

Article

Analysis of Fault Classifiers to Detect the Faults and Node Failures in a Wireless Sensor Network

S. Gnanavel ^{1,*}, M. Sreekrishna ², Vinodhini Mani ^{2,*}, G. Kumaran ³, R. S. Amshavalli ⁴, Sadeen Alharbi ⁵, Mashaal Maashi ⁵, Osamah Ibrahim Khalaf ⁶, Ghaida Muttashar Abdulsahib ⁷, Ans D. Alghamdi ⁸ and Theyazn H. H. Aldhyani ⁹

- ¹ Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology, Kattankulathur 603203, India
- ² Department of Computer Science and Engineering, School of Computing, Sathyabama Institute of Science and Technology, Chennai 600119, India; sreekrishna.cse@sathyabama.ac.in
- ³ Saveetha School of Engineering, Chennai 602105, India; kumarang.sse@saveetha.com
- ⁴ Sathyabama Institute of Science and Technology, Chennai 600119, India; amshavalli.cse@sathyabama.ac.in
- ⁵ Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia; sadalharbi@ksu.edu.sa (S.A.); mmaashi@ksu.edu.sa (M.M.)
- ⁶ Al-Nahrain Nano Renewable Research Center, Al-Nahrain University, Baghdad 10072, Iraq; usama81818@nahrainuniv.edu.iq
- ⁷ Department of Computer Engineering, University of Technology, Baghdad 10066, Iraq; 30834@uotechnology.edu.iq
- ⁸ Computer Engineering and Science Department, College of Computer Science and Information Technology, Al Baha University, Al-Baha 42331, Saudi Arabia; ans@bu.edu.sa
- ⁹ Applied College in Abqaiq, King Faisal University, P.O. Box 400, Al-Ahsa 31982, Saudi Arabia; taldhyani@kfu.edu.sa
- * Correspondence: gnanaves1@srmist.edu.in (S.G.); vinodhini.cse@sathyabama.ac.in (V.M.)



Citation: Gnanavel, S.; Sreekrishna, M.; Mani, V.; Kumaran, G.; Amshavalli, R.S.; Alharbi, S.; Maashi, M.; Khalaf, O.I.; Abdulsahib, G.M.; Alghamdi, A.D.; et al. Analysis of Fault Classifiers to Detect the Faults and Node Failures in a Wireless Sensor Network. *Electronics* **2022**, *11*, 1609. <https://doi.org/10.3390/electronics11101609>

Academic Editors: Celestine Iwendu, M. Poongodi, SenthilKumar Mohan and Mohit Mittal

Received: 13 April 2022

Accepted: 12 May 2022

Published: 18 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: Technology evaluation in the electronics field leads to the great development of Wireless Sensor Networks (WSN) for a variety of applications. The sensor nodes are deployed in hazardous environments, and they are operated by isolated battery sources. Network connectivity is purely based on power availability, which impacts the network lifetime. Hence, power must be used wisely to prolong the network lifetime. The sensor nodes that fail due to power have to detect quickly to maintain the network. In a WSN, classifiers are used to detect the faults for checking the data generated by the sensor nodes. In this paper, six classifiers such as Support Vector Machine, Convolutional Neural Network, Multilayer Perceptron, Stochastic Gradient Descent, Random Forest and Probabilistic Neural Network have been taken for analysis. Six different faults (Offset fault, Gain fault, Stuck-at fault, Out of Bounds, Spike fault and Data loss) are injected in the data generated by the sensor nodes. The faulty data are checked by the classifiers. The simulation results show that the Random Forest detected more faults and it also outperformed all other classifiers in that category.

Keywords: support vector machine (SVM); convolutional neural network (CNN); multilayer perceptron (MLP); stochastic gradient descent (SGD); random forest (RF) and probabilistic neural network (PNN)

1. Introduction

Wireless Sensor Network [1] is a fascinating technology to monitor and control a remote environment through wireless sensing of various parameters. The size of WSN is reduced with the help of VLSI, which leads to considerable improvements in the energy efficiency. Though the WSN is small in size, it incorporates three major functional units to complete its task. The core functional units are (1) the Environmental Sensing Unit; (2) the Data Processing Unit; and (3) The Communication Unit. In automation applications, the transducer plays an important role to read the physical parameters by converting them

into an electrical signal for further processing. In WSN, the sensing unit performs the functionality of a transducer with the help of tiny and powerful sensors.

The WSNs nodes are deployed in environments such as forests, rivers, and mountains to detect or sense various atmospheric parameters such as temperature, humidity, fire, and pressure. The information collected by the sensor nodes is converted as an electrical signal from signals such as sound, light, etc. by the transducers. Then, the electrical analog signal has to be converted into a digital signal as binary information. The sensed and converted binary information is passed to the next stage, where all the calculations and processing are carried out in WSNs. This processing unit has limited memory and hence, it will forward the binary information to the next sensor node, or it will process itself based on the requirement.

The processing unit has the control of a sensing module with limited capacities. The important and essential part of the sensor unit is the data communication module, which contains the transceiver circuit that will perform both transmitter and receiver functions for the data exchange. It is the responsibility of the communication module to ensure all the information passes through the routing protocols [2] and data communications. The components of the sensing module, processing module and communication module are connected in a Printed Circuit Board (PCB), which is made as small as possible for power constraints. This connected module is referred to as a sensor node. Figure 1 illustrates an overview of the sensor node architecture.

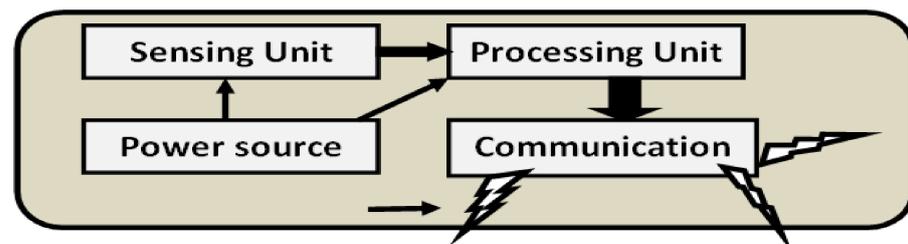


Figure 1. Sensor node architecture.

The sensor nodes sense parameters, such as pressure or temperature, at regular intervals. The WSNs are used in a hazardous environment, as they work in unmonitored fashions, and they are also deployed in extreme environmental conditions. This scenario leads the WSNs to be highly vulnerable to failures.

The failures in WSNs are as follows,

- Software failures, which lead to malfunction of sensor nodes;
- Hardware failures occur, due to a sensor fault or a signal conditioning circuit fault;
- Communication failures are due to a fault on wireless antenna circuits or wireless protocols.

Based on the sensed data, the faults are classified [3] into different categories as mentioned below.

Offset fault: Sensing circuits with sensors may have errors in them. It is found through the calibration process. This error is rectified by adding/subtracting with a displacement value in the real sensed data.

Gain fault: The sensed data are processed by a circuit in the sensor node to change the sensed data to the required level. The processing circuit may deviate from the actual processing level.

Stuck-at fault: In a digital circuit, the data are at a point to be changed based on input. However, the circuit is stuck with either zero or one permanently without changing.

Out of bounds: The range of sensed data varies beyond the range of the sensing circuit.

Spike fault: The rate of data sensing and processing of the sensor nodes varies abruptly with the actual data rate of sensing and processing.

Noise fault: Random numbers are added with the sensed data.

Data loss fault: Due to the sensor fault or communication fault, few data are missed with the actual system data rate.

Random fault: Fault occurs instantly without any other specific faults in the system

Besides the mentioned faults, faults can occur instantly and simultaneously from the sensor and communication units.

Monitoring the sensor node's functionality is important to achieve the expected goal of the designed application system. The fault-finding system [4,5] has to be effective and fast enough to get the required data without any error and delay. Vast amounts of research have been going on at different possibilities with constrained sensor resources. This research is based on multiple mechanisms available in the field of WSN development.

