INTELLIGENT MACHINE LEARNING APPROACHES TOWARDS SENSOR FAULT DIAGNOSTIC IN WIRELESS SENSOR NETWORKS

DISSERTATION

for the Degree of

MASTER OF PHILOSOPHY
(Electrical Engineering)

UMER SAEED

OCTOBER 2020

Intelligent Machine Learning Approaches Towards Sensor Fault Diagnostic in Wireless Sensor Networks

DISSERTATION

Submitted in Partial Fulfillment
of the Requirements for the
Degree of

MASTER OF PHILOSOPHY
(Electrical Engineering)

at the

UNIVERSITY OF ULSAN

by

Umer Saeed

October 2020

Publication No.______________

Intelligent Machine Learning Approaches Towards Sensor Fault Diagnostic in Wireless Sensor Networks

Approved by Supervisory Committee:

____

Prof. Hyung-Yun Kong, Chair

Insoo Koo
____

Prof. Insoo Koo, Supervisor

____

Prof. Sangjo Choi

School of Electrical Engineering
University of Ulsan, South Korea
Date: October, 2020

VITA

Umer Saeed was born in a small town of district Nowshera, situated in the Khyber-Pakhtunkhwa province in Pakistan. He received the bachelor's degree in Software Engineering from Comsats University Islamabad (CUI), Pakistan in February 2019.

Since March 2019, he is pursuing his master's degree from the University of Ulsan (UOU), South Korea, under the supervision of Professor Insoo Koo. His current research interests include machine learning algorithms development for sensors abnormal behavior detection and diagnosis, wireless sensor networks, and the Internet-of-Things.

Dedicated

To Mom;

To Dad;

To Sister;

To Friends.

ACKNOWLEDGMENTS

First and foremost, I owe all the accomplishments of my life to my parents, my father Mr. Aftab Saeed and mother Mrs. Zahida Begum. I will always be indebted of their exertions towards my education despite all the hardships.

I wish to express my gratitude to my elder sister Mrs. Nida Saeed who stood behind me everytime I came up with hardships in my life.

I am grateful to all my friends especially Hashim Khan, Abbas Gillani, and Zahoor Ahmad for their extreme support. Their love and kindness remained with me along the way.

I would like to express profound gratitude to my advisor, Professor Insoo Koo, for providing me the opportunity to be part of his research group. I am highly obliged to him for his kindness and accessible guidance throughout my research.

I wish to express special thanks to Dr. Sana Ullah Jan and Dr. Young Doo Lee for their helpful collaboration, encouragement, guidance, and care throughout this journey.

I gratefully acknowledge BK21+ for the financial support throughout my Master studies.

Extremely thankful to all the talented members of the Multimedia Communication Systems Laboratory. It was a pleasure to share the laboratory with all of you.

Finally, an extreme gratitude goes to the Pakistani community at UoU for their love, respect, and support.

Umer Saeed

Ulsan, South Korea. October - 2020

ABSTRACT

Intelligent Machine Learning Approaches Towards Sensor Fault Diagnostic in Wireless Sensor Networks

by

Umer Saeed

Supervisor: Professor Insoo Koo

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Philosophy (Electrical Engineering)

October 2020

Wireless Sensor Network (WSN) being highly diversified Cyber–Physical System makes it vulnerable to numerous failures. These failures due to abnormal behaviors in the network can cause serious threat towards safety, economy, and reliability of systems. Abnormal behaviors of sensors are primarily triggered by low-quality production, electromagnetic interference, and complex environments. The precise detection and diagnosis of abnormal behaviors in WSN is a challenging issue due to the diversity of deployment and limitations in the resources.

In this dissertation, a data-driven supervised machine learning-based techniques are considered to scrutinize the behavior of sensors through their data for the timely detection and diagnosis of abnormal behaviors (faults or anomaly). In this study, most of the faults that commonly occur in WSN are considered such as drift, hard-over, spike, erratic, data-loss, stuck, and random fault.

A trusted dataset published by the researchers at the University of North Carolina composed of temperature and humidity sensor healthy measurements of multi-hop scenario was acquired and the aforementioned faults were injected in the non-faulty (healthy) sensor measurements. This practise is common among researchers due to the lack in availability of defective datasets.

Events from fault occurrences were generated to replicate realistic scenarios of WSN. For instance, fault may occur in WSN for a short length as well as long, or it may occur in the combination of both. To detect and diagnose the faults in timely manner, an ensemble learning-based lightweight machine learning classification technique is adopted, which is known as Extremely Randomized Trees or Extra-Trees.

Furthermore, multiple data labelling approaches such as multi-label/multi-class were utilized in order to get the best performance out of machine learning classifiers. In this study, the proposed Extra-Trees-based detection and diagnosis scheme has shown the ability of robustness towards signal noise and strong reduction of bias and variance error.

The performance of the proposed scheme was compared with those of the state-of-the-art machine learning algorithms such as support vector machine, neural network, random forest, and decision tree. Performance evaluation shows the efficiency of the proposed scheme in terms of lightweightness and detection/diagnosis accuracy, precision, F1-score, and area value under the ROC curve. To achieve the lightweight measure, the proposed scheme training time was compared to the aforementioned state-of-the-art machine learning classifiers.

Contents

List of Figures

List of Tables

Nomenclature

Notation	Description
AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC-ROC	Area under the ROC Curve
CPS	Cyber-Physical Systems
CV	Cross Validation
CAFD	Context-Aware Fault Diagnostic
DA	Detection Accuracy
DT	Decision Tree
ET	Extremely Randomized Trees/Extra-Trees
FP	False Positives
FN	False Negatives
IoT	Internet-of-Things
ML	Machine Learning
MLP	Multi-Layer Perceptron
NN	Neural Network
RF	Random Forest
RBF	Radial-Basis Function
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
SCADA	Supervisory Control and Data Acquisition
TI	Tactile Internet
TP	True Positives
TN	True Negatives
WSN	Wireless Sensor Network

Chapter 1

Introduction

1.1 Motivation

Modern technologies such as Internet-of-Things (IoT) and Cyber-Physical Systems (CPS) plays a vital role in our everyday life. In the modern era, from the revolution of autonomous vehicles, smart homes, intelligent health care systems, advanced transportation, disaster management systems, modern agriculture, and energy management systems heading towards industry 4.0, the IoT and CPS has taken over the world. These advanced systems have the ability to address social challenges such as environmental sustainability and economic downfalls.

Encompassing human-to-machine and machine-to-machine communication, the Tactile Internet (TI) is considered an evolution of IoT. However, these advanced systems are based on the integration of diverse physical objects such as sensors. To monitor the physical conditions, sensors are spatially dispersed and data is collected at the central node engendering Wireless Sensor Network (WSN). Undeniably, sensors make our lives easier by their innumerable usages. Nevertheless, they come at a cost of being prone to failures.

According to one study, about 50 billion smart devices and one trillion sensors to be around and working by the year 2020. Deployment of sensors in complex environments while facing natural factors, electromagnetic interference and other relevant factors can lead towards the sensor abnormal behaviours. These abnormal behaviours (anomaly or faults) are serious threat towards the safety, economy, and system's reliability.

1.2 Thesis Objective

The primary objective of this thesis is to develop a lightweight effective system utilizing data-driven supervised machine leaning-based algorithms for the detection and diagnosis of abnormal behaviours occurring in WSN. In order to achieve this goal, multiple sensor observations in healthy and non-healthy state are given as input to the distinct machine learning classifiers while utilizing multi-label and multi-class classification approaches. With the aim of accurate detection and diagnosis of sensor faults, ensemble learning approach, Extremely Randomized Trees have been proposed that provides the robustness towards signal noise and strong reduction of bias/variance error.

1.3 Thesis Outline

This thesis consists of four chapters as follows:

- **Chapter 1** presents motivation, thesis objective, and thesis outline.
- **Chapter 2** provides an ensemble learning-based fault detection and diagnosis approach utilizing Extremely Randomized Trees in wireless sensor networks.
- **Chapter 3** proposes a context-aware fault diagnostic scheme for the sensor faults using intelligent machine learning classification approaches.
- **Chapter 4** concludes the thesis contributions and presents future works.

Chapter 2

Fault Diagnosis based on Extremely Randomized Trees in Wireless Sensor Networks[1]

2.1 Introduction

Wireless Sensor Network (WSN) often consist of hundreds of sensors connected through wireless channels. Sensors in the current era perform an essential role by covering a vast number of applications [2], [3]. Ground-breaking technologies like the Tactile Internet (TI) are considered an evolution of the Internet of Things (IoT), encompassing machine-to-machine and human-to-machine communication. The TI has to deal with interactive systems in real-time, with high densities for sensors and actuators [4]. In these modern times, from the revolution in smart grids, autonomous vehicles, smart homes, intelligent transportation systems, advanced agriculture, disaster management, and health systems heading towards

[1] The study in this chapter was published in Elsevier Reliability Engineering and System Safety. [1]

Industry 4.0, the cyber-physical system (CPS) has taken over the world [5], [6]. One estimate is for about 50 billion smart devices and one trillion sensors to be around and working by the year 2020 [7].

The potential of WSN has gained the attention of researchers from all over the world due to their minimal cost and the enormous domain of applications. In WSN, the collections of sensor data are smartly processed and communicated. However, technologies like sensors, without a doubt, make our lives easier because of usages ranging from health care systems to transportation. They also come with a vulnerability to distinct failures [8]. Sensors are prone to break down due to electromagnetic interference, deployment in complex environments, and from other natural factors. Any of these considerations make sensors susceptible to hardware, software, or communication failure [9].

WSN plays the role of interface between the physical and digital worlds. When sensors communicate faulty data to the sink node, it may lead to serious outcomes in terms of safety, economic impact, and system reliability. When a sensor is defective, it can completely stop signal generation, or it may send incorrect signals. It can unstably switch between normal and defective. Defects or faults can be expressed as an irregular property in the behavior of the system [10].

Studies have been conducted mainly from the 1980s for the detection and diagnosis of defects in physical facilities, i.e. industries. These approaches were restricted to certain conditions and environments. It was hard to determine many model parameters due to a system's complexity. Recently, due to the emergence of Artificial Intelligence (AI) techniques, data-driven modern approaches like Machine Learning (ML) have been widely considered for fault detection and diagnosis of faults in rolling elements or bearings [11], [12]. For example, Haedong et al. [13] studied fault detection and identification (diagnosis) of the machinery systems utilizing model-based approach. In [14], Wasim et al. proposed the prognostic (Remaining Useful Life) technique for rotatory machinery utilizing machine learning regression methods. However, faults also appear more frequently in sensors, and they trigger serious consequences. Therefore, timely detection and diagnosis of sensor faults in WSN are extremely important to ensure safety, strengthen network security, improve data quality, extend the network lifetime, and shorten response times [8], [15].

The general approaches to the detection of abnormalities in WSN are knowledge-based, signal-based, model-based, or a hybrid [16], [17]. Because of less equipment redundancy, the knowledge-based methods are becoming more appealing in industries [11], [18]. To gather historical data on industrial systems in large amounts, Supervisory Control and Data Acquisition (SCADA) systems are generally installed.

Sensors are deployed everywhere in our surroundings and their smooth workflow and reliable operations are compulsory to ensure safety and avoiding any economic loss. With the increase of WSN in recent times, many studies have been conducted for fault detection and identification in WSN to make it safe and reliable [19–25]. For constructing an intelligent WSN, there are many complications such as resource limitations, energy, and environmental constraints. Every technique considered to detect and diagnose faults in WSN must be efficient enough to reduce these limitations with high reliability [26, 27]. In [28], Xuedan et al. proposed an approach based on one-class SVM towards handling of streaming data in WSN utilizing online distributed method and to detect abnormalities over networks. Low misdetection and high true positives were achieved using the proposed solution. Minji et al. [29] considered sensor faults for their work and estimated the resilience of the system under sensor degradation and fault. In [15], Salah et al. proposed a technique of fault detection in WSN using SVM classifier. This work is based on binary classification (2-classes) which distinguishes only between normal and faulty signals and does not further predict the exact occurring fault in WSN. Likewise, Zainib et al. [30] considered a multi-hop scenario to detect faults in WSN and proposed an ensemble learning algorithm (RF classifier). They considered most of the faults commonly occur in WSN such as gain, offset, stuck-at, spike, data-loss, and out of bounds, but did not propose a solution to diagnose (classify) each fault.

Lately, supervised ML classification algorithms [31], such as the Support Vector Machine (SVM) [15], Neural Network (NN) [32], [33] and Random Forest (RF) [30] have become well-known topics for anomaly detection. Salah et al. [15] presented effective performance from an SVM, compared with other ML classifiers, while Zainib et al. [30] showed the effectiveness of RF classifier for fault detection in WSN. Samanta and Al-Balushi [12] revealed the efficiency of NN for rolling elements fault diagnostics. In this work, we utilize a novel classifier, Extremely Randomized Trees or Extra-Trees (ET) [34], and propose an ET-based diagnostic scheme for timely detection and diagnosis of sensor faults. As shown in other literature such as [35–38], the Extra-Trees algorithm has shown the ability of robustness towards noise and a strong reduction of bias and variance error. However, to the best of our knowledge, this sort of tree-based ensemble classifier has never been considered in prior research on the recognition of abnormalities in WSN. In addition, the randomization scheme in ET makes it computationally much faster than other ML classification algorithms. Subsequently, the proposed ET-based diagnostic scheme can achieve low training time when implemented in practical scenarios. The proposed algorithm is discussed in detail in Section 2.4.

The rest of this chapter is structured as follows. Section 2.2 presents the taxonomy of faults in WSN. In Section 2.3, the proposed scheme is discussed. A brief introduction to the ET algorithm is presented in Section 2.4. Data acquisition, data pre-processing, evaluation metrics, and simulation results are provided in Section 2.5. Finally, Section 2.6 concludes the chapter.

