Asim Lateef

ANOMALY DETECTION IN WIRELESS SENSOR NETWORKS

M.Sc. Thesis

UNIVERSITY OF JYVÄSKYLÄ
DEPARTMENT OF MATHEMETICAL INFORMATION TECHNOLOGY
2016

ABSTRACT

Author: Lateef, Asim
Topic: Anomaly Detection in Wireless Sensor Networks
Jyväskylä: University of Jyväskylä, 2016, *p* 78.
Mathemetical Information Technology, Master's Thesis
Supervisor: Hämäläinen, Timo

Wireless Sensor Network can be defined as a network of integrated sensors responsible for environmental sensing, data processing and communication with other sensors and the base station while consuming low power. Today, WSNs are being used in almost every part of life. The cost effective nature of WSNs is beneficial for environmental monitoring, production facilities and security monitoring. At the same time WSNs are vulnerable to security breaches, attacks and information leakage. Anomaly detection techniques are used to detect such activities over the network that do not conform to the normal behavior of the network communication. Supervised Machine learning approach is one way to detect anomalies where a normal model is developed with known responses called labels and this model is tested against new data sets. We experimented Supervised Machine Learning approach for the labelled sensor data set of Humidity and Temperature and the results show that KNN (K Nearest Neighbor) proves to be the best anomaly detection algorithm for this data set.

Keywords: Wireless sensor networks, anomaly detection, supervised machine learning

ACKNOWLEDGEMENT

Firstly, I would like to express my sincere gratitude to my advisor Professor Timo Hämäläinen for the continuous support of my MSc. study and thesis research, for his patience, motivation, and immense knowledge. Professor Hämäläinen's suggestion for my thesis topic and guidance has made me learn many new things in the field of Anomaly Detection and Wireless Sensor Networks.

My sincere thanks also goes to all the Professors and staff of the University of Jyvaskyla for their support and guidance.

Last but not the least, I would like to thank my family: my parents, wife and kids and to my brothers and sister for supporting me throughout writing this thesis and my life in general.

LIST OF ABBREVIATIONS

WS	Wireless Sensor
WSN	Wireless Sensor Networks
ML	Machine Learning
KNN	K Nearest Neighbor
AD	Anomaly Detection
ADS	Anomaly Detection System
ID	Intrusion Detection
IDS	Intrusion Detection System
DDIS	Distributed Intrusion Detection System
DT	Decision Trees
SVM	Support Vector Machines
NNC	Nearest Neighbor Classifiers
EC	Ensemble Classifiers

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1 Introduction

WSNs are composed of tiny embedded systems called sensors. These sensors have the ability to sense the environment, receive and process the data and communicate with other sensors and the end user. Sensors are being utilized around us for many purposes and benefits i.e. military target tracking and surveillance, detection of natural disasters, biomedical health monitoring, hazardous environment exploration and seismic sensing, in hospitals to monitor and collect the patient data, structural sensing and so on. These sensors are also placed in dense and harsh environments where usually the presence of human beings is difficult e.g. military purposes to sense and detect the environmental happenings. They operate with little battery power and size. Due to their presence in such environments and the value of the data being processed and communicated, they are also prone to outsider attacks. These attacks can be physical and virtual i.e. hacking.

Anomaly detection is one way to detect attacks and intrusions in WSNs. The main approaches of AD are Statistical. Machine Learning and Data mining. However there is no one technique to fulfill the purpose of AD in any given environment. The nature and purpose of WSN and its data processing and communication requirements need to be understood before applying any AD approach. Supervised Machine Learning is one recommended approach to detect intrusions and anomalies in WSNs.

The motivation behind our research is to first study the WSNs and AD Techniques used for WSNs. Second, we decided to use "Labelled" dataset of temperature and humidity for our analysis. The studies showed that Supervised AD Techniques are well recommended for Labelled data sets. Supervised AD Techniques use a model or classifier to determine the normal behavior of the data set. Then this model or classifier is applied to the new data set to find anomalies. Our experiments have

shown that K-nearest neighbor (KNN) is most suitable AD Technique for such data set.

This thesis work has been distributed in 6 chapters. The basic introduction about wireless sensor networks, the standards and operating systems used followed by architecture, applications and security issues are explained in Chapter 2. The Anomaly detection, the anomaly detection techniques used for WSN and applications of AD techniques for WSN are described in Chapter 3. The Chapter 4 describes the Supervised Anomaly Detection Techniques widely used for AD in WSN. The modelling of the data and results are discussed in Chapter 5. The conclusion is written in Chapter 6.

2 Literature Review: Wireless Sensor Networks

2.1 Introduction

This chapter introduces the reader about WSNs. Afterwards, WSN design architecture, communication topologies, radio standards, applications of WSNs, security issues and types of attacks in WSN are discussed.

2.2 Wireless Sensor Networks [5]

WSN can be defined as a network of integrated sensors responsible for environmental sensing, data processing and communication with other sensors and the base station while consuming low power.

WSNs are composed of individual embedded systems that have the capability of interaction with their environment, processing information and communication with their neighbor sensors and with the end user.

2.3 Components of a WSN [1]

A sensor node typically consists of **three components** and can be either an individual board or embedded into a single system:

Wireless motes

The essential parts of a mote consist of memory, communication and power units. Motes are responsible for communication in WSNs. A wide variety of motes have been developed in recent years. TelosB mote is one good example as shown in Figure 2.1

Sensor board

It is embedded with multiple types of sensors and can be integrated into the wireless module such as in the Telos like humidity, temperature and light sensors.

Programming board

The programming or gateway board, provides various interfaces such as Ethernet, WiFi, USB, or serial ports for connectivity purposes. The programming board can be to program the motes or gather data from them.

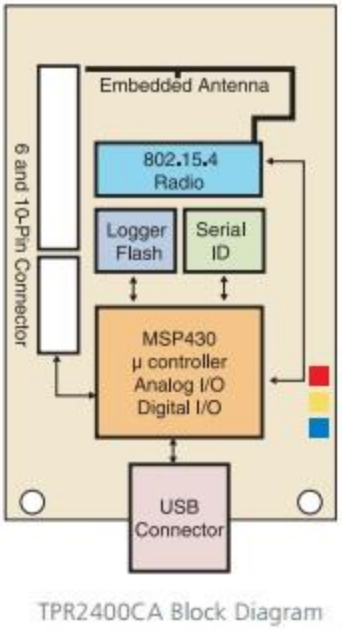

Figure 2.1 Sensor Board Block Diagram

2.4 WSN Radio Standards

The radio standards used in the WSN communication are discussed below.

IEEE802.11x [5]

This standard is used for high bandwidth data transfer between computers and other devices used in local area network. The data transfer rate range is between 1-50 Mbps and over. Typical transmission range is 300 feet with a standard antenna and can be improved with an antenna. WSN applications generally abstain from using this standard due to high data transfer rate.

Bluetooth (IEEE802.15.1 and 15.2) [5]

Bluetooth, also known as personal area network (PAN) standard is recommended for communication between personal computers and devices in a closed perimeter. Bluetooth standard is not recommended for sensor nodes as it consumes high power, has issues with synchronization, serves up to seven nodes only using star network and has a complex medium access controller layer (MAC).

IEEE 802.15.4 [5]

This standard has been specifically designed for WSN applications. It uses low power and supports multiple frequencies and multiple data rates. Star and Mesh Network topology is supported by this standard.

ZigBee [5]

The ZigBee alliance has specified the IEEE 802.15.4 standard as medium access controller layer (MAC) and physical layer. It also supports star and hybrid star-mesh network topologies.

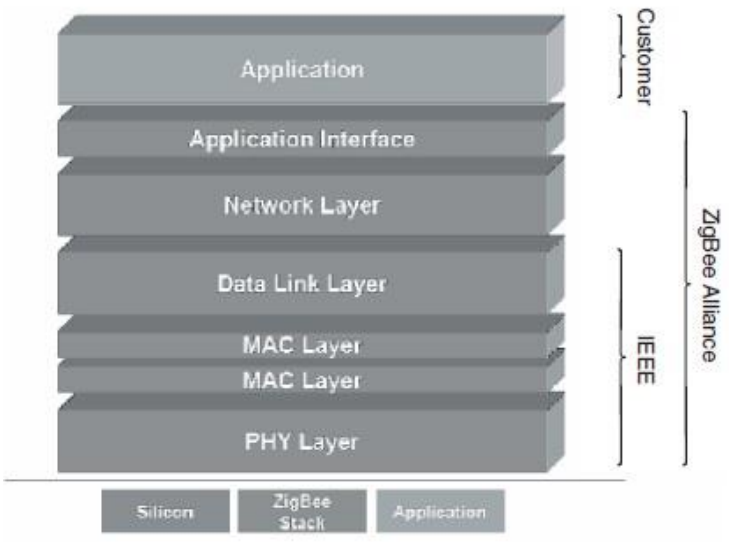

Figure 2.2 ZigBee Stack

IEEE 1451.5 [5]

The wireless sensor working group for IEEE1451.5 is working towards standardizing the interface of sensors to a wireless network. The IEEE802.15.4 physical layer has been chosen as the wireless networking communications interface.

WirelessHART [1] Highway Addressable Remote Transducer (HART) protocol is used mostly as a communication protocol in the automation and industrial applications. WirelessHart has been developed as a wireless extension to the HART protocol. WirelessHART relies on the IEEE 802.15.4 physical layer standard for the 2.4 GHz band. The Figure 2.x shows the network architecture of WirelssHART standard.

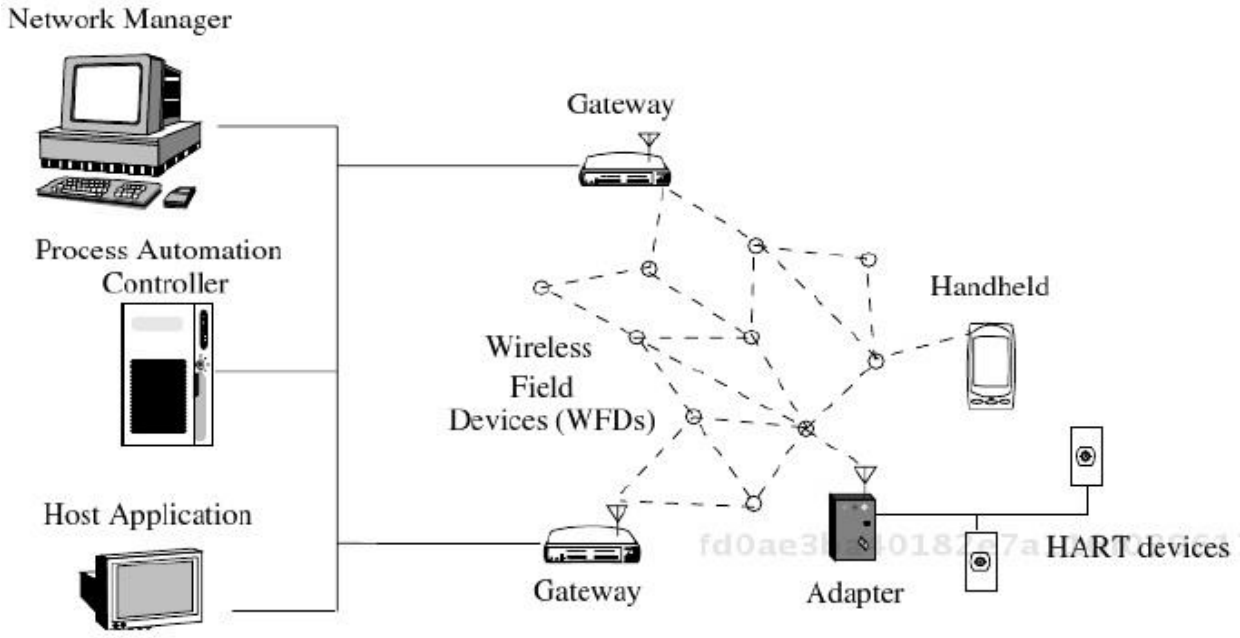

Figure 2.3 WirelessHART Activity

6L0WPAN [1]

Sensors need gateways to connect to the internet. The Internet Engineering Task Force (IETF) is developing the IPv6 over Low-power Wireless Personal Area Network (6L0WPAN) standard to integrate WSNs with the Internet. This standard uses the IPv6 stack on top of IEEE 802.15.4 to connect any device with the Internet.

2.5 WSN Architecture

Network Topologies

Different communication topologies used for WSN are discussed below. The purpose of this communication is that sensor nodes could send/receive the data to the base station or to the other nodes in the network based on the topology. Since the communication process is wireless, power consumption of the node plays an import role in order to keep the communication alive. [5]

Star Network (Figure 2.4)

The sensor nodes in a start network topology, communicate with the base station independently and cannot communicate with one another. The benifit of this type of communication is to save power but in case any node loses the communication with the base station then the data transamission is also lost. [5]

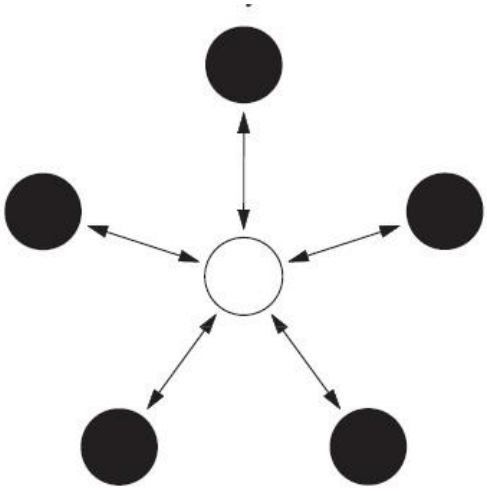

Figure 2.4 Start Network [5]

Mesh Network (Figure 2.5)

The sensor nodes in the mesh network are allowed to communicate with each other and with the base station to transfer the data. The advantage of this type of network is that sensor nodes can communicate with nearest nodes or the base station depending on the distance. Hence the communication does not break. However this type of communication consumes more power. The inter node communication also allows "multi-hop" communication system where a node can communicate with the desired node through an intermediate node. [5]

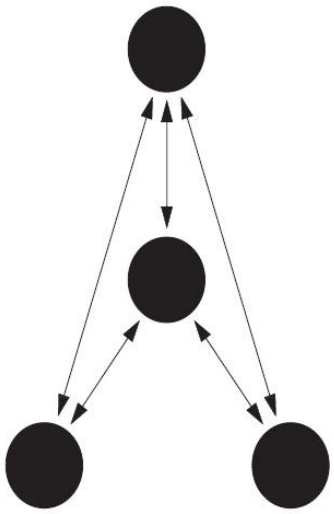

Figure 2.5 Mesh Network [5]

Hybrid Star – Mesh Network (Figure 2.6)

The hybrid network topology is benificial for multi-hop communication system as the high power nodes are used more than the ones with low power. Hence it saves power consumption. The mesh network standard "ZigBee" is used in this commuication topology. [5]

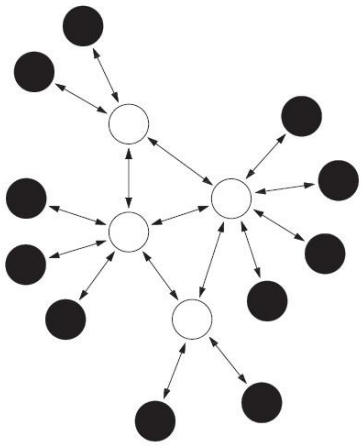

Figure 2.6 Hybrid Star-Mesh Network [5]

2.6 Network Protocol Stack [1]

WSs in a WSN are responsible for data processing and communication through a gateway to connect to their destination i.e. the end user. For this purpose there exists a protocol stack that provides multifunctional services for the WSs. The protocol stack (Figure 2.7) is the communication building block of sensor nodes. It is a combination of different units working in layers in order to manage the wireless communication, power and data processing.

The protocol stack consists of the physical, data link, network, transport and application layers. There are exist synchronization, localization, topology management, power management, mobility management and task management planes.

The management planes are responsible for power efficient collaboration and communication between WSs.

The **Power Management Plane** is responsible for managing the power level of sensor node. It can save power and communicate to other neighboring nodes about the power level.

The **Mobility Management Plane** is responsible for registering the movement of sensor node itself and neighboring nodes as well. It is also tasked to balance the power and task usage.

The responsibility of **Task Management Plane** is to manage the sensing tasks depending of the power level.

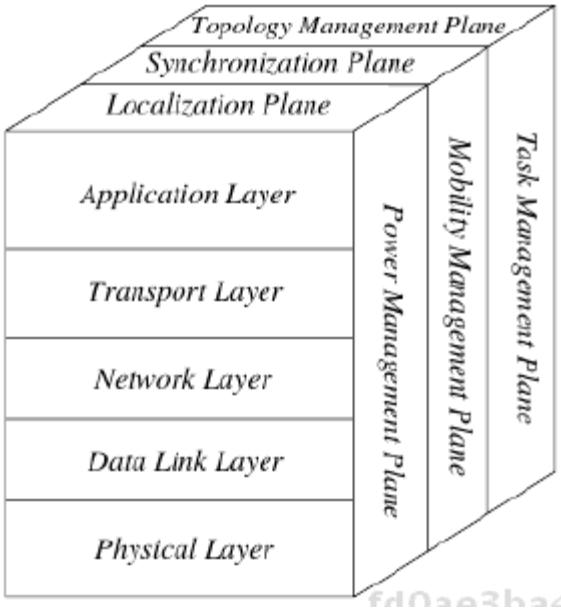

Figure 2.7 Network Protocol Stack

Physical Layer

The physical layer (Figure 2.7) is responsible for frequency management and signal processing.