In fault-finding mechanisms, three approaches are available. They are the Centralized approach, Distributed approach, and Hybrid approach [6,7]. These approaches are utilized due to the availability of computing algorithms such as machine learning and deep learning methodologies. Machine learning and deep learning are computerized programs with various possibilities of past behavior [8]. Artificial intelligence plays an important role in machine learning to estimate the outcome based on the past behavior. Data mining is the latest technological term extracted from machine learning and deep learning. It dismantles the data into multiple subsets and plays key support in the process of decision-making as opined by [9]. Machine learning is classified into the following categories based on the available information on given data.

- Supervised learning: This approach performs a data mining process with a predefined set of labeled classes;
- Unsupervised learning: This method classifies data into unlabeled datasets without prior knowledge;
- Semi-supervised learning: To achieve the best result, the advantages of supervisor learning and unsupervised learning are combined as a hybrid approach;
- The fault-finding mechanism (Sasmita Acharya and Tripathy, 2017) is performed with the following classifiers:
- Support Vector Machine (SVM).
- Convolutional Neural Network (CNN).
- Multilayer Perceptron (MLP).
- Stochastic Gradient Descent (SGD).
- Random Forest (RF).
- Probabilistic Neural Network (PNN).

In the work of the SVM technique, Spatially Organized Distributed Echo State Network (SODESN) and Hidden Markov Models (HMM) are used to find the fault and also for data classification. Instead of the above techniques, have used algorithms to detect the faulty nodes and helped to replace them in time to reduce the risk of network failure in WSN. Fault identification mechanisms are used to detect the fault in sensor nodes and replace them to maintain the connectivity in time. The data sensed and transmitted by these faulty nodes are considered for the evaluation purposes of further fault detection.

2. Works on Fault Detection

Nosha et al. have analyzed various random classifiers and tested the result of the classifiers using python. In this approach, the following process is performed as per the order of flow. The data are collected from the sensor nodes, faults are injected randomly on collected data and the faults are detected by all the classifiers used in this approach. The RF classifier outperforms most of the faults in WSN [10].

Yuan et al. (2018) have used three techniques to find faults and also to categorize them into different categories. The results of naive Bayes, SVM, and Gradient Lifting Decision Tree (GBDT) are compared and better results are obtained. This work has identified the fault due to low energy on sensor nodes' battery and also detected the short-term fault, noise fault and fixed fault [10].

Zidi et al. have implemented the SVM technique and this technique has divided the sensed datasets into two categories: 1 and -1 . The datasets are classified into 1 as error-free measurement and -1 as the faulty measurement. This is implemented in the cluster head to classify and for decision making [11].

Cheng et al. have deployed SVR mechanisms for fault identification with neighbor coordination. The conditions considered in this technique are lesser node densities and high node failure rates. This technique comes under the prediction model at the sensor node level and also mutual testing by every node in the network [12].

The detected faults have been analyzed in WSN by Muhammed et al. The faults are divided as (1) transient faults and (2) persistent faults. The faults that occur at sensor nodes due to node failure are permanently classified as Persistent faults. The faults are classified into transient faults due to the sensor node's misbehavior temporarily.

An algorithm based on the distribution method for effective handling of streaming of sensed data in WSNs. It is also used for anomalies detection throughout the networks and to find cost functions. They have achieved the lowest misdetection rate and highest fault detection rate [13].

Swain et al. (2018) have analyzed that the link failures are the major reason for the network connectivity issues in WSN. This is handled by them by proposing an analysis technique, namely Feed Forward Neural Network (FFNN), for comprehensive analysis. This FFNN measures the connection link quality to estimate the network lifetime. This technique finds the quality of links by measuring the Packet Drop Ratio (PDR) values in WSN [14,15].

Zhang et al. have effectively used machine learning in WSN to identify faults and for data analysis. The WSNs are more vulnerable to faults and failures due to their nature. The authors have used a data mining technique for fault finding and diagnosis [16].

Zhang et al. have identified that the functional cost of WSNs depends on sensor node failures. Hence, it has to be identified and rectified as much as earlier to reduce the operational cost. For fault finding, the RF has been used by the authors. The authors have proposed the technique of data-driven architecture and it has provided better results than SVM [17].

Two techniques for node failure identification and to separate the failure nodes. The technique used for fault finding is Trend Analysis incorporated with least Squares Support Vector Machine (TA-SSVM). To segregate the failed nodes from the WSN, the Error-Correcting Output Coding-Support Vector Machine (ECOC-SVM) technique is used [18].

Araya et al. have proposed a new classifier with a pattern-based approach. The energy consumption pattern in WSN is identified by the Collective Contextual Anomaly Detection using Sliding Window (CCAD-SW) architecture to differentiate the failure sensor nodes from WSN. The authors have also used another technique, namely Ensemble Anomaly Detection (EAD), to classify the sensor nodes [19].

Packet losses have been identified by the distributed filtering mechanism with a nonlinear system in WSN by Gao This technique has also been used for classification. The sensor nodes that failed in the WSN are identified and segregated by Bernoulli distribution. In addition to this, the Lyapunov functional technique is used for performance analysis on fault-finding stability.

Heterogeneous faults are also present in the WSN, and they have to be identified and rectified to increase the performance of WSN. It has three stages of operations which are used to find out hard permanent, soft permanent, intermittent and transient faults. Energy consumption is a major problem in WSN. The clustering approach has decreased this energy expenditure. Swain has used a time-out stats signal mechanism to find the fault nodes. Analysis of Variance (ANOVA) is used to check the sensor nodes that have equal values on parameters.

3. Challenges and Problem Statement

The Challenges and Difficulties of WSNs in fault finding are focused with the following points:

- The WSNs have very minimal resources at every sensor node. The classifiers are used to identify the faults, as they are computed by a simple calculation.
- The location of sensor nodes is hazardous and dangerous environments.
- The fault-finding method mentioned is prompt and quick to reject any loss. This fault-finding method must recognize the faulty data by concerning the correct data, and then the sensor nodes are replaced.

3.1. Faults in WSNs

As discussed previously, the faults are the data-centric approach. The calibration error and battery failure are system-centric approaches in WSN. Since WSNs are used in hazardous environments, e.g., flood level of a river, forest fire detection, rain, etc., the probability of failures in sensor nodes is high due to the nature of the environments. The faults of sensor nodes in the normal application do not have any severe impact. In tough situations such as calamities, forest fires, tsunami, etc., loss of human lives, environmental conditions and economic losses happen frequently due to node failures. Faults have to be prevented from these difficult situations to protect from losses. From the fault-finding mechanisms, faults are classified into different categories through various algorithms. The faults are categorized quickly. The corresponding actions have been taken to solve the problems by providing appropriate solutions [14]. A brief representation of fault classification is noted below with data collected from the sensor nodes.

The collected data are represented by the triplet function $d(n,t,f(t))$;

$$f(t) = \alpha + \beta x + \eta \quad (1)$$

where $f(t)$ is the sensed data at time t of node n .

α —Offset value of sensed dataset,

β —Gain value,

η —The noise affected in the sensed data.

3.1.1. Gain Fault

It represents the rate of variation of data and it is expected more than the actual data. This fault occurs due to the incremental nature of β at the data point x .

The equation for this fault is given below:

$$x^J = \beta x + \eta \quad (2)$$

3.1.2. Offset Fault

Sensing circuits with sensors may have errors in them. They are found by the calibration process. This error is rectified by addition/subtraction with a displacement value in the real sensed data. The offset fault is represented as below:

$$x^J = \alpha + x + \eta \quad (3)$$

where α —the offset value to be added in the actual reading.

3.1.3. Stuck-at Fault

In a digital circuit, the data at a point to be changed are based on input. However, the circuit is stuck with either zero or one permanently without changing.

The Stuck-at fault is derived in Equation (4) as:

$$x^J = \alpha \quad (4)$$

where x^J is the data collected by the sensor node at time instant t .