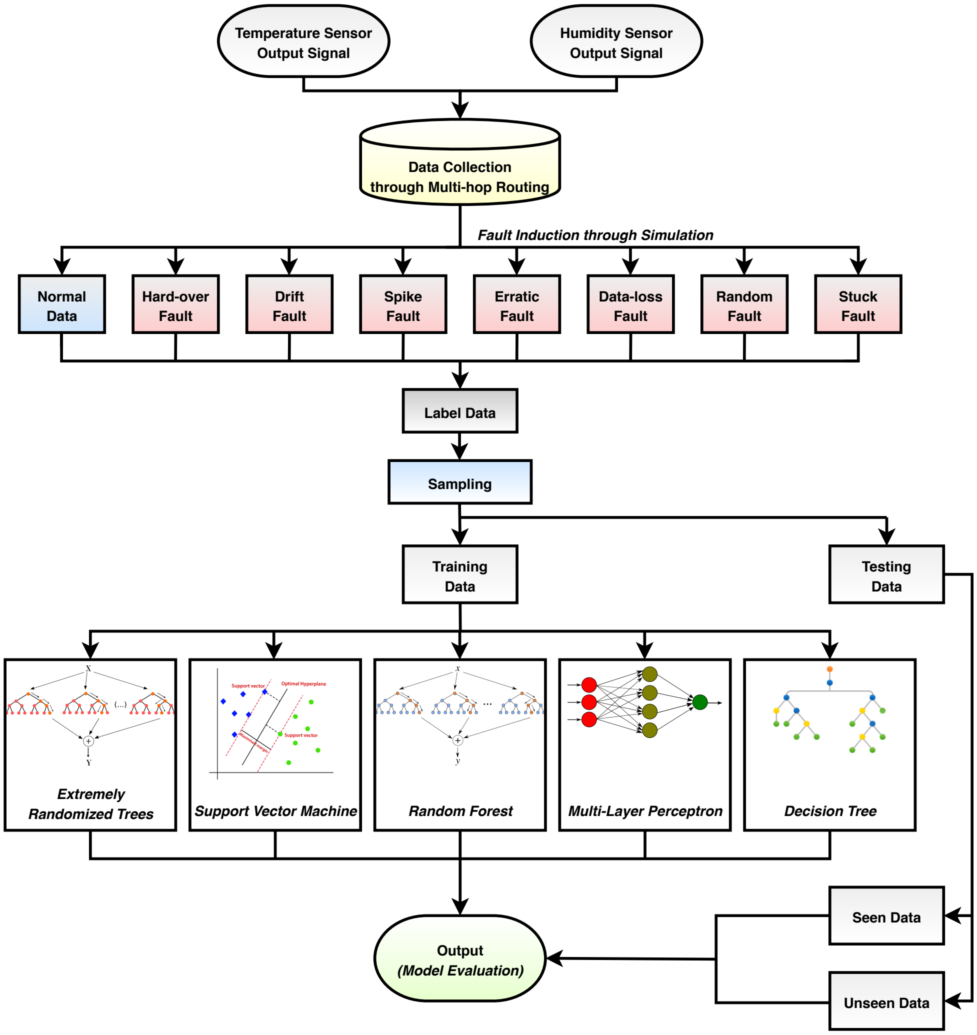


Figure 2.1: The framework of the applied model for fault diagnosis.

2.2 Fault Taxonomy

Figure 2.1 shows the overall flowchart of the applied model for fault diagnostics. As shown in Figure 2.1, the ET, SVM, RF, Multi-Layer Perceptron (MLP) and Decision Tree (DT)-based multi-class classifiers are used to investigate sensor fault diagnosis after preprocessing of data. In this work, the seven kinds of sensor faults such as hard-over, drift, spike, erratic, data loss, stuck and random faults are considered. Multi-dimensional data

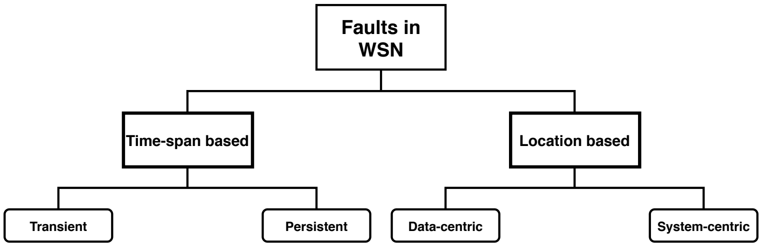


Figure 2.2: Fault taxonomy in WSN.

were given as input to the above-mentioned classifiers.

Faults in WSN are caused by various things which can be classified into two main categories: timespan-based faults and location-based faults [39], as shown in Figure 2.2. In the timespan-based faults, there are transient and persistent type faults based on the period of the fault. At first, transient faults are temporary and occur for a short period due to weather conditions, network congestion, etc [40, 41]. On the other hand, persistent faults are permanent and exist until recovery is carried out. The entire WSN is not normally defective. Instead of the entire global network, faults usually impact only a limited number of components. Therefore, the detection and identification of faults have to be based on the specific location rather than the overall global network [42]. In the location-based faults, there are data-centric and system-centric faults. At first, based on locality, data-centric faults, also known as soft faults, take the attributes of the sensed data into consideration when determining the specific fault. System-centric faults, also called hard faults, consider the characteristics of the systems used in the WSN. Moreover, the faults examined in this research can be considered in the context of soft faults. However, this categorization of faults is not disjoint and the categories can overlap one another.

In this research, seven distinctive WSN faults are considered so they can be diagnosed at an early stage to prevent serious consequences. The fault causes are related to either sensor functionality or the gathered data. In this work, the data accumulated from a sensor node can be modeled as a time series, shown in equation 2.1:

$$d(n, t, f(t)) \tag{2.1}$$

where n is the node id, t is the time instance of the sensed value, and $f(t)$ indicates the sensed value in node n during time t as modeled in equation 2.2:

$$\alpha + \beta x + \eta \tag{2.2}$$

where α is an additive constant (offset), β is a multiplicative constant (gain), x is the normal (non-faulty) sensor value, and η denotes noise in the data. In an ideal scenario, $f(t)$ will be only x, but in real-world situations, a non-faulty node will have $f(t) = x + \eta$.

In the present study, the types of faults include hard-over, drift, spike, erratic, data-loss, stuck and random faults. Sample plots of these faults are illustrated in Figure 2.3. These faults in the WSN can be described as follows.

2.2.1 Hard-over/Bias Fault

This fault occurs when the output of the sensor shifts from normal to a higher state. In other words, adding a constant bias to the normal signal [43]. Hard-over fault signal $S_n^{hardover}$ can be acquired by adding a high constant bias value b to all non-faulty S_n^{normal} signal elements. This fault is represented by equation 2.3:

$$S_n^{hardover} = S_n^{normal} + b, \quad b = constt \tag{2.3}$$

2.2.2 Drift Fault

This nature of fault appears when the output signal of the sensor keeps increasing linearly over time, starting from the normal state [6], [44]. Fault signal S_n^{drift} can be acquired

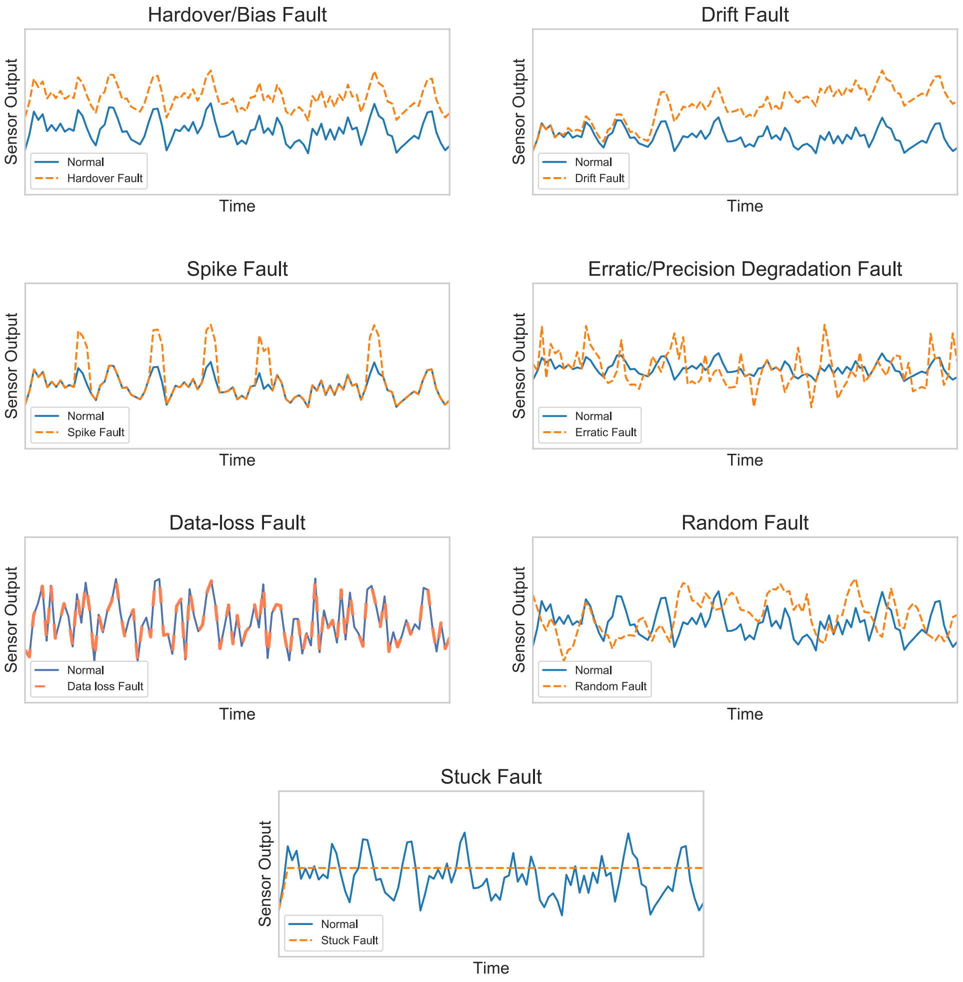


Figure 2.3: Sample plots of faulty and non-faulty signals.

in a non-faulty signal by adding a linearly rising bias term, where the added bias to the n^{th} element is n times the constant initial bias b_0. This fault can be modeled by equation 2.4:

$$S_n^{drift} = S_n^{normal} + b_n, \quad b_n = nb_0, \quad b_0 = constt \tag{2.4}$$

2.2.3 Spike Fault

In the output signal, this kind of fault is observed intermittently in the form of high-amplitude spikes [8], [28]. Periodically, a constant bias is added to the n^{th} element of the non-faulty signal to obtain spike fault S_n^{spike}, where $n = v \times \eta$ is the elements index in the signal, with $v = (1, 2, ...,)$ as natural numbers, and $\eta \geq 2$ as a positive integer. It is modeled in equation 2.5:

$$S_n^{spike} = S_n^{normal} + b_n,$$
$$b_n = \begin{cases} b, & n = v \times \eta, \quad v = (1,2,...,), \quad \eta = constt \\ \\ 0, & otherwise \end{cases} \tag{2.5}$$

2.2.4 Erratic/Precision Degradation Fault

The sensor's output variance increases significantly above the usual state [45]. To acquire erratic fault $S_n^{erratic}$ in non-faulty signal S_n^{normal}, a signal $\dot{S}_n$ of mean 0 and high variance, $\dot{\delta^2} \gg \delta^{2^{\text{normal}}}$, where $\delta^{2^{\text{normal}}}$ is the variance of the non-faulty signal is added to the raw non-faulty signal. This type of fault can be defined in equation 2.6:

$$S_n^{erratic} = S_n^{normal} + \dot{S}_n, \quad \dot{S}_n \sim N(0, \dot{\delta^2}), \quad \dot{\delta^2} \gg \delta^{2^{\text{normal}}} \tag{2.6}$$

2.2.5 Stuck Fault

This fault can be transient or persistent, according to the situation. There can be nil, or almost nil variations in the output signal of the sensor for a period of time [39]. In the case of complete failure, the output is stuck persistently at a constant value [46], [47]. To acquire stuck fault S_n^{stuck} in a non-faulty signal, a fixed value α is simply kept in all indices of the non-faulty signal. A stuck fault is revealed by equation 2.7:

$$S_n^{stuck} = \alpha, \quad \alpha = constt \tag{2.7}$$

2.2.6 Data-loss Fault

Occurring mainly due to a hardware or calibration defect, the data-loss fault (as the name suggests) is a null value sensed from a time series for a node. The detection of the data-loss fault has been considered in several kinds of research [30], [48].

2.2.7 Random Fault

This kind of fault in sensors can be explained simply as an instantaneous error where, for instance, the signal output is interrupted [15]. Not commonly explored by researchers, the random fault can be described as multiple positive or negative rapid peaks, which can affect WSN data.

2.3 Proposed Methodology

2.3.1 Data-Driven Approach

The data-driven approach has a wide number of real-world applications, which are common to constructing an appropriate ML model [49]. In the case of classification, the model is used to identify patterns and for structure discovery in the data. While AI methods and statistical approaches are two dissimilar techniques in the data-driven world, it is now common to apply intelligent techniques to solve fault detection and diagnosis problems. The classification-based approach is considered one of the best solutions for the categorization and identification of faults occurring in WSN. Figure 2.4 illustrates the common steps taken towards fault classification or diagnosis. After data acquisition and the identification of certain classes, intelligent models are employed.

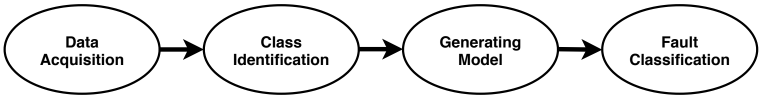


Figure 2.4: Steps towards fault classification.

