent a realistic scenario for sensor network deployment in the field.

An efficient fault detection algorithm based on clonal selection principle (FDCSP) of AIS has been proposed to detect faulty sensor nodes and then the faults are classified into respective types using the probabilistic neural network approach. After the actual fault status is detected, the faulty sensor nodes are isolated in the isolation phase. The performance of the algorithm is evaluated by using the performance metrics where it is found that the fault detection accuracy of the proposed FDCSP algorithm is improved by 1.8% over Mohapatra et al. [25], 5.06% over Panda et al. [9] and 6.45% over Elhadef et al. [26] algorithm. The false alarm rate of the proposed algorithm is improved by 0.43%, 0.78% and 1.88% over

Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. The false positive rate of the proposed algorithm is improved by 1.81%, 5.07% and 6.46% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. The fault classification performance is measured by fault classification accuracy and false classification rate. The simulation result also shows that the FDCSP algorithm provides less fault detection latency i.e., 4.11%, 6.19% and 10.35% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively and consumes less energy i.e., 11.40%, 20.95% and 49.03% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively.

An improved negative selection algorithm (INSA) has been proposed to diagnose the faulty sensor nodes and classified into soft permanent, soft intermittent and soft transient using the support vector machine. The performance of the algorithm is evaluated by using the performance metrics such as FDA, FAR, FPR, FCA, FCR, FDL and EC. It is shown that, the fault detection accuracy of the proposed INSA algorithm is improved by 1.55% over ,4.97% over Panda et al. [9] and 6.49% over Elhadef et al. [26] algorithm. The false alarm rate of the proposed algorithm is improved by 0.33%, 0.85% and 1.86% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. The false positive rate of the proposed algorithm is improved by 1.55%, 4.97% and 6.49% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. The fault classification phase gives the average classification accuracy approximately 97% and the average misclassification rate 0.03. The simulation result also shows that the proposed algorithm provides less fault detection latency i.e., 4.26%, 8.75% and 13.09% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively and consumes less energy i.e., 11.64%, 21.81% and 50% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively.

Fault diagnosis using dendritic cell algorithm (FDDCA) has been proposed to detect the faulty nodes. The important feature of this algorithm is that no training data is required. The performance of the algorithm is evaluated by using the performance metrics where it is shown that, the FDDCA algorithm gives better result as compared to the existing algorithms in terms of fault detection accuracy, false alarm rate, false positive rate, fault detection latency and energy consumption. The fault detection accuracy of the proposed FDDCA algorithm is improved by 1.51%, 4.7% and 6.46% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. The false alarm rate of the proposed algorithm is improved by 0.25%, 0.9% and 1.85% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. The false positive rate of the proposed algorithm is improved by 1.51%, 4.7% and 6.46% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. The proposed algorithm provides less fault detection latency i.e., 7.48%, 12.54% and 16.56% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively and consumes less energy i.e., 18.49%, 26.99% and 53.51% over Mohapatra et al., Panda et al. [9] and Elhadef et al. [26], respectively.

An artificial immune network based fault diagnosis algorithm (AINFDA) has been

proposed to diagnose all the hard and soft faulty sensor nodes. In this algorithm to train and optimize the fault samples, learning, memory and suppression mechanism of immune network is used. Fault type information has been added to memory antibodies so that it can learn and memorize the same types of faults. So, that without performing any computation, the faults can be detected and hence, classification accuracy is improved. Experimental result shows that the fault detection accuracy of the proposed AINFDA algorithm is improved by 1.51%, 5.01% and 6.46% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. The false alarm rate of the proposed algorithm is improved by 0.39%, 0.75% and 1.74% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. The false positive rate of the proposed algorithm is improved by 1.51%, 5.01% and 6.46% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively. The proposed algorithm provides less fault detection latency i.e., 4.44%, 6.52% and 10.94% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively and consumes less energy i.e., 12.16%, 22.15% and 49.80% over Mohapatra et al. [25], Panda et al. [9] and Elhadef et al. [26], respectively.

7.2 Comparison of Proposed Algorithms

Motivated by the challenges and need of efficient fault diagnosis algorithms in wireless sensor network four algorithms are proposed based on artificial immune system. The first proposed algorithm is based on clonal selection principle of AIS. For the convergence of the algorithm the proposed work uses cloning and mutation. After the detection of faulty nodes the faults are classified into their respective types using PNN approach. The decision making process will be affected in the presence of faulty nodes so, the faults are isolated in the isolation phase. The proposed algorithm gives better performance over Mohapatra et al., [25], Panda et al. [9] and Elhadef et al. [26] in terms of fault detection accuracy (FDA), false alarm rate (FAR), false positive rate (FPR), fault detection latency (FDL) and energy consumption (EC). However, the computational overhead is more and to reduce that, the second algorithm was proposed.