Data Link Layer

The data link layer (Figure 2.7) is responsible for establishing and maintaining the communication network. It also manages data processing.

MAC

The MAC protocol is responsible for establishing communication network and sharing of resources in multi-hop self-organizing WSNs.

Error Control

Data link layer is also responsible for error control of the data transmission. Forward error correction (FEC) and automatic repeat request (ARQ) are important modes of error control.

Network Layer

WSs require multi-hop routing protocols for the data communication by using neighbor sensor nodes as gateways. The network layer (Figure 2.7) is designed to handle such communication by providing efficient power and to facilitate the routing not only between neighbor nodes but also to neighboring WSNs, Internet and to command and control systems.

Transport Layer

The transport layer (Figure 2.7) is required by the external networks or Internet to connect to the WSNs. For internal communication of WSN, the transport layer protocols provide reliability and congestion control.

Application Layer

The application layer (Figure 2.7) includes the main application as well as several management functionalities. In addition to the application code that is specific for each application, query processing and network management functionalities also reside at this layer.

2.7 Applications of WSNs

With the development of sensors and their integration into WSNs, their use and effectiveness has become so important that they are being used in almost every field of life. WSs are able to monitor various types of conditions such as humidity, temperature pressure, direction, speed, movement, light, noise, objects, stress, event detection and much more. This provides the opportunity to develop different types of applications to monitor security & intelligence, space, environment, health, industrial, weather and climate etc. Some of the applications of WSNs are explained below. Figure 2.8 shows the famous WSN applications developed for military, environmental, health home and Industrial sectors. [1]

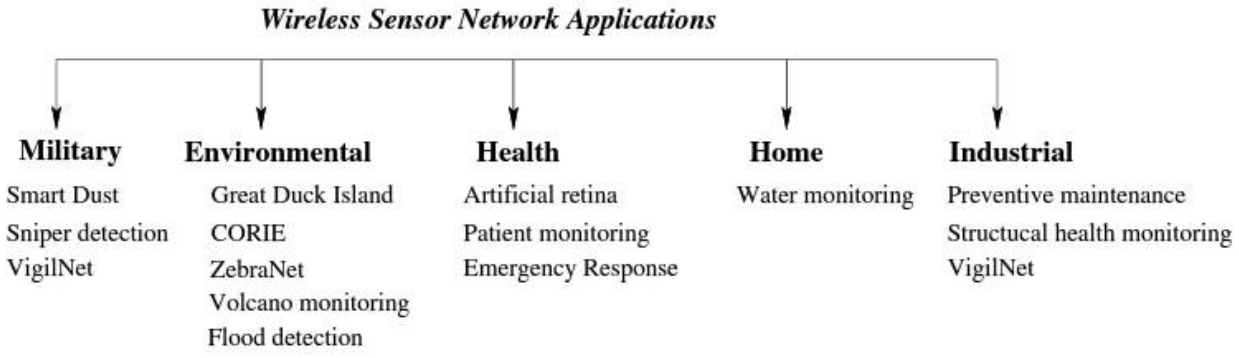

Figure 2.8 WSN Applications [1]

Military applications[1]

WSNs have become essential in the development of applications for C4ISRT systems i.e. command, control, communications, computing, intelligence, surveillance, reconnaissance and targeting systems for the military. The monitoring applications of WSN are used mostly for surveillance, reconnaissance and targeting purposes.

Smart Dust is a WSNs application system that works in hostile environments where human access is dangerous. The tiny cubic millimeter sized sensors are used to detect, monitor and track activities. Sniper Detection System (Figure 2.9) is a sniper location detection system that is used my military and law-enforcement agencies. VigilNet (Figure 2.10) is WSN based target tracking surveillance network.

Figure 2.9 Sniper Detection System

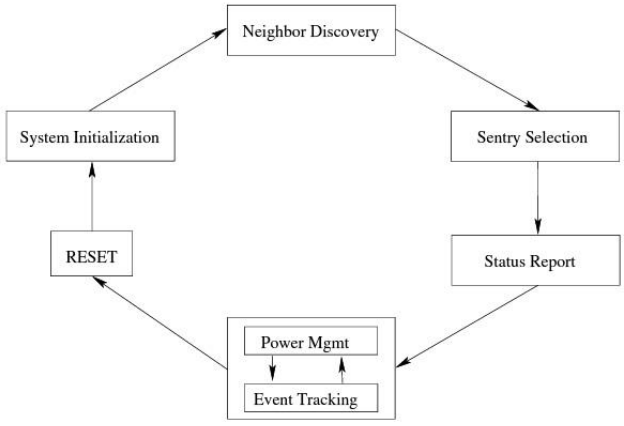

Figure 2.10 VigilNet

Environmental applications[1]

The WSN environmental applications are used for earth and environmental monitoring, tracking, detection, mapping and research etc.

The Great Duck Island project (Figure 2.11) was used to measure the occupancy of nesting burrows and the role of microclimatic factors in the habitat selection of seabirds in Maine. The environmental observation and forecasting system (EOFS) knows as (CORIE) was developed to measure velocity temperature, salinity and depth of water along with wind and air properties for the Columbia River. Flood detection and prediction system (Figure 2.13) was developed by MIT and was tested in Honduras. ZebraNet system (Figure 2.12) was deployed in Kenya for the tracking of two species of

zebras. WSN applications were used to monitor for volcano monitoring of "Volcán Tangurahua" in Ecuador in 2004-2005.

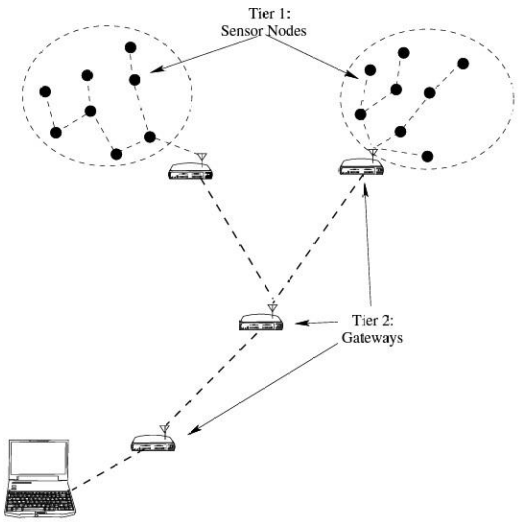

Figure 2.11 Great Duck Island project

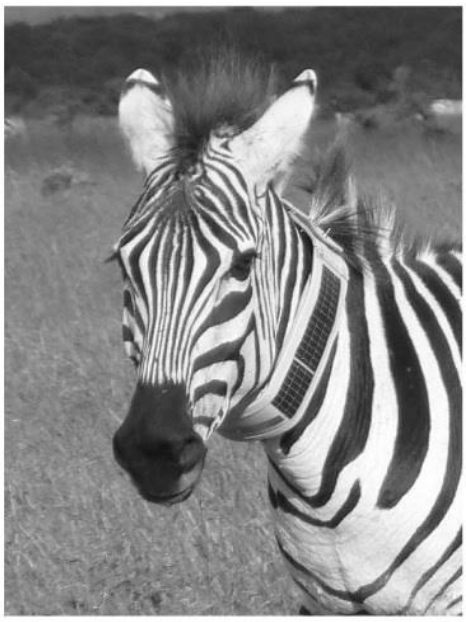

Figure 2.12 ZebraNet

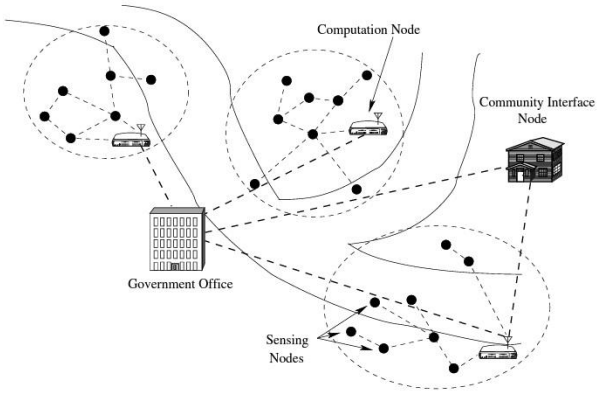

Figure 2.13 Flood Detection

Health applications [1]

The health industry is also benefiting from the WSN applications. These applications help in monitoring, diagnosis, administration, tracking of different activities in hospitals and elsewhere.

Visually impaired people can benefit from the Artificial Retina project (Figure 2.14) for curing the diseases i.e. age-related muscular degeneration (AMD) and retinitis pigmentosa (RP). The CodeBlue project (Figure 2.15, 2.16) requires wearable sensors for

monitoring of the pulse rate, blood oxygen, heart and muscles activity etc. of the patients. The medical personnel can access all the information received from wearable sensors on a hand held PDAs (Figure 2.17)

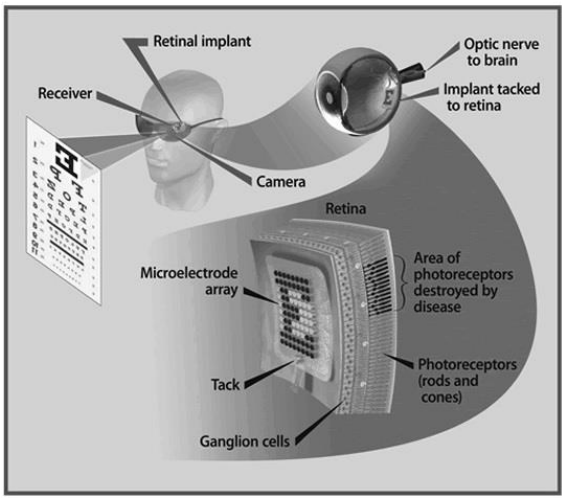

Figure 2.14 AR Project

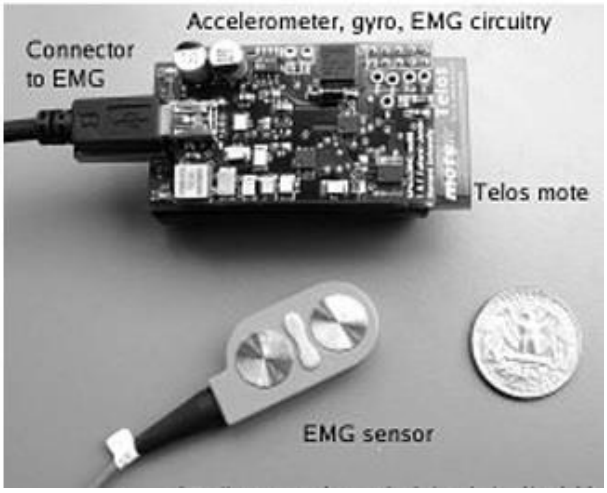

Figure 2.15 Code Blue Project

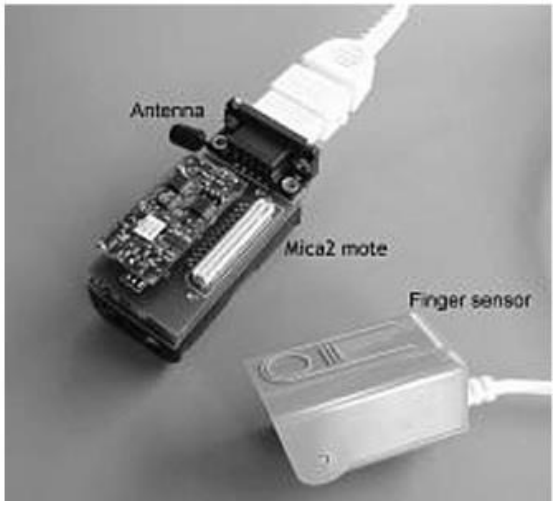

Figure 2.16 Code Blue Project

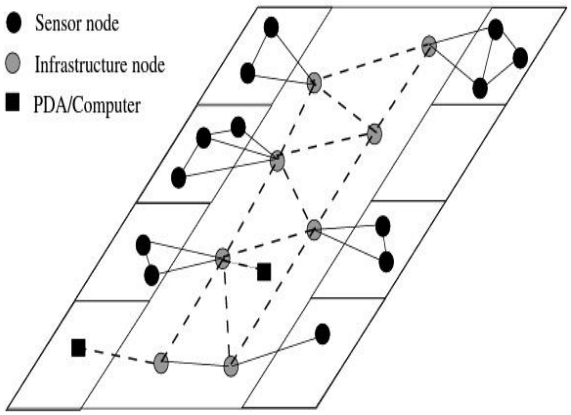

Figure 2.17 Code Blue Project

Home applications[1]

WSNs can be integrated in the devices used at home such as refrigerators, ovens, TVs etc. These devices can not only be monitored but also controlled by WSNs through internet. The Nonintrusive Autonomous Water Monitoring System (NAWMS) (Figure 2.18) is a detection and monitoring system for the water usage at homes.

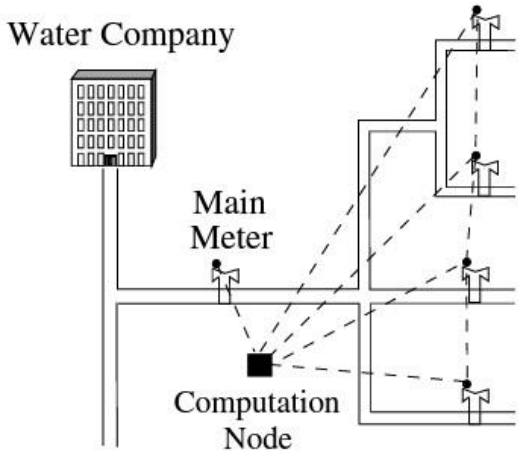

Figure 2.18 NAWMS

Industrial applications[1]

WSNs have proved to be cost effective, accurate and efficient than wired sensors for industrial use. Industrial WSNs are used for monitoring, control, detection, diagnosis, instrumentation, tracking and sensing activities.

The data automation for preventive industrial maintenance has been developed in the form of sensors and gateways as FabApp architecture (Figure 2.19)

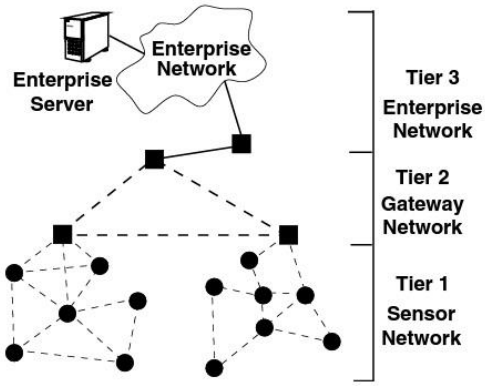

Figure 2.19 FabApp Architecture

Industrial Automation [5]

Wireless sensors are providing very cost effective solutions by replacing lead wires in the industry. WSN have been very effective in various sectors like Health monitoring for wind turbines, environmental monitoring, health care and location based services. A production line example is shown in Figure 2.20 below. Wireless sensors are used to measure gaps in rubber seals.

