TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

THESIS NO.: 072/MSCS/659

Sensor Network Anomaly Detection Model by cascading Inverse Weight Clustering and C5.0 Decision Tree

by

Pramod Kumar Chaudhary

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SYSTEM AND KNOWLEDGE ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

LALITPUR, NEPAL

NOVEMBER, 2019

Sensor Network Anomaly Detection Model by cascading Inverse Weight Clustering and C5.0 Decision Tree

by

Pramod Kumar Chaudhary

072/MSCS/659

Thesis Supervisor

Dr. Arun Kumar Timalsina

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer System and Knowledge Engineering

Department of Electronics and Computer Engineering

Institute of Engineering, Pulchowk Campus

Tribhuvan University

Lalitpur, Nepal

NOVEMBER, 2019

COPYRIGHT

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

RECOMMENDATION

The undersigned certify that it has been read and recommended to the Department of Electronics and Computer Engineering for acceptance, a report of thesis entitled **"Sensor Network Anomaly Detection Model by cascading Inverse Weight Clustering and C5.0 Decision Tree"**, submitted by **Mr. Pramod Kumar Chaudhary** in partial fulfillment of the requirement for the award of the degree of "**Master of Science in Computer System and Knowledge Engineering**".

_____

Supervisor, Dr. Arun Kumar Timalsina
Department of Electronics and Computer Engineering

_____

External Examiner, Saurav Dhungana
Lead Engineer, Leapfrog technologies, Inc.

_____

Committee Chairperson, Dr. Aman Shakya
Program Coordinator
Department of Electronics and Computer Engineering
Institute of Engineering, Pulchowk Campus

Date ----------

DEPARTMENTAL ACCEPTANCE

The thesis entitled **"Sensor Network Anomaly Detection Model by cascading Inverse Weight Clustering and C5.0 Decision Tree",** submitted by **Mr. Pramod Kumar Chaudhary** in partial fulfillment of the requirement for the award of the degree of "**Master of Science in Computer System and Knowledge Engineering**" has been accepted as a bonafide record of work independently carried out by him in the department.

Dr. Surendra Shrestha
Head of the Department
Department of Electronics and Computer Engineering,
Pulchowk Campus,
Institute of Engineering,
Tribhuvan University, Nepal.

ACKNOWLEDGEMENTS

First I would like to express my sincere thanks to supervisor, Dr**. Arun Kumar Timalsina** for his immense guidance, inspiration and encouragement as it was great pleasure to work under his supervision. I would like express special thanks to our Program Coordinator **Dr. Aman Shakya,** Head of Department **Dr. Surendra Shrestha, Prof. Dr. Sashidhar Ram Joshi, Prof. Dr. Subarna Shakya, Dr. Dibakar Raj Panta**, **Dr. Sanjeeb Prasad Panday, Dr. Basanta Joshi,** and **Mr. Baburam Dawadi** for helping along the way giving me the precious guidance during the thesis title selection phase.

Last but not the least, I would like to thank Mr. Pravesh Koirala, Mr. Raju Shrestha, and all my class mates for their direct and indirect support and encouragement.

ABSTRACT

Wireless Sensor Network is a network of integrated sensors responsible for environmental sensing, data processing and communication with other sensors and the base station while consuming low power. 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 does not conform to the normal behavior of the network communication.

Anomaly detection in wireless sensor network using Inverse Weighted Clustering and C5.0 Decision tree, a method for classifying anomalous and normal activities have been proposed. The IWC clustering method is first used to partition the training instances into k clusters using Euclidean distance similarity. On each cluster, representing a density region of normal or anomaly instances, decision trees are built using C5.0 decision tree algorithm. The decision tree on each cluster refined the decision boundaries by learning the subgroups within the cluster. The experiment was carried out on three datasets (University of North Carolina Greensboro (UNCG), Intel Berkeley Research Lab (IBRL) and Bharatpur Airport WSN). The results show that proposed method achieved detection rate of 98.9% at false alarm-rate of 0.31% on IBRL; detection rate of 99.57 % at false alarm-rate of 0.35% on Bharatpur Airport.

Keywords: WSN; Anomaly detection; IWC clustering; C5.0 decision tree;

TABLE OF CONTENTS

List of Figures

List of Tables

List of Acronyms

ADP	Anomaly Detection Process
ADS	Anomaly Detection System
AI	Artificial Intelligent
ARQ	Automatic Repeat Request
ART	Adaptive Resonance Theory
CHAID	Chi-squared Automatic Interaction Detector
DARPA	Defense Advanced Research Projects Agency
DT	Decision Tree
EM	Expectation-Maximization Meta Algorithm
FCM	Fuzzy C-Means
FEC	Forward Error Correction
HIDS	Host Based IDS
IBRL	Intel Berkeley Research lab
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
IWC	Inverse Weight Clustering
KNN	K-Nearest Neighbor
MAC	Media Access Control
ML	Machine Learning
NIDS	Network Based IDS
OSI	Open System Interconnection
ROC	Receiver Operating Curves
SOM	Self-Organizing Map
SVM	Support Vector Machine
UNC	Unsupervised Niche Clustering
UNCG	University of North Carolina at Greensboro
WSN	Wireless Sensor Network

CHAPTER 1: INTRODUCTION

1.1 Background

Wireless sensor networks (WSNs) have become a popular area of research in recent years due to their huge potential to be used in various applications. They have been used with success in critical application scenarios, such as remote patient health monitoring, environmental monitoring, structural monitoring of engineering structures and military surveillance, where the dependability of WSNs becomes an important factor. A number of sensors can be used to monitor and collect information from the environment and send the information to a central location. WSNs can be densely distributed over a geographical area and individual nodes can autonomously communicate and interact with each other over the wireless medium [1].

WSNs are highly vulnerable to attacks, due to their open and distributed nature and limited resources of the sensor nodes. Moreover, in WSNs packets broadcasting have to be done frequently, sensor nodes can be deployed randomly in an environment so an attacker adversary can be easily injected to a WSN [2].

The information obtained from the WSNs has to be accurate and complete. Analysis of data collected from sensor at timely manner is of high importance. Raw data collected from WSN often suffer from inaccuracy and incompleteness. Inaccurate or incomplete data measurements of WSN are often known as WSN anomalies. Anomalies in WSNs can be caused by errors, malfunctioning or failure of nodes and attacks [1]. Anomaly may be caused by not only faulty sensor node but also security threats in the network or unusual phenomena in the monitoring scope. Therefore, it is very important that the anomaly of sensor node is detected in order to obtain accurate information and make effective decisions by information gatherers [2].