The second proposed algorithm is based on the negative selection principle of AIS. It discriminates between the self and non-self. The major advantage of this approach is that, it does not need any prior knowledge of fault patterns and able to adopt the changes in the faulty and fault free situations. The accuracy is increased as there is no need of cloning and mutation and computational complexity is also less than the first proposed algorithm.

The third proposed algorithm is based on the dendritic cell algorithm of AIS. It reduces the false positive rate and responds to danger rather than non-self or, fault. The important feature of this algorithm is that no training data is required. Hence it gives better result as compared to the second proposed algorithm. This proposed algorithm requires less number of message exchanges and computational complexity than the other proposed algorithms

The fourth proposed algorithm is based on the concept of artificial immune network of AIS. In this algorithm to train and optimize the fault samples, learning, memory and suppression mechanism of immune network is used. Fault type information has been added to memory antibodies so that it can learn and memorize the same types of faults. Hence, classification accuracy is also improved.

The overall comparison of the proposed algorithms is shown in Table 7.1. The proposed algorithms are compared in terms of different metrics such as methodology, accuracy, false alarm rate, false positive rate, fault detection latency, energy consumption, training, affinity, data set, message complexity, diagnosis approach and implementation. It is noted that the solutions for clonal and artificial immune network depends on computation of affinity values whereas, the methods such as negative selection and dendritic cell algorithms do not use affinity evaluation for finding the solution. As affinity evaluation incurs more computational overhead, clonal selection and artificial immune network algorithm is avoided. Negative selection and dendritic cell is preferred with respect to computational overhead. The proposed algorithm negative selection has a requirement of training whereas, other three proposed algorithms do not depend on training. In order to satisfy the training requirement, the negative selection depends on real data set. When the real data set is not available and computational overhead can be compromised, the proposed algorithms such as clonal selection, dendritic cell, and artificial immune network algorithms are preferred. In fact, dendritic cell algorithm gives efficient solution as compared to that of other remaining three proposed algorithms in terms of FDA, FAR, FPR, FDL, EC and message complexity as it does not need training and evaluation of affinity. The comparison shows the positive and negative aspects of the proposed algorithms.

The existing Mohapatra et al. [25] and Panda et al. [9] dependent upon their neighboring nodes. So, in this case each sensor node send their sensed data for m number of times to its neighboring nodes within the transmission range. The degree of a node is defined as the number of neighbor nodes present in its transmission range. If d is considered as the average degree of the network, then the message complexity for n number of nodes is defined as $\mathcal{O}(n \times m \times d)$. The message complexity for Elhadef et al. [26] is $\mathcal{O}(n \times h \times t)$ where, n is the number of sensor nodes and h is the number of hidden node and t is the number of iteration. Algorithms 1 and 4 proposed in chapter 3 and 6 consumes messages of $\mathcal{O}(n \times d)$ where, n is the number of sensor nodes and d is the degree of a sensor node. The message complexity of algorithm 2 proposed in chapter 4 is $\mathcal{O}(n \times m)$ where, n is the number of sensor nodes and m is the number of matching. The message complexity of algorithm 3 proposed in chapter 5 is $\mathcal{O}(1)$ as the proposed algorithm exchanges a fixed number of messages to diagnose a faulty or fault free node. These message complexities of proposed algorithms are better as compared to existing algorithms.

Table 7.1: Comparison of proposed algorithms

Criteria	**1st proposed algorithm**	**2nd proposed algorithm**	**3rd proposed algorithm**	**4th proposed algorithm**
Methodology	Clonal selection principle	Negative selection algorithm	Dendritic cell algorithm	Artificial immune network
Topology	Arbitrary network	Arbitrary network	Arbitrary network	Arbitrary network
No. of nodes	1000	1000	1000	1000
Accuracy	99.11%	99.25%	99.31%	99.16%
FAR	3.55%	3.3%	3.21%	3.46%
FPR	0.88%	0.75%	0.69%	0.84%
FDL	3.03 s	2.92 s	2.72 s	3.01 s
EC	1.49 J	1.46 J	1.19 J	1.30 J
Training	Not required	Required	Not required	Required
Affinity	Required	Not required	Not required	Required
Data set	Not required	Required	Not required	Not required
Message complexity	$\mathcal{O}(n \times d)$	$\mathcal{O}(n \times m)$	$\mathcal{O}(1)$	$\mathcal{O}(n \times d)$
Diagnosis approach	Distributed	Distributed	Distributed	Distributed
Network	Static	Static	Static	Static
Implementation	Simulation	Simulation and test-bed	Simulation	Simulation

7.3 Future Scope

The proposed algorithms are based on the assumption that the network topology is static. In the future, we will consider the dynamic network topology in which sensor nodes join and leave the network during the diagnosis time. The fault recovery during the diagnosis is to be also considered. In this work, only sensor nodes are assumed to be faulty and links are fault free. In future, the fault diagnosis algorithm in the occurrence of failure of nodes as well as links will be developed. Furthermore, the work can be extended to handle byzantine fault, in which the behavior of the faulty node changes dynamically. Heterogeneous types of sensor nodes will be considered in the future. The wireless sensor network having heterogeneity such as different processing power, storage and equipped with different types of sensors will be considered in developing fault diagnosis algorithm. In order to see the feasibility of the proposed algorithms in terms of their performance, our next work will be to develop real time implementation of WSN in the application environment such as forest fire monitoring, under water sensor network, battle field and agricultural monitoring.