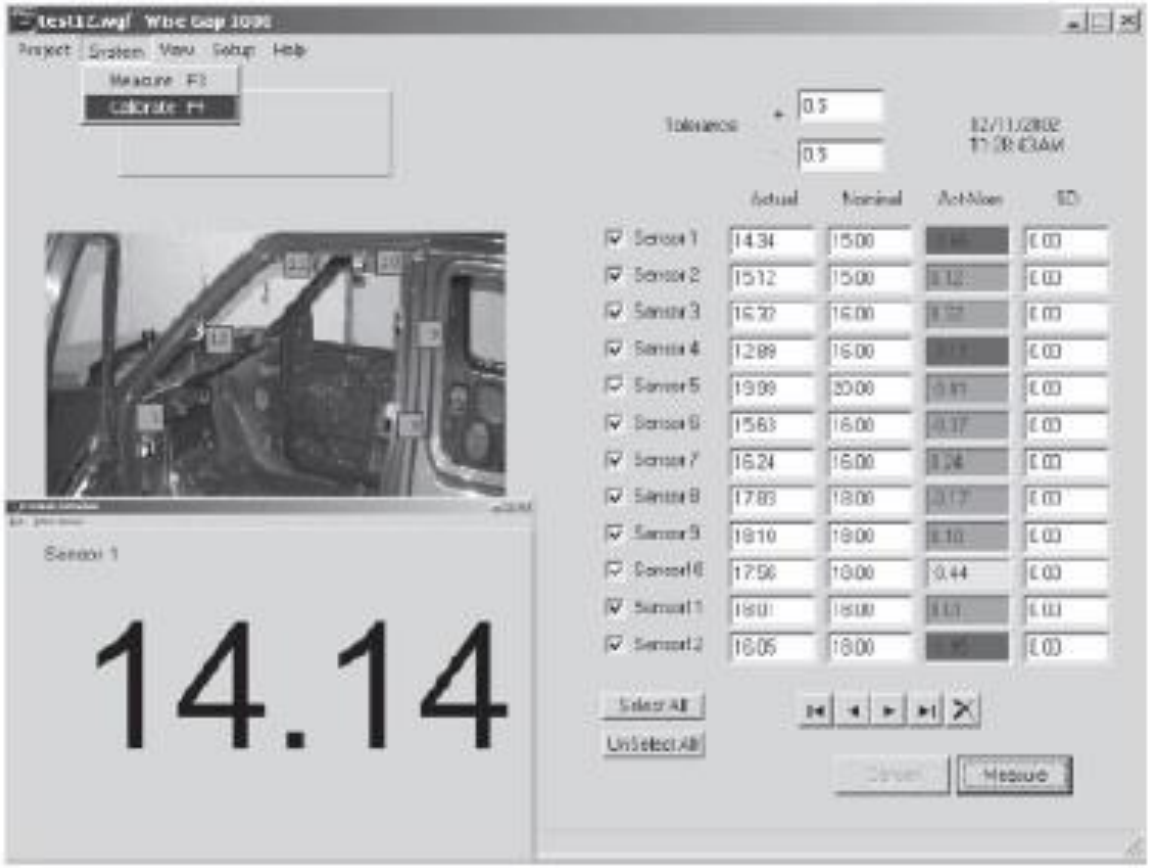

Figure 2.20 Production Line

Civil Structure Monitoring [5]

This is the example of sensors deployed at Ben Franklin Bridge. Since high speed trains crossed over the bridge. The requirement was to monitor the strains on the structure. Once the train arrives on the bridge, the strain waveform is logged by the sensor nodes. The example in the Figure 2.21 shows analysis of the data collected by one node.

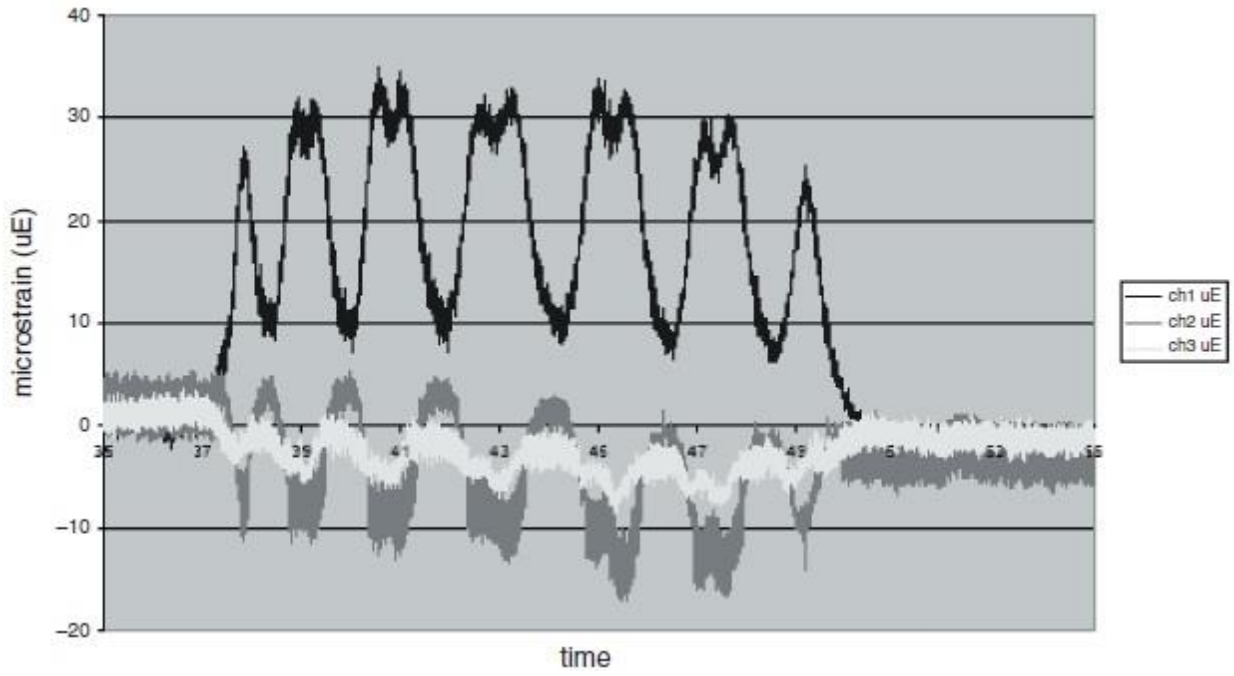

Figure 2.21 Ben Franklin Bridge Sensor Information

2.8 Security issues in WSN

WSNs and their adoption in every part of life also poses risks to their security, privacy and operations. WSNs are at risk in the same way as any other wired network especially when it is connected with the internet to transfer the information to its destination. WSNs can be hacked for the purpose of jamming the network, altering or stealing the information and jeopardizing the operations etc. In this section we will discuss different types of security issues related to WSNs.

Security Principals [7]

The fundamental security principals of the data communication in WSNs can be classifies as follows:

- **Authentication**: Data is initiated from original source.
- **Confidentiality**: Only authorized sensor nodes receive the data.
- **Integrity**: Data is original and not modified.
- **Availability**: Services of the sensor node are available as an when required.
- **Freshness**: Fresh data is transmitted always.

Types of Attacks in Wireless Sensor Networks [3] [4][6]

- **Jamming** Jamming the carrier frequency using high-energy signals and preventing successful communication over the wireless medium.
- **Tampering** Causing physical damage to the node
- **Sinkhole Attack** Altering of routing information in order to lure traffic toward the compromised node
- **Denial of service (DOS)** Flooding the network with unwanted traffic in order to exhaust its resources, resulting in a denial of service for legitimate traffic. E.g. Hello Flooding and exhaustion attack
- **Spoofing** Impersonating another node using a false identity
- **Selective Forwarding** Dropping some or all packets that pass through the compromised node.
- **Sybil** A malicious node pretending that it is a number of the other existing nodes in the network.
- **Wormholes** Capturing network traffic by a malicious node and tunneling or diverting it to another node in the network.
- **Eavesdropping** Listening to a message belonging to others.
- **HELLO flooding**. HELLO flooding. This attack is used to make the sensor network confused. In a node to node communication, Hello packet is used by the node to announce itself. An attacker with strong network transmission power can make every node send Hello packet to its neighbor and make the network confused.

The layering based attacks and the countermeasures can be viewed in the Figures below:

Physical Layer Attacks [7]

Threat	*Countermeasure*
Interference	Channel hopping and Blacklisting
Jamming	Channel hopping and Blacklisting
Sybil	Physical Protection of devices
Tampering	Protection and Changing of key

Figure 2.22 Physical Layer Attacks

Data Link Layer Attacks [7]

Threat	*Countermeasure*
Collision	CRC and Time Diversity
Exhaustion	Protection of Network ID and other Information that is required to joining device
Spoofing	Use different path for re-sending the message
Sybil	Regularly changing of key
De-synchronization	Using different neighbors for time synchronization
Traffic analysis	Sending of dummy packet in quite hours: and regular monitoring WSN network
Eavesdropping	Key protects DLPDU from Eavesdropper

Figure 2.23 Data Link Layer Attacks

Network Layer Attacks [7]

Threat	*Countermeasure*
Eavesdropping	Session Keys protect NPDU from Eavesdropper.
DoS	Protection of network specific data link network ID etc. Physical protection and inspection of network.
Selective forwarding	Regular network monitoring using Source Routing.
Sybil	Resetting of device and changing of session keys.
Traffic Analysis	Sending of dummy packet in quite hours: and regular monitoring WSN network.
Wormhole	Physical monitoring of Field devices and regular monitoring of network using Source Routing. Monitoring system may use packet leach techniques.

Figure 2.24 Network Layer Attacks

Transport Layer Attacks [7]

The transport layer is vulnerable to flooding attack. Too many connection requests are sent to a vulnerable node in order to exhaust and make it useless for communication any more.

2.9 Conclusion

WSNs have become an integral part of our everyday life. While benefiting from its uses and making our life easy, these sensor networks are also prone to attacks that can not only jeopardize their operations but may also risk everyone attached to them. Hence, the need to overcome this risk, scientists are continuously developing systems to secure the WSNs. Anomaly detection is one way to detect any intrusions to make the WSN safe. We will discuss AD in WSNs in the following chapter.

3 Literature Review: Anomaly detection in Wireless Sensor Networks

3.1 Introduction

This chapter discusses the basic definitions of AD, types of Anomalies, AD techniques used for WSN and the applications of AD techniques for WSN.

3.2 Anomaly Detection

The study of outliers or anomalies was conducted by [Edgeworth 1887] as early as in 19th century. Barnett and Lewis [42] define that, "an anomaly or outlier in a set of data as "an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data."

AD is a process of detecting such patterns that deviate from the normal outcome of the data. Such patterns are called anomalies or outliers. [9]

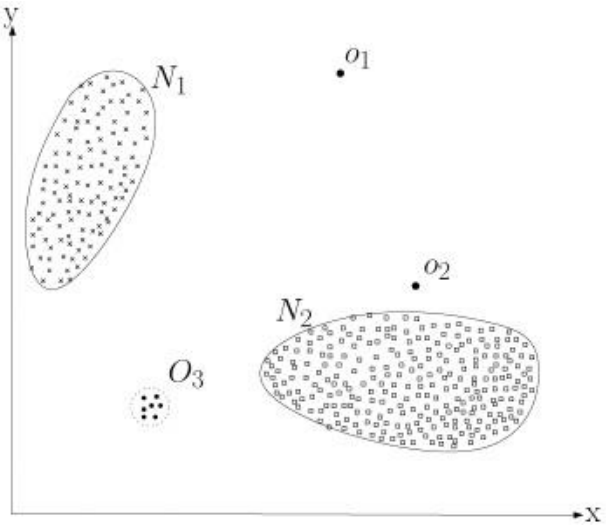

Figure 3.1 Anomaly Detection [9]

The process of Ad involves various challenging. The biggest challenge is to define the normal behavior of the data. It provides challenges for different data environments. The nature of the data and types of anomalies to be detected proves to be a difficult process. For example in malicious activities the adversaries tend to behave in a normal manner. Noise is also defined as a type of data that is unwanted in data analysis and it should be removed. Anomaly detection is widely used in detecting faults, frauds, misuses and intrusions in financial, health, cyber-security, critical systems and military sectors.

A simple illustration has been drawn in Figure 3.1 explains a two-dimensional data set. The normal data regions are N1 and N2 and points o1, o2 and O3, are anomalies.

The Figure 3.2 explains the applications domains, data challenges and the AD techniques representing different research areas.

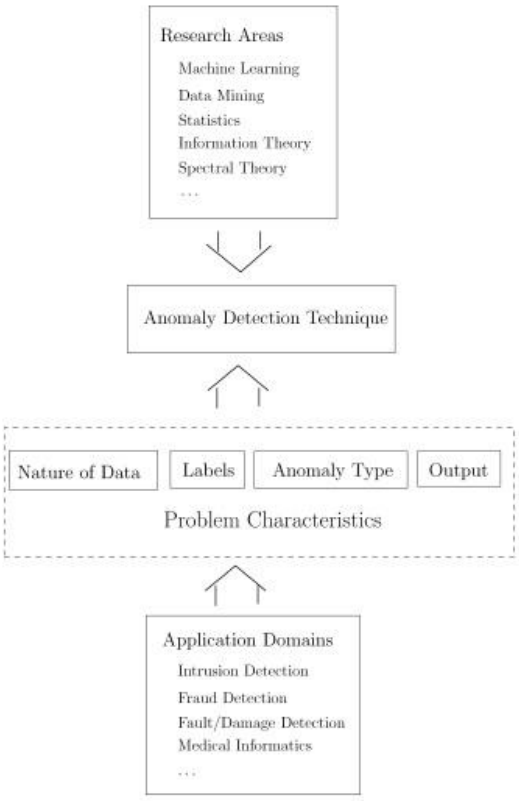

Figure 3.2 Key components associated with an anomaly detection technique

Patcha et al. [8] have explained that the AD systems, model the normal system or network behavior which enables them to be effective in detection of known, unknown and also "zero day" attacks. An AD system first creates a normal profile of the system, network, or program activity. Thereafter, any activity deviating from the normal is treated as an anomaly. The advantage of ADSs is that the profiles of normal activity are customized for individual systems, applications and networks, hence making it very difficult for attackers to carry out their activities without getting detected. That is why AD systems have the capability to detect insider attacks. An anomaly detection system also has the ability to detect previously unknown attacks.

3.3 Types of Anomalies [9]

Point Anomaly

When the individual data instance with respect of the rest of the data is considered as anomalous then this instance is called point anomaly. In Figure 3.x above the instances o1, o2 and O3 are examples of point anomalies.

Contextual Anomaly

If a data instance is anomalous in a specific context then it is termed a contextual anomaly. An example of Contextual anomaly is in Figure 3.3 of time-series temperature data. The temperature at time t1 and t2 is same but in a different context. Hence it is not considered as an anomaly. However the temperature t1 of 35◦F would be considered normal in winters but it would be an anomaly i.e. t2 in summers.

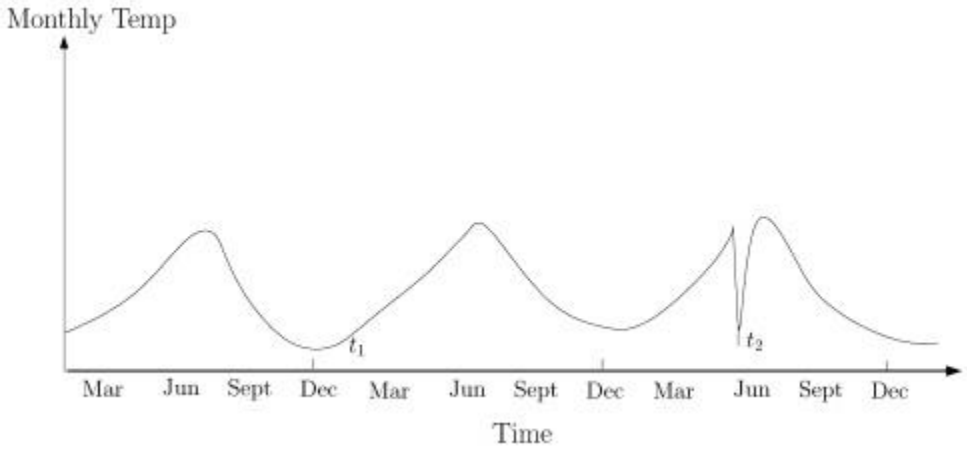

Figure 3.3 Contextual Anomaly [9]

Collective Anomaly

When a group of data instances are anomalous with respect to the rest of the data set, then it is called collective anomaly. The Figure 3.4 of an electrocardiogram shows that the low value exists for an abnormally long time, hence it is an anomaly.

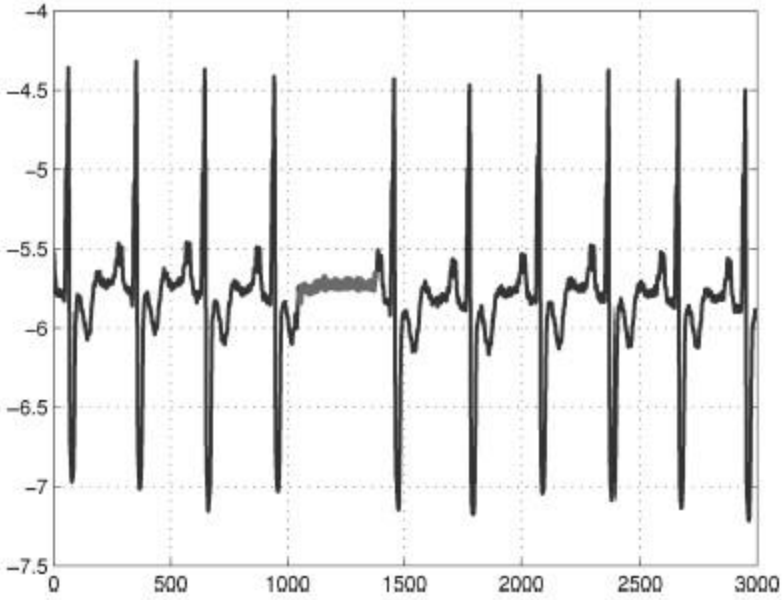

Figure 3.4 Collective Anomaly

3.4 Anomaly Detection Techniques

An anomaly detection approach usually comprises of two phases. 1. Training phase to create a normal behavior profile of the data. 2. Testing phase to apply the learned behavior to the new data. [8]

The result of any detection technique is to report whether an anomaly is found or not. Kumar and Stafford[23] have suggested four possibilities of the outcome of the detection process:

- **Intrusive but not anomalous**: False-negative. False report of absence of anomaly.
- **Not intrusive but anomalous**: False-positive. Intrusion detected but is not anomaly.
- **Not intrusive and not anomalous**: True-negative. Activity is not reported and is not anomaly.
- **Intrusive and anomalous**: True-positive. Activity is reported and is an anomaly.