3.1.4. Spike Fault

The rate of data sensing and processing of the sensor nodes varies abruptly with the actual data rate of sensing and processing.

The Spike fault is mentioned in Equation (5).

$$\frac{|f(t) - f_p(t)|}{t} > \lambda \quad (5)$$

where:

$f(t)$ —The actual sensed data.

$f_p(t)$ —The predicted data at time t .

While λ —The sensed data changing speed with time t .

3.1.5. Data Loss Fault

A Data Loss fault occurs due to the sensor fault or communication fault and few data are missed with the actual system data rate. Equation (6) represents the Data Loss fault.

$$f(t) = \Phi, t > \tau \quad (6)$$

where τ - is the maximum time to sense the data in a sensor node.

Φ —is a null set value of the sensor node.

3.1.6. Out of Bounds

The range of sensed data varies beyond the range of the sensing circuit. This fault is discussed by [15].

The mathematical representation of out-of-bounds fault is Equation (7).

$$x^J > \theta \text{ or } x^J < \theta_1 \quad (7)$$

where x^J is the data-variable threshold for the sensed value.

θ and θ_1 —established the range of data variables.

3.2. Classifiers

The sensed and collected data have been categorized into predefined categories, as discussed in the introduction. As many works have been analyzed in the literature review, the following classifiers have been implemented in this work. For fault detection from the sensed data, SVM has been implemented in WSN, and for the classification of sensed data, the CNN has been implemented and have used the RF mechanism to find and category the faults of sensed data, have used PNN to find the faulty sensor nodes from the WSN.

3.2.1. SVM

In SVM, the sensor nodes are identified and classified in hyperplane in a non-probabilistic manner. The data collected are given to training and the same is labeled as a supervised learning mechanism. This hyperplane has divided the plane into two sub-classes. The SVM classifier has parameters such as the epsilon (ϵ), regularization, and kernel parameters.

3.2.2. MLP

MLP (Swain et al.) has been implemented in WSN to identify the faults from the category of Feed-Forward Artificial Neural Networks (FFANN) methodology. This technique processes the sensor nodes in multiple layers. Every sensor node is active as a neuron that transfers the data from one point to other points by the way of nonlinear fashion. This technique has been categorized under the classification of supervised learning and it has also used the detected data for training purposes [14,15].

3.2.3. CNN

Every sensor node in a WSN (Yuan et al.) is the same as a neuron in our nervous system. Sensor nodes pass data to a base station as neurons do in our nervous system. The neural networks are inspired by our nervous system. In this system, the brain has the ability to collect and process various datasets of text files, images, and video files. Our nervous system has the following layers of neurons namely input layer sensing cells, output layer controlling by commands and hidden layers.

In neural networks, every neuron receives many inputs, and consolidates all inputs. After the consolidation, the information is passed to the next neuron.

The following steps are followed in the CNN mechanism to receive, process and transmit signals to the neighboring neurons:

- In the first step, the data sensed by the sensor node are to be reduced to a level of acceptance or minimum required level. At this input level, many sensors sense the data at high rates. This high data rate of sensing is not needed to get meaningful full data. This data reduction makes the system faster with an efficient method of sensing.
- The next step in CNN is to process the sensed and received data. Data processing in sensor nodes involves a lot of power and resources. In a neural network, the sensed data are processed on a common point.
- The final step in CNN is to provide communication between all the sensor nodes in the entire network. This process is used to predict the type of fault and classify the sensor nodes in the network. After the classification, the faulty nodes are replaced or connected with other nodes in WSN.

3.2.4. RF

Classification of sensed data and sensor nodes is performed in the RF method through collective or collaborative learning classification and decision making. It creates a group of decision points at training time. This also generates a class for predicting the network structures. The number of trees and its accuracy of classifications are proportional (Yuan et al.) [10]. It works on two steps of operation functionally in WSN as fault finding and failure node identification:

RF tree creation process is the first step.

- The number of total features available in the system is referred to as m . From these available features, some of them are selected randomly and they are represented as K .
- Few nodes are chosen from the selected features K , by the method of best split-point. The selected nodes are divided into daughter nodes using the best split method. The above procedure is repeated to get the best tree output from the entire network.

After the RF tree formation, the data sensed and communicated by these RF Trees are to be classified. This classification is computed by the following stages for classification.

In the RF tree, sensed data and communication processes are given for voting in the entire WSN.

- The prediction target where the maximum voted has been considered as the best result of this RF algorithm.

3.2.5. SGD

Fault finding in the sensed data from a network is completed by estimation method on SGD as a linear model. Estimation of fault on the data sample on every predefined time interval is calculated on WSN. The method used to find the rate of faults is by the way of learning. The mean and the variance must be in the predefined range of estimation. Zero mean and unit variance assure the best result in the rate of faults. In a fault function, the predefined offset value is added as a penalty with the loss function, and this makes this fault model an error-free system. This has been implemented by the combination of two functions such as squared Euclidean Norm (EN) L2 and the Absolute Norm (AN) L1. This combined procedure is called the elastic net.

3.2.6. PNN

PNN has been derived from the neural network's methodology. This method is highly sensitive in one path, and it has a greater value than the other path. The sensed parameter variables should have a normalized scale within themselves; otherwise, they must be normalized to an acceptable level before they are used. If smaller datasets are used in applications and classification schemes, then the PNN algorithm is an efficient one.

The PNN is used to calculate the Probability Distributed Function (PDF) by the Parzen Window (PW) and this PW is based on Kernel Density Estimation (KDE). The PNN is a multi-layered algorithm based on Feed Forward Network (FFN) at four levels as follows:

- **Input layer:** As the PNN is based on the neural network, it has multiple layers of operation in it. The starting point of this algorithm is the input layer. The neurons act as a predictor variable among the available network structures. In WSN, the sensor node's communication is considered as neurons. In PNN, N number of neurons as groups are available and N−1 neurons are considered as logical variables. For each group, the range of the variable is standardized by adjusting with the offset variable. These calculated values are utilized as input to the following layer of neurons.
- **Pattern layer:** The next layer of PNN is the pattern layer. At the training phase, each dataset contains a neuron. In every case, the predictor variables and expected target values are stored. After the target case is set, an individual neuron computes the Euclidean Distance (ED) from the midpoint of all other neurons in the layer. The entire process is repeated to get the best ED. Finally, the calculated ED is used in the process with the radial kernel function and sigma values.
- **Summation layer:** The pattern layer's output is given as input to the summation layer for further processing in PNN. Each category of output variable has one existing neuron pattern. All the hidden neurons aggregate the collected target data on every training case. These are weighted average values of the pattern neurons. All the collected values are aggregated on the pattern neurons which are forwarded to the output layer.
- **Output layer:** Data collected from the summation layer are passed to the output layer with the weighted votes for comparisons. The result of these comparisons generates the weighted votes of the target category for each target class. This is referred to as the calculated output of the second layer.

4. System Model

The proposed system consists of two sensor node modules connected with a computer for the database by wireless connectivity such as a Bluetooth protocol. Each sensor node senses the data independently and communicates this to the computer for further processing and the database as shown in Figure 2.

The proposed system functionally includes three categories.

- Data sensing phase;
- Fault injection phase;
- Fault estimation and classification.