References

[1] Yick, J., Mukherjee, B., and Ghosal, D., 2008. "Wireless sensor network survey". *Computer networks,* **52**(12), pp. 2292–2330.

[2] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E., 2002. "Wireless sensor networks: a survey". *Computer networks,* **38**(4), pp. 393–422.

[3] Mahapatro, A., and Khilar, P. M., 2013. "Fault diagnosis in wireless sensor networks: A survey". *IEEE Communications Surveys & Tutorials,* **15**(4), pp. 2000–2026.

[4] Avizienis, A., Laprie, J.-C., Randell, B., and Landwehr, C., 2004. "Basic concepts and taxonomy of dependable and secure computing". *IEEE transactions on dependable and secure computing,* **1**(1), pp. 11–33.

[5] Barooah, P., Chenji, H., Stoleru, R., and Kalmar-Nagy, T., 2011. "Cut detection in wireless sensor networks". *IEEE transactions on parallel and distributed systems,* **23**(3), pp. 483–490.

[6] Bondavalli, A., Chiaradonna, S., Di Giandomenico, F., and Grandoni, F., 2000. "Threshold-based mechanisms to discriminate transient from intermittent faults". *IEEE Transactions on Computers,* **49**(3), pp. 230–245.

[7] Elhadef, M., Boukerche, A., and Elkadiki, H., 2008. "A distributed fault identification protocol for wireless and mobile ad hoc networks". *Journal of parallel and distributed computing,* **68**(3), pp. 321–335.

[8] Barborak, M., Dahbura, A., and Malek, M., 1993. "The consensus problem in fault-tolerant computing". *ACM Computing Surveys (CSur),* **25**(2), pp. 171–220.

[9] Panda, M., and Khilar, P. M., 2015. "Distributed self fault diagnosis algorithm for large scale wireless sensor networks using modified three sigma edit test". *Ad Hoc Networks,* **25**, pp. 170–184.

[10] Malek, M., 1980. "A comparison connection assignment for diagnosis of multiprocessor systems". In Proceedings of the 7th annual symposium on Computer Architecture, ACM, pp. 31–36.

[11] Blough, D. M., and Brown, H. W., 1999. "The broadcast comparison model for on-line fault diagnosis in multicomputer systems: theory and implementation". *IEEE Transactions on Computers,* **48**(5), pp. 470–493.

[12] Yang, X., Megson, G. M., and Evans, D. J., 2005. "A comparison-based diagnosis algorithm tailored for crossed cube multiprocessor systems". *Microprocessors and Microsystems,* **29**(4), pp. 169–175.

[13] Yang, X., and Tang, Y. Y., 2007. "Efficient fault identification of diagnosable systems under the comparison model". *IEEE Transactions on computers,* **56**(12), pp. 1612–1618.

[14] Hsieh, S.-Y., and Chen, Y.-S., 2008. "Strongly diagnosable product networks under the comparison diagnosis model". *IEEE Transactions on Computers,* **57**(6), pp. 721–732.

[15] Chang, G.-Y., Chen, G.-H., and Chang, G. J., 2006. "(t, k)-diagnosis for matching composition networks under the mm* model". *IEEE Transactions on Computers,* **56**(1), pp. 73–79.

[16] Tsai, C.-H., 2011. "A quick pessimistic diagnosis algorithm for hypercube-like multiprocessor systems under the pmc model". *IEEE Transactions on Computers,* **62**(2), pp. 259–267.

[17] Chang, G.-Y., 2011. "Conditional ({t}, k)-diagnosis under the pmc model". *IEEE transactions on parallel and distributed systems,* **22**(11), pp. 1797–1803.

[18] Duarte, E. P., Weber, A., and Fonseca, K. V., 2011. "Distributed diagnosis of dynamic events in partitionable arbitrary topology networks". *IEEE Transactions on Parallel and Distributed Systems,* **23**(8), pp. 1415–1426.

[19] Chessa, S., and Santi, P., 2001. "Comparison-based system-level fault diagnosis in ad hoc networks". In Proceedings 20th IEEE Symposium on Reliable Distributed Systems, IEEE, pp. 257–266.