The AD techniques can be categorized in Statistical, Machine Learning and Data Mining. These techniques along with examples are explained below.

3.4.1 Statistical anomaly detection

In statistical anomaly detection methods, the system creates behavior profiles. Typically, two profiles are kept for each object: the current profile and the stored behavior profile. The detection system updates the current profile and calculates an anomaly score regularly by comparing the current profile with the stored behavior profile. If the anomaly score is higher than a certain point, the intrusion detection system generates an alert. [8]

Statistical Anomaly Detection Systems:

Examples of Statistical ADSs are explained below.

- Haystack [8] [10]It uses descriptive statistics behavior model for AD.
- NIDES [8][11] A DIDS having anomaly as well as signature detection modules.
- Staniford [8][12] Statistical AD technique that calculates an anomaly score for individual packets to be tested in an IDS.
- Hotellings T2 [8][13] In order to detect host-based intrusions, this test is used to analyze the audit trails of activities in an computer system. Data-mining based AD methods

3.4.2 Machine learning based techniques

Machine learning can be defined as a process of analyzing a system over time and improving the performance or various tasks involved.

Machine learning based anomaly detection systems

Examples of ML based ADSs are explained below.

- Forrest et al. [14] developed a system that builds a normal profile by finding correlations in fixed length sequences for anomaly detection.
- Eskin et al. [15] developed a system to improve system call modeling.
- Valdes et al. [16] developed a system that uses Bayesian inference techniques to detect traffic attacks.
- Shyu et al. [17] developed a system that uses principal component analysis to detect intrusions.
- Yeung et al. [18] presented a hidden Markov models based dynamic modeling approach for modeling system calls.

- PHAD [19] It examines IP headers by connecting various ports.
- ALAD [19] This system scans ports and inbound TCP connections to detects anomalies.
- LERAD [21] It uses a learning algorithm to anomaly detection.

3.4.3 Data mining based techniques

In the context of AD, data mining can be defined as a process of analyzing the patterns in the data for the purpose of detection of anomalies.

Data mining based anomaly detection systems.

Examples of data mining systems are explained below:

- RIPPER [22] It uses inductive rule generation for important and infrequent events.
- FIRE [24] In order to detect individual attacks, t generates fuzzy sets to define fuzzy rules.
- ANDSOM [25] It is used to track and classify the connection behavior.
- MINDS [26] To detect intrusions, uses clustering data and density based local outliers.
- ADAM [27] Uses a classification technique to performs anomaly detection.

3.5 Anomaly Detection in WSN.

In the context of AD, sensors networks have generally two issues while operating in any environment i.e. faulty sensors or intrusion detection or both. The data can also consist of noise i.e. non-related or non-important data. The AD techniques used for detection of anomalies in WSNs have to handle noise and are recommended to be light weight due to resource constraints.

Anomaly Detection Techniques for WSNs

Chandola at el. [9] and Rajasegarar et al. [28] have recommended both supervised and unsupervised AD techniques for WSNs.

3.5.1 Classification based AD Techniques

Classification based AD Technique uses a two stage process. First stage is to create a model or classifier from the labeled data set and then in the second stage, the trained model is applied to the new data set in order to find any anomalies.

Classification based anomaly detection techniques can be divided in two categories i.e. multi class and one-class (Figure 3.5) Multi-class AD techniques assume that the there are multiple trained models or classifiers contains labelled data belonging to multiple normal classes (Figure 3.5a). One-class AD techniques assume that there is one trained model or classifier that contains labelled data belonging to single normal classes (Figure 3.5b).

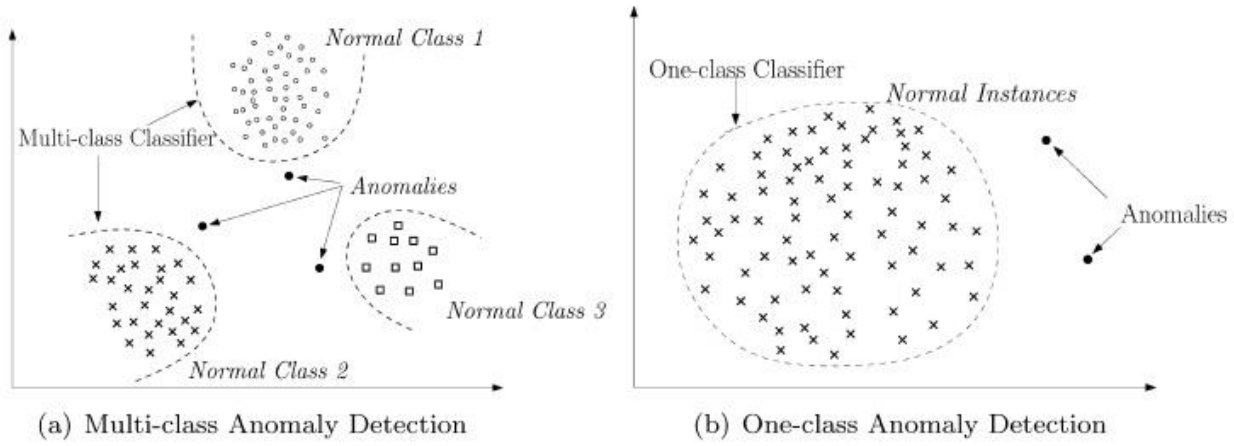

Figure 3.5 Classification Based AD

Bayesian Networks [43][35]

Bayesian networks can be defined as nodes encompassing the nodes and arcs. If variable X denotes a node then P(X) means the probability of this variable. An arc from node X to Y describes an effect of X on Y i.e. P(Y | X) and it is called conditional probability. The nodes encompassing the nodes and arcs can be considered the architecture of a typical Bayesian network and along with the conditional probability as the parameter.

Support Vector Machines [9][41]

SVM AD Techniques are used in one-class network scenarios. These techniques learn the region encompassing the instances of training data. Then the techniques determine if the test instance falls within or out of the learned region. If it falls out of the test region, it is declared as an anomaly.

SVM has been used for AD in audio signals and temporal sequences.

Rule-Based AD Techniques [9][36]

Rule-based AD techniques are applied in one class and multi class environment. The Rule-based AD Techniques learn the normal behavior rules of a system. Any instance of the data that has not been covered in the rule is considered as an anomaly. It is s two step technique i.e. learn rules from the training data and for each test instance find a rule that has not been captured during the training phase. Rule learning algorithms i.e. RIPPER or Decision Trees etc. are used in the process.

Rule-based techniques are used for intrusion detection and credit card fraud detection.

The advantages of classification-based techniques

- Can use powerful algorithms that can distinguish between instances of different classes.
- Since each instance is tested against a trained model hence the testing phase is fast.

The disadvantages of classification-based techniques

- Multi-class techniques rely on the availability of accurate labels that is usually not possible.
- The techniques assign a label to each test instance, that can also become a disadvantage when a meaningful anomaly score is desired for the test instances.

3.5.2 Nearest Neighbor-Based Techniques [9][37][38][39]

Nearest neighbor-based AD Techniques are based on the concept that any normal instance will occur in a crowded neighborhood and anomalous instance will occur far from their closest neighbors. Therefore the distance or similarity measure between two data instances is required. The popular distance measurement tool is Euclidean distance.

The techniques can be grouped into two categories:

- The distance of a data instance to its kth nearest neighbor as the anomaly score.
- Compute the relative density of each data instance to compute its anomaly score.

Distance to kth Nearest Neighbor

The distance of a data instance to its kth nearest neighbor is important in deciding if it is an anomaly or not. Some scientists use the value of k=1 and some select n instances as anomaly. A different way to compute the anomaly score of a data instance is to count the number of nearest neighbors (n) that are not more than (d) distance apart from the given data instance.

Advantages of Nearest Neighbor-Based Techniques:

- Being unsupervised in nature, they do not make any assumptions because they are purely data driven.
- Semi supervised techniques perform better than unsupervised ones.
- Adapting nearest neighbor-based techniques to a different data type is easy.

Disadvantages of Nearest Neighbor-Based Techniques:

- There could be missed anomalies if normal data lacks the neighbors or anomalous data has neighbors that are not labelled.
- For semi supervised techniques, false-positive rate will be is high if the test data does not have enough similar normal instances.
- It is very challenging to compute all the distances of test and training data.
- For complex data, distance measuring can be challenging.

3.5.3 Spectral Anomaly Detection Techniques [9][40]

Spectral techniques can work in an unsupervised as well as a semi supervised setting. The general approach of these techniques lies in determining the subspaces to find anomalies. It is assumed that normal instances and anomalies appear differently in lower dimensional subspace, hence data can be embedded into it.

Advantages of spectral AD Techniques:

- These techniques are suitable for handling high dimensional data sets.
- Can be used in an unsupervised setting.

Disadvantages of spectral AD Techniques:

- Are useful only if the normal and anomalous instances are separable in the lower dimensional embedding of the data.

- Have high computational complexity.

3.6 Conclusion

Anomaly or outlier detection is a process to determine any activity that does not conform to the normal behavior of the system. There are many types of approaches of AD namely Statistical, Machine Learning and Data Mining based etc. Each AD approach is suitable for a specific system. Hence scientists and researchers have to apply different AD Techniques to various systems and scenarios under observation. WSNs have a specific setup of operations. Most important and vulnerable systems to outsider attacks are Military and Health systems. WSNs are deployed in a distributed manner to sense and collect data and transfer it to the end user. Due to the unique nature of WSNs, scientists and researchers recommend specific AD Techniques. Still no one technique can be most suitable for detection purposes. The data collection environment, type and nature of data are few important considerations for selection and applying the AD Technique.

4 Anomaly detection of Labelled Wireless Sensor Network Data using Machine Learning Techniques

4.1 Introduction

This chapter is divided into following sections. The data labels and their significance in the AD Techniques is described in section 4.2 The data acquisition and data collection environment in section 4.3 The section 4.4 is about data analysis. The graph plots have been drawn based on the dataset and then discussed. The discussion in section 4.4 is about Supervised Machine Learning Technique and its importance and relation with the "Labelled Data set" being analyzed. This section also explains "Classification Learner App", a Matlab Software tool for modelling the dataset. Section 4.5 is Data Modelling and Conclusion of the chapter is in section 4.6. Finally the results of data modelling and conclusion are discussed in Chapter 5 and 6 respectively.

The chapter starts with the introduction of the data collection and its environment. Then we discuss data processing techniques to train the data in order to understand the normal behavior of the data. Afterwards, we apply Machine Learning Techniques to find out the anomalies. The anomalous data was introduced during the data collection process in order to develop a labeled data set for the purpose of examining the detection process by various techniques.

4.2 Labelled data [9] [28]

The data labels explain whether that instance is normal or anomalous. Expert professionals are required to obtain the labeled training data set and it is a difficult task. It is also very difficult task to obtain a labeled set of data with anomalies that covers all possible type of anomalous behavior. The anomalous behavior of the data is often dynamic and new types of anomalies could arise, for which there is no labeled training data.

Modes of Anomaly Detection Techniques

Based on the extent to which the labels are available, anomaly detection techniques can operate in one of the following three modes:

Supervised Anomaly Detection

Supervised AD approach is used for predictive modelling as well as in anomaly detection. The process involves training a normal model based on the specific data set. This model defines both normal and anomalous data instances. Once the model is trained, it is applied to the new data set to find out anomalies or to make predictions.

Semi supervised Anomaly Detection

The semi supervised AD approach involves the training model for normal data only. Hence the anomalous data is not previously modelled. It is said to be more used as it is not an easy task to model all the anomalous data types.

Unsupervised Anomaly Detection

The Unsupervised AD approach does not involve training any data model. The techniques based on this approach assume that there is not high risk of detection of anomalies in the data. Due to this approach there exist high false alarm occurring.

4.3 Data collection and environment

The dataset [30] consists of measurements of Humidity and Temperature. As a hardware platform for sensors, TelosB motes [31] were used to collect the data. The description and working of the motes has already been explained in Chapter 2. The duration time of data collection was 6 hours with time intervals of 5 seconds per reading. Two network schemes were adopted i.e. Single-hop and multi-hop. The data labels were denoted as “0” and “1”. The normal behavior of the data is “0” whereas as “1” is an

anomaly that is introduced. Hot water steam was used to increase the humidity and temperature.

The data collection, hardware and software environment and introduction of the anomalies as has been described by Suthaharan et al. [29]. The dataset is available at The University of North Carolina at Greensboro website [30].

Dataset

The dataset [30] includes 4 tables of Single-hop dataset and 4 tabled of Milti-hop dataset. The elements of the data are "Reading", "MoteID", "Humidity", "Temperature" and "Label". Reading is the number of readings collected in different intervals of time. MoteID defines the mote used in a specific data collection scheme. Humidity and Temperature are the corresponding values. Label has the value 0 or 1. "0" value means normal data whereas the value "1" means an anomaly.

Crossbow TelosB Mote Platform

The datasheet of the TelosB Mote [31] explains its design and architecture. It is an open source platform and was developed by University of California, Berkeley. The basic features include the following. IEEE 802.15.4 compatibility. Data transmission capability of 250kbps. USB support. Open source TinyOS operating system. Sensor support for temperature, light and humidity.

Figure 4.1 Crossbow's TelosB mote

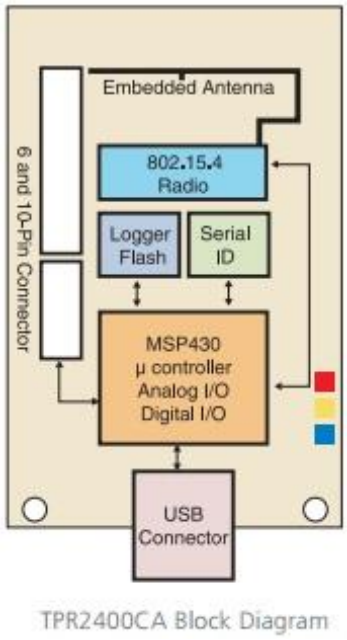

Figure 4.2 Block Diagram of Mote

Data Collection

The single and multi-hop data transmission was designed as a distributed system. The humidity and temperature data collected by the base station was highly correlated and in elliptical shapes, thus to create parameters for a global ellipse. The global ellipse was then used by the sensors in order to detect anomalies.

Single-Hop Data Collection

The distributed Single-hop system consisted of two indoor sensor nodes and two outdoor sensor nodes. Anomalies were introduced in the middle of the 6 hours data collection time frame to one sensor node in each set up (indoor and outdoor). A hot water kettle was used to increase the temperature and the humidity simultaneously.

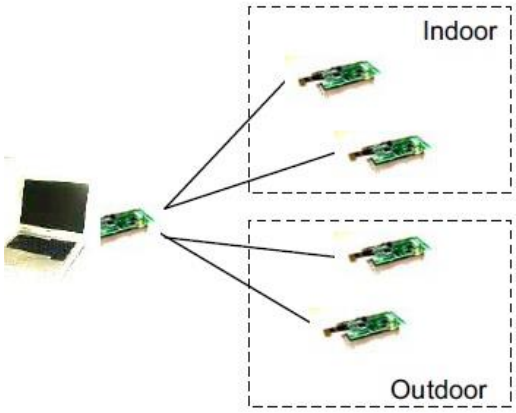

Figure 4.3 Single Hop Data Collection System

Multi-Hop Data Collection

Due to the limitation of transmission range, the sensors might not reach the base station directly. The multi-hop sensor network system was designed keeping in view the wide area network scenario. The customized environment had the following measurements. 1. One meter range for the sensor nodes. 2. Distance between the sensor and the base station was set to about 3 meters. 3. Distance between the router and the sensor was less than 1 meter. In order to reduce data traffic, each sensor node collects ten readings at the interval of 5 seconds and sends it to the base station in one packet. The time period for collecting the readings is 6 hours, during which anomalies were introduced to one sensor node in each scenario by using a water kettle which altered the temperature and humidity appropriately.[29]

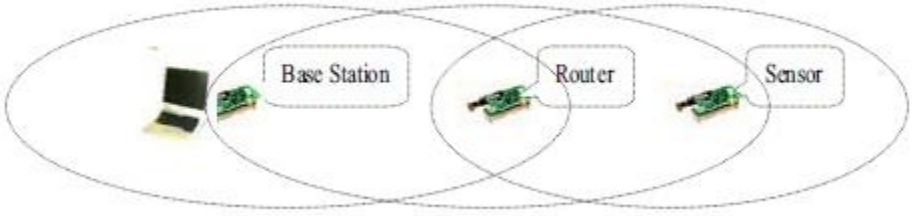

Figure 4.4 Multi Hop Data Collection System

4.4 Data Analysis

The dataset consists of 5 elements i.e. Reading, MoteID, Humidity, Temperature and Label. The Reading is the time interval of the sensor to collect data. MoteID represents the sensor ID. Humidity and Temperature values are collected according to the MoteID and at certain time interval as described in section 4.2

Following graphs have been plotted using Matlab for the datasets as described in section 4.2 above. The graphs have been plotted in respect to the Reading and Time values to show the Humidity and Temperature values.