A key function of a Sensor Network is the analysis of data that is generated in the form of measurements by sensor nodes. One objective of data analysis is anomaly detection. The aim of anomaly detection is to identify data that do not conform to the patterns exhibited by the majority of the data set. An anomaly or outlier is defined as

“an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data” [3]. Algorithms that perform anomaly detection construct a model using a set of data measurements. The model is then used to classify data as either normal or anomaly.

1.2 Intrusion Detection System (IDS)

In literature, any kinds of illegitimate or unapproved behavior in a network or a system will be considered as intrusions [4]. An Intrusion Detection System (IDS) is a set of the tools, methods, and resources to facilitate distinguish, evaluate, and description intrusions.

Based on deployment, Intrusion Detection Systems can be classified into two categories [4]:

i. **Host-Based IDS (HIDS)** – It exists on the individual devices in the network. It tracks the incoming and outgoing packets from the device on which it is installed and notifies the administrator if any suspicious activity is found.

ii. **Network-Based IDS (NIDS)** – It exists at certain points in the network to monitor traffic to and from all the devices in the network. It analyses the network traffic and matches it to the library of known attacks. If an attack is detected, or any abnormal activity is sensed, an alert is sent to the administrator.

Based on detection methodologies, Intrusion detection systems come down to two patterns of detection [4].

i. **Misuse (signature/rule) Based Detection**: This technique compares the observed behavior with known attack patterns (signatures). Action patterns that may pose a security threat have to be defined and stored in the system. The advantage of this technique is that it can accurately and efficiently detect instances of known attacks, but it lacks an ability to detect an unknown type of attack.

ii. **Anomaly Detection**: The detection is based on monitoring changes in behavior, rather than searching for some known attack signatures. Before the anomaly detection based system is deployed, it usually must be taught to recognize normal system activity (usually by automated training). The system then watches for activities that differ from the learned behavior by a statistically significant amount. The main disadvantage of this type of system is high false positive rate. The system also assumes that there are no intruders during the learning phase.

Anomaly detection systems (ADS) monitor the behaviour of a system and flag significant deviations from the normal activity as anomalies. A more recent class of ADS developed using machine learning techniques like artificial neural-networks, fuzzy classifiers, multivariate analysis, and others have become popular because of their high detection accuracies at low false positive rates [5].

A model for anomaly detection in WSN using Inverse Weighted Clustering (IWC) for clustering jobs whereas C5.0 decision tree for classification jobs to classify the data either it is normal or abnormal have been proposed. It is effectively used to detect anomalies with high true positive rate and low false positive rate on WSN datasets. The proposed method is robust, effective and also retains its good detection performance.

1.3 Wireless Sensor Network

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to monitor physical or environmental conditions. A WSN system incorporates a gateway that provides wireless connectivity back to the wired world and distributed nodes [6].

1.3.1 Sensor network architecture

Most common architecture for WSN follows the OSI Model. Basically in sensor network we need five layers: application layer, transport layer, network layer, data

link layer and physical layer. Added to the five layers are the three cross layers planes as shown in Figure 1.1.

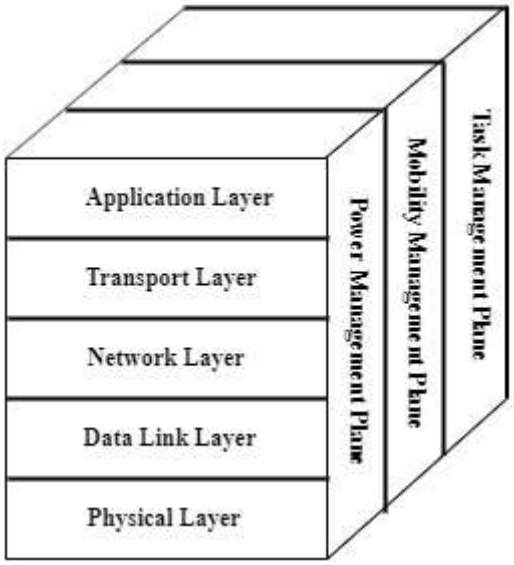

Figure 1. 1 *WSN Architecture [6]*

The three cross planes or layers are; power management plane, mobility management plane and task management plane. These layers are used to manage the network and make the sensors work together in order to increase the overall efficiency of the network [6].

- **Power management plane:** It is responsible for managing the power level of a sensor node for sensing, processing and communication.
- **Mobility management plane**: It is responsible for configuration and reconfiguration of sensor nodes to establish or maintain network connectivity.
- **Task management plane**: It is responsible for task distribution among sensor nodes to improve energy and prolong network lifetime.

Physical Layer

Responsible for frequency selection, carrier frequency generation, signal detection, modulation and data encryption.

Data Link Layer

The data link layer 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

WSNs require multi-hop routing protocols for the data communication by using neighbor sensor nodes as gateways. The network layer 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 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 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.

The Figure 1.2 shows a view of simple wireless sensor network. Wireless sensor network consists of one or more base stations known as gateways, a number of sensor nodes and end user. The output generated by one node is wirelessly transmitted to the base station for data collection, analysis and logging. Each and every node in the Wireless Sensor Network acts as router for transmitting the information from source node to sink node [7]. The end users are facilitated with the data from the sensor via some website or some application in the console terminal.

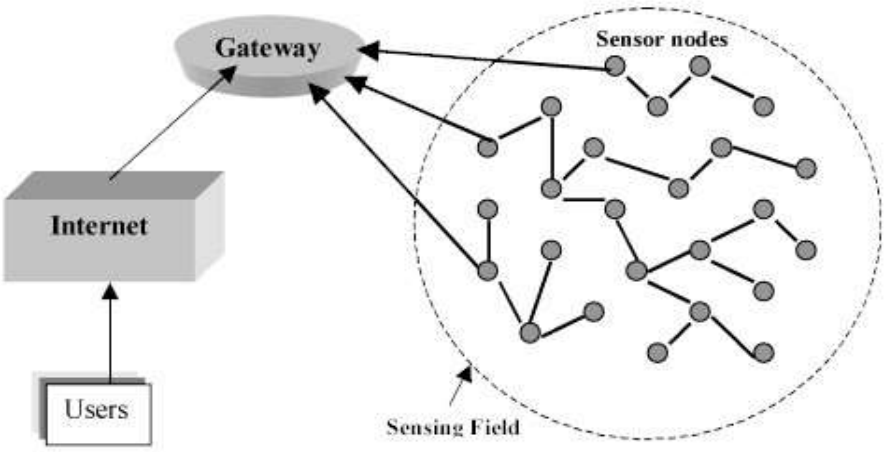

Figure 1. 2 *Illustration of a WSN with a number of sensors [7]*
Source: http://article.sapub.org/10.5923.j.jwnc.20150501.03.html

1.3.2 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. WSNs 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 [8].