2.3.2 Fault Diagnosis System Model

As the data preparation block illustrates in Figure 2.5, to construct an observation vector or data sample (X_τ), four successive instances or data measurements were created: $(t_n, t_{n+1} ... t_{n+N})$. Each data measurement consisted of two temperature (T) and two humidity (H) sensor measurements. In each instance, the sensed readings were taken from both mote 1 and mote 2 of the network, as explained in Section 2.5. The faults induced in the data for experimental purposes were hard-over, drift, spike, erratic, data-loss, random and stuck faults. The prepared datasets were labeled with each faulty and non-faulty observation.

The classification function is based on data learning. In scenarios like fault detection and diagnosis in WSN, having accurate data is an important key, which can give meaningful information to resolve certain problems in the network. Therefore, to address the problems discussed in the previous section, data must be classified so accurately that it can classify any new observation (data) in real-time with tremendous accuracy. In this research, our proposed solution is based on an ensemble learning technique called Extremely Randomized Trees, which generates a decision function using a collection of decision trees. The classifier takes an input feature vector and classifies it with each tree in a forest-like structure and, based on a majority of vote, outputs the labeled class. A WSN is a collection of multiple interconnected nodes having a cluster head that communicates with other layers of the

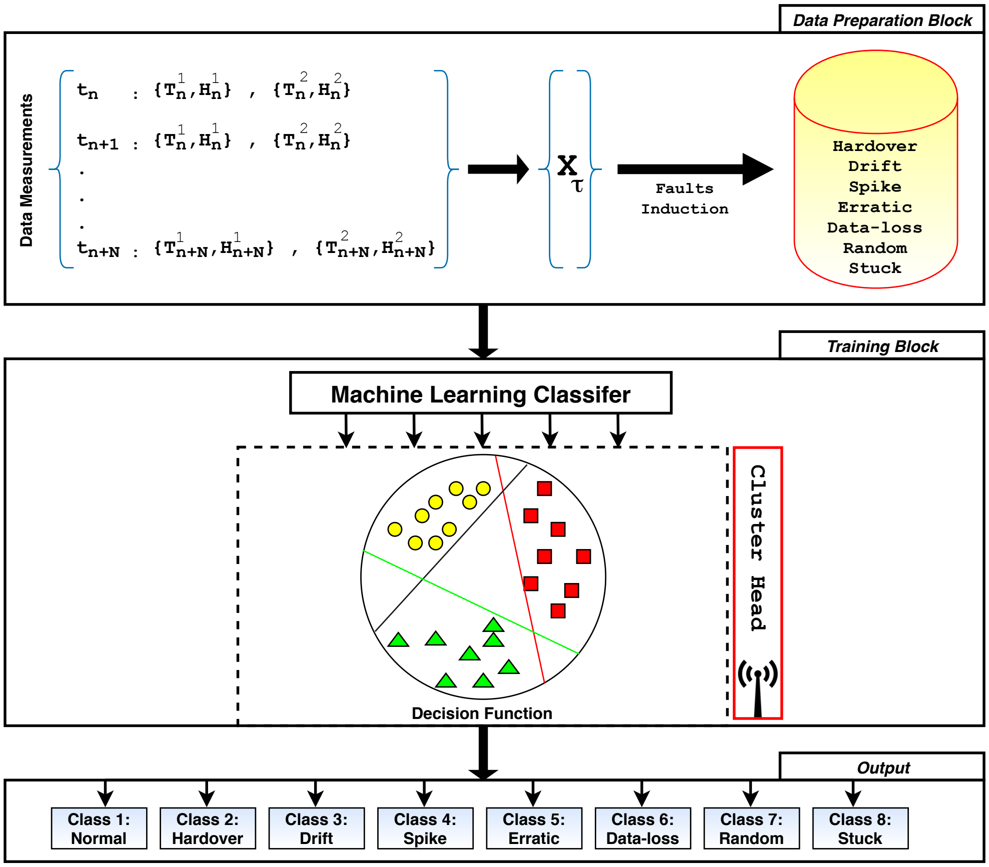


Figure 2.5: Proposed system model for timely detection and diagnosis of faults in WSN.

network/nodes. For classifying data in the WSN scenario, this decision function, also known as separation function, is deployed in the cluster head. After the deployment, to diagnose the data, the output is classified into eight distinctive classes, which are composed of normal and faulty functionalities, as shown in Figure 2.5.

2.4 Extremely Randomized Trees

Introduced in 2006, the Extremely Randomized Trees or Extra-Trees algorithm is an ensemble approach based on a large number of decision trees [34]. The ensemble technique is used in a vast number of applications for classification and regression tasks [50].

The idea behind the ensemble technique is to combine the decisions of distinct models and make a judgment based on that combination, which essentially results in better performance, compared to the achievements of a single decision or model. The DT-based ensemble technique can achieve high performance when the base learners are independent, and that can be attained through randomization. When growing the trees, randomization entails better tree diversity, and facilitates reducing the correlation [51]. One might say that ensemble learning methodologies work on the principles of divide-and-conquer approach (or the wisdom of the crowd) to achieve enhanced performance. In supervised ML tasks, we can get a stable and more robust classifier (model) with precise predictions using an ensemble technique because it reduces the factors, i.e. noise, bias, and variance. However, an ensemble learner can cause a notable raise in computational costs due to the need to train a number of individual classifiers. Consequently, we highlight the ET algorithm, which works almost similar to, yet faster than, the tree-based ensemble method, i.e. random forest.

The ET algorithm consists of number of DT, where each tree is composed of a root node, child/split nodes, and leaf nodes, as shown in Figure 2.6. Given a dataset X, at the root node, ET selects a split rule based on a random subset of features and a partially random cut point. In each child node, this procedure is repeated until reaching a leaf node. Furthermore, the three most important parameters of ET can be outlined as the number of trees in the ensemble (k), the number of attributes/features to select randomly (f), and the minimum number of samples/instances required to split a node (n_{min}).

As an ensemble of individual trees, the ET algorithm is similar to the regular RF, but with two key differences. First, instead of training a bootstrap sample, the entire learning sample is used to train each tree. Second, the top-down splitting of nodes in the tree is with completely random splits, not the best splits. A random cut-point is used instead of calculating the locally optimal cut-point for each attribute being considered, based on

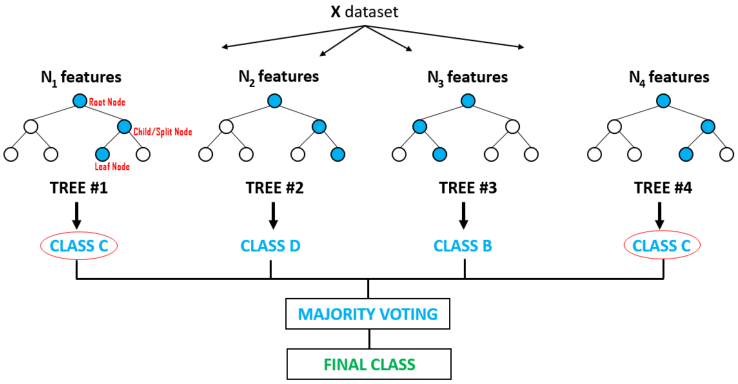


Figure 2.6: Illustration of the Extremely Randomized Trees algorithm (an ensemble of decision trees).

gini impurity or information gain. This value is selected from a uniform distribution within the attribute's empirical range (in the training set of the tree). Subsequently, the split that produces the highest score of all the randomly generated splits is selected for splitting the node. Since finding the best split at every node for each attribute or feature is highly time-consuming when growing a DT, the process of ET makes it much faster to train than an ordinary RF algorithm. Also, ET outperforms RF when there are noisy points in the data, which is usually the case with sensors, as detailed in the findings in Section 2.5.

Furthermore, in the testing process, a test sample proceeds through each of the DT and to each child node, choosing the best splits and forwarding the test sample to the right/left child node of the tree before a leaf node is reached. Class for the test sample in any DT is determined by the leaf node and the final prediction is called as the majority of votes by the k decision trees of the ET algorithm (Figure 2.6).

The generalization error of the ML model can be declared as the sum of unique errors, i.e. bias and variance. A high bias can give rise to underfitting, which can be calculated as the ability to generalize unseen data accurately. In other circumstances, a

Table 2.1: Steps towards the Extra-Trees algorithm.

```
1: Construct a training set of size $S$.
2: Randomly select $n$ learning samples without replacement from training set $S$ (Bootstrap=False).
3: Build a tree from the entire learning sample. At each node:
   3.1: Randomly select $f$ features without replacement.
   3.2: Split the node by random cut-points.
4: Repeat, $k$ times, steps 2-3.
5: Aggregate the results of each tree to assign the respective class (majority voting).
```

high variance can arouse overfitting, which is provoked by the intense sensitivity of the model to inconsequential variations in the training set. The ET algorithm has the ability to strongly reduce bias and variance error better than any other randomization method, i.e. random forest. The variance is minimized by the selection of the cut-point and the explicit randomization of the subset of attributes, whereas the bias is minimized due to the complete use of the original training set to learn the individual DT [34].

Furthermore, a major advantage of ET during implementation is that it does not need immense concentration towards the selection of hyperparameter values. The ET model is quite robust to noise from an individual DT such that, typically, there is no need to prune. The general working steps of the ET algorithm are summarized in Table 2.1. In practice, the number of trees k (step 4) is considered to be the single parameter that needs to be taken care of while constructing the ET model.

2.5 Experimental Results

To conduct this research, we did simulations in Python [52]. For data preparation, we used the NumPy, and Pandas libraries. To evaluate the performance of the classifiers, we performed simulations on a system with an Intel Core i5 CPU and 8 GB RAM. Details on the datasets and simulations are as follows.

2.5.1 Data Acquisition

In our research, a data-driven approach was adopted for fault classification, which used historical data for training. For this research, we used a dataset published online by researchers at the University of North Carolina at Greensboro. They collected temperature and humidity sensor data using Telos B motes, generating single-hop and multi-hop WSN scenarios [53]. For our research, we considered the data from the multi-hop scenario (Figure 2.7).

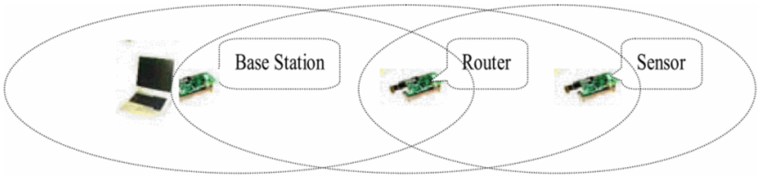


Figure 2.7: Multi-hop scenario.

2.5.2 Data Preparation

To prepare our dataset, we considered the multi-hop indoor dataset. Following a set of observations, we generated 16-dimension data. Each vector or data sample contained measurements in four successive instances $(t0, t1, t2, t3)$. We considered two temperature measurements $(T1, T2)$ and two humidity measurements $(H1, H2)$ for each instance. In each instance, $T1$ and $H1$ belonged to the first mote, while $T2$ and $H2$ belonged to the second mote of the WSN.

In total, 600*16 normal data samples (data observations or vectors) were generated, and then, several types of sensor faults (hard-over, drift, spike, erratic, data-loss, random, stuck) were infused into the normal (no-fault) data with different fault parameters $(0.2, 0.4, 0.6, 0.8, 1.0)$. In each faulty dataset, $\frac{1}{2}$ of the observations were used to introduce faults in mote 1 measurements, with the other $\frac{1}{2}$ for mote 2, so a realistic scenario could be introduced into the WSN.

Considering five different fault parameters and a normal dataset with seven different faulty datasets, a total of 8*5 datasets were prepared to conduct this research. From each dataset, 400*16 data samples were used to train the classifiers, and 200*16 were used for testing. Following the techniques of multi-class classification, labels were assigned to the normal and faulty observations. The resultant dataset of each fault parameter consisted of 4800*16 samples.

2.5.3 Results

To evaluate the performance of the classifiers in our proposed scheme, we selected three different metrics:

- Accuracy
- Precision
- F1-score/F-measure

It is usually not a good practice in ML to use one metric as an evaluation point. If the classifier does not report numerous sensor faults, the accuracy may still be higher, i.e. high false positives (FP), low true positives (TP). However, the precision will decrease significantly.

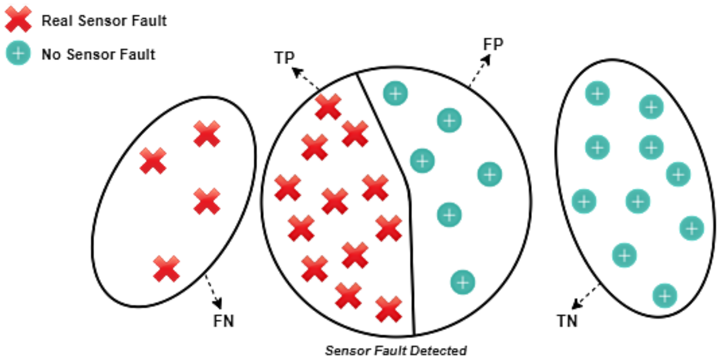


Figure 2.8: Assessment metrics terminology for sensor fault detection.

Thus, it is inadequate to use only accuracy to determine a classifier's performance. To efficiently deal with sensor failures, sensor fault detection and diagnosis systems use some common expressions to classify the data points (data samples) effectively. A typical example shown in Figure 2.8 elaborates on the terminologies generally used in sensor fault detection systems [54].

- *True Positives (TP):* Data points stated as positive (faulty) and are in fact positive.
- *True Negatives (TN):* Data points stated as negative (non-faulty) and are in fact negative.
- *False Positives (FP):* Data points stated as positive (faulty) and are in fact negative (non-faulty).
- *False Negatives (FN):* Data points stated as negative (non-faulty) and are in fact positive (faulty).

The accuracy ratio can be defined as, the number of correct predictions to the total number of predictions, i.e. TP, TN, FP, FN, which can be evaluated with equation 2.8:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.8}$$

where TP and TN refer to the accurately identified data samples, whereas FP and FN are samples incorrectly identified.