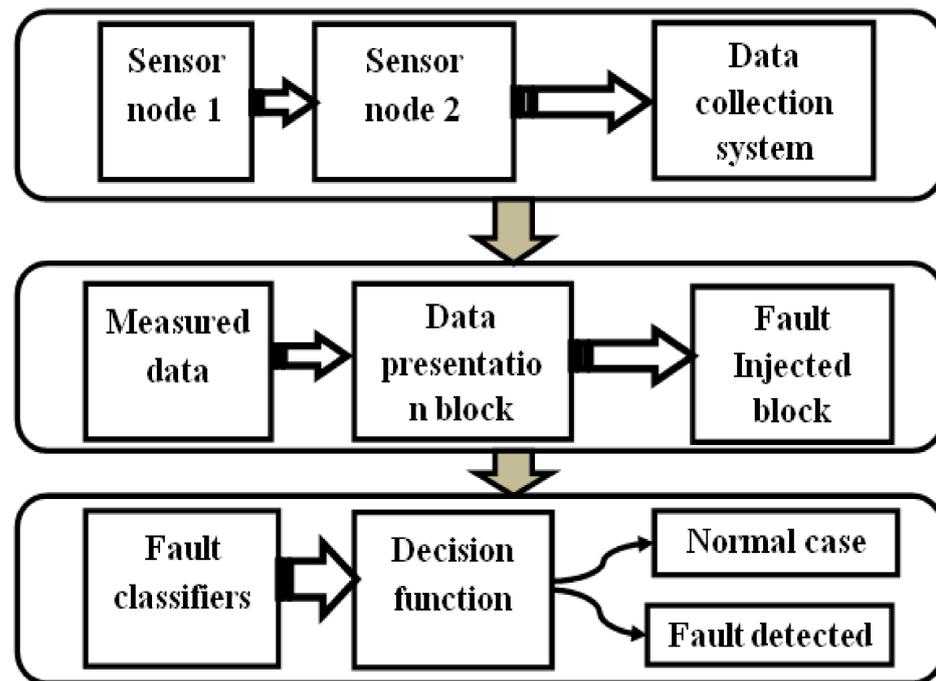


Figure 2. The system model of fault detection.

Step1

Data sense is involved in this Step of operation. The output of this sensed data is used for the data-generation block that creates an observation vector V_t . This vector incorporates two functions each from humidity functions of measurements, i.e., H_1 , H_2 , and two temperature functions of measurements, i.e., T_1 and T_2 . V_t , V_{t-1} and V_{t-2} are the measured datasets aggregated successively. These are contributed to the new observation vector.

Step2

In this Step, faults are infused on the actual sensed datasets. Gain fault, Offset fault, Out-of-Bounds fault and Stuck-at fault are infused as referred from [1]. The data loss and spike faults are infused in the dataset at the second Step of this proposed system model.

Step3

This is the Step where the collected data are checked and classified based on the results. As pointed out in the introduction, the WSNs are formed with many clusters of sensor nodes connected wireless with other sensor nodes. A cluster head is selected from every cluster of sensor node, and it collects data from the own cluster nodes and passes the collected data to the next cluster head node for successful communication.

In this Step, the classifiers which have been discussed before are used for the classification of faults based on the estimated result. The techniques used for classification such as SVM, CNN, SGD, MLP, RF and PNN, are implemented in the cluster head node of every cluster. A vector function is formed by each of these techniques to generate a decision function. Due to the limitations of WSN, the vector functions are not complex in the aspect of computational complexity. After the evaluation of the decision function of the sensor nodes, the collected data are classified into two classes. If the decision function is positive, the data are categorized as a normal class (fault-free case); otherwise, the data are classified as a faulty case. Data are sensed by two sensors nodes at the generation level. These sensed data are injected with predefined faults in the generation phase itself. In this phase, all the classifiers are run in the data with faults to detect the injected faults. Then, the faults are identified at different levels with different detection accuracy levels. From these results, the graph is plotted for better understanding.

5. Simulations and Results

The performances of the Classifiers are evaluated in the simulator using Python programming. The datasets derived from the proposed system model are given as input to the simulation process.

5.1. Datasets

As mentioned in the system, two multi-hop sensors are used for data sensing and communication. These sensors are placed on the outside of the data gathered. Temperature and humidity are sensed by the sensors. The data are collected at time instant of t_0 , t_1 and t_2 consecutively. At every instant of time, the sensors measure the temperature and humidity twice at every instance (T_1, T_2 and H_1, H_2). After the measurement of temperature and humidity, the various faults (Offset, Gain, Stuck-at, Out of Bounds, Spike, and Data loss) are induced randomly at the predefined rates (10%, 20%, 30%, 40% and 50%). The classified datasets are tabulated as normal data (error-free) and marked as one, whereas faulty datasets are denoted as abnormal with -1.

5.2. Results—Scenario I

The performances of the analyzed classifiers are evaluated by four different matrices. Among the four different matrices, the first one is Detection Accuracy (DA) and the DA is derived as mentioned in Equation (8):

$$DA = \frac{\text{Detected faulty observations}}{\text{Total faulty observations}} \tag{8}$$

The SVM classifier detects the faults in the sensor nodes as shown in Figure 3. It detects the Data loss fault effectively by 99% of the total faults occurring in the sensor network. It detects fewer spike faults as they happen in a short period of time.

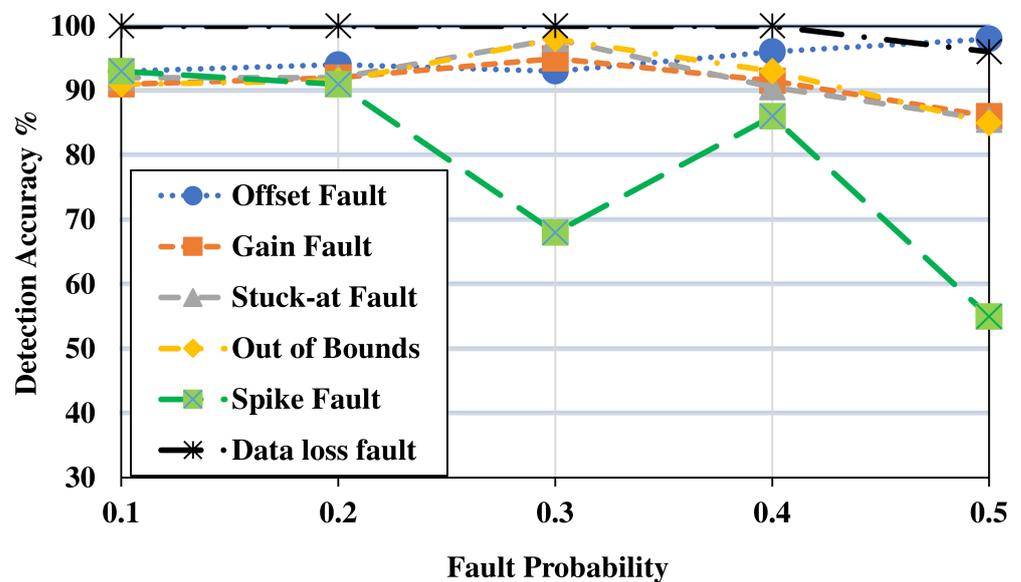


Figure 3. Percentage of fault detection accuracy of SVM.

The MLP classifier detects the faults in the sensor nodes as shown in Figure 4. It detects the Data loss fault effectively by 93% of the total faults occurring in the sensor network. The MLP classifier detects fewer faults than the SVM Classifier, as is discussed before this.