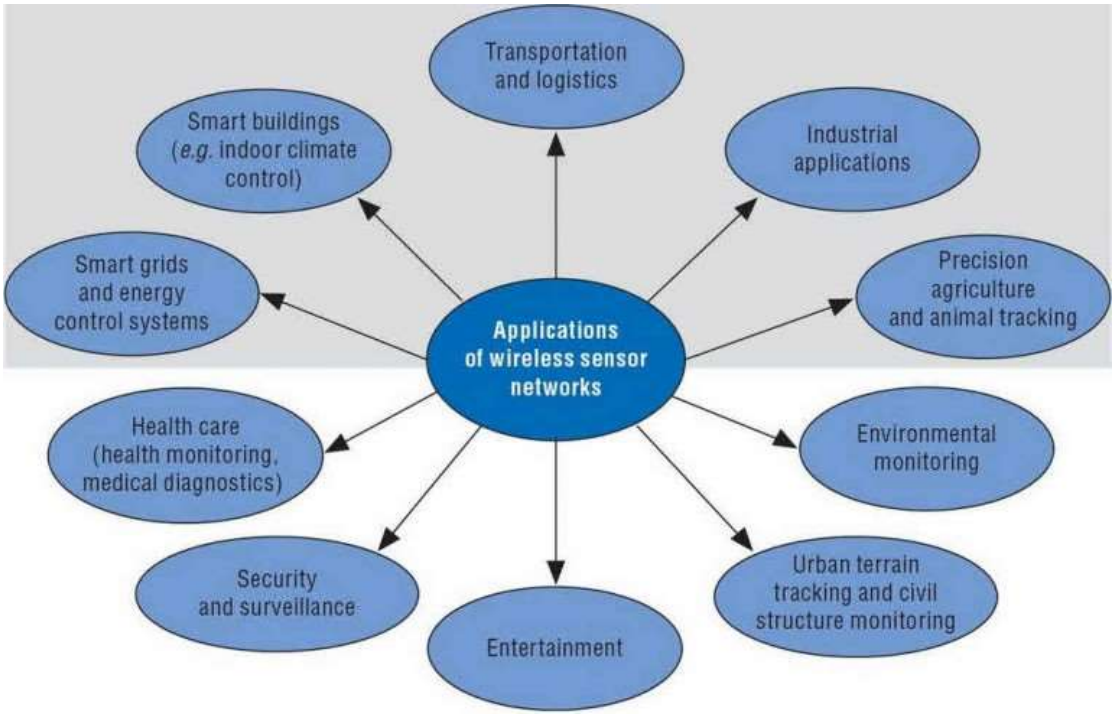

Figure 1. 3 *WSN Application [8]*

Source: https://www.elprocus.com/architecture-of-wireless-sensor-network-and-applications/

1.3.3 Security issues in WSN

WSNs and their adoption in every part of life also pose 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.

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 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.

1.4 Problem Statement

Anomaly detection has been widely used in many application domains. The most common techniques fall into the scope of statistics, clustering and machine learning. Depending on the types of samples necessary to process the data, these techniques are divided into supervised, semi-supervised or unsupervised.

Anomaly based Intrusion Detection System and Intrusion Prevention System relies on artificial intelligent (AI) and machine learning (ML) to detect anomalies. The idea behind AI and ML is to make a machine capable of learning by itself and distinguish between normal and abnormal behavior on the system.

K-means is one of the most popular algorithms for clustering data into groups. A decision tree algorithm such as C5.0 is the most popular classification algorithm used to distinguish between normal and abnormal behavior or patterns in data. Most of the intrusion detection systems use such a combination of algorithms to cluster sample data into groups, label them, and then use a classifier to train the intrusion detection systems to distinguish between these groups.

K-means has limitations; one of which is the initial selection of data before starting clustering. The inverse weight clustering (IWC) which is an enhanced version of K-means overcomes the problems of traditional K-means such as sensitivity to initial conditions [9].

1.5 Objective

The main objectives of this research work are:

i. To develop a model for anomaly detection of WSN datasets based on Inverse Weighted Clustering and Decision Tree.
ii. To evaluate the model.

1.6 Motivation

Nowadays, much attention has been paid to intrusion detection system which is closely linked to the safe use of network services. Several machine learning

paradigms have been investigated for the design of IDS. The motivation for using the combined approach is to improve the accuracy of the anomaly detection system when compared to using individual clustering and classification. Internet has become the main factor of global information that contains commercial, social and cultural activities. As the internet plays an important role in our day to day life there is increase in number of attacks and threats to the network.

1.7 Scope of Work

The spatiotemporal datasets used for this approach are numerical values such as temperature and humidity recorded from the wireless sensor network.

1.8 Limitation

The proposed model is limited to the WSN datasets only. It does not work in textual data. Further, this model detects contextual data anomalies at application level and not in network layer.

1.9 Organization of Thesis Report

There are 6 chapters in this thesis and are organized as follows**:** Chapter 1 is an introductory chapter which includes background of the study, problem statement, objectives of thesis and scope and limitations of the work. Chapter 2 highlights related works carried out in this same kind of research as literature review. Chapter 3 presents the general overview of the Anomaly Detection Techniques in WSN. It also, describes about Machine Learning Algorithms for Anomaly Detection in WSN**.** Chapter 4 presents the proposed methodology for IWC and C5.0 Decision tree. In Chapter 5, experimental results and performance comparison is discussed. Finally, conclusion and the future work of the research are presented in Chapter 6.

CHAPTER 2: LITERATURE REVIEW

This chapter presents a literature survey of various models and techniques used to detect the intrusion. How IDS developed and various kinds of changes take place in existing and new models.

2.1 Related Work

Extensive research has been done in the field of anomaly detection; various techniques are there for detecting anomalies in a dataset. Machine learning algorithms were introduced in the field of anomaly detection to discover predictability of data behavior, either it is normal or abnormal. In addition, better accuracy rate can be achieved using the merged approach, in which two or more machine learning algorithms from different clustering and classification techniques reintegrated to perform anomaly detection. However, reducing false alarms remains as a challenging task for researchers in the area.

Researches carried out by various researchers in the field of intrusion detection in Wireless Sensor Networks were explored. Some of the relevant research methods and their limitations are discussed in this section.

S. R. Gaddam, Vir V. Phoha and et al. (2007)[10], presented "K-Means+ID3," a method to cascade K-Means clustering and ID3 decision tree learning methods for classifying anomalous and normal activities in a computer network, an active electronic circuit, and a mechanical mass-beam system. Experimental results on three datasets show that the detection accuracy of the K Means+ID3 method is as high as 96.24 percent at a false-positive-rate of 0.03 percent on Network Anomaly Data; the total accuracy is as high as 80.01 percent on Mechanical System Data and 79.9 percent on Duffing Equation Data.