[20] Lee, M.-H., and Choi, Y.-H., 2008. "Fault detection of wireless sensor networks". *Computer Communications,* **31**(14), pp. 3469–3475.

[21] Krishnamachari, B., and Iyengar, S., 2004. "Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks". *IEEE Transactions on Computers*(3), pp. 241–250.

[22] Swain, R. R., and Khilar, P. M., 2017. "Composite fault diagnosis in wireless sensor networks using neural networks". *Wireless Personal Communications,* **95**(3), pp. 2507–2548.

[23] Heinzelman, W. R., Chandrakasan, A., and Balakrishnan, H., 2000. "Energy-efficient communication protocol for wireless microsensor networks". In Proceedings of the 33rd annual Hawaii international conference on system sciences, IEEE, pp. 10–pp.

[24] Issariyakul, T., and Hossain, E., 2009. "Introduction to network simulator 2 (ns2)". In *Introduction to network simulator NS2*. Springer, pp. 1–18.

[25] Mohapatra, S., and Khilar, P. M., 2017. "Artificial immune system based fault diagnosis in large wireless sensor network topology". In TENCON 2017-2017 IEEE Region 10 Conference, IEEE, pp. 2687–2692.

[26] Mourad, E., and Nayak, A., 2011. "Comparison-based system-level fault diagnosis: A neural network approach". *IEEE Transactions on Parallel and Distributed Systems,* **23**(6), pp. 1047–1059.

[27] Panda, R. R., Gouda, B. S., and Panigrahi, T., 2014. "Efficient fault node detection algorithm for wireless sensor networks". In 2014 International Conference on High Performance Computing and Applications (ICHPCA), IEEE, pp. 1–5.

[28] Maronna, R., Martin, D., and Yohai, V., 2006. Robust statistics (pp. 978-0).

[29] Jin, X., Chow, T. W., Sun, Y., Shan, J., and Lau, B. C., 2015. "Kuiper test and autoregressive model-based approach for wireless sensor network fault diagnosis". *Wireless Networks,* **21**(3), pp. 829–839.

[30] Anděl, J., 1976. "Autoregressive series with random parameters". *Mathematische Operationsforschung und Statistik,* **7**(5), pp. 735–741.

[31] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., 1990. "Numerical recipes in pascal". *Cambridge University, Cambridge, UK*.

[32] Kar, C., and Mohanty, A., 2004. "Application of ks test in ball bearing fault diagnosis". *Journal of sound and vibration,* **1**(269), pp. 439–454.

[33] Guo, S., Zhong, Z., and He, T., 2009. "Find: faulty node detection for wireless sensor networks". In Proceedings of the 7th ACM conference on embedded networked sensor systems, ACM, pp. 253–266.

[34] Guo, S., Zhang, H., Zhong, Z., Chen, J., Cao, Q., and He, T., 2014. "Detecting faulty nodes with data errors for wireless sensor networks". *ACM Transactions on Sensor Networks (TOSN),* **10**(3), p. 40.

[35] Kamal, A. R. M., Bleakley, C. J., and Dobson, S., 2014. "Failure detection in wireless sensor networks: A sequence-based dynamic approach". *ACM Transactions on Sensor Networks (TOSN),* **10**(2), p. 35.

[36] Lau, B. C., Ma, E. W., and Chow, T. W., 2014. "Probabilistic fault detector for wireless sensor network". *Expert Systems with Applications,* **41**(8), pp. 3703–3711.

[37] Tang, P., and Chow, T. W., 2016. "Wireless sensor-networks conditions monitoring and fault diagnosis using neighborhood hidden conditional random field". *IEEE Transactions on Industrial Informatics,* **12**(3), pp. 933–940.

[38] Dhal, R., Torres, J. A., and Roy, S., 2015. "Detecting link failures in complex network processes using remote monitoring". *Physica A: Statistical Mechanics and its Applications,* **437**, pp. 36–54.

[39] Abid, A., Kachouri, A., Guiloufi, A. B. F., Mahfoudhi, A., Nasri, N., and Abid, M., 2015. "Centralized knn anomaly detector for wsn". In 2015 IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD15), IEEE, pp. 1–4.

[40] Yang, Y., Su, L., Khan, M., Lemay, M., Abdelzaher, T., and Han, J., 2015. "Power-based diagnosis of node silence in remote high-end sensing systems". *ACM Transactions on Sensor Networks (TOSN),* **11**(2), p. 33.

[41] Rabiner, L. R., and Juang, B.-H., 1986. "An introduction to hidden markov models". *ieee assp magazine,* **3**(1), pp. 4–16.

[42] Warriach, E. U., and Tei, K., 2013. "Fault detection in wireless sensor networks: A machine learning approach". In 2013 IEEE 16th International Conference on Computational Science and Engineering, IEEE, pp. 758–765.