Single-Hop Indoor Sensor Data Analysis

In the single-hop data collection scenario, two motes are placed indoors. One mote collects normal measurements whereas the other mote is introduced with anomalous data. The graph in Figure 4.5 shows that the MoteID -2 has collected normal data whereas Figure 4.6 shows that MoteID -1 collected the data with anomalies.

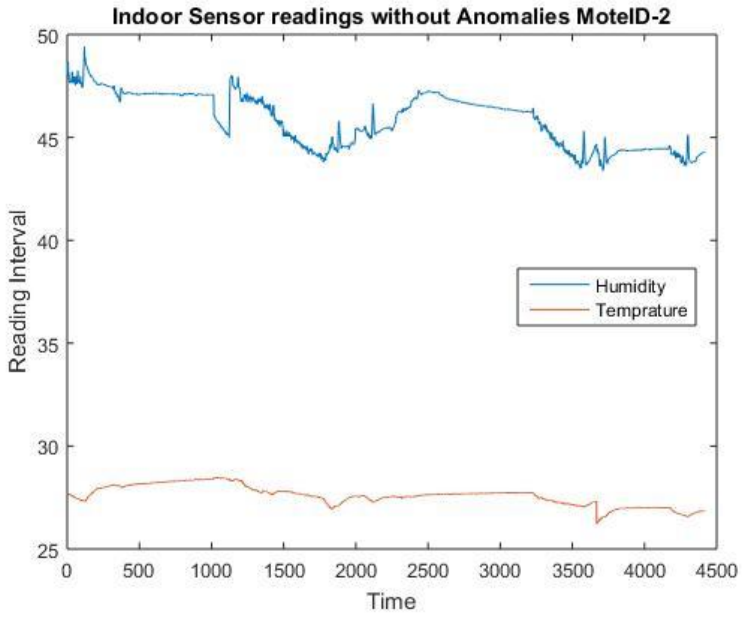

Figure 4.5 MoteID-2

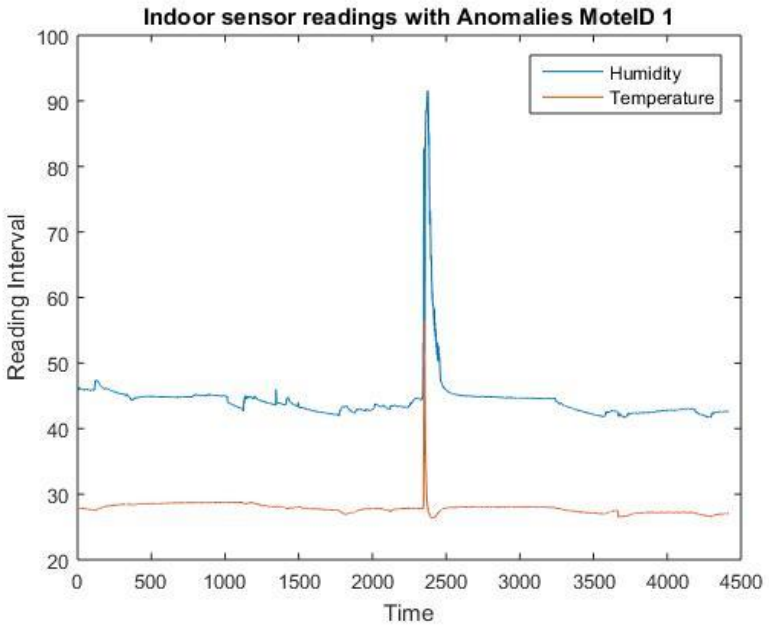

Figure 4.6 MoteID-1

Single-Hop Outdoor Sensor Data Analysis

The outdoor single-hop data collection scenario also consists of two motes placed outdoors. MoteID -3 (Figure 4.7) collects normal measurements whereas MoteID -4 (Figure 4.8) is introduced with anomalous data.

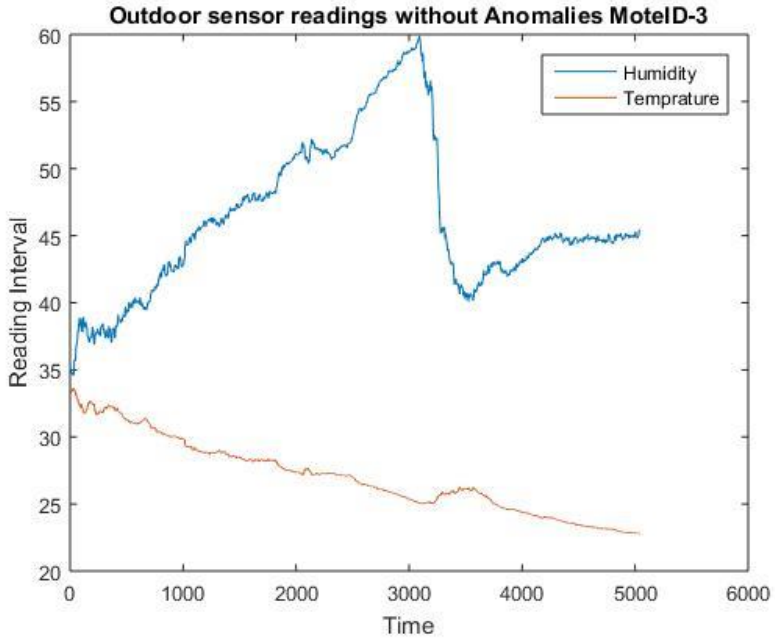

Figure 4.7 MoteID- 3

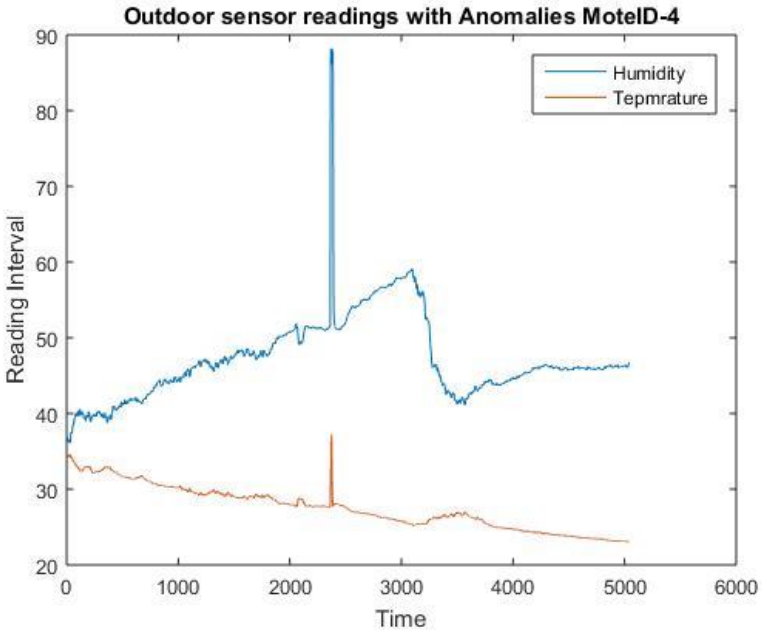

Figure 4.8 MoteID- 4

Multi-Hop Indoor Sensor Data Analysis

The indoor multi-hop data collection scenario consists of two motes placed outdoors. MoteID -4 (Figure 4.9) collects normal measurements whereas MoteID -5 (Figure 4.10) is introduced with anomalous data.

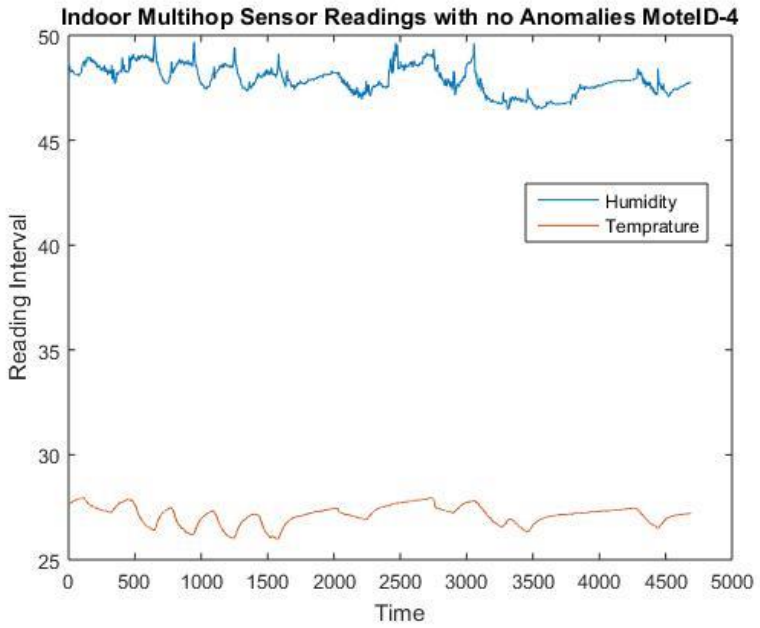

Figure 4.9 MoteID-4

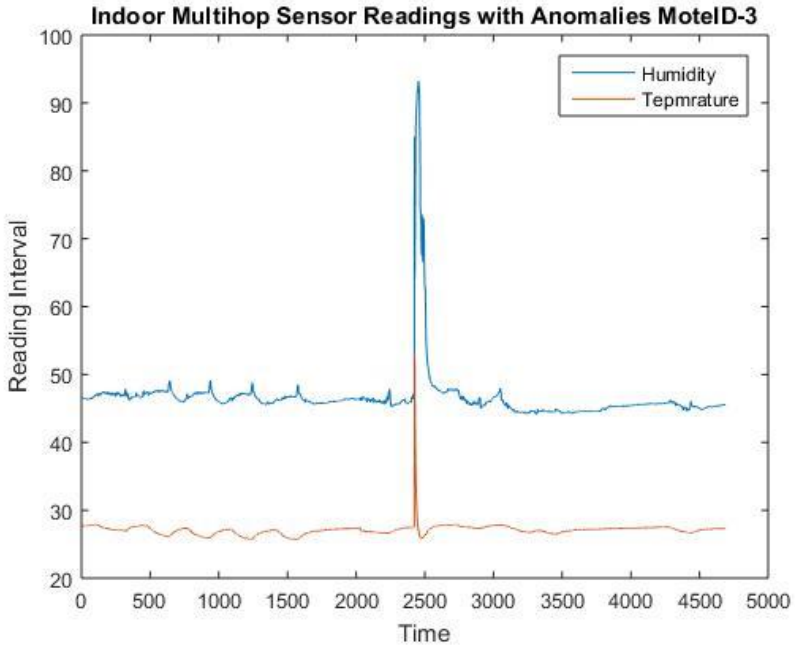

Figure 4.10 MoteID-3

Multi-Hop Outdoor Sensor Data Analysis

The outdoor multi-hop data collection scenario also consists of two motes placed outdoors. MoteID -2 (Figure 4.11) collects normal measurements whereas MoteID -1 (Figure 4.12) is introduced with anomalous data.

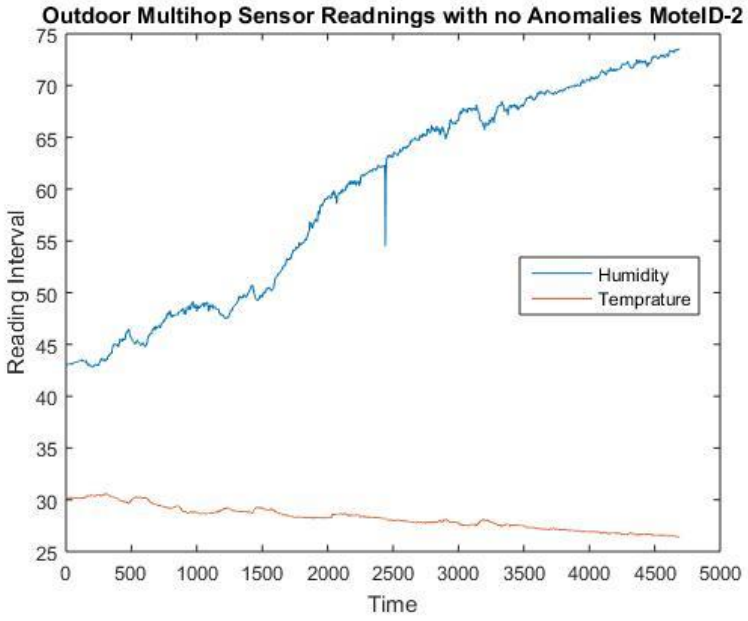

Figure 4.11 MoteID-2

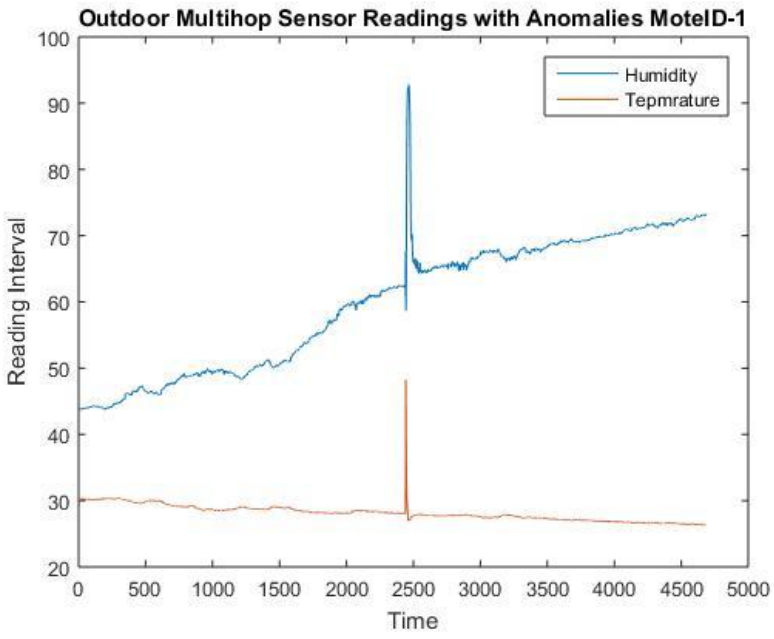

Figure 4.12 MoteID-1

4.5 Modeling the Data: Supervised Machine Learning

Supervised Anomaly detection has already been explained in the section 4.2 as an approach to build a predictive model for normal vs. anomaly classes. Any unseen data instance is compared against the model to determine which class it belongs to.

Supervised learning algorithm is also explained in [33] as "an algorithm that takes a known set of input data and known responses to the data (output), and trains a model to generate reasonable predictions for the response to new data." The Figure 4.13 explains the modelling process.

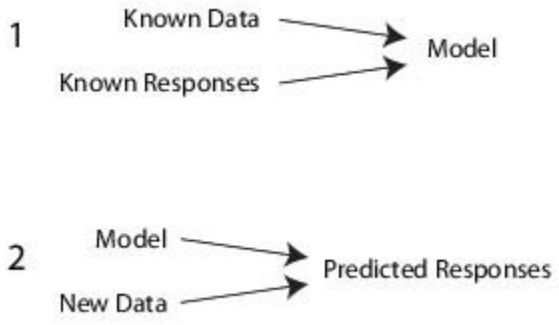

Figure 4.13 Supervised AD

Supervised learning can be categorized in two groups as:

A. Classification. A class or label from a finite set of classes is assigned to an observation.

B. Regression. It is used for prediction or forecasting by continuously measuring an observation.

4.5.1 Classification Learner App [34]

The Classification Learner app provided by Matlab Software is used to train models to classify data using supervised machine learning. The Classification Learner app is used to train models in order to classify data. Supervised Machine Learning can be performed by applying following parameters:

i. Import data. Imports the dataset into Classification Learner app.

ii. Feature Selection. Selection of Predictors and Response variable

iii. Chose a Classifier. In order to train the data, any or all of the classifiers listed below can be selected to see which classifier provides the best model.

a. Decision Trees

b. Support Vector Machines

c. Nearest Neighbor Classifier

d. Ensemble Classifier

iv. Generate code. The Matlab code can be generated for the best trained model

v. Export Model. The best trained model can also be exported in the Matlab workspace. This model can be used for predictive analysis with a different dataset.

Features Selection

The dataset as explained in the section 4.2 above is imported in the Classification Learner app for the purpose of modelling. For the feature selection process Predictors and Response has to be selected. Since our dataset is based on the collection of "Humidity" and "Temperature", hence these are our prime Predictors and the Response value is based on the "Label". After the selection of Predictors and Response, validation option is selected.