Precision can be defined as the measure of correctly identified positive classes from all the positive predicted classes. Thus, when the costs of false positives are high, precision is a better measure, which is defined in equation 2.9:

$$Precision = \frac{TP}{TP + FP} \tag{2.9}$$

where TP is the number of true positives and FP is the number of false positives. TP are the observations accurately identified, according to their respective class, e.g. data samples classified as hard-over faults are, in fact, hard-over faults. FP are the observations inaccurately identified, e.g. data samples classified as hard-over faults are, in fact, normal. Classifiers with the lowest false positives will have a high precision rate.

F1-score/F-measure is the weighted average or harmonic mean of recall and precision. As a statistical measure, this metric is used based on FN and FP to check the performance of the classifier. F1-score is defined with equation 2.10:

$$F1 - score = \frac{2 \times (Recall \times Precision)}{(Recall + Precision)} \tag{2.10}$$

The parameters used in this work are related to datasets generation or ML classifiers. Firstly, the fault parameter is related to the rate of faults (0.2, 0.4, 0.6, 0.8, 1.0) in the datasets. For example, the value 0.2 in the fault parameter corresponds to the lowest fault value while the value 1.0 is the highest one. Secondly, the algorithm parameters (or hyperparameters) are used for training classifiers. The algorithm parameters are selected by exhaustive grid-search approaches. Table 2.2 summarizes the algorithm parameters for training the models used in this research.

In Figure 2.9, SVM performance can be noted on testing data in terms of precision for different fault types and normal data at different fault parameters (0.2, 0.4, 0.6, 0.8,

Table 2.2: Algorithm parameters selected by grid-search for training the models.

Classifiers	**Parameters**

Support Vector Machine	C=1.0
	kernel=rbf
	decision_function_shape=ovr
	gamma=auto

Multilayer Perceptron	activation=relu
	solver=lbfgs
	hidden_layer_sizes=30
	learning_rate=constant

Random Forest	bootstrap=True
	n_estimators=30
	criterion=gini
	max_features=auto
	max_depth=None
	min_samples_split=2

Decision Tree	splitter=best
	criterion=gini
	max_depth=30
	min_samples_split=2

Extra-Trees	bootstrap=False
	n_estimators=30
	criterion=gini
	max_features=auto
	max_depth=None
	min_samples_split=2

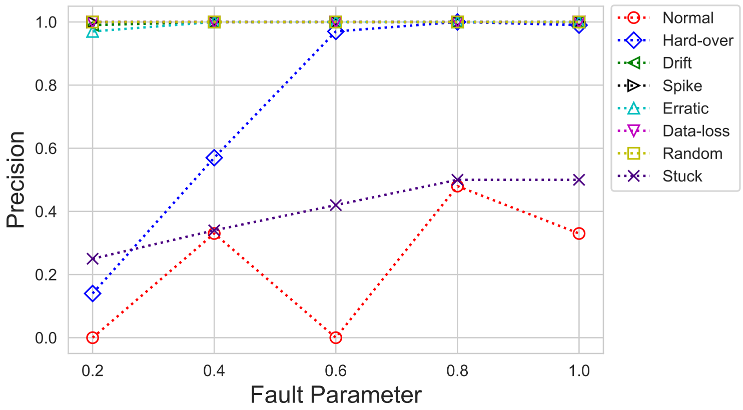


Figure 2.9: Support Vector Machine classifier precision rate vs. fault parameters of distinctive classes.

1.0). After extensive parameter tuning, the best results were obtained using a *radial-basis function (rbf)* kernel with cost parameter $C = 1$ based on *one-versus-rest* manner. The SVM performed well on drift, spike, erratic, data-loss, and random faults. Hard-over faults were gradually caught by the SVM as the fault parameter increased, whereas it showed unusual and poor precision with stuck fault and normal samples. Misclassification occurred between normal samples and stuck fault samples due to high resemblances in the data samples, which makes it difficult for every classifier to differentiate between these two classes.

In Figure 2.10, from among all the classifiers, MLP showed abnormal behavior. For training MLP, there is no concrete rule to declare a specific number of *hidden layers*, and they vary based on each distinct dataset. Extensive tuning of the number of neurons in a hidden layer with *relu* as an activation function at a default *constant* learning rate, we concluded our results. At different fault parameters, unusual oscillations can be noted, caused by slow convergence and network paralysis because MLP uses a backpropagation technique for training.

In Figure 2.11, RF showed tremendous performance on drift, spike, data-loss, and

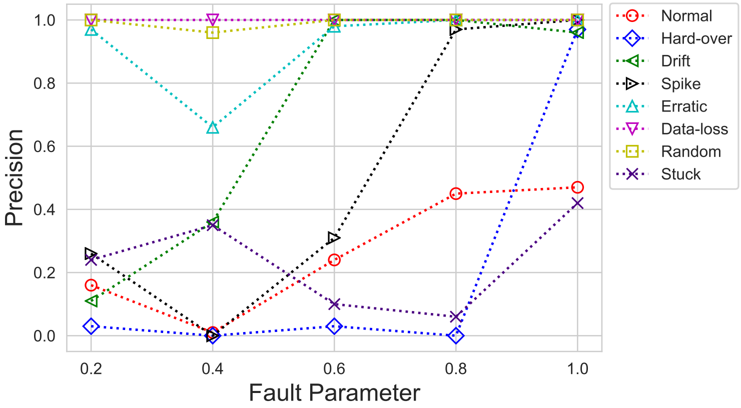


Figure 2.10: Multi-Layer Perceptron classifier precision rate vs. fault parameters of distinctive classes.

random faults, while hard-over fault detection gradually increased as the fault parameter increased. Furthermore, for erratic faults at a 0.8 fault parameter, the classifier reached a maximum 87% precision, whereas RF misclassified normal samples and stuck fault samples. However, optimal learning rates were obtained at *n_estimators=30*, which describes the number of trees in the forest; but RF did not overfit the model on generating enough trees.

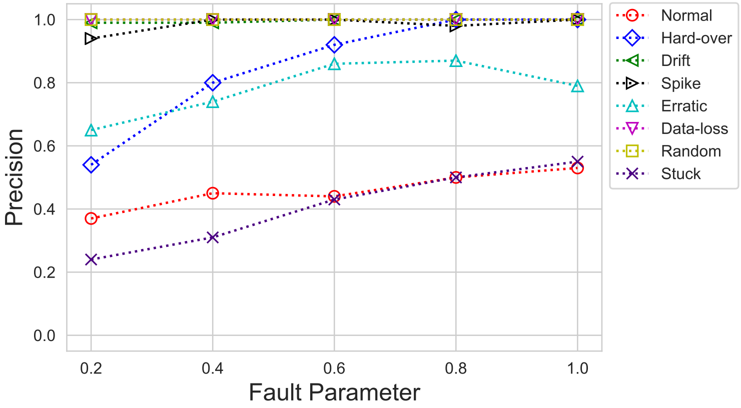


Figure 2.11: Random Forest classifier precision rate vs. fault parameters of distinctive classes.

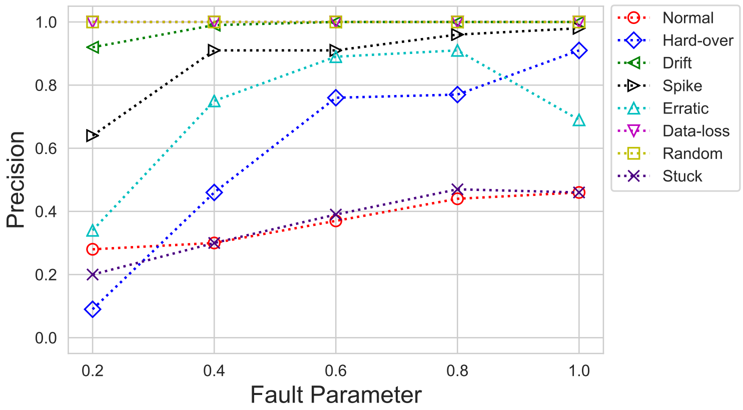


Figure 2.12: Decision Tree classifier precision rate vs. fault parameters of distinctive classes.

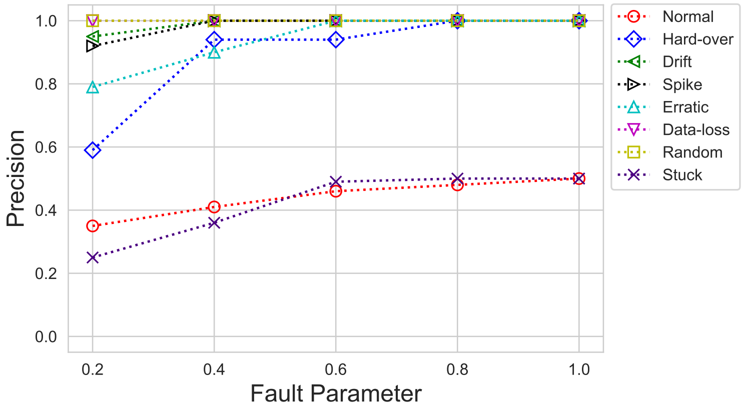


Figure 2.13: Extra-Trees classifier precision rate vs. fault parameters of distinctive classes.

In Figure 2.12, DT performance can be seen, which is the all-purpose ML algorithm, and usually performs well in classification tasks. The DT precision rate drastically changed as the fault parameter increased. We used the default *Gini* criterion to measure the quality of a split. Between *random* and *best* splitter functions, *best* was used. Drift, data-loss, and random faults were classified well, while hard-over and spike fault detection improved as the fault parameters increased. At 1.0, detection of erratic faults went down by up to 69%, showing uncommon behavior in the classifier. Moreover, like other classifiers, DT also misclassified normal and stuck samples, which are, in fact, the most difficult ones for a

Table 2.3: F1-Score of classifiers against eight distinct classes, taking the average from each fault parameter considered. ET had the highest F1-score, on average.

Classes	**Classifiers**				
	ET	*SVM*	*RF*	*MLP*	*DT*
Normal	44.6	0.8	47	36.6	31.6
Hard-over	85.6	54.6	80.8	19.2	59.8
Drift	98	99.4	99	61	92.2
Spike	88.2	81.2	87.2	43.4	72.8
Erratic	93	99.6	84	92.8	70
Data-loss	100	100	100	100	100
Random	100	100	100	99.6	100
Stuck	46	56	39.4	19.4	40.4

classifier to distinguish.

In Figure 2.13, after a fault parameter of 0.4, ET revealed an immense precision rate, more than 90% for six different classes. It reached 100% at 0.8 and 0.1 fault parameters. Normal samples and stuck fault samples were still a problem for the classifier to distinguish. Furthermore, to train ET, we used the same optimal learning rates as RF *(n_estimators=30)*. While generating sufficient trees did not overfit the model, it showed the same behavior as it had for an optimum learning point.

In Table 2.3, the F1-score for five different classifiers are shown against non-faulty and faulty classes. ET had the highest F1-score for all classes, an average of 81.92%, as illustrated in the bar graph (Figure 2.14).

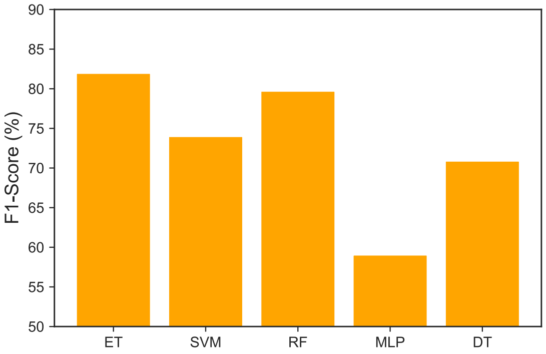
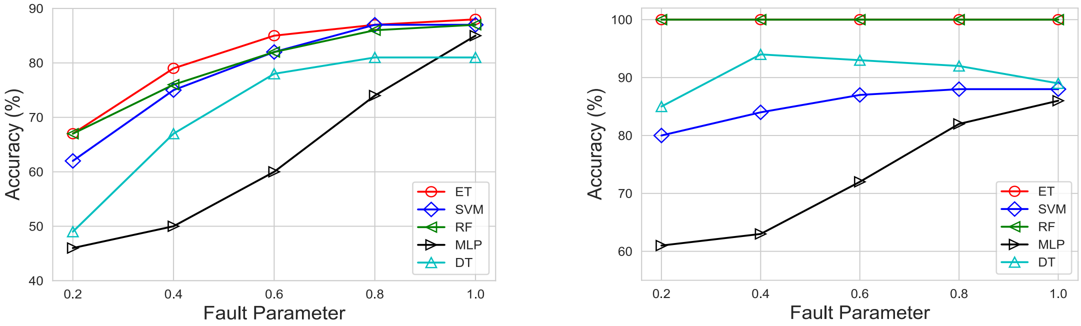


Figure 2.14: F1-Score for ET, SVM, RF, MLP, and DT.



(a)

(b)

Figure 2.15: Classification accuracy comparison of Extra-Trees with state-of-the-art ML approaches on (a) unseen data and (b) seen data.

As illustrated in Figure 2.15, we compared the average classification accuracy of ET, SVM, RF, MLP, and DT for eight differing classes, i.e. normal samples plus hard-over, drift, spike, erratic, data-loss, random, and stuck faults with individual fault parameters, i.e. 0.2, 0.4, 0.6, 0.8, 1.0. We tested the performance of the classifiers on testing data (unseen), as well as training data (seen).

With testing data, MLP and DT did not perform well with low fault parameters,

Table 2.4: Performance comparison between ET and other classifiers on training (seen) and testing (unseen) data.

Classifiers	**Average Classification Accuracy on Unseen-Data**	**Average Classification Accuracy on Seen-Data**
ET	81.20%	100%
SVM	78.60%	85.40%
RF	79.60%	100%
MLP	63%	72.80%
DT	71.20%	90.60%

but as the parameters increased, accuracy rose. MLP significantly improved in its accuracy, whereas DT showed the same behavior on fault parameters of 0.8 and 1.0. SVM and RF accuracy increased with the number of fault parameters, but overall, ET outperformed the rest of the classifiers. As we can see with the performance on the seen data, ET and RF noticeably surpassed SVM, MLP, and DT.