[43] Chen, J., Kher, S., and 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, ACM, pp. 65–72.

[44] Xu, X., Geng, W., Yang, G., Bessis, N., and Norrington, P., 2014. "Ledfd: A low energy consumption distributed fault detection algorithm for wireless sensor networks". *International Journal of Distributed Sensor Networks,* **10**(2), p. 714530.

[45] Saihi, M., Boussaid, B., Zouinkhi, A., and Abdelkrim, M. N., 2013. "Decentralized fault detection in wireless sensor network based on function error". In 10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13), IEEE, pp. 1–5.

[46] Panda, M., and Khilar, P. M., 2012. "Distributed soft fault detection algorithm in wireless sensor networks using statistical test". In 2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, IEEE, pp. 195–198.

[47] Panda, M., and Khilar, P. M., 2015. "Distributed byzantine fault detection technique in wireless sensor networks based on hypothesis testing". *Computers & Electrical Engineering,* **48**, pp. 270–285.

[48] Yuan, H., Zhao, X., and Yu, L., 2015. "A distributed bayesian algorithm for data fault detection in wireless sensor networks". In 2015 international conference on information networking (ICOIN), IEEE, pp. 63–68.

[49] Zhao, M., Tian, Z., and Chow, T. W., 2019. "Fault diagnosis on wireless sensor network using the neighborhood kernel density estimation". *Neural Computing and Applications,* **31**(8), pp. 4019–4030.

[50] Sharma, K. P., and Sharma, T. P., 2017. "rdfd: Reactive distributed fault detection in wireless sensor networks". *Wireless Networks,* **23**(4), pp. 1145–1160.

[51] Sahoo, M. N., and Khilar, P. M., 2014. "Distributed diagnosis of permanent and intermittent faults in wireless sensor networks". In *Advanced Computing, Networking and Informatics-Volume 2*. Springer, pp. 133–141.

[52] Sahoo, M. N., and Khilar, P. M., 2014. "Diagnosis of wireless sensor networks in presence of permanent and intermittent faults". *Wireless personal communications,* **78**(2), pp. 1571–1591.

[53] Chanak, P., and Banerjee, I., 2016. "Fuzzy rule-based faulty node classification and management scheme for large scale wireless sensor networks". *Expert Systems with Applications,* **45**, pp. 307–321.

[54] Ghorbel, O., Jmal, M. W., Abid, M., and Snoussi, H., 2015. "Distributed and efficient one-class outliers detection classifier in wireless sensors networks". In International Conference on Wired/Wireless Internet Communication, Springer, pp. 259–273.

[55] Obst, O., 2014. "Distributed fault detection in sensor networks using a recurrent neural network". *Neural processing letters,* **40**(3), pp. 261–273.

[56] Obst, O., 2009. "Distributed fault detection using a recurrent neural network". In Proceedings of the 2009 International Conference on Information Processing in Sensor Networks, IEEE Computer Society, pp. 373–374.

[57] Mahapatro, A., and Panda, A. K., 2014. "Choice of detection parameters on fault detection in wireless sensor networks: A multiobjective optimization approach". *Wireless personal communications,* **78**(1), pp. 649–669.

[58] Wang, N., and Chen, Y.-X., 2013. "A fault-event detection model using trust matrix in wsn". *Sensors & Transducers,* **158**(11), p. 190.

[59] Afsar, M. M., 2014. "Maximizing the reliability of clustered sensor networks by a fault-tolerant service". In 2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE), IEEE, pp. 1–8.

[60] Zafar, A., Wajid, B., and Akram, B. A., 2015. "A hybrid fault diagnosis architecture for wireless sensor networks". In 2015 International Conference on Open Source Systems & Technologies (ICOSST), IEEE, pp. 7–15.

[61] Nitesh, K., and Jana, P. K., 2015. "Dfda: a distributed fault detection algorithm in two tier wireless sensor networks". In Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014, Springer, pp. 739–746.

[62] Titouna, C., Aliouat, M., and Gueroui, M., 2015. "Outlier detection approach using bayes classifiers in wireless sensor networks". *Wireless Personal Communications,* **85**(3), pp. 1009–1023.

[63] Titouna, C., Aliouat, M., and Gueroui, M., 2016. "Fds: fault detection scheme for wireless sensor networks". *Wireless Personal Communications,* **86**(2), pp. 549–562.

[64] Wu, J.-Y., Duh, D.-R., Wang, T.-Y., and Chang, L.-Y., 2007. "Fast and simple on-line sensor fault detection scheme for wireless sensor networks". In International Conference on Embedded and Ubiquitous Computing, Springer, pp. 444–455.

[65] Kaur, A., and Sharma, T. P., 2010. "Afdep: agreement based ch failure detection and election protocol for a wsn". In International Conference on Advances in Information and Communication Technologies, Springer, pp. 249–257.