Training the data

Once the feature selection is done. Now it is the stage to model the dataset. The Classification Learner app provides the following Machine Learning Techniques or classifiers to analyze and create a model.

a. Decision Trees
b. Support Vector Machines
c. Nearest Neighbor Classifier
d. Ensemble Classifier

4.6 Conclusion

The labelled dataset of Humidity and Temperature has provided a good opportunity for learning and experiment. The graphs are plotted based on the datasets and normal vs anomalous data can visually be seen in the graphs. The normal behavior of the data collection has label = 0 whereas the anomaly is defined as label =1. Supervised Anomaly Detection approach is best suited for such data collection environment where a normal model can be established based on the response feature. The trained model can be applied to the new dataset for the purpose of AD.

5 Data Modelling and Results

5.1 Introduction

This chapter discusses the data modelling process and the results obtained. Section 5.2 explains how the dataset of two scenarios i.e. Single-hop and Multi-hop dataset is modelled using the "Classifier Learner App". The steps are explained graphically as much as possible.

5.2 Data Models

The data modelling results for the different data collection scenarios by using Classifier Learning App are detailed below. The corresponding code will be available in the Appendix section.

Single-Hop Indoor Sensor Data

The Classifier Learning App provides the option to select from various classifiers in order to find out the best model for future analysis with the same or new dataset. Figure 5.1 shows the various classifiers available. This process is used for our first scenario i.e. Single-Hop Indoor dataset.

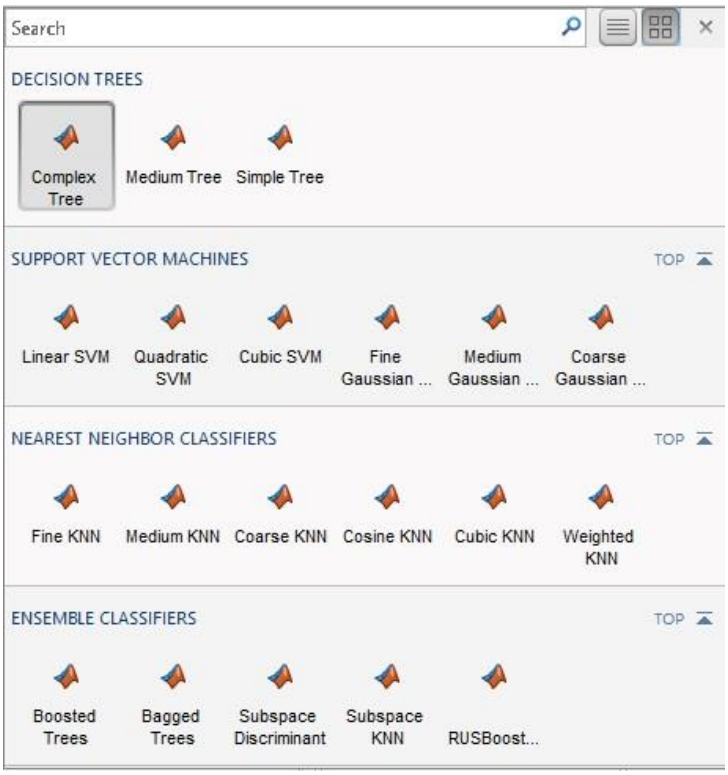

Figure 5.1 Classifiers

The Figure 5.2 shows the results of the tested classifiers. The variables of x-asis is Humidity and on y-axis is Temperature. Four classifiers i.e. Complex Tree, SVM (Support Vector Machine), KNN (K Nearest Neighbor) and Ensemble were selected for modelling. Out of the tested classifiers, the "Classifier Learning App" recommends best model as KNN (K Nearest Neighbor) based on the dataset.

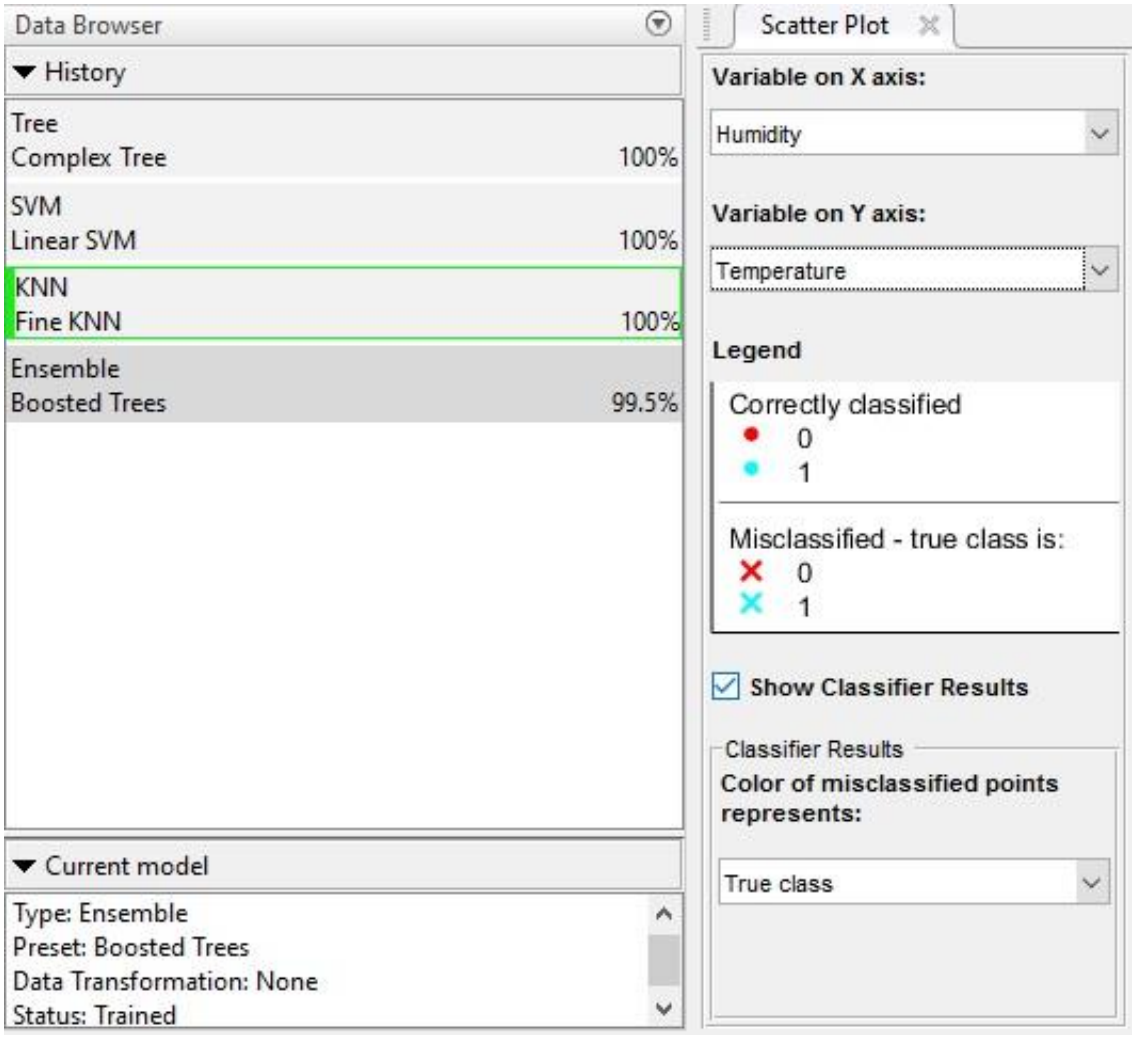

Figure 5.2 Classifier Learner App: Modelling options for Single-hop Indoor Sensor dataset

The graph below is based on the dataset of "Single-hop Indoor MoteID-1" with Anomalies. It has been drawn according to the KNN classifier as recommended best algorithm by Classifier Learner App.

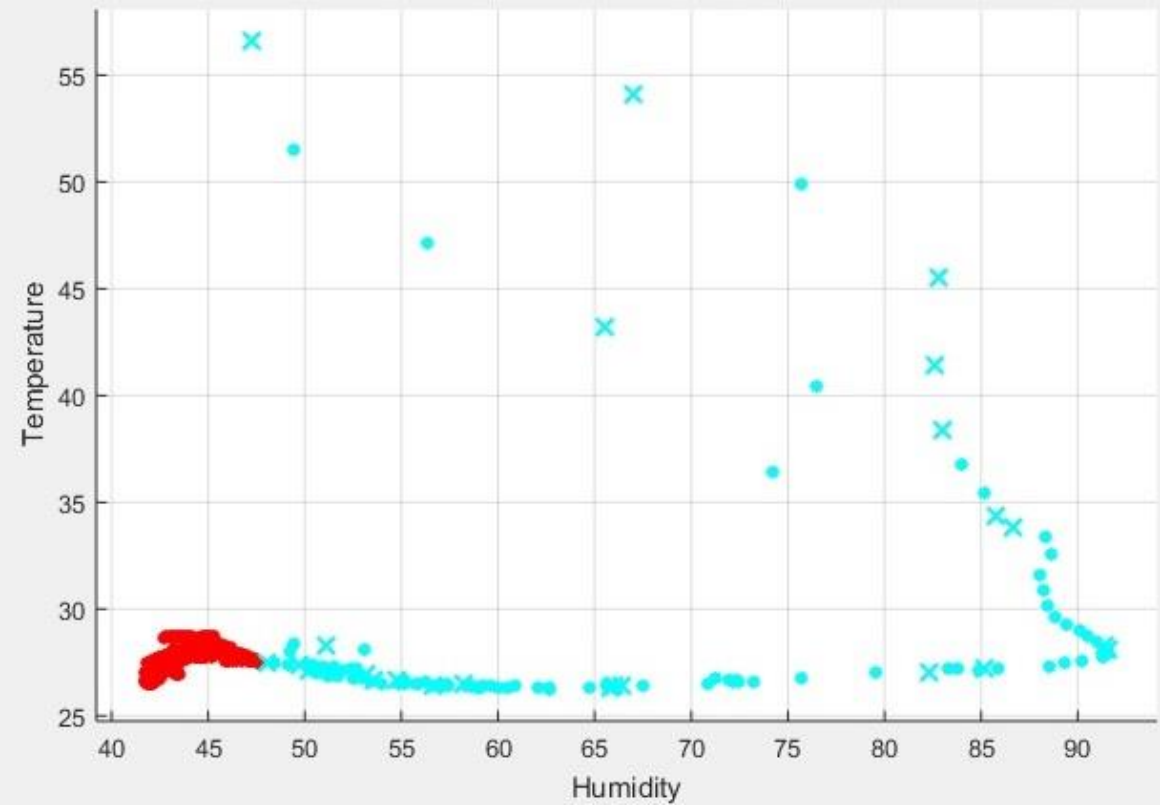

Figure 5.3 Single-hop Indoor MoteID-1 with Anomalies

Single-Hop Outdoor Sensor Data

The Figure 5.4 shows the results of the tested classifiers for Single-hop outdoor dataset. Out of the four classifiers KNN (K Nearest Neighbor) is recommended best model.

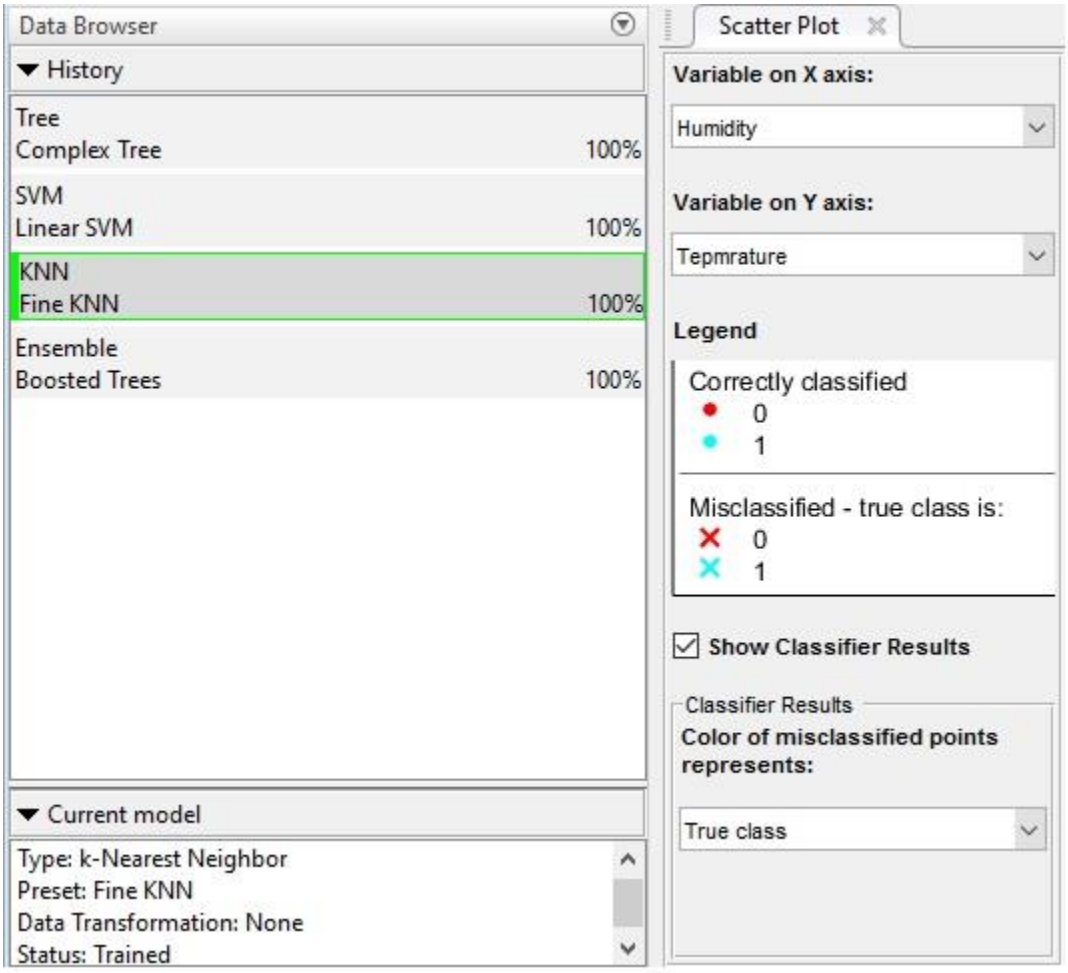

Figure 5.4 Classifier Learner App: Modelling options for Single-hop Outdoor Sensor dataset

The graph below (Figure 5.5) is based on the dataset of "Single-hop Outdoor MoteID-4" with Anomalies. It has been drawn according to the KNN classifier as recommended best algorithm by Classifier Learner App.

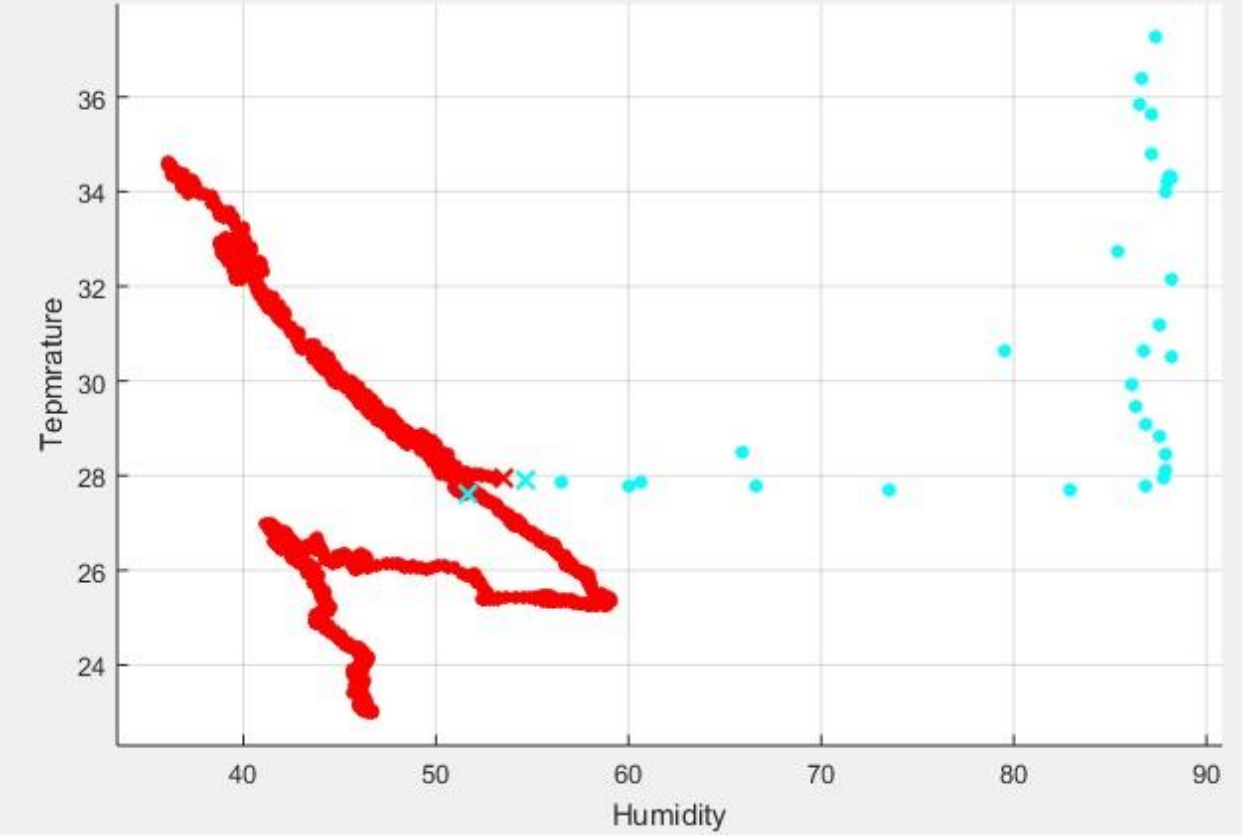

Figure 5.5 Single-hop Outdoor MoteID-4 with Anomalies

Multi-Hop Indoor Sensor Data

The Figure 5.6 shows the results of the tested classifiers for Multi-hop Indoor dataset. Out of the four classifiers KNN (K Nearest Neighbor) is recommended best model.