R. M. Elbasiony, T. E. Eltobely and et al. (2013)[11], used weighted K-means and Random Forest classification, the experiment worked very well except that KDD CUP99 dataset was used and the results were 98.3% detection rate and 1.6% false alarm rate.

M. Wazid and A. K. Das (2016)[12], proposed a robust and efficient secure intrusion detection approach which uses the K-means clustering in order to extend the lifetime of a WSN. They propose a new intrusion detection technique for hybrid anomaly; K-means built patterns of attacks automatically over training data for the detection purpose. After that intrusions are detected by matching network activities against the detection patterns. The authors assess the approach over a WSN dataset that is created using Opnet modeler, which contains a range of attributes, such as end- to- end delay, traffic sent and traffic received. The training dataset contains the normal values of the network parameters. The testing dataset is created in actual working mode consists of normal and abnormal values of the network parameters. Authors claim that proposed scheme achieves 98.6 % detection rate and 1.2 % false positive rate and the technique has the ability to detect two types of malicious nodes: blackhole and misdirection nodes.

Y. Li and J. Xia (2012)[13], proposed an efficient Intrusion detection system based on Support vector machines and gradually feature removal method, combination of clustering method, ant colony algorithm and support vector machine.

V. Golmah (2014)[14], proposed an efficient hybrid intrusion detection method based on C5.0 and SVM. This method achieves a better performance compared to the individual SVM. Evaluate the proposed method using DARPA dataset.

W. Yassin, N. I. Udzir and et al. (2013)[15], proposed integrated machine algorithms and Naïve Bayes to minimize false alarm rate and improve accuracy rate. The results show significant improvement in the accuracy rate with 99.0% when compared with previous studies with the same approach. However, false alarm rate was high at 2.2%.

In **H. M. Tahir, W. Hassan and et al.** (2015)[16], K-means clustering algorithms was combined with support vector machine to form hybrid intelligent system, the Authors were able to obtain 96.24% accuracy and 3.715% alarm rate.

K. H. Rao, G. Srinivas and et al. (2011)[17], proposed a technique by cascading K-means with different classification techniques, this removes the anomalies from K-means using id3, it overcome the disadvantage of both ID3 and K-means but integrating K-means +id3 is a time consuming process.

P. C. Yong, C. Xiang and et al. (2008)[18], proposed a multiple-level hybrid classifier, a novel intrusion detection system, which combines the supervised tree classifiers and unsupervised Bayesian clustering to detect intrusion. This approach provides the high detection rate and false alarm rate in comparison of Kernel miner, Three level tree classifier, Bagged boosted C5.0 trees.

G. Kim and S. Lee (2014)[19], presented a new hybrid intrusion detection method hierarchically integrates a misuse detection and anomaly detection in a decomposed structure. The misuse detection model is built based on C4.5 decision tree algorithm and is used to decompose the normal training data into smaller subsets. The one-class SVM is used to create anomaly detection for the decomposed region. C4.5 decision tree does not form a cluster, which can degrade the profiling ability.

J. Wang, Q. Yang and et al. (2009)[20], presented an intrusion detection system based on decision tree technology. In the process of constructing intrusion rules, information gain ratio is used in place of information gain. The experiment results show that the C4.5 decision tree is feasible and effective, and has a high accuracy rate. His experimental study shows that the C4.5 decision tree is an effective technique for the implementation of decision tree and it gives almost 90% of classifier accuracy. But in this approach the error rate remains the same.

A. P. Muniyandi, R. Rajeshwori and et al. (2012)[21], presented an anomaly detection method using K-Means+C4.5, a method to cascade k-means clustering and the C4.5 decision tree methods. This method achieves better performance in comparison to the K-Means, ID3, Naïve Bayes, K-NN, and SVM.

In recent years, integrated approaches have been widely explored. For instance, Naive Bayes, Bayesian network, Support Vector Machine (SVM), Self-Organizing Map

(SOM) which is based on neural networks, have all been used in existing anomaly detection mechanisms.

So far, various methods in anomaly detection domain have been employed; but interestingly most of them evaluate their approaches with the KDD Cup99 dataset. In short, various techniques have been proposed in the field of intrusion detection, but there is still room to improve detection rate and accuracy, and reducing false alarm rate. In contrast, the proposed approach has been tested on WSN datasets of IBRL and Bharatpur Airport to demonstrate that it is able to increase the detection rate while minimizing the false alarm rate.

In this research, Inverse Weighted K-mean and C5.0 was used; the reason for choosing Inverse Weighted K-mean is time complexity O(nkt) where n-total is number of patterns, k is number of clusters, t is number of iterations, its space complexity O(k+n) and its scalability, its order independent. The reason of choosing C5.0 is it is more efficient, its decision tree is smaller in cooperation with C4.5 and unnecessary attributes have been automatically removed by C5.0

CHAPTER 3: RELATED THEORY

3.1 Anomaly Detection in Wireless Sensor Network

Anomalies are observations that do not correspond to a well-defined notion of normal behaviors. In WSNs, anomalies can occur in the nodes, networks, transmission channels and application data and can be caused by systematic errors, random errors and malicious attacks [1].

WSN collect Spatial, Temporal or Spatiotemporal data and these data may have three types of anomalies, namely Point, Contextual and Collective anomalies [3].

- Point anomaly: An individual data instance that is considered anomalous with respect to the data set.
- Contextual anomaly: A data instance that is considered an anomaly in the current context. In a different context the same data instance might be considered normal.
- Collective anomalies: A collection of related anomalies.

Challenges in deploying Anomaly Detection System in WSNs

Traditional Intrusion detection systems cannot appropriately detect suspicious activities in a WSN. For distributed nature of WSN infrastructure the ability of traditional intrusion detection system to handle and block large malicious attacks from offender may not be sufficient.

The WSN architecture is highly dynamic, scalable and distributed in nature. The anomaly detection system to be successfully deployed in WSN, the anomaly detection systems have to cop up with the scalability of the wireless sensor network environment [4]. The deployment strategy of anomaly detection system is a big challenge in WSN environment.

So, the anomaly detection system for wireless sensor network should be light weight and no necessary information is passed between the client and server. The time taken by Anomaly detection system for detection and responding back to network intrusion in WSN is very high.

3.2 Anomaly Detection Techniques

The general architecture of all anomaly based intrusion detection systems methods is similar. Generally, all of them consist of the following basic modules or stages. These stages are parameterization, training and detection [22]. Parameterization includes collecting raw data from a monitored environment. The raw data should be representative of the system to be modeled, (e.g. Packet data from a network). The training stage seeks to model the system using manual or automatic methods.

Anomaly detection techniques for WSNs can be categorized into statistical based, Machine learning-based, Data mining-based approaches. These techniques along with examples are explained below.