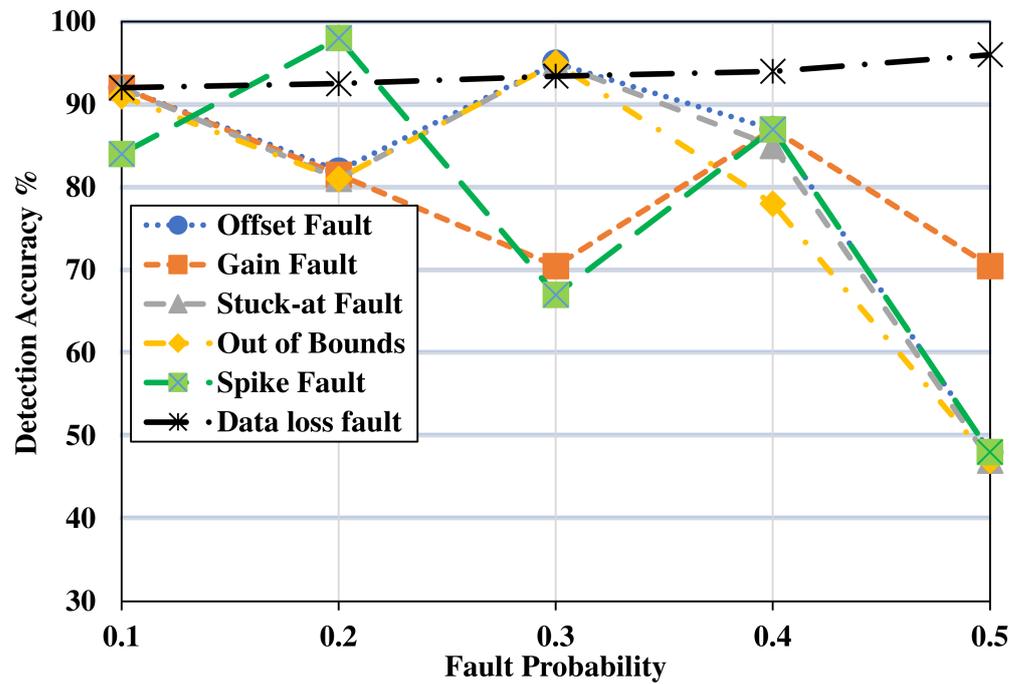


Figure 4. Percentage of fault detection accuracy of MLP.

The SGD classifier detects the faults in the sensor nodes as shown in Figure 5. It detects the data loss fault effectively by 81% of the total faults occurring in the sensor network. The average fault detection is less than the previously discussed Classifiers such as SVM and MLP.

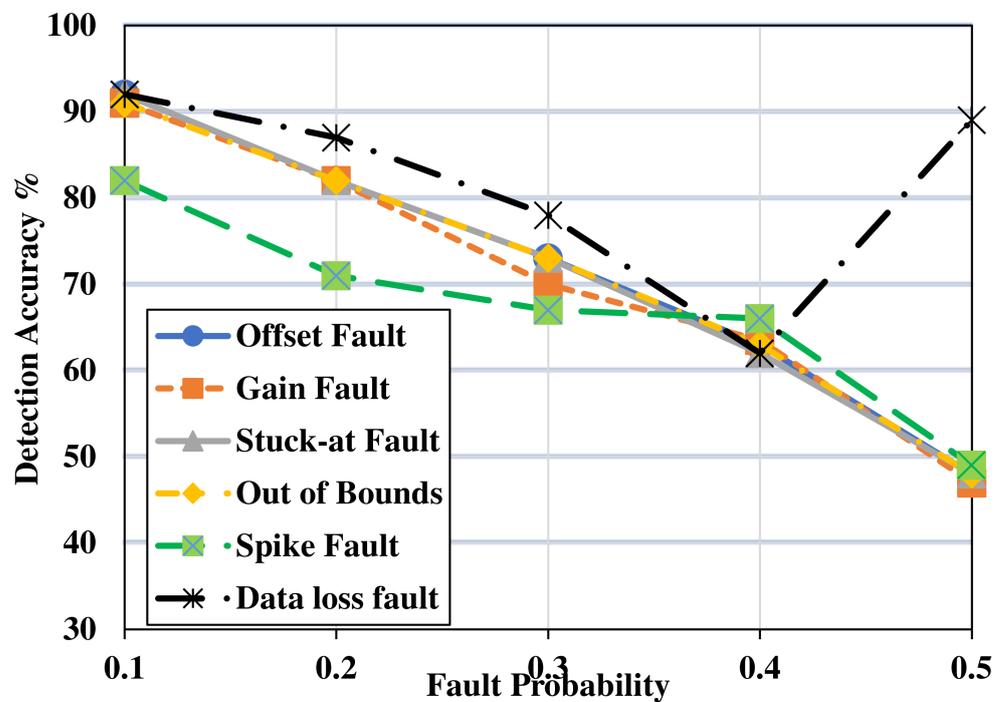


Figure 5. Percentage of fault detection accuracy of SGD.

The CNN classifier detects the faults in the sensor nodes, as shown in Figure 6. It detects all the faults in the same range, and the accuracy rate is decreased. The Out-of

Bounds fault is as low as 46%, and the same fault is detected with an accuracy of 90% on the probability of 0.1.

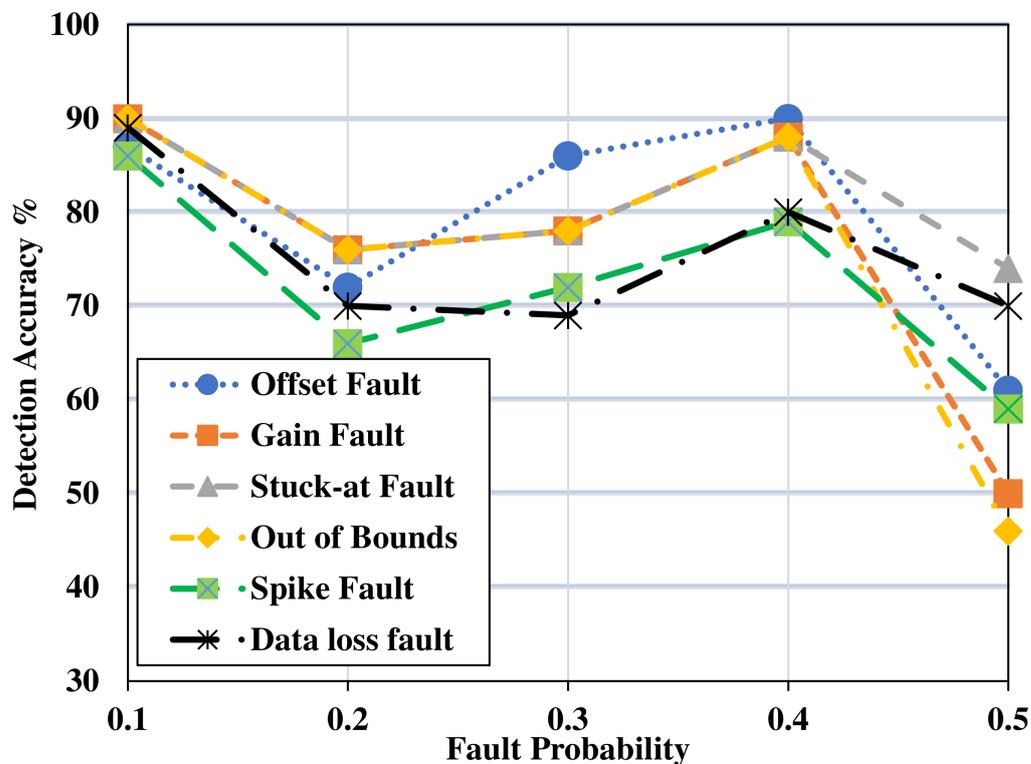


Figure 6. Percentage of fault detection accuracy of CNN.

The RF classifier detects the faults in the sensor nodes, as shown in Figure 7. It detects all the faults in the same range as well as with a high accuracy rate. The Out-of-Bounds faults and offset faults are detected 100% on the probability of 0.1 and 0.2, respectively.

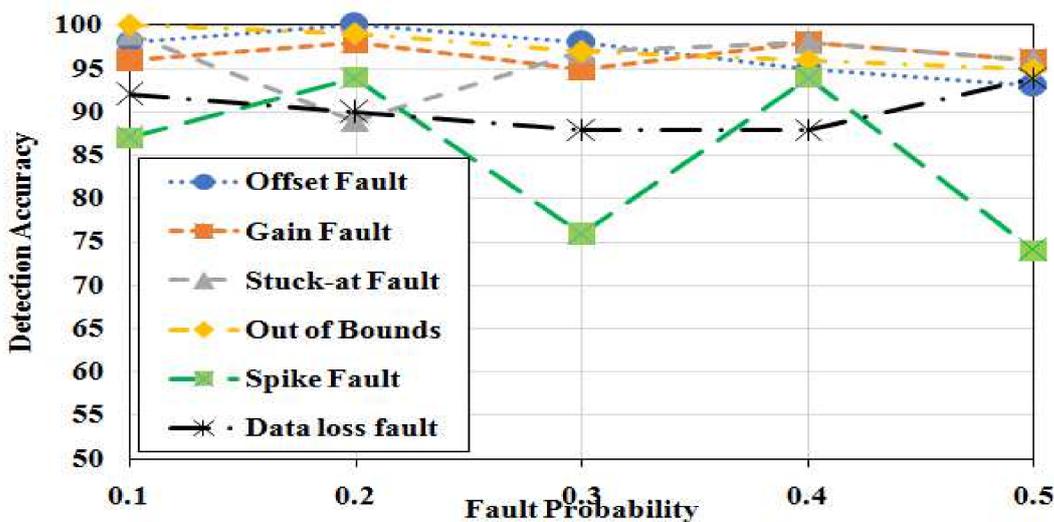


Figure 7. Percentage of fault detection accuracy of RF.