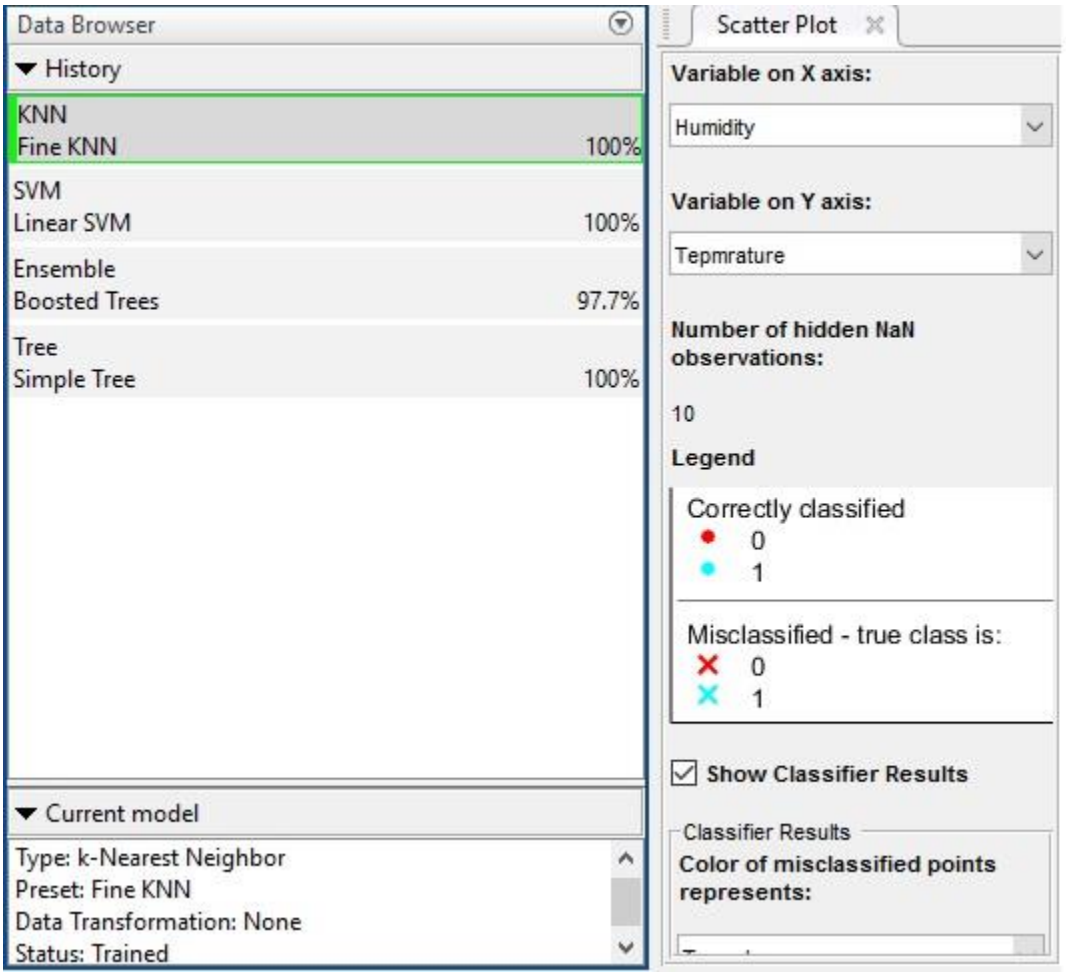

Figure 5.6 Classifier Learner App: Modelling options for Multi-hop Indoor Sensor dataset

The graph below (Figure 5.7) is based on the dataset of "Multi-hop Indoor MoteID-3" with Anomalies. It has been drawn according to the KNN classifier as recommended best algorithm by Classifier Learner App.

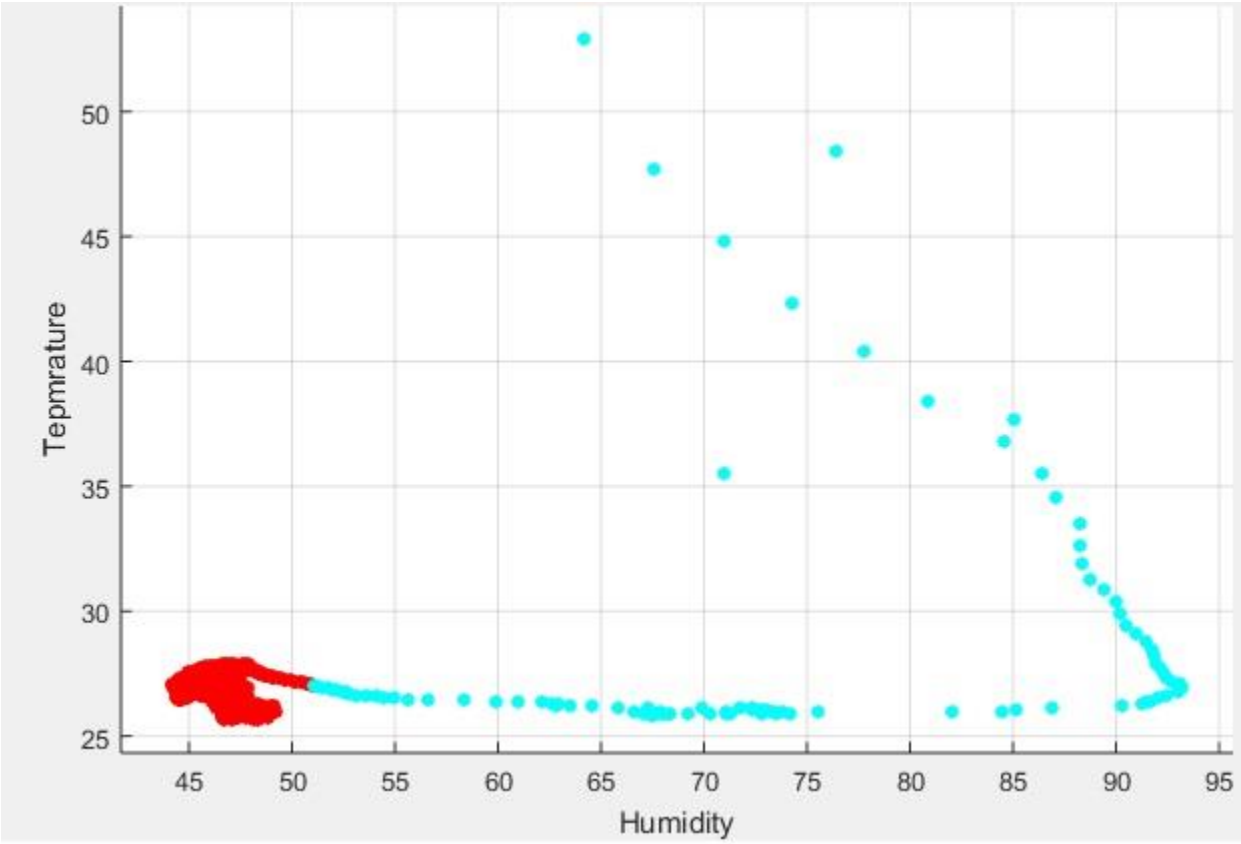

Figure 5.7 Multi-hop Indoor MoteID-3 with Anomalies

Multi-Hop Outdoor Sensor Data

The Figure 5.8 shows the results of the tested classifiers for Multi-hop Outdoor dataset. Out of the four classifiers KNN (K Nearest Neighbor) is recommended best model.

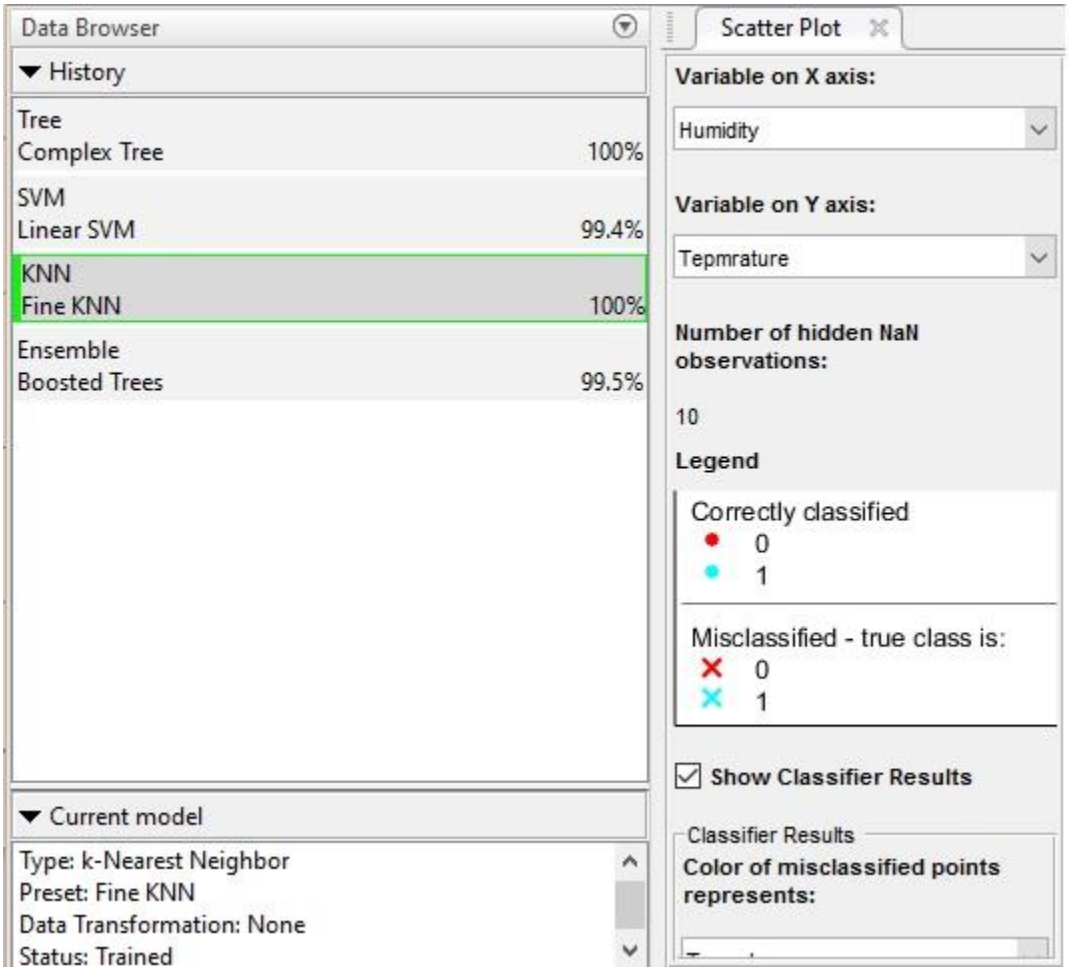

Figure 5.8 Classifier Learner App: Modelling options for Multi-hop Outdoor Sensor dataset

The graph below (Figure 5.9) is based on the dataset of "Multi-hop Outdoor MoteID-1" with Anomalies. It has been drawn according to the KNN classifier as recommended best algorithm by Classifier Learner App.

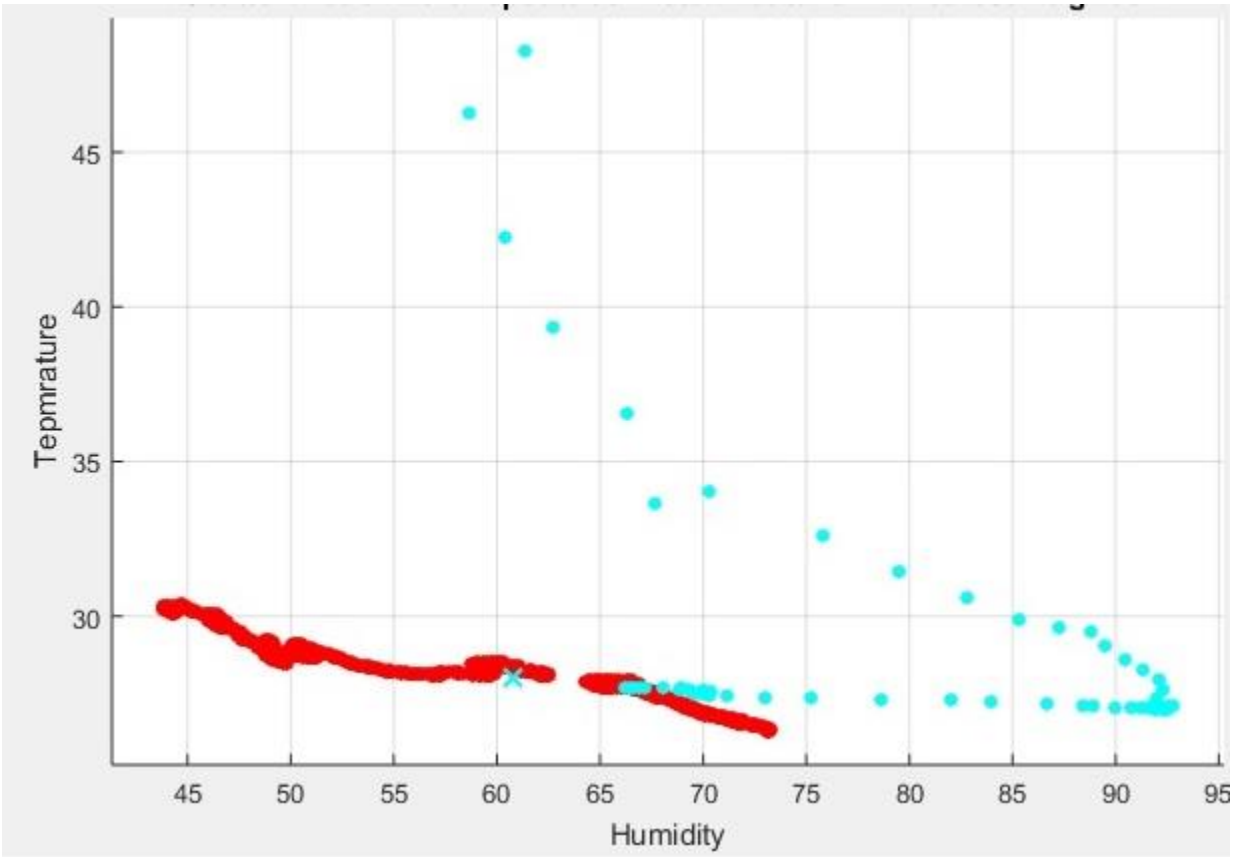

Figure 5.9 Multi-hop Outdoor MoteID-1 with Anomalies

5.3 Comparison of the results

The Classification Learner App provides the machine learning techniques i.e. Decision Trees (DT), Support Vector Machines (SVM), Nearest Neighbor Classifiers (NNC) and Ensemble Classifiers (EC) for training the data. All of the above

mentioned approaches were applied in the training phase. Hence the analysis results in Table 1 show that KNN gives best AD results in our labelled data environment.

Table 1 Data Analysis Results

Index	WSN Environment	AD Techniques Applied	AD Technique Recommended
1	Single hop Indoor MoteID-1	DT, SVM, NNC, EC	KNN
2	Single hop Outdoor MoteID-4	DT, SVM, NNC, EC	KNN
3	Multi-hop Indoor MoteID-3	DT, SVM, NNC, EC	KNN
4	Multi-hop Outdoor MoteID-1	DT, SVM, NNC, EC	KNN

6 Conclusion

In this thesis work, the author studied the WSNs in detail. The components, radio standards, architecture, protocol stack, applications and security issues of WSNs. The study has showed that sensor technology has revolutionized our life so much that nowadays we carry sensors almost all the times with us in the form of smartphones that can actually sense our body motions etc. At the same time they benefit us being part of the security surveillance networks as well as in hospitals measuring and processing patient data. Hence they are also prone to all types of hacker attacks especially once they are connected to the internet.

Anomaly detection techniques are very interesting to study. However no one technique is suitable for every type of WSN. The background information about the WSN especially the type of network and the data and purpose of the system analysis guides us about which AD Technique is to be selected.