3.2.1 Statistical based Techniques

Statistical based techniques build data reference model and evaluate each data pattern with respect to that reference model. Any deviation from the reference model is considered as anomaly. There are two types statistical based techniques i) parametric ii) non parametric techniques. In parametric techniques, known data distribution builds reference model against which parameters are evaluated. In nonparametric, as data distribution is not known a priori, some distribution estimation methods are used to build the reference model against which parameters are evaluated [1]. The limitations of this technique are: dynamic nature of WSN makes it difficult to select appropriate threshold value for evaluation, non-parametric statistical models are not suitable for real time applications, and computational cost of handling multivariate data is more.

3.2.2 Machine Learning based Techniques

Machine learning systems have capability to learn and improve their performance on the basis of certain tasks. This technique is having the capability to change their execution process on the basis of newly learned information. Machine learning algorithms focus on increasing the performance on the basis of previous results, but not on understanding the process unlike statistical approaches [22].

Most machine learning algorithms fall into the categories of supervised, unsupervised and reinforcement learning. In the first category, machine learning algorithms are provided with a labeled training data set. This set is used to build the system model representing the learned relation between the input, output and system parameters. In contrast to supervised learning, unsupervised learning algorithms are not provided with labels (i.e., there is no output vector). Basically, the goal of an unsupervised learning algorithm is to classify the sample sets to different groups (i.e., clusters) by investigating the similarity between the input samples. The third category includes reinforcement learning algorithms, in which the agent learns by interacting with its environment (i.e., online learning). Finally, some machine learning algorithms do not naturally fit into this classification since they share characteristics of both supervised and unsupervised learning methods. These hybrid algorithms (often termed as semi supervised learning) try to inherit the strengths of these main categories, while minimizing their weaknesses [22].

Supervised learning -- In supervised learning, a labeled training set (i.e., predefined inputs and known outputs) is used to build the system model. This model is used to represent the learned relation between the input, output and system parameters. In fact, supervised learning algorithms are extensively used to solve several challenges in WSNs such as localization and objects targeting, security and intrusion detection, data integrity and fault detection [22]. The most common supervised algorithms are, Supervised Neural Networks, Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), Bayesian Networks and Decision Tree [23].

Unsupervised learning -- Unsupervised learners are not provided with labels (i.e., there is no output vector). Basically, the goal of an unsupervised learning algorithm is to classify the sample set into different groups by investigating the similarity between them. This theme of learning algorithms is widely used in node clustering and data aggregation problems. Indeed, this wide adoption is due to data structures (i.e., no labeled data is available) and the desired outcome in such problems [22]. The most common unsupervised algorithms are, K-Means, Self-organizing maps (SOM), C-means, Expectation-Maximization Meta algorithm (EM), Adaptive resonance theory (ART), Unsupervised Niche Clustering (UNC) and One-Class Support Vector Machine [23].

Reinforcement learning -- Reinforcement learning enables an agent (e.g., a sensor node) to learn by interacting with its environment. The agent will learn to take the best actions that maximize its long-term rewards by using its own experience. The most well-known reinforcement learning technique is Q-learning [22].

3.2.3 Data mining based Techniques

In data mining technique the main concerns are with detecting uncovered patterns, anomalies, changes, associations and statistically significant structures. To eliminate the manual process of data profiles or updating of database data mining techniques is widely used nowadays for detecting the anomalies. It has the ability to detect deviation from normal behavior by creating a boundary value of network activity between normal and abnormal behavior [24].

3.3 ML Algorithms for anomaly detection in WSNs

In machine learning there is a need for clustering and classification algorithm. These algorithms are used to cluster after then classify and finally decide what to do with the information gathered. There are many different algorithms with several different purposes, and algorithms can be used for many different purposes. In this thesis, algorithm is going to be used as the decision maker for anomaly detection using machine learning techniques.

3.3.1 Clustering Algorithms

Clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. Clustering is the process of grouping similar objects into different groups, or more precisely, the partitioning of a data set into subsets, so that the data in each subset according to some defined distance measure. The following are the most popular clustering algorithms [25]:

- K-Means
- Fuzzy C-Means (FCM)

- Hierarchical Clustering
- Expectation-Maximization Meta Algorithm (EM)

K-means

The K-means algorithm assigns each point to the cluster whose center also called centroid is nearest. The center is the average of all the points in the cluster that is, its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster [25].

Euclidian distance is the most popular method to measure the distance. The formula for the Euclidian distance is:

P= (p_1, p_2, p_3) and Q= (q_1, q_2, q_3) are the two points in Euclidian space

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2} \qquad (3.1)$$

Input: numerical, there must be a distance metric defined over the variable space

->Euclidian distance

Output: the centres of each discovered cluster, and the assignment of each input datum to a cluster.

->centroid

The pseudo code of the k-means algorithm is to explain how it works:

A. Choose K as the number of clusters.

B. Initialize the codebook vectors of the K clusters (randomly, for instance)

C. For every new sample vector:

a. Compute the distance between the new vector and every cluster's codebook vector.

b. Re-compute the closest codebook vector with the new vector, using a learning rate that decreases in time.

3.3.2 Decision Tree Algorithms

Quinlan [26] defined Decision Trees as “powerful and common tools for classification and prediction. A decision tree is a tree that has three main components: nodes, arcs and leaves. Each node is labeled with a feature attribute, which is most

informative among the attributes not yet considered in the path from the root. Each arc out of a node is labeled with a feature value for the node's feature, and each leaf is labeled with a category or class. A decision tree can then be used to classify a data point by starting at the root of the tree and moving through it until a leaf node is reached. The leaf node provides the classification of the data point.

Decision tree algorithm is a classification method for predicting labels of data by iterating the input data through a learning tree. During this process, the feature properties are compared relative to decision conditions to reach a specific category [27]. There are many specific decision-tree algorithms. Notable ones include:

- ID3 (Iterative Dichotomiser 3)
- C4.5 and C5.0
- CART (Classification and Regression Tree)
- CHAID (CHi-squared Automatic Interaction Detector)
- C5.0/Sec 5

C5.0/Sec 5

C5.0 algorithm is an extension of C4.5 algorithm which is also extension of ID3. It is better than C4.5 on the speed, memory and the efficiency. The C5.0 rule sets have lower error rates on unseen cases. So comparing with C4.5 the accuracy of result is good with C5.0 algorithm. C5.0 automatically allows removing unhelpful attributes. C5.0 model works by splitting the sample based on the field that provides the maximum information gain [28]. The C5.0 model can split samples on basis of the biggest information gain field. The sample subset that is get from the former split will be split afterward. The process will continue until the sample subset cannot be split and is usually according to another field. Finally, examine the lowest level split, those sample subsets that don't have remarkable contribution to the model will be rejected. C5.0 is easily handled the multi value attribute and missing attribute from data set.