The PNN classifier detects the faults in the sensor nodes, as shown in Figure 8. It detects all the faults in the same range as well as with a high accuracy rate. A 99% spike fault is detected on the probability of 0.2.

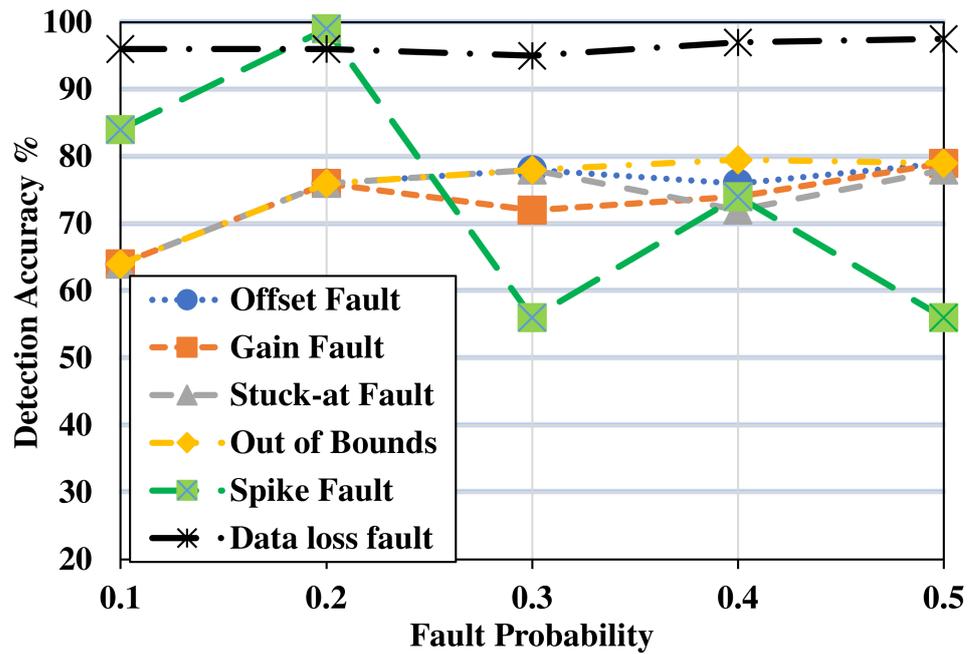


Figure 8. Percentage of fault detection accuracy of PNN.

5.3. Results—Scenario II

The performances of the classifiers are evaluated fault-wise for better understanding and recommendations for the classifier selection process in the WSN implementation.

The Offset fault is detected with higher accuracy by the RF classifier, as shown in Figure 9. It has been detected as 96.8% by RF, 94.8% by MLP, 94.5% by SVM, 79.2% by CNN, 74.6% by PNN and 71.6% by SGP. It is evident that RF outperforms all other classifiers in this fault as shown in Table 1.

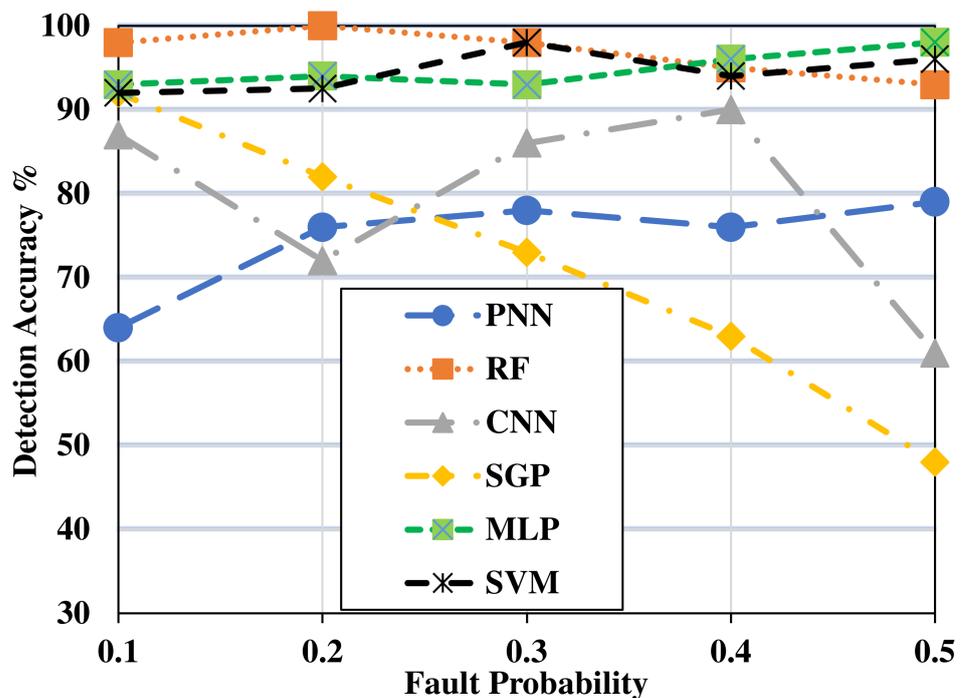


Figure 9. Offset fault detection accuracy.

Table 1. Offset fault detection accuracy.

Fault Probability	Fault Detection Accuracy in %					
	PNN	RF	CNN	SGP	MLP	SVM
0.1	64	98	87	92	93	92
0.2	76	100	72	82	94	93
0.3	78	98	86	73	93	98
0.4	76	95	90	63	96	94
0.5	79	93	61	48	98	96

The Gain fault has been detected with higher accuracy by the RF classifier, as shown in Figure 10. It has been detected as 96.6% by RF, 91.1% by MLP, 80.3% by SVM, 76.4% by CNN, 73% by PNN and 70.7% by SGP. It is evident that RF outperforms all other classifiers in this fault.

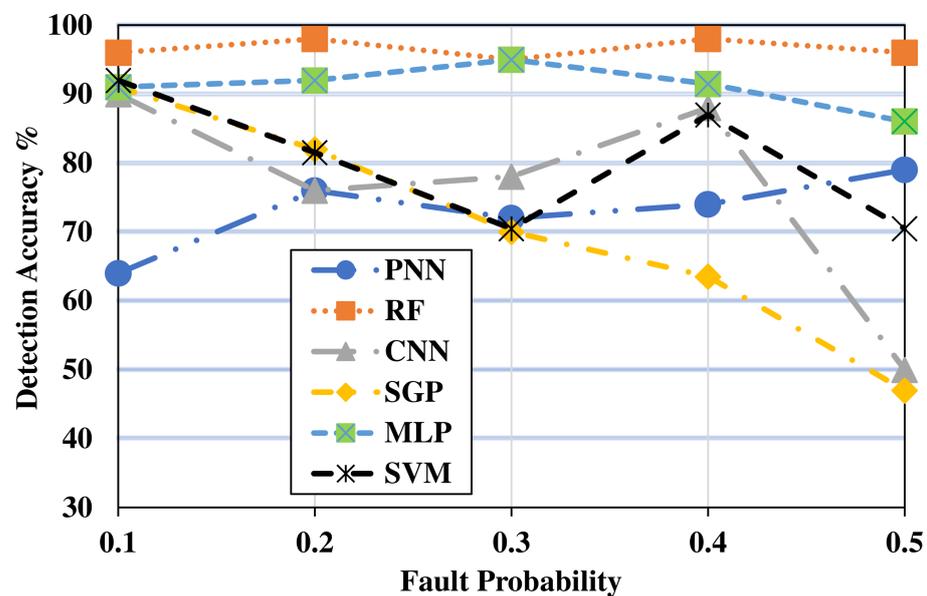


Figure 10. Gain fault detection accuracy.