Table 2.4 presents a performance comparison of the models with both training and testing data. ET clearly outperformed the rest of the classifiers, while RF showed competitive performance but did not surpass ET accuracy. SVM, MLP, and DT did not perform well in terms of diagnosing the desired non-faulty and faulty classes, even on the seen data.

Figure 2.16 illustrates the training time taken by the classifiers on each dataset. We observed that DT can achieve the lowest training time against other algorithms, but at the cost of low performance. ET achieved the second-best training time, and also performed tremendously well in terms of accuracy. It is easy to highly recommend the ET classifier for lightweight systems, which can achieve low computational power as well as high accuracy compared to state-of-the-art ML approaches.

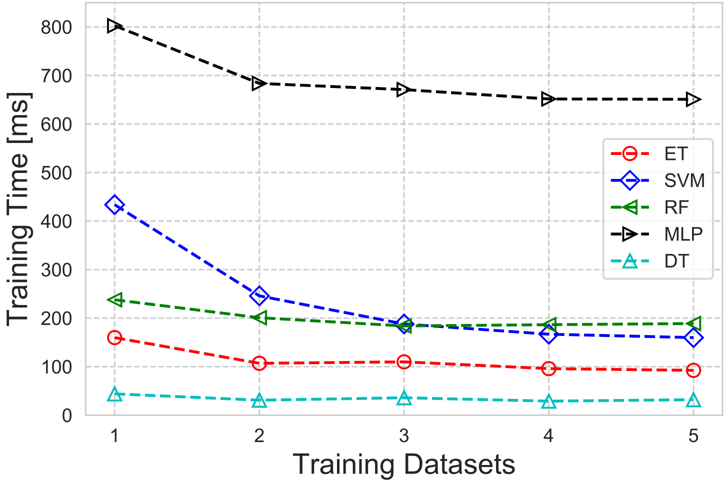


Figure 2.16: Computational training time vs. different training datasets for the proposed Extra-Trees classifier against state-of-the-art ML classifiers.

2.6 Conclusions

In this work, we proposed an Extra-Trees-based diagnostic scheme to detect and diagnose faults in a timely fashion for WSN. In addition, we compared the performances of the proposed scheme with those of several ML classification algorithms such as SVM, RF, MLP, and DT. For performance analysis, we used an online dataset that is trusted and has been used by others in the research community. The dataset, which was published by researchers at the University of North Carolina, contains sensor measurements collected by single-hop and multi-hop networks. Several faults commonly observed in WSN are hard-over, drift, spike, erratic, data-loss, random, and stuck faults. Subsequently, these faults were injected into the real dataset for experimental purposes. The simulation results showed that the proposed ET-based diagnostic scheme outperformed other ML approaches, in terms of performance metrics such as accuracy, precision, and F1-score. It is also observed that the proposed scheme has lower training time than other ML approaches. Therefore, the prospective usages include scrutinizing the behavior of sensors through their data for the timely detection and diagnosis of faults in WSN.

Chapter 3

CAFD: Context-Aware Fault Diagnostic Scheme towards Sensor Faults utilizing Machine Learning[1]

3.1 Introduction

The Cyber-Physical Systems (CPS) such as Industrial Control System (ICS), Smart Grids (SG), and Wireless Sensor Networks (WSN) often consist of hundreds of sensors that may be deployed in relatively harsh and complex environments. Natural factors, electromagnetic interference, and system defect can affect the performance of the sensors. When the sensor becomes faulty, it may completely stop generating signals or produce incorrect signals. In some cases, it can unstably jump between normal and faulty states. To improve safety, reliability, shorten response time, strengthen network security, and prolong network lifespan, many studies have focused on sensor fault detection. These faults or

[1] The study in this chapter is to be submitted.

anomalies can be expressed as an unusual property or behavior of a system [10].

Studies have been carried out mainly since the 1980s for the detection and diagnosis of defects in industrial facilities, i.e., physical-based or mathematical. These approaches were limited to specific environments and conditions. It is difficult to determine extreme model parameters due to system complexities. To overcome these limitations, data-driven approaches using Machine Learning (ML) techniques have been proposed lately, which analyses data to develop the best-fit models. These models in fact use historical data to find hidden patterns and identify expected outcomes. As modern systems are becoming complex, previous approaches are becoming difficult to implement. On the other hand, the data-driven approach can be developed to adequately approximate real systems based on the collected data.

In recent times, machine learning classification techniques such as Support Vector Machine (SVM) [33] and Neural Networks (NN) [32] have gain eminence in fault detection and diagnostic systems. In the past, algorithms for fault detection and diagnosis in rolling elements of machines have been explored in a vast number of studies reporting efficient results [11], [33], [12], [18]. However, sensors as well can be defective frequently leading to serious consequences in terms of safety, economy, and operation. The techniques used for bearing fault detection and sensor fault detection are homogeneous, however, the signal characteristics of sensor faults are different from the rolling elements. Therefore, using similar approaches for both does not guarantee the same accuracy in results. Precise detection and diagnosis of sensor anomalies in a timely manner is extremely important to ensure the safety and reliability of systems.

A typical sensor along with the data flow through major components is shown in Figure 3.1. These major components are primarily responsible for the abnormal behavior of sensors. Each component is associated with certain static limiting properties, which can be

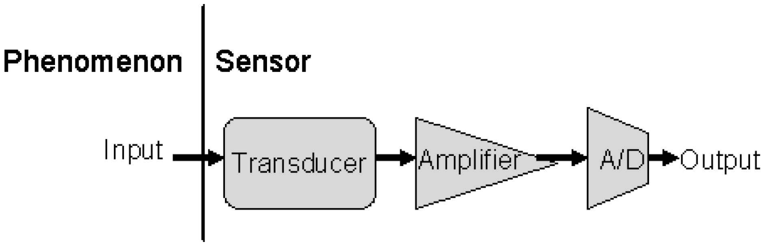


Figure 3.1: Illustration of sensor and its key components.

described by specifications and may affect the resulting data. However, the sensor's output can also be affected by the external environment such as communication or battery defect, which are common to occur in WSN.

Figure 3.2 represents the abnormal behaviors (or faults) in sensor according to its context. These contexts can be explained as the internal or external environment of sensor, which are accountable for the abnormal behaviours. As shown in Figure 3.2, there can be a single or multiple causes for a fault to arise. For instance, drift and hard-over fault is primarily caused by the calibration defect (or error), while data-loss fault is caused by either calibration or hardware defect. Moreover, spike fault can appear as a consequence of hardware, communication, or battery defect. The erratic fault is the result of battery defect, whereas stuck-at fault is triggered by several causes, such as hardware, communication, battery, and clipping defect. Further detail about drift, hard-over, data-loss, spike, erratic, and stuck-at fault is given in Section 2.2.

In this chapter, a new Context-Aware Fault Diagnostic (CAFD) scheme towards the detection and diagnosis of fault or anomaly in sensors has been employed. First, the data under consideration is multi-labelled according to the context of faults. Then, these data samples are given as an input to the context-based ML classifier to diagnose. Upon diagnosing (or classification), the output of the context-based classifier along with the original data samples (or sensor output signal) are given as an input to the fault-based ML

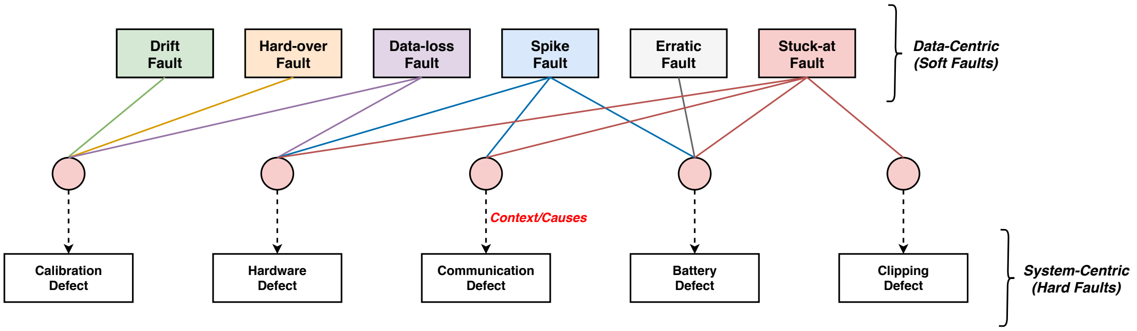


Figure 3.2: Representation of sensor abnormal behaviours (or faults) according to their context.

classifier. Finally, fault-based classifier categorizes the data samples to detect and diagnose any abnormal behaviour occurred in the network. The proposed scheme is discussed in detail in Section 3.3.

The rest of this chapter is organized as follows. Section 2.2 presented the taxonomy of faults in sensors. A brief introduction towards the machine learning classifiers and classification techniques under consideration is given in Section 3.2. In Section 3.3, the proposed CAFD scheme is discussed. Section 3.4 provides simulations and results. Finally, Section 3.5 concludes the chapter.

3.2 Machine Learning Classifiers and Classification Techniques

Classification is a supervised machine learning approach, which can be defined as a means of categorizing some unknown items into a discrete set of classes. In this work, multi-label and multi-class classification approaches are used, which identifies the hidden patterns between normal and faulty states. The classification algorithms and approaches used in this work are explained as follows.

3.2.1 Classification Algorithms

3.2.1.1 Support Vector Machine

Developed in the 1970s, SVM deals with the concept of statistical learning theory. In the field of machine learning, precisely for fault detection and diagnosis, SVM is one of the well-known algorithms [8]. Linear line or hyper-plane is generated as a decision boundary for classification tasks between data points of the distinct classes. The nearest data points to the hyper-plane, which impart construction of the hyper-plane are called support vectors (as presented in Figure 3.3). The optimized hyper-plane can be mathematically expressed by equation 3.1:

$$w^T x + b = 0 \tag{3.1}$$

where w is the vector of weights, x is an input vector, and b represents the bias.

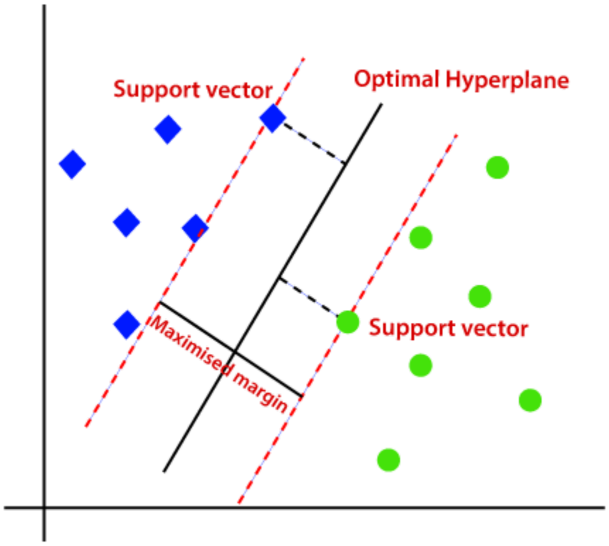


Figure 3.3: Illustration of Support Vector Machine Classifier.

The equation of the support vectors of each class is given as

$$\begin{aligned} w^T x + b &= +1, \quad for \quad d_i = +1 \\ w^T x + b &= -1, \quad for \quad d_i = -1 \end{aligned} \tag{3.2}$$

where di corresponds to the respective class, i.e., $di = +1$ for class A, and $di = -1$ for class B. In this research, multi-label as well as multi-class SVM-based classifier is used to analyze the results for sensor fault classification. The cost parameter C was set to default ($C = 1$).

3.2.1.2 Artificial Neural Network

A class of feed-forward Artificial Neural Network (ANN), Multi-Layer Perceptron (MLP) consists of, at least, three layers of nodes: an input layer, hidden layer, and an output layer. Each node is a neuron that uses a nonlinear activation function except for the input nodes. For training, MLP utilizes supervised learning technique known as backpropagation. An instance of MLP is shown in Figure 3.4.

The number of nodes in hidden layers of ANN can be decided according to the nature of data. There are no rules defined to declare a specific number of layers. Large number of nodes can over-fit the training data, while fewer nodes can lead towards under-

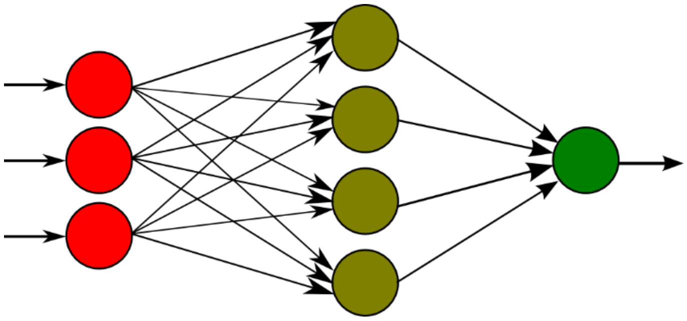


Figure 3.4: Illustration of Feed-forward Neural Network.

fitting. In both cases, the classifier will not be able to categorize the data accurately. An optimal number of hidden layers, as well as nodes, shall be chosen to minimize the misclassification.

3.2.1.3 Extremely Randomized Trees

Extremely Randomized Trees or Extra-Trees (ET) algorithm is generally used for classification and regression tasks. A large number of integrated Decision Trees (DT) creates an ensemble approach known as ET. As shown in Figure 3.5, ET algorithm operates on number of DT, where each DT is composed of root node, child nodes, and leaf nodes. Given a data point x, ET selects a split rule based on a random subset of features and a partially random cut point at the root node. This phenomena makes the process faster and exhibits strong reduction towards bias and variance error.

The concept behind the ensemble technique is to generate several DT models and make a judgement based on the association of it. This approach is based on principal 'wisdom of the crowd', which eventually results in better performance compared to a single DT model. The ET algorithm is explained in detail in Section 2.4.