[66] Nguyen, T. A., Bucur, D., Aiello, M., and Tei, K., 2013. "Applying time series analysis and neighbourhood voting in a decentralised approach for fault detection and classification in wsns". In Proceedings of the Fourth Symposium on Information and Communication Technology, ACM, pp. 234–241.

[67] Chanak, P., Banerjee, I., and Sherratt, R. S., 2016. "Mobile sink based fault diagnosis scheme for wireless sensor networks". *Journal of Systems and Software,* **119**, pp. 45–57.

[68] Abo-Zahhad, M., Ahmed, S. M., Sabor, N., and Sasaki, S., 2015. "Mobile sink-based adaptive immune energy-efficient clustering protocol for improving the lifetime and stability period of wireless sensor networks". *IEEE Sensors Journal,* **15**(8), pp. 4576–4586.

[69] Castro, L. N., De Castro, L. N., and Timmis, J., 2002. *Artificial immune systems: a new computational intelligence approach*. Springer Science & Business Media.

[70] Janeway, C. A., 2001. "The immune system in health and disease". *http://www. garlandscience. com*.

[71] Rizwan, R., Khan, F. A., Abbas, H., and Chauhdary, S. H., 2015. "Anomaly detection in wireless sensor networks using immune-based bioinspired mechanism". *International journal of distributed sensor networks,* **11**(10), p. 684952.

[72] de Castro, L. N., and Timmis, J., 2002. "Artificial immune systems: a novel approach to pattern recognition".

[73] Dasgupta, D., and González, F., 2002. "An immunity-based technique to characterize intrusions in computer networks". *IEEE Transactions on evolutionary computation,* **6**(3), pp. 281–291.

[74] Dasgupta, D., KrishnaKumar, K., Wong, D., and Berry, M., 2004. "Negative selection algorithm for aircraft fault detection". In International Conference on Artificial Immune Systems, Springer, pp. 1–13.

[75] Burnet, S. F. M., et al., 1959. "The clonal selection theory of acquired immunity".

[76] Mazhar, N., and Farooq, M., 2008. "A sense of danger: dendritic cells inspired artificial immune system for manet security". In Proceedings of the 10th annual conference on Genetic and evolutionary computation, ACM, pp. 63–70.

[77] Matzinger, P., 1994. "Tolerance, danger, and the extended family". *Annual review of immunology,* **12**(1), pp. 991–1045.

[78] Jerne, N. K., 1974. "Towards a network theory of the immune system". *Ann. Immunol.,* **125**, pp. 373–389.

[79] Jegadeeshwaran, R., and Sugumaran, V., 2015. "Brake fault diagnosis using clonal selection classification algorithm (csca)–a statistical learning approach". *Engineering Science and Technology, an International Journal,* **18**(1), pp. 14–23.

[80] Gan, Z., Zhao, M.-B., and Chow, T. W., 2009. "Induction machine fault detection using clone selection programming". *Expert Systems with Applications,* **36**(4), pp. 8000–8012.

[81] Mohapatra, S., Khilar, P. M., and Swain, R. R., 2019. "Fault diagnosis in wireless sensor network using clonal selection principle and probabilistic neural network approach". *International Journal of Communication Systems*, p. e4138.

[82] Chen, G., Zhang, L., and Bao, J., 2013. "An improved negative selection algorithm and its application in the fault diagnosis of vibrating screen by wireless sensor networks". *Journal of Computational and Theoretical Nanoscience,* **10**(10), pp. 2418–2426.

[83] Gao, X. Z., Wang, X., and Zenger, K., 2014. "Motor fault diagnosis using negative selection algorithm". *Neural Computing and Applications,* **25**(1), pp. 55–65.

[84] Laurentys, C., Ronacher, G., Palhares, R. M., and Caminhas, W. M., 2010. "Design of an artificial immune system for fault detection: A negative selection approach". *Expert Systems with Applications,* **37**(7), pp. 5507–5513.

[85] Li, D., Liu, S., and Zhang, H., 2015. "Negative selection algorithm with constant detectors for anomaly detection". *Applied Soft Computing,* **36**, pp. 618–632.

[86] Zeeshan, M., Javed, H., Haider, A., and Khan, A., 2015. "An immunology inspired flow control attack detection using negative selection with r-contiguous bit matching for wireless sensor networks". *International Journal of Distributed Sensor Networks,* **11**(11), p. 169654.

[87] Taylor, D. W., and Corne, D. W., 2003. "An investigation of the negative selection algorithm for fault detection in refrigeration systems". In International Conference on Artificial Immune Systems, Springer, pp. 34–45.

[88] Alizadeh, E., Meskin, N., and Khorasani, K., 2016. "A negative selection immune system inspired methodology for fault diagnosis of wind turbines". *IEEE transactions on cybernetics,* **47**(11), pp. 3799–3813.