The Stuck-at fault has been detected with higher accuracy by the RF classifier, as shown in Figure 11, and it has been detected as 95.8% by RF, 91.6% by MLP, 81.2% by CNN, 80% by SVM, 73.6% by PNN and 71.4% by SGP. It is evident that RF outperforms all other classifiers in this fault.

The Out-of-Bounds fault has been detected with higher accuracy by the RF classifier, as shown in Figure 12. The RF has detected 97.4%, whereas it is detected as 91.7% by MLP, 78.4% by SVM, 75.6% by CNN, 75.3% by PNN and 71.4% by SGP. It is evident that RF outperforms all other classifiers in this fault.

The Spike fault has been detected with higher accuracy by the RF classifier, as shown in Figure 13. It has been detected as 85% by RF, 78.6% by MLP, 76.8% by SVM, 73.8% by CNN, 72.4% by PNN and 67% by SGP. It is evident that RF outperforms all other classifiers in this fault.

The Data loss fault has been detected with higher accuracy by the MLP classifier, as shown in Figure 14. It has been detected as 99.2% by MLP, 96.3% by PNN, 94.5% by SVM, 90% by RF, 81.6% by SGP and 75.6% by CNN. It is evident that MLP outperforms all other classifiers in this fault.

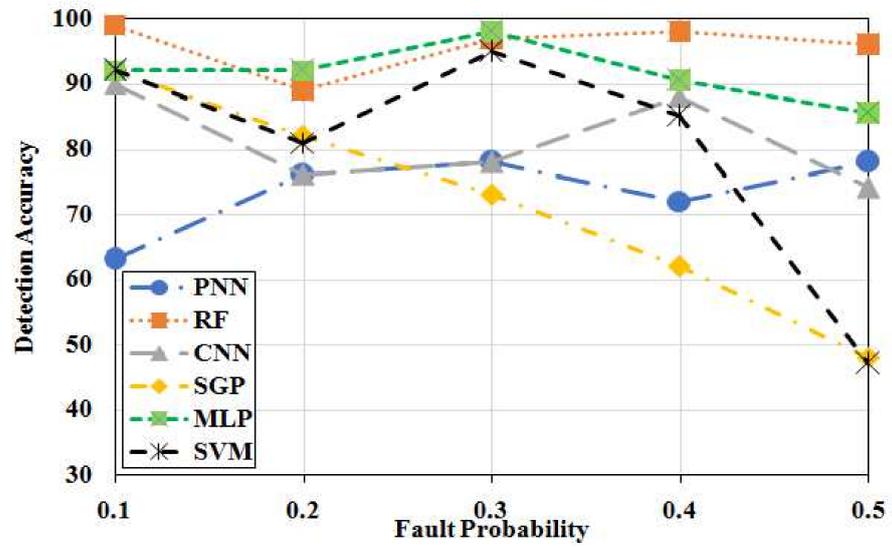


Figure 11. Stuck-at fault detection accuracy.

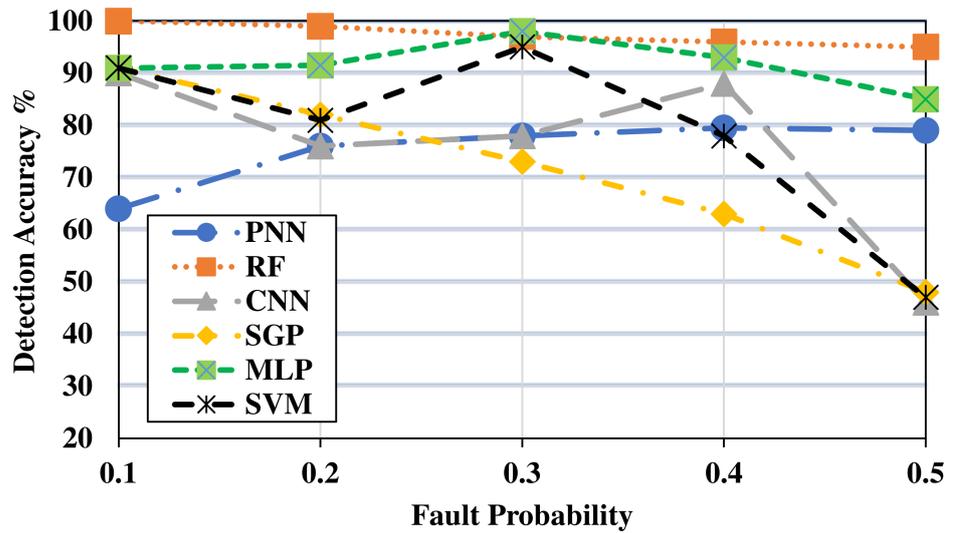


Figure 12. Out-of-Bounds fault detection accuracy.

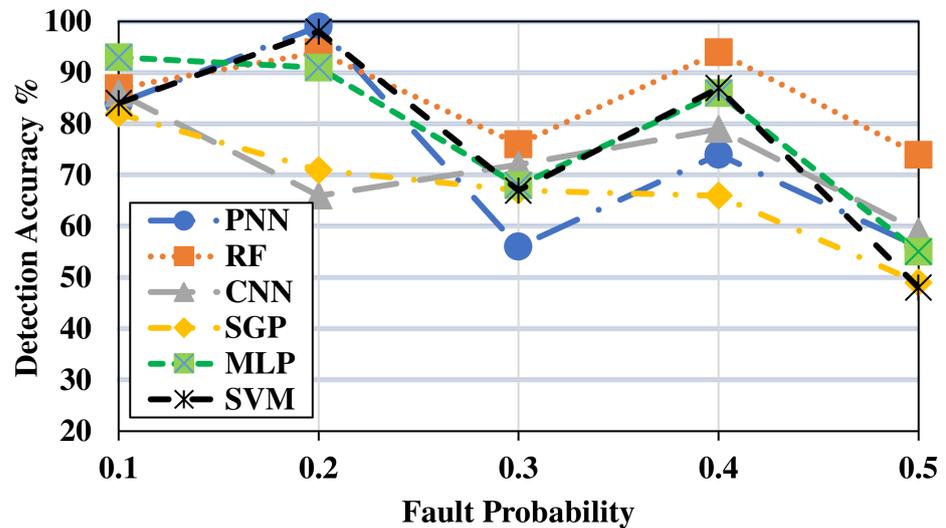


Figure 13. Spike fault detection accuracy.