3.3.3 K-Nearest Neighbor (K-NN)

In supervised learning algorithm, a test data sample is classified based on the labels (or output values) of nearest data samples. By computing an average of readings

within its neighbourhood, the missing or unknown test sample measurement is predicted. Determination of nearest set of nodes is done by using different methods. One of simplest method to determine the neighbourhood is by using the Euclidean distance between different sensors. As the distance measure is computed using few local points, with k normally a small positive integer, the K-NN approach does not need high computational power [22].

3.3.4 Support Vector Machines (SVMs)

Support Vector Machine is supervised learning method used for prediction and classification. In the context of WSN, they have been used for intrusion detection, or detecting malicious behaviour of sensor nodes, security and localisation. With SVM, it is possible to uncover the spatio-temporal correlations in data, as the algorithm involves constructing a set of hyperplanes (or optimizing a quadratic function with linear constraints) separating WSN data measurements in feature space, by as wide as possible margins. Figure 3.1 show the schematic of SVM classifies WSN measurements [22].

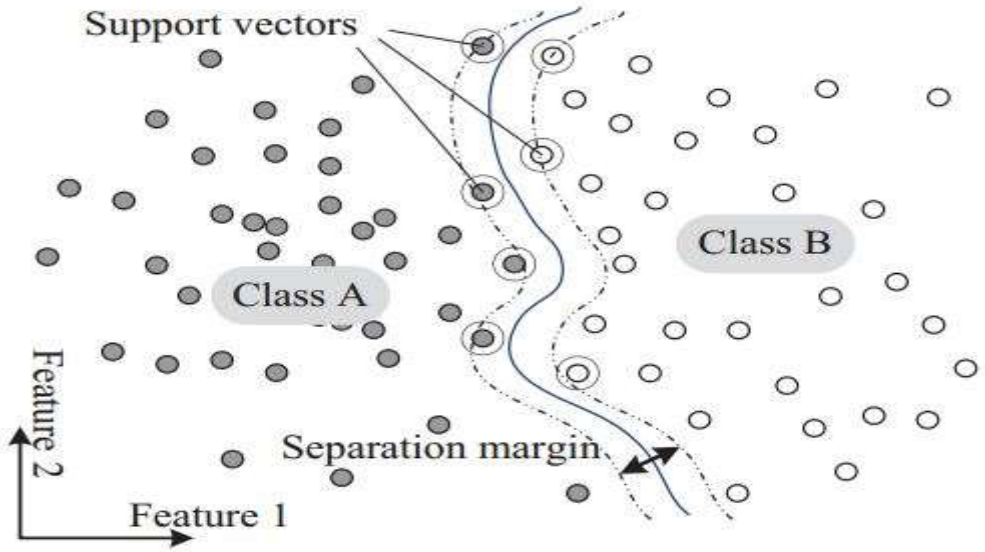

Figure 3. 1 *Schematic of SVM Classification Process [22]*

3.3.5 Naive Bayes classifier

Naive Bayes classifier is probabilistic classifier. It predicts the class according to membership probability. To derive conditional probability, it analyzes the relation between independent and dependent variable [22].

Bayes Theorem:

$$P(H/X) = P(X/H).P(H)/P(X) \quad (3.2)$$

Where, X is the data record and H is hypothesis which represents data X and belongs to class C. P(H) is the prior probability, P(H/X) is the posterior probability of H conditioned on X and P(X/H) is the posterior probability of X conditioned on H.

Construction of Naive Bayes is easy without any complicated iterative parameter. It may be applied to large number of data points but time complexity increases.

CHAPTER 4: METHODOLOGY

The methodologies used in this research work describe the datasets used, as well as the working of Weighted Clustering and Decision tree algorithms and finally it also describes the evaluation criteria; by which the accuracy of the model in detecting anomalies in WSN can be measured.

4.1 System Model

The proposed model for anomaly detection in WSN is to provide the mechanism for improving the detection precisely whereas reducing the false alarm rate. The model describes the flow data to detect anomaly in WSN. The model employs three stage processes before resulting output as normal or abnormal. The model consists of the following steps:

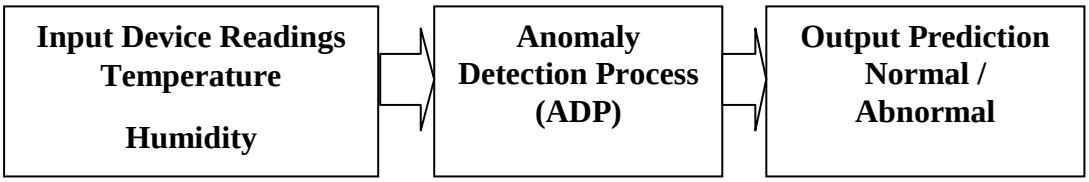

Figure 4. 1 *Model overview*

The datasets collected from wireless sensor networks was used as an input to the clustering algorithm. The clustering algorithm cluster the datasets into two groups based on similarities measurements. After clustering, the groups are labeled as "normal" and "abnormal". The input data is split into two parts; training data (for classification) and testing data (for predictions).

The process used in Anomalies Detection Process (ADP) consists of three stages that are clustering, classification, and predictions. The three stages are:

- **Data preprocessing**: This stage makes the original WSN datasets applicable input for the further steps. Preprocessing also reduces vagueness and produce accurate information to detection engine.
- **Processing**: This stage is concerned with clustering input data into groups and for classification.

- **Prediction**s: This stage is concerned with using the Decision Tree for predictions.

Figure 4.2 describes ADP in details. The input data after preprocessing is used by IWC algorithm to build two clusters; one cluster combines normal data and the other one combine abnormal data. Finally, C5.0 decision tree algorithm is used for classification purpose to distinguish normal and abnormal behaviour of the testing data.