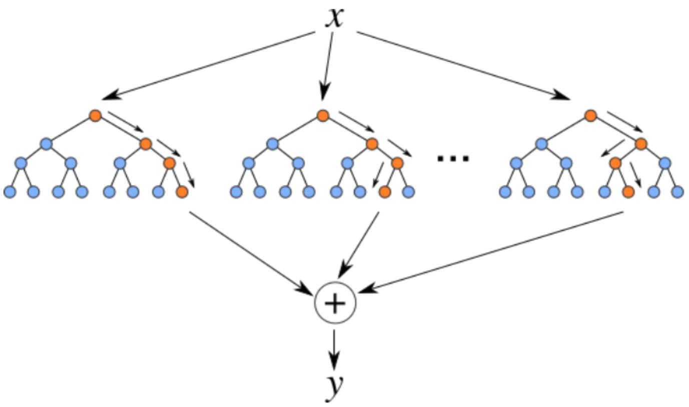


Figure 3.5: Illustration of Extremely Randomized Trees (an ensemble of decision trees).

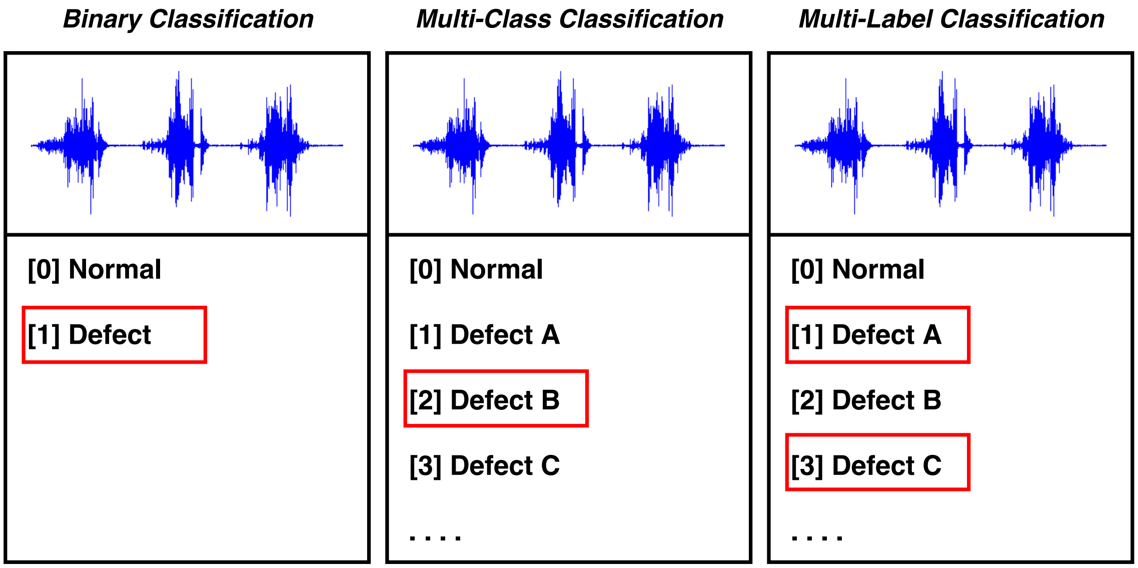


Figure 3.6: Types of classification.

3.2.2 Classification Techniques

An instance of supervised learning, classification is a technique to identify data observations according to where it belonged on the basis of training data. Classification is primarily divided into three categories: binary, multi-class, and multi-label. Figure 3.6 exemplifies the types of classification.

- **Binary Classification** involves two classes. A set of data observations (or data sample) can only be assigned to one of two classes. For instance, in the case of sensor fault detection, data samples are categorized to either normal or abnormal class.

- **Multi-Class Classification** problem comprises more than two classes, which are mutually exclusive. A single data observation can belong to only one class. For example, sensor output signal may belong to multiple classes such as normal, drift fault, hard-over fault and so forth. Nevertheless, the output of multi-class classifier can only belong to one class of the target variables.

- **Multi-Label Classification** is contrary to multi-class classification. In case of multi-label, a single data observation can concurrently belong to two or more classes of the target variables. For instance, in the circumstances of stuck-at sensor fault, a data sample under observation can simultaneously belong to multiple defective classes (as shown in Figure 3.2).

3.3 Proposed CAFD Scheme

The term *context* in the Context-Aware Fault Diagnostic (CAFD) scheme refers towards the interior or exterior conditions (or environment) of sensors, whereas *aware* terminology relates to the conscious intelligent ML algorithm. The idea behind CAFD system is to utilize the context of sensors, which are primarily responsible for the occurring abnormal behaviours (anomalies or faults).

In Figure 3.2, the data-centric or soft faults (i.e., drift, hard-over, data-loss, spike, erratic, and stuck-at faults) are represented by lines to their context, respectively. The system-centric or hard defects, (i.e., calibration, hardware, communication, battery, and clipping defects) which can be declared as the context of sensors, are the prime causes of data-centric sensor faults. The framework of the proposed CAFD system is presented in Figure 3.7.

Following data acquisition and preparation explained in Section 3.4.1, the data samples are given as an input to the ML classifier for training. Firstly, each data sample is labelled according to the corresponding context. For instance, hard-over fault samples are labelled 1 for calibration, while 0 for rest of the contexts. Since hard-over fault substantially transpire due to calibration. Table 3.1 explicates the labels for each class under consideration in this work.

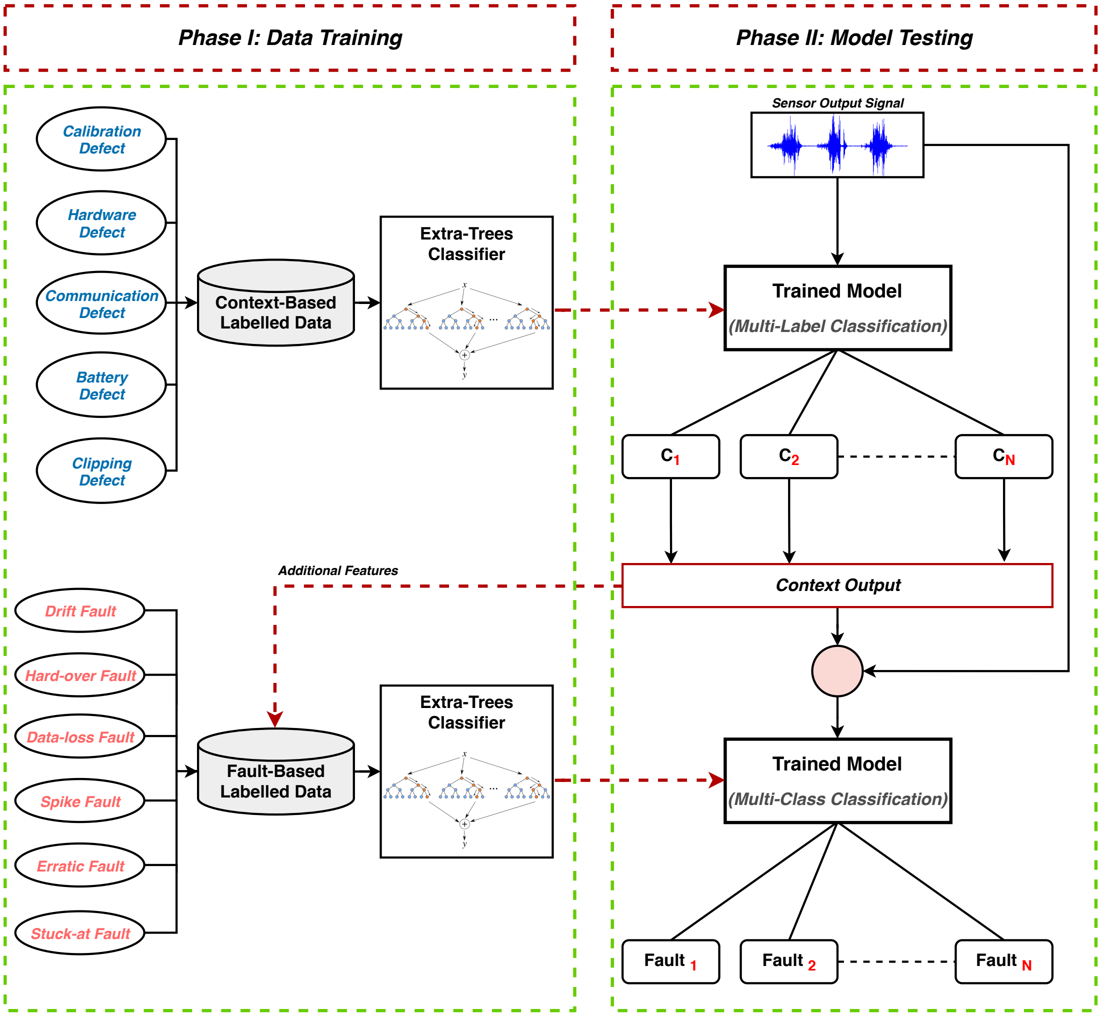


Figure 3.7: Framework of the proposed Context-Aware Fault Diagnostic (CAFD) scheme.

Furthermore, the context-based multi-label data is first trained using Extra-Trees algorithm for classification purposes. Subsequently, the context-based classifier (Extra-Trees) were given distinct sensor output signals or data samples to identify. This technique classified each sample according to its context $(C_1, C_2, ...C_n)$, which belonged to any one of the aforementioned contexts such as calibration.

Afterwards, the output of context-based classifier in the form of labels were utilized as an input features in the fault-based classifier. The fault-based multi-class data is consisted of legitimate and faulty data samples along with the additional features (or data points) from the context-based classifier. Further, the fault-based multi-class data, labelled with normal and data-centric faulty classes as above-mentioned, was trained by ET classifier.

To detect and diagnose the data-centric faults, the fault-based classifier were given the sensor output signals S_n in the form of data observations. The final classification was performed by fault-based classifier (Extra-Trees), which led towards the diagnostics of faults in sensors.

3.4 Simulations and Results

3.4.1 Data Acquisition and Preparation

To evaluate the performance of ML classifiers, the data under consideration plays an essential role. Unfortunately, it is ideal to get data with genuine faults obtained from realistic scenarios. There are no publicly available datasets, which in fact addresses all the faults in sensor. Therefore, to conduct this research, we acquired a dataset (healthy measurements), which is published by the researchers of the University of North Carolina [53]. This dataset is composed of temperature and humidity sensor measurements. The data was acquired using Telos B motes, while creating single-hop/multi-hop WSN scenario. For our research, we first obtained the multi-hop data (healthy state), and then injected it with the six diverse faults (such as hard-over, drift, spike, erratic, data-loss, and stuck faults). This approach is common among researchers to obtain faulty datasets [8] [15] [30].

To prepare the dataset, we generated 16-dimensional data samples (measurements or vectors). Each sample was composed of 16 data points in 4 successive instances ($t0, t1, t2, t3$). Each instance was constructed from 2 temperature sensor measurements ($T1, T2$), and 2 humidity sensor measurements ($H1, H2$). In each instance, $T1$ and $H1$ measurements

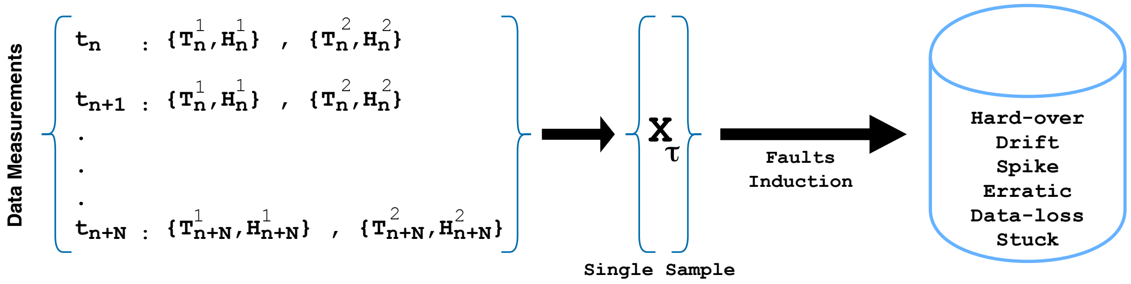

Data Measurements
t_n : $\{T^1_n, H^1_n\}$, $\{T^2_n, H^2_n\}$
t_{n+1} : $\{T^1_n, H^1_n\}$, $\{T^2_n, H^2_n\}$
.
.
.
t_{n+N} : $\{T^1_{n+N}, H^1_{n+N}\}$, $\{T^2_{n+N}, H^2_{n+N}\}$
X_τ
Single Sample
Faults Induction
Hard-over
Drift
Spike
Erratic
Data-loss
Stuck

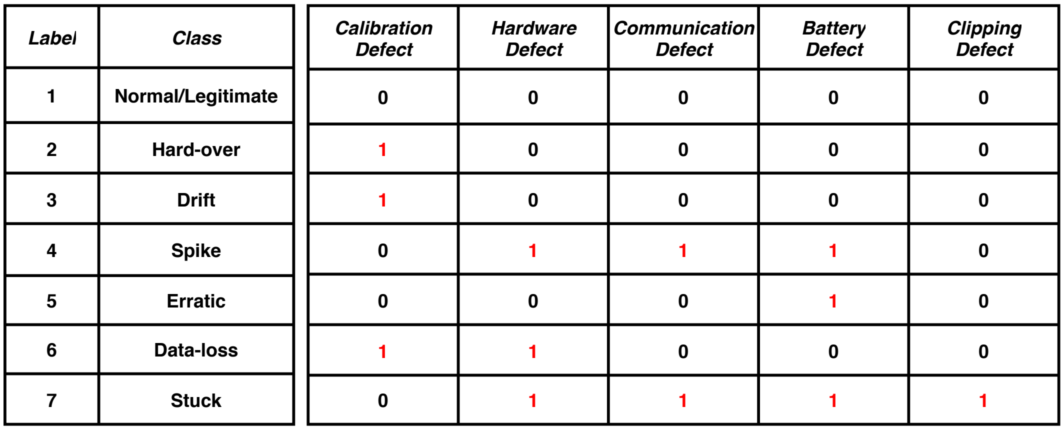
Figure 3.8: Illustration of data preparation.

belonged to the first node, whereas $T2$ and $H2$ to the second node of the multi-hop scenario.