[89] de Abreu, C. C. E., Duarte, M. A. Q., and Villarreal, F., 2017. "An immunological approach based on the negative selection algorithm for real noise classification in speech signals". *AEU-International Journal of Electronics and Communications,* **72**, pp. 125–133.

[90] Aydin, I., Karakose, M., and Akin, E., 2010. "Chaotic-based hybrid negative selection algorithm and its applications in fault and anomaly detection". *Expert Systems with Applications,* **37**(7), pp. 5285–5294.

[91] Alizadeh, E., Meskin, N., and Khorasani, K., 2017. "A dendritic cell immune system inspired scheme for sensor fault detection and isolation of wind turbines". *IEEE Transactions on Industrial Informatics,* **14**(2), pp. 545–555.

[92] Xiao, X., and Zhang, R., 2017. "Study of immune-based intrusion detection technology in wireless sensor networks". *Arabian Journal for Science and Engineering,* **42**(8), pp. 3159–3174.

[93] Jiang, W. K., Chen, Y. J., and Zhang, J., 2013. "A fault diagnosis method based on artificial immune network". In Applied Mechanics and Materials, Vol. 385, Trans Tech Publ, pp. 658–662.

[94] Wang, F. Z., Shao, S. M., and Dong, P. F., 2014. "Research on transformer fault diagnosis method based on artificial immune network and fuzzy c-means clustering algorithm". In Applied Mechanics and Materials, Vol. 574, Trans Tech Publ, pp. 468–473.

[95] Ishiguro, A., Watanabe, Y., and Uchikawa, Y., 1994. "Fault diagnosis of plant systems using immune networks". In Proceedings of 1994 IEEE International Conference on MFI'94. Multisensor Fusion and Integration for Intelligent Systems, IEEE, pp. 34–42.

[96] Hao, X., and Cai-Xin, S., 2007. "Artificial immune network classification algorithm for fault diagnosis of power transformer". *IEEE Transactions on Power Delivery,* **22**(2), pp. 930–935.

[97] Dasgupta, D., Yu, S., and Nino, F., 2011. "Recent advances in artificial immune systems: models and applications". *Applied Soft Computing,* **11**(2), pp. 1574–1587.

[98] Bayar, N., Darmoul, S., Hajri-Gabouj, S., and Pierreval, H., 2015. "Fault detection, diagnosis and recovery using artificial immune systems: A review". *Engineering Applications of Artificial Intelligence,* **46**, pp. 43–57.

[99] Jabbari, A., Jedermann, R., and Lang, W., 2007. "Application of computational intelligence for sensor fault detection and isolation". *World academy of science, engineering and technology,* **33**, pp. 265–270.

[100] Swain, R. R., Khilar, P. M., and Bhoi, S. K., 2018. "Heterogeneous fault diagnosis for wireless sensor networks". *Ad Hoc Networks,* **69**, pp. 15–37.

[101] Specht, D. F., 1990. "Probabilistic neural networks". *Neural networks,* **3**(1), pp. 109–118.

[102] Swain, R. R., Dash, T., and 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), p. e3273.

[103] Forrest, S., Perelson, A. S., Allen, L., and Cherukuri, R., 1994. "Self-nonself discrimination in a computer". In Proceedings of 1994 IEEE computer society symposium on research in security and privacy, Ieee, pp. 202–212.

[104] González, F., Dasgupta, D., and Gómez, J., 2003. "The effect of binary matching rules in negative selection". In Genetic and Evolutionary Computation Conference, Springer, pp. 195–206.

[105] Deng, F., Guo, S., Zhou, R., and Chen, J., 2015. "Sensor multifault diagnosis with improved support vector machines". *IEEE Transactions on Automation Science and Engineering,* **14**(2), pp. 1053–1063.

[106] Nilsson, N. J., 1965. "Learning machines.".

[107] Hastie, T., and Tibshirani, R., 1998. "Classification by pairwise coupling". In Advances in neural information processing systems, pp. 507–513.

[108] Allwein, E. L., Schapire, R. E., and Singer, Y., 2000. "Reducing multiclass to binary: A unifying approach for margin classifiers". *Journal of machine learning research,* **1**(Dec), pp. 113–141.

[109] Yin, A.-r., Xie, X., and Kuang, J.-m., 2008. "Application of hadamard ecoc in multi-class problems based on svm". *Acta Electronica Sinica,* **36**(1), p. 122.

[110] Greensmith, J., 2007. "The dendritic cell algorithm". PhD thesis, Citeseer.

[111] Alizadeh, E., Meskin, N., and Khorasani, K., 2016. "A sensor fault detection and isolation strategy by using a dendritic cell algorithm". In 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, pp. 001171–001177.