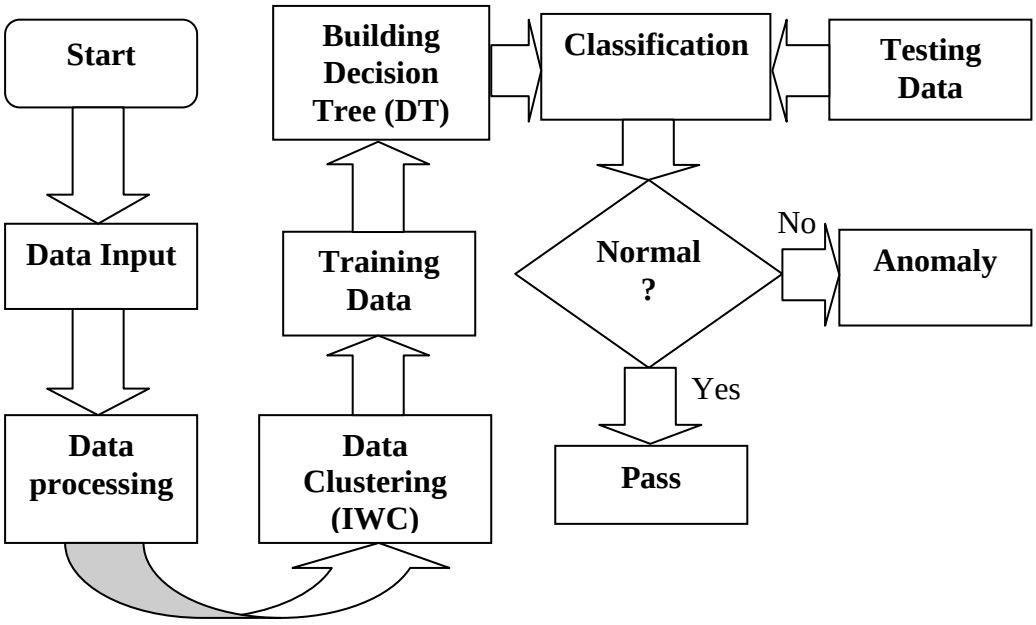

Figure 4. 2 *Flow chart of the model*

4.2 Experimental Datasets

To validate the proposed model real labeled sensor data was needed**.** However, synthetic datasets from University of North Carolina, Greensboro (UNCG), real Intel Berkeley Research Laboratory (IBRL) and local Bharatpur Airport WSN datasets were used to test the model, because no real sensor data with labeled anomalies are currently available.

Standard datasets

- UNCG(with class label)
- IBRL(without class label)

Local dataset

- Bharatpur Airport WSN

A labeled UNCG wireless sensor network dataset [29] collected from a multi-hop wireless sensor network deployment using TelosB motes. The data consists of humidity and temperature measurements collected during 6 hour period at intervals of 5 seconds. The multi-hop data was collected on 10th July 2010.

The IBRL datasets contain information collected from 54 sensors deployed in the Intel Berkeley research lab, between 28 February and 5 April 2004. Mica2Dot sensors with weatherboards collected time-stamped topology information, along with humidity, temperature, light, and voltage values once every 31s. The data were collected using the TinyDB in-network query processing system, built on the TinyOS platform [30].

Bharatpur Airport wireless sensor network data contain humidity and temperature collected during January 2016 at interval of 15 min was obtained from Department of Meteorology, Kathmandu, Nepal.

4.3 Data pre-processing

A random sample of Sensor Network Devices readings was taken deleting the unnecessary attributes such as date and time of reading, sensor ID, light and voltage data since the proposed anomaly detection model is concerned with detecting anomalies at applications layer.

Anomalies (contextual anomalies) were manually added to the Bharatpur Airport WSN data to make the input data somehow labeled and qualified for the research purpose. Intel lab used the same way to manipulate WSN input data by manually adding records with anomalies in the dataset. The type of manually introduced anomaly in Bharatpur Airport data represent irregularities in sensed data due to changes in data values over time and space at node location (spatiotemporal data) in which the contextual anomalies tried to find out. UNCG data was already labeled and fit for input to classifier; and was only used for classification result.

1	Timestam	TZ	WS (m/s)	WD (deg)	RH (%)	T (C)	P_ST (hPa	P_SEA (hP	SM_5 (%)	TS_5 (C)	SM_20 (%	TS_20 (C)	GLOB (W/	RSUM (mr	ZZZ_BATT (V)
12430	00:08.0	n			62.51715	19.95955	997	1019.6	23.55556	16.7	17.11111	15.5	488.567	0	13.62214
12431	15:08.0	n			59.16174	20.58485	996.6	1019.2	24.88889	16.8	17.11111	15.7	477.6804	0	13.54216
12432	30:08.0	n			56.40384	21.02314	996.3	1018.9	24.88889	16.8	17.11111	15.7	472.9281	0	13.59058
12433	45:08.0	n			55.52871	21.31202	995.9	1018.5	25.55556	16.8	17.11111	15.7	458.6565	0	13.57857
12434	00:08.0	n			51.55354	22.09106	995.6	1018.1	25.55556	16.8	17.11111	15.7	437.0996	0	12.78332
12435	15:08.0	n			52.2469	22.28198	995.3	1017.8	24.88889	16.8	17.11111	15.7	416.0277	0	13.55068
12436	30:08.0	n			50.09828	22.55926	995	1017.6	25.55556	16.8	17.11111	15.8	388.1266	0	13.51458
12437	45:08.0	n			47.11441	22.96992	994.8	1017.4	24.88889	16.8	17.11111	15.8	348.2421	0	13.47681
12438	00:08.0	n			43.7905	23.00945	994.6	1017.2	24.88889	16.8	17.11111	15.8	327.3436	0	13.49933
12439	15:08.0	n			39.44626	23.24251	994.5	1017.1	24.88889	16.8	17.11111	15.8	293.3134	0	12.89708
12440	30:08.0	n			43.36163	22.82566	994.3	1016.8	25.55556	16.8	17.11111	16	256.4288	0	13.47503

(a) Raw data

1	RH (%)	T (C)
12430	62.51715	19.95955
12431	59.16174	20.58485
12432	56.40384	21.02314
12433	55.52871	21.31202
12434	51.55354	22.09106
12435	52.2469	22.28198
12436	50.09828	22.55926
12437	47.11441	22.96992
12438	43.7905	23.00945
12439	39.44626	23.24251
12440	43.36163	22.82566

(b) Processed data

Figure 4. 3 *(a) and (b) show the sample data*

4.4 Proposed Model

Anomaly Detection in WSN using IWC+C5.0 Method

Machine learning method used in anomaly-based detection in recent years promises high detection and accuracy rate. However, the rate for false alarms also increases accordingly. IWC+C5.0 is able to detect intrusive activities and focuses to achieve high detection and accuracy rate with lower false alarm.

An anomaly detection model using two machine learning algorithms IWC and C5.0 have been proposed. Initially IWC was used for partitioning the dataset into K closest cluster using Euclidean distance formula and then C5.0 techniques was applied on each closest cluster to build decision tree for each cluster and classify the each

instance into normal or anomaly using decision tree result. The model consist of two phases selection phase and classification phase

i. *Selection phase:* The closest cluster is selected for each test instance. In the selection cluster the decision tree corresponding to the cluster is generated
ii. *Classification phase:* The test instance is classified into normal and anomaly using the C5.0 decision tree result and the cluster label as normal or anomaly

There two modules in IWC+C5.0; are namely the pre-classification module and the classification module. The first module, involving Inverse Weighted Clustering iteration function where similar data are grouped into two clusters based on their behavior. The entire data are labeled with the K-th clusters set accordingly. Next, the labeled clustered data are classified into abnormal and normal classes using the Decision tree classifier to recover the misclassified data from the first module. And found that IWC+C5.0 is able to classify the normal and abnormal data more accurately at the subsequent classification module.