Overall, $400 * 16$ normal (legitimate/healthy) data points or observations were initiated. Afterwards, above-mentioned six distinct sensor fault types were injected in the normal (non-faulty) data via simulations using equation 2.3, 2.4, 2.5, 2.6, 2.7. Figure 3.8 illustrates the data wrangling process. To replicate realistic scenario of WSN, half of the data points were used to introduce faults in the first node, while another half in the second node of the multi-hop network. Some of the faults (such as hard-over, drift, spike, erratic) were induced with different intensity of fault $(0.1, 0.2 \ldots, 1.0)$, whereas, in case of data-loss and stuck fault, the sensor's output is either null, or unchanging constant value. The higher the fault intensity value, the higher the rate of fault in the data. For instance, the value 0.1 in the fault intensity corresponds to the lowest rate of fault, while 1.0 is the highest. However, the data-loss and stuck fault samples remained unchanged throughout. Intuitively, the accuracy of classifier improves with the increase in fault intensity.

Considering normal class (or data) and the six above mentioned faulty classes, the final dataset was composed of $7 * 400 * 16$ data points. In each class, 60% of the data samples were used to train ML classifier, whereas 40% for testing. In this work, two different labelling techniques were used to classify data (i.e. multi-class classification, multi-label classification), as explained in Section 3.2.2. In the case of multi-class, a single column of

Table 3.1: Representation of labels for each class according to the context.



Label	*Class*	*Calibration Defect*	*Hardware Defect*	*Communication Defect*	*Battery Defect*	*Clipping Defect*
1	Normal/Legitimate	0	0	0	0	0
2	Hard-over	1	0	0	0	0
3	Drift	1	0	0	0	0
4	Spike	0	1	1	1	0
5	Erratic	0	0	0	1	0
6	Data-loss	1	1	0	0	0
7	Stuck	0	1	1	1	1

labels was introduced, while for multi-label, five distinct columns of binary numbers were taken into consideration according to the context of each class. Table 3.1 shows the labels in terms of numerical value for each class.

3.4.2 Results

To perform the experiments, all the algorithms under consideration in this work were simulated in Python using Scikit-learn and NumPy libraries. The Grid-Search Cross-Validation (CV) technique with $CV = 5$ was used on the dataset to obtain the optimal hyperparameters for each algorithm to train. This technique works on the principles of *fit* and *score* method in order to determine the best parameters, which can be used to train the ML models.

Generally using a single performance evaluation metric for ML models is not considered a good practice. In this work, three distinct metrics were taken into consideration to assess the performance of the classification algorithms. These metrics are defined as follows.

- **Diagnostic Accuracy** can be defined as the ratio of correctly identified faulty or defective data samples to the total number of defective samples.

$$Diagnostic\ Accuracy = \frac{Number\ of\ correctly\ identified\ defective\ samples}{Total\ number\ of\ defective\ samples} \tag{3.3}$$

- **F1-Score** or F-Measure is the harmonic mean of recall and precision. This weighted average is commonly used to assess the performance of ML classification models.

$$F1 - Score = 2 \times \left(\frac{Recall \times Precision}{Recall + Precision} \right) \tag{3.4}$$

where,

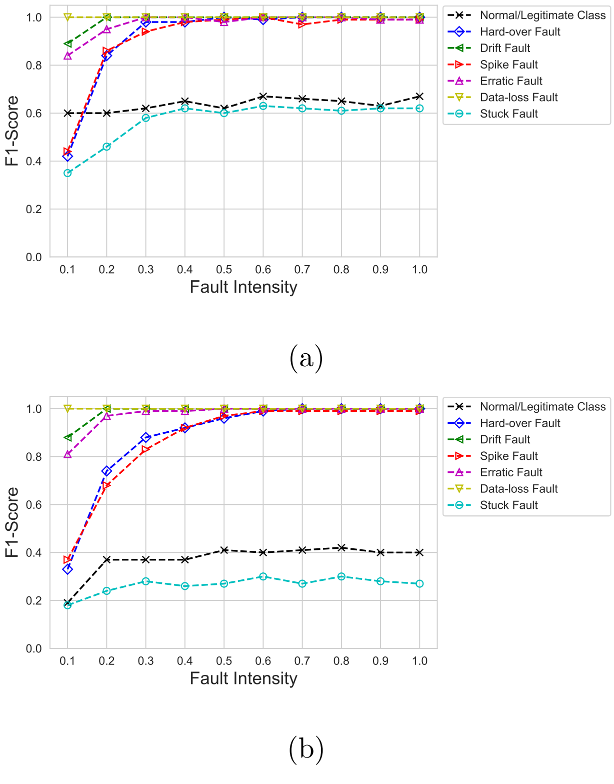
$$Recall = \frac{True\ Positives}{Actual\ Positives} \quad , \tag{3.5}$$

$$Precision = \frac{True\ Positives}{Predicted\ Positives} \tag{3.6}$$

- **Area value under the ROC Curve (ROC-AUC)** is an evaluation metric, which calculates a scalar value in the range of $[0, 1]$. This measure determines how accurately the ML classifier can distinguish between faulty and non-faulty data observations. An accurate classifier can have the $ROC - AUC$ value up to 1.0.

In this work, without-context (or traditional) approach can be simply distinguished from the context-aware approach as the technique, where sensor output signals are given in its genuine state to the ML classifier without considering the additional features extracted through multi-label classification techniques. Furthermore, fault intensity depicts the rate of fault injected in the datasets. For instance, 0.1 fault intensity corresponds to the lowest fault rate, whereas 1.0 is the highest. As the fault intensity increases, the performance of the classifier is also expected to increase.

ET performance can be noted in terms of F1-score for normal and distinct fault classes at various fault intensities in Figure 3.9. Context-aware approach has revealed the ability to distinguish between different set of classes precisely compared to without context approach. However, normal and stuck fault class have highly identical data points, which makes it hard for the classifier to discriminate in both cases.



(a)

(b)

Figure 3.9: Extra-Trees F1-score comparison of individual class on (a) context-aware approach vs. (b) without context.

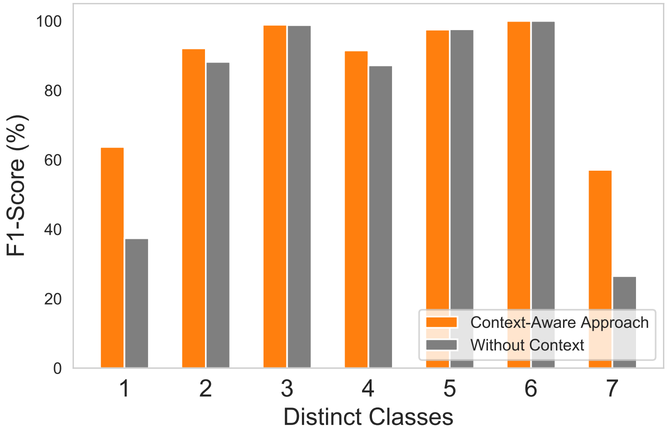


Figure 3.10: F1-score of context-aware vs. without context approach for Extra-Trees on distinctive classes.

Nevertheless, the bar graph displayed in Figure 3.10 explicitly provides the performance differences between the two approaches in terms of F1-score average. Each number in the graph depicts different class: (1) Normal, (2) Hard-over, (3) Drift, (4) Spike, (5) Erratic, (6) Data-loss, and (7) stuck fault. While some of the classes have shown somewhat similar F1-score on both approaches, most of them were improved on context-aware approach.

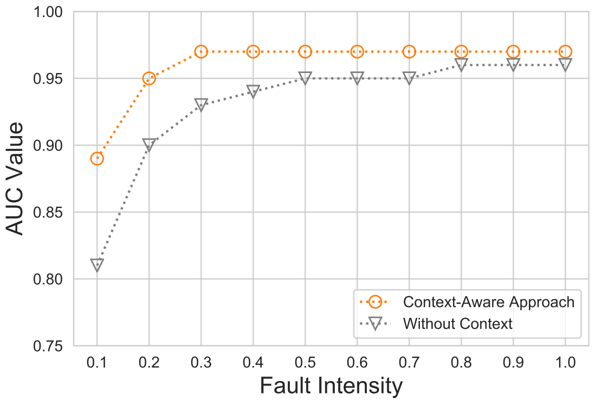


Figure 3.11: ROC-AUC against diverse fault intensity of the proposed ET-based context-aware approach compared with traditional approach.

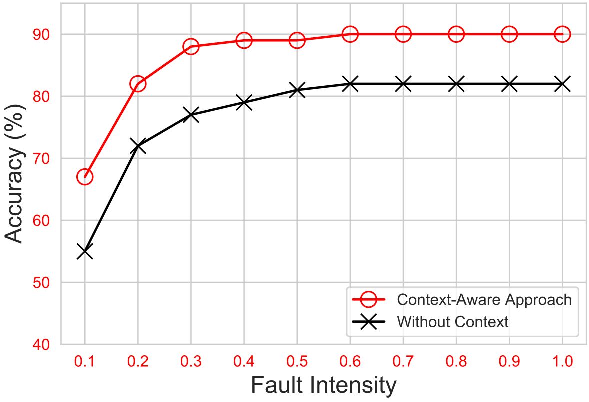


Figure 3.12: Diagnostic accuracy of context-aware vs. without context approach for Extra-Trees on diverse fault intensity.

In Figure 3.11, the ROC-AUC of ET versus different fault intensity of the proposed scheme is revealed. Starting from the lowest fault intensity 0.1 up to 0.3, the ET-based context-aware approach AUC value considerably increased. The lowest AUC value noticed was 0.89. However, from 0.3 to 1.0, proposed approach constantly achieved maximum AUC value up to 0.97. On the other hand, same classifier (ET) in without context approach with the identical hyperparameters (as context-aware approach) revealed the performance as low up to 0.81, whereas with the increase in fault intensity, ROC-AUC also elevated.

Figure 3.12 discloses the average diagnostic accuracy of the approaches under consideration. As can be seen, both approaches significantly improved with the rise in fault intensity. Nonetheless, utilizing ET classifier, the proposed context-aware approach achieved maximum accuracy up to 90%, whereas the contrary approach had utmost diagnostic accuracy up to 81%.

Furthermore, we also tested the proposed approach on other state-of-the-art ML algorithms such as SVM and NN and compared the performance with traditional approach. As shown in Table 3.2, exploiting the proposed CAFD approach, ET provides the best

Table 3.2: Performance comparison between ET and state-of-the-art ML classifiers in terms of average classification accuracy.

Classifiers	**Context-Aware Approach**	**Without Context Approach**
ET	**86.5%**	77.4%
SVM	83.1%	81.4%
NN	80.4%	78.1%

performance on average compared to other classifiers. Additionally, the diagnostic accuracy of each classifier on different fault intensity is presented in Figure 3.13. Indubitably, ET achieves the highest diagnostic accuracy under the proposed scheme.

Finally, the time taken by each classifier to train on the number of training samples is illustrated in Figure 3.14. We observed that, ET is computationally inexpensive compared to SVM and NN. It is easy to state, utilizing ET for the lightweight systems under the

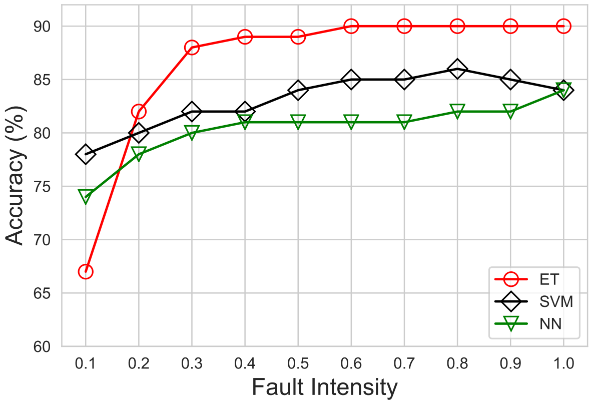


Figure 3.13: Accuracy comparison of ET with state-of-the-art approaches on the proposed CAFD scheme.

proposed CAFD scheme can achieve high performance by precisely detecting and diagnosing sensor faults.

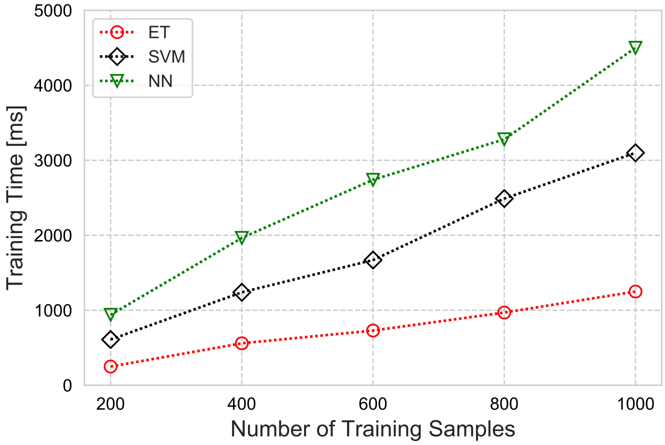


Figure 3.14: Training time against number of training samples for distinct ML classifiers.

3.5 Conclusions

In this work, a lightweight CAFD scheme is proposed for the timely detection and diagnosis of low intensity faults in sensors. First, a dataset composed of healthy temperature and humidity sensor measurements was acquired. Afterwards, the commonly occurred faults in sensors (i.e., hard-over, drift, spike, erratic, data-loss, and stuck) were injected with different intensity into the healthy dataset in order to generate realistic defective WSN scenario. Healthy and faulty data observations were labelled utilizing multi-label/multi-class classification techniques for experimental purposes. These data observations were then used to train ML classifiers. An extensive simulation study revealed that, using the context of sensors as additional features in the original data observations can significantly improve the classifiers' performance. Furthermore, the proposed ET classifier in the CAFD scheme has shown efficiency over SVM and NN in terms of diagnostic accuracy and training time.