[112] Chelly, Z., and Elouedi, Z., 2016. "A survey of the dendritic cell algorithm". *Knowledge and Information Systems,* **48**(3), pp. 505–535.

[113] Bachmayer, S., 2008. "Artificial immune systems". *Artificial immune systems,* **5132**, pp. 119–131.

[114] De Castro, L. N., and Von Zuben, F. J., 1999. "Artificial immune systems: Part i–basic theory and applications". *Universidade Estadual de Campinas, Dezembro de, Tech. Rep,* **210**(1).

[115] De Castro, L. N., and Von Zuben, F. J., 2002. "Learning and optimization using the clonal selection principle". *IEEE transactions on evolutionary computation,* **6**(3), pp. 239–251.

[116] Deng, Z., Zhu, X., Cheng, D., Zong, M., and Zhang, S., 2016. "Efficient knn classification algorithm for big data". *Neurocomputing,* **195**, pp. 143–148.

[0]This reference format follows ASME style. You are advised to follow one reference format of any dominant journal of your field.

Dissemination

SCI indexed journals [1]

1. Mohapatra, Santoshinee, Pabitra M. Khilar, and Rakesh R. Swain. ”Fault diagnosis in wireless sensor network using clonal selection principle and probabilistic neural network approach.” International Journal of Communication Systems 32, no. 16 (2019): e4138.

2. Mohapatra, Santoshinee, and Pabitra Mohan Khilar. ”Fault diagnosis in wireless sensor network using negative selection algorithm and support vector machine.” Computational Intelligence 36.3 (2020): 1374-1393.

Book chapters [1]

1. Mohapatra, Santoshinee, and Pabitra Mohan Khilar. ”Immune Inspired Fault Diagnosis in Wireless Sensor Network.” In Nature Inspired Computing for Wireless Sensor Networks, pp. 103-116. Springer, Singapore, 2020.

Conferences [1]

1. Mohapatra, Santoshinee, and Pabitra Mohan Khilar. ”Artificial immune system based fault diagnosis in large wireless sensor network topology.” In TENCON 2017-2017 IEEE Region 10 Conference, pp. 2687-2692. IEEE, 2017.

2. Mohapatra, Santoshinee, and Pabitra Mohan Khilar. ”Forest fire monitoring and detection of faulty nodes using wireless sensor network.” In 2016 IEEE Region 10 Conference (TENCON), pp. 3232-3236. IEEE, 2016.

3. Mohapatra, Santoshinee, and Pabitra Mohan Khilar. ”Fault Diagnosis in Wireless Sensor Network Using Self/Non-self Discrimination Principle.” In Progress in Computing, Analytics and Networking, pp. 161-168. Springer, Singapore, 2020.

[1] Articles already published, in press, or formally accepted for publication.

4. Senapati, B. R., Mohapatra, Santoshinee, and Pabitra Mohan Khilar. ”Fault Detection for VANET using Vehicular Cloud.” International Conference on Intelligent and Cloud Computing (ICICC-2019). Springer, 2019.

Article under preparation [2]

1. Mohapatra, Santoshinee, Pabitra M. Khilar. ”Fault Diagnosis in Wireless Sensor Network using Dendritic Cell Algorithm.” Engineering Applications of Artificial Intelligence.

2. Mohapatra, Santoshinee, Pabitra M. Khilar. ”Artificial Immune Network based Fault Diagnosis in Wireless Sensor Network.” Journal of Experimental & Theoretical Artificial Intelligence.

[2] Articles under review, communicated, or to be communicated.

BIO-DATA

Santoshinee Mohapatra

Date of Birth: 19^{th} June, 1990

Correspondence

Department of Computer Science and Engineering
National Institute of Technology, Rourkela – 769008, India.
Ph: +91 8093355662 (M)
e-mail: santoshinee88@gmail.com, 514CS1012@nitrkl.ac.in

Qualification

- Ph.D. (Computer Science and Engineering)
  National Institute of Technology (NIT), Rourkela, Odisha, India
- M.Tech. (Computer Science and Engineering)
  Veer Surendra Sai University of Technology (VSSUT), Burla, Odisha, India
- B.Tech. (Computer Science and Engineering)
  Ghanashyam Hemalata Institute of Technology and Management, Puri
  Biju Patnaik University of Technology (BPUT), Odisha, India
- 12^{th} (Science)
  Ramadevi Women's Junior College, Bhubaneswar
  Council of Higher Secondary Education (CHSE), Odisha, India
- 10^{th}
  Saraswati Sishu/Vidya Mandir, Bhubaneswar
  Board of Secondary Education (BSE), Odisha, India

Publications

- 02 Journal Articles
- 04 Conference Articles
- 01 Book Chapter

Permanent Address

Santoshinee Mohapatra
GA-585, Sailashree Vihar, Chandrasekhar Pur
Bhubaneswar-751021, Odisha, India