Hence for the purpose of AD in WSN, Labelled data set was selected. The literature review recommended that Supervised Machine Learning Techniques provide best AD results. The author has used Matlab Software that provides state of the art integrated development environment for the purpose of data analysis. The "Classification Learner app" provides data modelling facility. The modelling phase allows various techniques to train the data. Once the data is trained, the best model can be exported and used to test against new data set. Our experience has showed that K-Nearest Neighbor (KNN) proves to be the most suitable technique to detect anomalous data.

In future, more complex and multiclass data set can be tested for learning purposes.

REFERENCES

[1] Akyildiz, I. F., & Vuran, M. C. (2010). Wireless sensor networks (Vol. 4). John Wiley & Sons.

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

[3] Perrig, A., Stankovic, J., & Wagner, D. (2004). Security in wireless sensor networks. Communications of the ACM, 47(6), 53-57.

[4] Li, Y. X., Qin, L., & Liang, Q. (2010, December). Research on wireless sensor network security. In Computational Intelligence and Security (CIS), 2010 International Conference on (pp. 493-496). IEEE.

[5] Chris, T., & Steven, A. (2011). Wireless Sensor Networks: Principles and Applications.

[6] Benenson, Z., Cholewinski, P. M., & Freiling, F. C. (2008). Vulnerabilities and attacks in wireless sensor networks. Wireless Sensors Networks Security, 22-43.

[7] Modares, H., Salleh, R., & Moravejosharieh, A. (2011, September). Overview of security issues in wireless sensor networks. In 2011 Third International Conference on Computational Intelligence, Modelling & Simulation (pp. 308-311). IEEE.

[8] Patcha, A., & Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer networks, 51(12), 3448-3470.

[9] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), 15.

[10] Smaha, S. E. (1988, December). Haystack: An intrusion detection system. In Aerospace Computer Security Applications Conference, 1988., Fourth (pp. 37-44). IEEE.

[11] Anderson, D., Frivold, T., & Tamaru, A. (1994). A. Valdes, Next Generation Intrusion Detection Expert System (NIDES). Software Users Manual, Beta-Update release, Computer Science Laboratory, SRI International, Menlo Park, CA, USA, Technical Report SRI-CSL-95-0. Chicago

[12] Staniford, S., Hoagland, J. A., & McAlerney, J. M. (2002). Practical automated detection of stealthy portscans. Journal of Computer Security, 10(1-2), 105-136. Chicago

[13] Ye, N., Emran, S. M., Chen, Q., & Vilbert, S. (2002). Multivariate statistical analysis of audit trails for host-based intrusion detection. IEEE Transactions on Computers, 51(7), 810-820.

[14] Forrest, S., Hofmeyr, S. A., Somayaji, A., & Longstaff, T. A. (1996, May). A sense of self for unix processes. In Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on (pp. 120-128). IEEE.

[15] Eskin, E., Lee, W., & Stolfo, S. J. (2001). Modeling system calls for intrusion detection with dynamic window sizes. In DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01. Proceedings (Vol. 1, pp. 165-175). IEEE.

[16] Valdes, A., & Skinner, K. (2000, October). Adaptive, model-based monitoring for cyber attack detection. In International Workshop on Recent Advances in Intrusion Detection (pp. 80-93). Springer Berlin Heidelberg.

[17] Shyu, M. L., Chen, S. C., Sarinnapakorn, K., & Chang, L. (2003). A novel anomaly detection scheme based on principal component classifier. MIAMI UNIV CORAL GABLES FL DEPT OF ELECTRICAL AND COMPUTER ENGINEERING.

[18] Yeung, D. Y., & Ding, Y. (2003). Host-based intrusion detection using dynamic and static behavioral models. Pattern recognition, 36(1), 229-243.

[19 Mahoney, M. V., & Chan, P. K. (2001). PHAD: Packet header anomaly detection for identifying hostile network traffic.

[20] Mahoney, M. V., & Chan, P. K. (2002). Learning models of network traffic for detecting novel attacks.

[21] Mahoney, M. V., & Chan, P. K. (2002, July). Learning nonstationary models of normal network traffic for detecting novel attacks. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 376-385). ACM.

[22] Lee, W., & Stolfo, S. J. (1998, January). Data Mining Approaches for Intrusion Detection. In Usenix security.

[23] Kumar, S., & Spafford, E. H. (1994). An application of pattern matching in intrusion detection.

[24] Dickerson, J. E., & Dickerson, J. A. (2000). Fuzzy network profiling for intrusion detection. In Fuzzy Information Processing Society, 2000. NAFIPS. 19th International Conference of the North American (pp. 301-306). IEEE.

[25] Ramadas, M., Ostermann, S., & Tjaden, B. (2003, September). Detecting anomalous network traffic with self-organizing maps. In *International Workshop on Recent Advances in Intrusion Detection* (pp. 36-54). Springer Berlin Heidelberg.

[26] Ertoz, L., Eilertson, E., Lazarevic, A., Tan, P. N., Kumar, V., Srivastava, J., & Dokas, P. (2004). Minds-minnesota intrusion detection system. Next generation data mining, 199-218. Chicago

[27] Barbará, D., Couto, J., Jajodia, S., & Wu, N. (2001). ADAM: a testbed for exploring the use of data mining in intrusion detection. *ACM Sigmod Record*, *30*(4), 15-24.

[28] Rajasegarar, S., Leckie, C., & Palaniswami, M. (2008). Anomaly detection in wireless sensor networks. *IEEE Wireless Communications*, *15*(4), 34-40.

[29] Suthaharan, S., Alzahrani, M., Rajasegarar, S., Leckie, C., & Palaniswami, M. (2010, December). Labelled data collection for anomaly detection in wireless sensor networks. In *Intelligent sensors, sensor networks and information processing (ISSNIP), 2010 sixth international conference on* (pp. 269-274). IEEE.

[30] http://www.uncg.edu/cmp/downloads/

[31] http://www.willow.co.uk/TelosB_Datasheet.pdf

[32] http://tinyos.stanford.edu/tinyos-wiki/index.php/TinyOS_Tutorials

[33] https://se.mathworks.com/help/stats/supervised-learning-machine-learning-workflow-and-algorithms.html

[34] https://se.mathworks.com/help/stats/classification-learner-app.html?searchHighlight=classification%20learner

[35] Janakiram, D., Reddy, V. A., & Kumar, A. P. (2006, January). Outlier detection in wireless sensor networks using Bayesian belief networks. In 2006 1st International Conference on Communication Systems Software & Middleware (pp. 1-6). IEEE.

[36] Branch, J. W., Giannella, C., Szymanski, B., Wolff, R., & Kargupta, H. (2013). In-network outlier detection in wireless sensor networks. Knowledge and information systems, 34(1), 23-54.

[37] Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V., & Gunopulos, D. (2006, September). Online outlier detection in sensor data using non-parametric models. In Proceedings of the 32nd international conference on Very large data bases (pp. 187-198). VLDB Endowment.

[38] Zhang, K., Shi, S., Gao, H., & Li, J. (2007, August). Unsupervised outlier detection in sensor networks using aggregation tree. In International Conference on Advanced Data Mining and Applications (pp. 158-169). Springer Berlin Heidelberg.

[39] Idé, T., Papadimitriou, S., & Vlachos, M. (2007, October). Computing correlation anomaly scores using stochastic nearest neighbors. In Seventh IEEE International Conference on Data Mining (ICDM 2007) (pp. 523-528). IEEE.

[40] Chatzigiannakis, V., Papavassiliou, S., Grammatikou, M., & Maglaris, B. (2006, June). Hierarchical anomaly detection in distributed large-scale sensor networks. In 11th IEEE Symposium on Computers and Communications (ISCC'06) (pp. 761-767). IEEE.

[41] Eskin, E., Arnold, A., Prerau, M., Portnoy, L., & Stolfo, S. (2002). A geometric framework for unsupervised anomaly detection. In Applications of data mining in computer security (pp. 77-101). Springer US.

[42] Bamnett, V., & Lewis, T. (1994). Outliers in statistical data. Chicago

[43] Alpaydin, E. (2014). Introduction to machine learning. MIT press.

APPENDIX

A The Code for Graph Plots

Figure 4.5

```
function createfigure(YMatrix1)
%CREATEFIGURE(YMATRIX1)
%  YMATRIX1:  matrix of y data

%  Auto-generated by MATLAB

% Create figure
figure1 = figure;

% Create axes
axes1 = axes('Parent',figure1);
box(axes1,'on');
hold(axes1,'on');

% Create multiple lines using matrix input to plot
plot1 = plot(YMatrix1);
set(plot1(1),...
   'DisplayName','Singlehopindoormoteid1data_WithAnomalies.Humidity');
set(plot1(2),...
   'DisplayName','Singlehopindoormoteid1data_WithAnomalies.Temperature');
```

Figure 4.6

```
function createfigure(YMatrix1)
%CREATEFIGURE(YMATRIX1)
%  YMATRIX1:  matrix of y data

%  Auto-generated by MATLAB

% Create figure
figure1 = figure;

% Create axes
axes1 = axes('Parent',figure1);
box(axes1,'on');
hold(axes1,'on');

% Create multiple lines using matrix input to plot
plot1 = plot(YMatrix1);
set(plot1(1),'DisplayName','singlehopindoormoteid2data.Humidity');
set(plot1(2),'DisplayName','singlehopindoormoteid2data.Temprature');
```

Figure 4.7

```
function createfigure(YMatrix1)
%CREATEFIGURE(YMATRIX1)
%  YMATRIX1:  matrix of y data
```

```
%  Auto-generated by MATLAB

% Create figure
figure1 = figure;

% Create axes
axes1 = axes('Parent',figure1);
box(axes1,'on');
hold(axes1,'on');

% Create multiple lines using matrix input to plot
plot1 = plot(YMatrix1);
set(plot1(1),'DisplayName','singlehopoutdoormoteid3data.Humidity');
set(plot1(2),'DisplayName','singlehopoutdoormoteid3data.Temprature');
```

Figure 4.8

```
function createfigure(YMatrix1)
%CREATEFIGURE(YMATRIX1)
%  YMATRIX1:  matrix of y data

%  Auto-generated by MATLAB

% Create figure
figure1 = figure;

% Create axes
```

```
axes1 = axes('Parent',figure1);
box(axes1,'on');
hold(axes1,'on');

% Create multiple lines using matrix input to plot
plot1 = plot(YMatrix1);
set(plot1(1),'DisplayName','singlehopoutdoormoteid4data.Humidity');
set(plot1(2),'DisplayName','singlehopoutdoormoteid4data.Tepmrature');
```

Figure 4.9

```
function createfigure(YMatrix1)
%CREATEFIGURE(YMATRIX1)
%  YMATRIX1:  matrix of y data

%  Auto-generated by MATLAB

% Create figure
figure1 = figure;

% Create axes
axes1 = axes('Parent',figure1);
box(axes1,'on');
hold(axes1,'on');

% Create multiple lines using matrix input to plot
plot1 = plot(YMatrix1);
```

```
set(plot1(1),'DisplayName','multihopindoormoteid4data.Humidity');
set(plot1(2),'DisplayName','multihopindoormoteid4data.Temprature');
```

Figure 4.10

```
function createfigure(YMatrix1)
%CREATEFIGURE(YMATRIX1)
%  YMATRIX1:  matrix of y data

%  Auto-generated by MATLAB

% Create figure
figure1 = figure;

% Create axes
axes1 = axes('Parent',figure1);
box(axes1,'on');
hold(axes1,'on');

% Create multiple lines using matrix input to plot
plot1 = plot(YMatrix1);
set(plot1(1),'DisplayName','multihopindoormoteid3data.Humidity');
set(plot1(2),'DisplayName','multihopindoormoteid3data.Tepmrature');
```

Figure 4.11

```
function createfigure(YMatrix1)
```

```
%CREATEFIGURE(YMATRIX1)
%  YMATRIX1:  matrix of y data

%  Auto-generated by MATLAB on 19-Dec-2016 23:50:54

% Create figure
figure1 = figure;

% Create axes
axes1 = axes('Parent',figure1);
box(axes1,'on');
hold(axes1,'on');

% Create multiple lines using matrix input to plot
plot1 = plot(YMatrix1);
set(plot1(1),'DisplayName','multihopoutdoormoteid2data.Humidity');
set(plot1(2),'DisplayName','multihopoutdoormoteid2data.Temprature');
```

Figure 4.12

```
function createfigure(YMatrix1)
%CREATEFIGURE(YMATRIX1)
%  YMATRIX1:  matrix of y data

%  Auto-generated by MATLAB on 19-Dec-2016 23:51:57

% Create figure
```

```
figure1 = figure;

% Create axes
axes1 = axes('Parent',figure1);
box(axes1,'on');
hold(axes1,'on');

% Create multiple lines using matrix input to plot
plot1 = plot(YMatrix1);
set(plot1(1),'DisplayName','multihopoutdoormoteid1data.Humidity');
set(plot1(2),'DisplayName','multihopoutdoormoteid1data.Tepmrature');
```

B Code generated after training the classifier models

Figure 5.3

```
function [trainedClassifier, validationAccuracy] = trainClassifier(datasetTable)
% Extract predictors and response
predictorNames = {'Reading', 'MoteID', 'Humidity', 'Temperature'};
predictors = datasetTable(:,predictorNames);
predictors = table2array(varfun(@double, predictors));
response = datasetTable.Label;
% Train a classifier
trainedClassifier = fitensemble(predictors, response, 'AdaBoostM1', 200, 'Tree', 'Type',
'Classification', 'LearnRate', 1.000000e-01, 'PredictorNames', {'Reading' 'MoteID'
'Humidity' 'Temperature'}, 'ResponseName', 'Label', 'ClassNames', [0 1]);
```

```
% Perform cross-validation
partitionedModel = crossval(trainedClassifier, 'KFold', 5);

% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun', 'ClassifError');

%% Uncomment this section to compute validation predictions and scores:
% % Compute validation predictions and scores
% [validationPredictions, validationScores] = kfoldPredict(partitionedModel);
```

Figure 5.5

```
function [trainedClassifier, validationAccuracy] = trainClassifier(datasetTable)
% Extract predictors and response
predictorNames = {'Reading', 'MoteID', 'Humidity', 'Tepmrature'};
predictors = datasetTable(:,predictorNames);
predictors = table2array(varfun(@double, predictors));
response = datasetTable.Label;
% Train a classifier
trainedClassifier = fitensemble(predictors, response, 'AdaBoostM1', 200, 'Tree', 'Type',
'Classification', 'LearnRate', 1.000000e-01, 'PredictorNames', {'Reading' 'MoteID'
'Humidity' 'Tepmrature'}, 'ResponseName', 'Label', 'ClassNames', [0 1]);

% Perform cross-validation
partitionedModel = crossval(trainedClassifier, 'KFold', 5);

% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun', 'ClassifError');
```

```
%% Uncomment this section to compute validation predictions and scores:
% % Compute validation predictions and scores
% [validationPredictions, validationScores] = kfoldPredict(partitionedModel);
```

Figure 5.7

```
function [trainedClassifier, validationAccuracy] = trainClassifier(datasetTable)
% Extract predictors and response
predictorNames = {'Reading', 'MoteID', 'Humidity', 'Tepmrature'};
predictors = datasetTable(:,predictorNames);
predictors = table2array(varfun(@double, predictors));
response = datasetTable.Label;
% Train a classifier
trainedClassifier = fitcknn(predictors, response, 'PredictorNames', {'Reading' 'MoteID'
'Humidity' 'Tepmrature'}, 'ResponseName', 'Label', 'ClassNames', [0 1], 'Distance',
'Euclidean', 'Exponent', '', 'NumNeighbors', 1, 'DistanceWeight', 'Equal',
'StandardizeData', 1);


% Perform cross-validation
partitionedModel = crossval(trainedClassifier, 'KFold', 5);


% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun', 'ClassifError');


%% Uncomment this section to compute validation predictions and scores:
% % Compute validation predictions and scores
% [validationPredictions, validationScores] = kfoldPredict(partitionedModel);
```

Figure 5.9

```
function [trainedClassifier, validationAccuracy] = trainClassifier(datasetTable)
% Extract predictors and response
predictorNames = {'Reading', 'MoteID', 'Humidity', 'Tepmrature'};
predictors = datasetTable(:,predictorNames);
predictors = table2array(varfun(@double, predictors));
response = datasetTable.Label;
% Train a classifier
trainedClassifier = fitcknn(predictors, response, 'PredictorNames', {'Reading' 'MoteID'
'Humidity' 'Tepmrature'}, 'ResponseName', 'Label', 'ClassNames', [0 1], 'Distance',
'Euclidean', 'Exponent', '', 'NumNeighbors', 1, 'DistanceWeight', 'Equal',
'StandardizeData', 1);


% Perform cross-validation
partitionedModel = crossval(trainedClassifier, 'KFold', 5);


% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun', 'ClassifError');


%% Uncomment this section to compute validation predictions and scores:
% % Compute validation predictions and scores
% [validationPredictions, validationScores] = kfoldPredict(partitionedModel);
```