Chapter 4

Summary of Contributions and Future Works

4.1 Introduction

This chapter provides the contribution of this dissertation. The problem statement, objective, methodologies, and results carried out by the proposed solutions are presented in chapter 2 and 3. The first section 4.2 of the current chapter summarizes the primary contributions of those investigations, whereas the outline of the future direction is given in section 4.3.

4.2 Summary of Contributions

This dissertation investigated and addressed problems related to the abnormal behaviors originated in the domains of Wireless Sensor Network. Efforts were made to timely detect and diagnose the low-intensity abnormal behaviors utilizing data-driven supervised machine learning-based techniques. The contribution of this dissertation, in the context of machine learning-based effective detection and diagnosis of sensors abnormal behaviors in the WSN is outlined as follows.

- Most of the abnormal behaviors or faults that commonly occur in WSN are considered: hard-over, drift, spike, erratic, data-loss, stuck, and random fault.
- Dataset composed of temperature and humidity sensor measurements of multi-hop scenario was acquired and the aforementioned faults were injected into the healthy dataset at distinct intensities.
- The intensity of fault (or fault parameter) is simulated in a way to replicate realistic scenarios of WSN. For instance, fault can occur in WSN for a short length as well as long, or it can be occurred in a combination of both.
- Following that, Extremely Randomized Trees-based detection and diagnostic scheme was proposed, which is an ensemble-based machine learning algorithm. Additionally, multiple classification techniques such as multi-label/multi-class were utilized to generate a context-aware diagnostic system for the timely detection and diagnosis of faults with high precision.
- Furthermore, an extensive simulation experiments are conducted on the prepared datasets to demonstrate the efficiency of the proposed schemes. Four different machine learning algorithms such as support vector machine, neural network, random forest, and decision tree are compared with the proposed classifier.
- Finally, the performance of the proposed technique is evaluated by widely used measures such as accuracy, F1-score, precision, and area value under the ROC curve. These performance evaluations revealed the efficiency of the proposed scheme to detect and diagnose the abnormal behaviors in a timely fashion.

- In addition, the training time taken by distinct classifiers were calculated in order to describe the lightweight measure.

4.3 Future Direction

In the future, it is aimed to work on further improvement of the abnormal behavior detection and diagnostic systems. Identification of abnormal behaviors precisely at the node level is needed alternatively to the central node. Furthermore, the robustness of the proposed schemes will be verified by deploying a large number of sensors in the network, while focusing on extreme low-intensity faults. Finally, it is intended to generate a prognostic system, which will help to estimate the remaining useful life of sensors in the network.

Publications

International Journal

[1] **Umer Saeed**, Sana Ullah Jan, Young-Doo Lee, and Insoo Koo. “Fault diagnosis based on extremely randomized trees in wireless sensor networks.”, *Reliability Engineering and System Safety* (2020): 107284.

Journal Paper Draft under Preparation

[2] **Umer Saeed**, Young-Doo Lee, Sana Ullah Jan, and Insoo Koo. “CAFD: Context-Aware Fault Diagnostic Scheme towards Sensor Faults utilizing Machine Learning.”, *to be submitted.*

International Conference

[3] **Umer Saeed**, Sana Ullah Jan, Young-Doo Lee, and Insoo Koo. “Machine Learning-based Real-Time Sensor Drift Fault Detection using Raspberry Pi.”, *In 2020 International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1-7. IEEE, 2020.

[4] Jan, Sana Ullah, **Umer Saeed**, and Insoo Koo. “Machine Learning for Detecting Drift Fault of Sensors in Cyber-Physical Systems.”, *In 2020 17th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pp. 389-394. IEEE, 2020.

Bibliography

[1] U. Saeed, S. U. Jan, Y.-D. Lee, and I. Koo, "Fault diagnosis based on extremely randomized trees in wireless sensor networks," *Reliability Engineering & System Safety*, p. 107284, 2020.

[2] D. Puccinelli and M. Haenggi, "Wireless sensor networks: applications and challenges of ubiquitous sensing," *IEEE Circuits and systems magazine*, vol. 5, no. 3, pp. 19–31, 2005.

[3] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[4] G. P. Fettweis, "The tactile internet: Applications and challenges," *IEEE Vehicular Technology Magazine*, vol. 9, no. 1, pp. 64–70, 2014.

[5] S. K. Khaitan and J. D. McCalley, "Design techniques and applications of cyberphysical systems: A survey," *IEEE Systems Journal*, vol. 9, no. 2, pp. 350–365, 2014.

[6] S. U. Jan, U. Saeed, and I. Koo, "Machine learning for detecting drift fault of sensors in cyber-physical systems," in *2020 17th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*. IEEE, 2020, pp. 389–394.

[7] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, “A comparative study of lpwan technologies for large-scale iot deployment,” *ICT express*, vol. 5, no. 1, pp. 1–7, 2019.

[8] S. U. Jan, Y.-D. Lee, J. Shin, and I. Koo, “Sensor fault classification based on support vector machine and statistical time-domain features,” *IEEE Access*, vol. 5, pp. 8682–8690, 2017.

[9] Z. Zhang, A. Mehmood, L. Shu, Z. Huo, Y. Zhang, and M. Mukherjee, “A survey on fault diagnosis in wireless sensor networks,” *IEEE Access*, vol. 6, pp. 11 349–11 364, 2018.

[10] Z. Gao, C. Cecati, and S. X. Ding, “A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.

[11] J. Tian, C. Morillo, M. H. Azarian, and M. Pecht, “Motor bearing fault detection using spectral kurtosis-based feature extraction coupled with k-nearest neighbor distance analysis,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, pp. 1793–1803, 2015.

[12] B. Samanta and K. Al-Balushi, “Artificial neural network based fault diagnostics of rolling element bearings using time-domain features,” *Mechanical systems and signal processing*, vol. 17, no. 2, pp. 317–328, 2003.

[13] H. Jeong, B. Park, S. Park, H. Min, and S. Lee, “Fault detection and identification method using observer-based residuals,” *Reliability Engineering & System Safety*, vol. 184, pp. 27–40, 2019.

[14] W. Ahmad, S. A. Khan, M. M. Islam, and J.-M. Kim, “A reliable technique for

remaining useful life estimation of rolling element bearings using dynamic regression models," *Reliability Engineering & System Safety*, vol. 184, pp. 67–76, 2019.

[15] S. Zidi, T. Moulahi, and B. Alaya, "Fault detection in wireless sensor networks through svm classifier," *IEEE Sensors Journal*, vol. 18, no. 1, pp. 340–347, 2017.

[16] C. Titouna, M. Aliouat, and M. Gueroui, "Outlier detection approach using bayes classifiers in wireless sensor networks," *Wireless Personal Communications*, vol. 85, no. 3, pp. 1009–1023, 2015.

[17] G.-X. Zhong and G.-H. Yang, "Fault detection for discrete-time switched systems with sensor stuck faults and servo inputs," *ISA transactions*, vol. 58, pp. 196–205, 2015.

[18] Y. Wang, J. Xiang, R. Markert, and M. Liang, "Spectral kurtosis for fault detection, diagnosis and prognostics of rotating machines: A review with applications," *Mechanical Systems and Signal Processing*, vol. 66, pp. 679–698, 2016.

[19] R. R. Swain, P. M. Khilar, and S. K. Bhoi, "Underlying and persistence fault diagnosis in wireless sensor networks using majority neighbors co-ordination approach," *Wireless Personal Communications*, vol. 111, no. 2, pp. 763–798, 2020.

[20] S. K. Bhoi, M. S. Obaidat, D. Puthal, M. Singh, and K.-F. Hsiao, "Software defined network based fault detection in industrial wireless sensor networks," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.

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

[22] S. K. Bhoi and P. M. Khilar, "Self soft fault detection based routing protocol for vehicular ad hoc network in city environment," *Wireless Networks*, vol. 22, no. 1, pp. 285–305, 2016.

[23] J. K. Rout, S. K. Bhoi, and S. K. Panda, "Sftp: a secure and fault-tolerant paradigm against blackhole attack in manet," *arXiv preprint arXiv:1403.0338*, 2014.

[24] S. K. Bhoi, S. K. Panda, and P. M. Khilar, "A density-based clustering paradigm to detect faults in wireless sensor network," in *Proceedings of International Conference on Advances in Computing.* Springer, 2013, pp. 865–871.

[25] S. K. Bhoi and P. M. Khilar, "Sst: A secure fault-tolerant smart transportation system for vehicular ad hoc network," in *2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing.* IEEE, 2012, pp. 545–550.

[26] Y.-G. Yue and P. He, "A comprehensive survey on the reliability of mobile wireless sensor networks: Taxonomy, challenges, and future directions," *Information Fusion*, vol. 44, pp. 188–204, 2018.

[27] L. Cao, Y. Cai, and Y. Yue, "Swarm intelligence-based performance optimization for mobile wireless sensor networks: Survey, challenges, and future directions," *IEEE Access*, vol. 7, pp. 161 524–161 553, 2019.

[28] X. Miao, Y. Liu, H. Zhao, and C. Li, "Distributed online one-class support vector machine for anomaly detection over networks," *IEEE transactions on cybernetics*, vol. 49, no. 4, pp. 1475–1488, 2018.

[29] M. Yoo, T. Kim, J. T. Yoon, Y. Kim, S. Kim, and B. D. Youn, "A resilience measure formulation that considers sensor faults," *Reliability Engineering & System Safety*, vol. 199, p. 106393, 2020.

[30] Z. Noshad, N. Javaid, T. Saba, Z. Wadud, M. Q. Saleem, M. E. Alzahrani, and O. E. Sheta, "Fault detection in wireless sensor networks through the random forest classifier," *Sensors*, vol. 19, no. 7, p. 1568, 2019.

[31] C. Zhang, C. Liu, X. Zhang, and G. Almpanidis, "An up-to-date comparison of state-of-the-art classification algorithms," *Expert Systems with Applications*, vol. 82, pp. 128–150, 2017.

[32] B. Sreejith, A. Verma, and A. Srividya, "Fault diagnosis of rolling element bearing using time-domain features and neural networks," in *2008 IEEE Region 10 and the Third international Conference on Industrial and Information Systems.* IEEE, 2008, pp. 1–6.

[33] B. Samanta, "Gear fault detection using artificial neural networks and support vector machines with genetic algorithms," *Mechanical systems and signal processing*, vol. 18, no. 3, pp. 625–644, 2004.

[34] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

[35] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018.

[36] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Information Fusion*, vol. 37, pp. 132–156, 2017.

[37] M. R. C. Acosta, S. Ahmed, C. E. Garcia, and I. Koo, "Extremely randomized trees-based scheme for stealthy cyber-attack detection in smart grid networks," *IEEE Access*, vol. 8, pp. 19 921–19 933, 2020.

[38] M. Goetz, C. Weber, J. Bloecher, B. Stieltjes, H.-P. Meinzer, and K. Maier-Hein, "Extremely randomized trees based brain tumor segmentation," *Proceeding of BRATS challenge-MICCAI*, pp. 006–011, 2014.

[39] T. Muhammed and R. A. Shaikh, "An analysis of fault detection strategies in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 78, pp. 267–287, 2017.

[40] K. P. Sharma and T. P. Sharma, "rdfd: Reactive distributed fault detection in wireless sensor networks," *Wireless Networks*, vol. 23, no. 4, pp. 1145–1160, 2017.

[41] A. Mahapatro and A. K. Panda, "Choice of detection parameters on fault detection in wireless sensor networks: A multiobjective optimization approach," *Wireless personal communications*, vol. 78, no. 1, pp. 649–669, 2014.

[42] S. Kutten and D. Peleg, "Fault-local distributed mending," in *Proceedings of the fourteenth annual ACM symposium on Principles of distributed computing*, 1995, pp. 20–27.

[43] S. U. Jan and I. Koo, "A novel feature selection scheme and a diversified-input svm-based classifier for sensor fault classification," *Journal of Sensors*, vol. 2018, 2018.

[44] R. Dunia, S. J. Qin, T. F. Edgar, and T. J. McAvoy, "Identification of faulty sensors using principal component analysis," *AIChE Journal*, vol. 42, no. 10, pp. 2797–2812, 1996.

[45] J.-l. Yang, Y.-s. Chen, L.-l. Zhang, and Z. Sun, "Fault detection, isolation, and diagnosis of self-validating multifunctional sensors," *Review of Scientific Instruments*, vol. 87, no. 6, p. 065004, 2016.

[46] J. Kullaa, "Detection, identification, and quantification of sensor fault in a sensor network," *Mechanical Systems and Signal Processing*, vol. 40, no. 1, pp. 208–221, 2013.

[47] Y. Yu, W. Li, D. Sheng, and J. Chen, "A novel sensor fault diagnosis method based

on modified ensemble empirical mode decomposition and probabilistic neural network," *Measurement*, vol. 68, pp. 328–336, 2015.

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

[49] D. Solomatine, "Applications of data-driven modelling and machine learning in control of water resources," in *Computational intelligence in control.* IGI Global, 2003, pp. 197–217.

[50] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, 2011.

[51] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.* O'Reilly Media, 2019.

[52] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[53] S. Suthaharan, M. Alzahrani, S. Rajasegarar, C. Leckie, and M. Palaniswami, "Labelled data collection for anomaly detection in wireless sensor networks," in *2010 sixth international conference on intelligent sensors, sensor networks and information processing.* IEEE, 2010, pp. 269–274.

[54] N. E. ElHady and J. Provost, "A systematic survey on sensor failure detection and fault-tolerance in ambient assisted living," *Sensors*, vol. 18, no. 7, p. 1991, 2018.