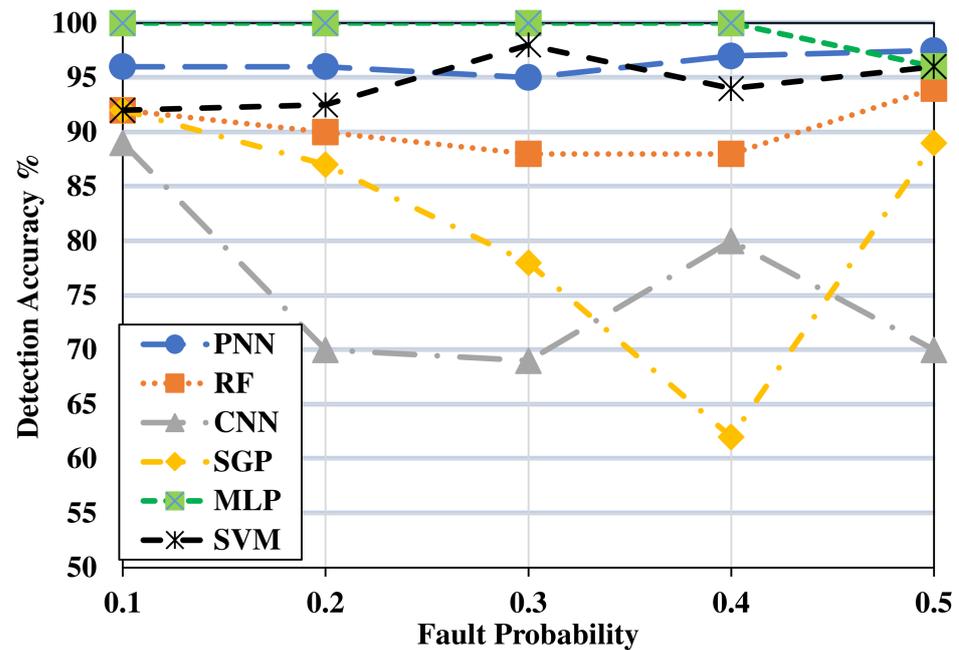


Figure 14. Data loss fault detection accuracy.

This chapter explains the different types of faults (Offset fault, Gain fault, Stuck-at fault, Out-of-Bounds, Spike fault and Data loss) that occur in WSN with their mathematical model. This gives the details of the classifiers (Support Vector Machine (SVM), Convolutional Neural Network (CNN), Multilayer Perceptron (MLP), Stochastic Gradient Descent (SGD), Random Forest (RF) and Probabilistic Neural Network (PNN)) used to detect faults. The simulation results show that the RF classifier outperforms all other classifiers in this category.

6. Conclusions

Six classifiers were used and tested with WSN. The simulated result is shown in Table 2. It depicts that the RF Classifier detects the Offset fault, Gain fault, Out-of-Bounds fault and Spike faults with 97% accuracy each and the Data loss fault with 90% accuracy. The MLP detects the Data loss fault with 99% accuracy. So, from the above results, the RF Classifier detects five faults out of six faults more accurately. The RF classifier outperforms all other classifiers in this category. Fault-wise analysis and classifier-wise analysis suggests that the Random Forest classifier is best suited for faultfinding in wireless sensor networks.

Table 2. Percentage of fault detection accuracy of classifiers.

Classifier	Fault Detection Accuracy in %					
	Offset	Gain	Stuck-at	Out-of-Bounds	Spike	Data Loss
PNN	75	73	74	75	74	96
RF	97	97	96	97	97	90
CNN	79	76	81	76	78	76
SGP	72	71	71	71	71	82
MLP	95	91	92	92	92	99
SVM	95	80	80	78	83	95

Author Contributions: Conceptualization, S.G. and M.S.; methodology, V.M.; software, G.K.; validation, R.S.A., S.A. and M.M.; formal analysis, O.I.K.; investigation, G.M.A.; resources, A.D.A.; data curation, T.H.H.A.; writing—original draft preparation, S.G.; writing—review and editing, V.M.; visualization, G.K.; supervision, G.M.A.; project administration, T.H.H.A.; funding acquisition, S.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless sensor networks: A survey. *Comput. Netw.* **2002**, *38*, 393–422. [[CrossRef](#)]
2. Acharya, S.; Tripathy, C.R. A reliable fault-tolerant ANFIS model based data aggregation scheme for Wireless Sensor Networks. *J. King Saud Univ. Comput. Inf. Sci.* **2017**, *32*, 741–753. [[CrossRef](#)]
3. Miao, X.; Liu, Y.; Zhao, H.; Li, C. Distributed Online One-Class Support Vector Machine for Anomaly Detection Over Networks. *IEEE Trans. Cybern.* **2018**, *99*, 1–14. [[CrossRef](#)]
4. Gao, Y.; Xiao, F.; Liu, J.; Wang, R. Distributed Soft Fault Detection for Interval Type-2 Fuzzy-model-based Stochastic Systems with Wireless Sensor Networks. *IEEE Trans. Ind. Inform.* **2018**, *15*, 334–347. [[CrossRef](#)]
5. Gharamaleki, M.M.; Babaie, S. A New Distributed Fault Detection Method for Wireless Sensor Networks. *IEEE Syst. J.* **2020**, *14*, 4883–4890. [[CrossRef](#)]
6. Gu, X.; Deng, F.; Gao, X.; Zhou, R. An Improved Sensor Fault Diagnosis Scheme based on TA-LSSVM and ECOC-SVM. *J. Syst. Sci. Complex* **2018**, *31*, 372–384. [[CrossRef](#)]
7. Kumaran, G.; Yaashuwanth, C. E2MR-HOA: Conservation of Energy through Multi-Hop Routing Protocol for WSN'S Using Hybrid Optimization Algorithm. *J. Circuits Syst. Comput.* **2021**, *30*, 2150041. [[CrossRef](#)]
8. Muhammed, T.; Shaikh, R.A. An analysis of fault detection strategies in wireless sensor networks. *J. Netw. Comput. Appl.* **2017**, *78*, 267–287.
9. Noshad, Z.; Javaid, N.; Saba, T.; Wadud, Z.; Saleem, M.Q.; Alzahrani, M.E.; Sheta, O.E. Fault Detection in Wireless Sensor Networks through the Random Forest Classifier. *Sensors* **2019**, *19*, 1568. [[CrossRef](#)] [[PubMed](#)]
10. Yuan, Y.; Li, S.; Zhang, X.; Sun, J. A Comparative Analysis of SVM, Naive Bayes and GBDT for Data Faults Detection in WSNs. In Proceedings of the 2018 IEEE International Conference on Software Quality, Reliability and Security Companion, Lisbon, Portugal, 16–20 July 2018; pp. 394–399.
11. Zidi, S.; Moulahi, T.; Alaya, B. Fault detection in wireless sensor networks through SVM classifier. *IEEE Sens. J.* **2017**, *18*, 340–347. [[CrossRef](#)]
12. Cheng, Y.; Liu, Q.; Wang, J.; Wan, S.; Umer, T. Distributed Fault Detection for Wireless Sensor Networks Based on Support Vector Regression. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 1–8. [[CrossRef](#)]
13. Sohrabi, K.; Gao, J.; Ailawadhi, V.; Pottie, G. Protocols for Self-Organization of a Wireless Sensor Network. *IEEE Pers. Commun. Mag.* **2000**, *7*, 16–27. [[CrossRef](#)]
14. Swain, R.R.; Khilar, P.M.; Bhoi, S.K. Heterogeneous fault diagnosis for wireless sensor networks. *Ad Hoc Netw.* **2018**, *69*, 15–37. [[CrossRef](#)]
15. Swain, R.R.; Khilar, P.M.; Dash, T. Neural network based automated detection of link failures in wireless sensor networks and extension to a study on the detection of disjoint nodes. *J. Ambient. Intell. Hum. Comput.* **2018**, *10*, 593–610. [[CrossRef](#)]
16. Zhang, D.; Qian, L.; Mao, B.; Huang, C.; Huang, B.; Si, Y.A. Data-Driven Design for Fault Detection of Wind Turbines using Random Forests and XGboost. *IEEE Access* **2018**, *6*, 21020–21031. [[CrossRef](#)]
17. Zhang, X.; Zou, J.; He, K.; Sun, J. Accelerating very deep convolutional networks for classification and detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 1943–1955. [[CrossRef](#)] [[PubMed](#)]
18. Aliakbarisani, R.; Ghasemi, A.; Wu, S.F. A data-driven metric learning-based scheme for unsupervised network anomaly detection. *Comput. Electr. Eng.* **2019**, *73*, 71–83. [[CrossRef](#)]
19. Araya, D.B.; Grolinger, K.; ElYamany, H.F.; Capretz, M.A.; Bitsumlak, G. An ensemble learning framework for anomaly detection in building energy consumption. *Energy Build* **2017**, *144*, 191–206. [[CrossRef](#)]