4.5 Inverse Weighted Clustering (IWC)

The K-Means algorithm is one of the most frequently used investigatory algorithms in data analysis. The algorithm attempts to locate K prototypes or means throughout a dataset in such a way that the K prototypes in some way best represent the data. However the algorithm is known to suffer from the defect that the means or prototypes found depend on the initial values given to them at the start of the simulation [9].

IWC in essence is built upon K-means algorithm. However, it relies on running the k-means many times until the centroids and clusters become stable. In other words, while the k-means stops once k centroids and clusters are formulated, IWC takes the resulting centroids and rerun k-means over the same data by computing the distance between each record and the centroids [9].

$$m_k = \frac{\sum_n r_{kn} x_n}{\sum_{j,n} r_{jn}} \quad (4.1)$$

Where e.g. $r_{kn} = \frac{\exp(-\beta d(x_n, m_k))}{\sum_j \exp(-\beta d(x_n, m_j))}$ (4.2)

Inverse Weighted Clustering Algorithm (IWC) expands k-means as following:

$$J_I = \sum_{i=1}^{N} \sum_{k=1}^{K} \frac{1}{[\![x_i - m_k]\!]^p} \quad (4.3)$$

$$\frac{\partial J_I}{\partial m_k} = \sum_{i=1}^{N} P(X_i - m_k) \frac{1}{\|X_i - m_k\|^{P+2}} \quad (4.4)$$

$$\frac{\partial J_I}{\partial m_k} = 0 \Rightarrow$$

$$m_k = \frac{\sum_{i=1}^{N} \frac{1}{\|X_i - m_k\|^{P+2}} X_i}{\sum_{i=1}^{N} \frac{1}{\|X_i - m_k\|^{P+2}}}$$

$$= \frac{\sum_{i=1}^{N} b_{ik} X_i}{\sum_{i=1}^{N} b_{ik}} \quad (4.5)$$

Where

$$b_{ki} = \frac{1}{\|X_i - m_k\|^{P+2}} \quad (4.6)$$

The partial derivative of J_I with respect to $\mathbf{m}_k$ will maximize the performance function J_I. So the implementation of (4.5) will always move $\mathbf{m}_k$ to the closest data point.

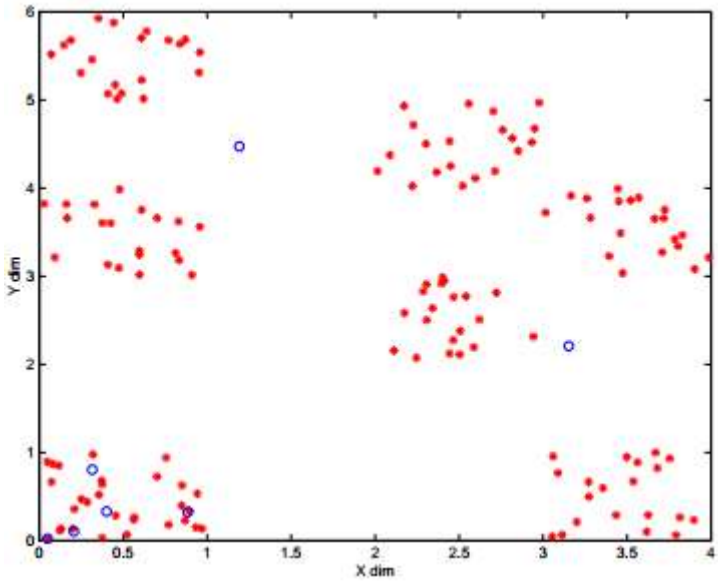

Figure 4. 4 *K-means failed in identifying all the clusters [9]*

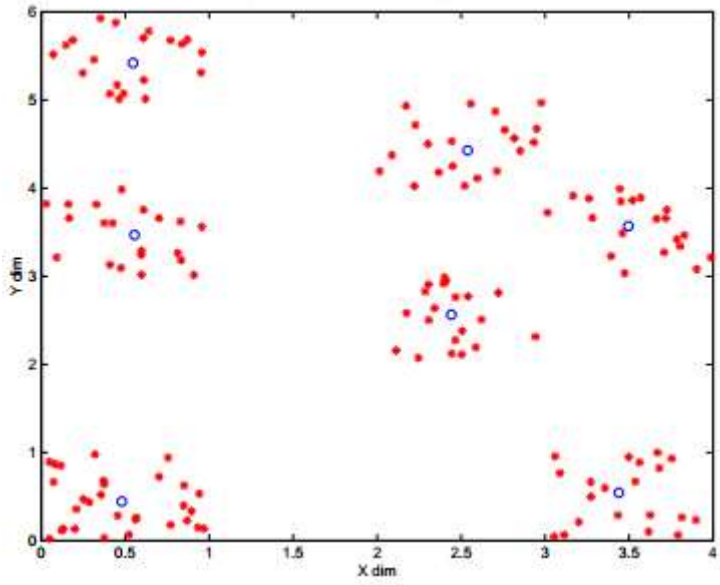

Figure 4. 5 *IWC algorithm identified all the clusters successfully [9]*

Here, the value of K have been predefined for portioning the datasets into K clusters, representing cluster -1, cluster -2 and so. After running clustering algorithm on datasets, certain activities or data are alike to either normal or abnormal behaviour. The IWC algorithm is unable to differentiate this behaviour precisely. Thus, Decision tree classifier is applied to re-classify clustered labeled data to improve the shortcoming.

4.6 C5.0 Decision tree

The input to a classifier is a training set of record, each of which is a tuple of attribute values tagged with a class label. A set of attribute values defines each record. A decision tree has the root and each internal node labeled with a question. The arcs emanating from each node represent each possible answer to the associated question [14]. Each leaf node represents a prediction of solution to the problem under consideration.

C5.0 supports boosting of decision trees. Boosting is a technique for generating and combining multiple classifiers to give improved final predictive accuracy. C5.0 incorporates variable misclassification costs. It supports sampling and cross-validation. C5.0 models are quite robust in the presence of problems such as missing data and large numbers of input fields [14]. It does not require long training times to

estimate. In addition, it is easier to understand than some other model types, since the rules derived from the model have a very straightforward interpretation. C5.0 tree or rule sets are usually smaller than C4.5.

A C5.0 model is based on the information theory. Decision trees are built by calculating the information gain ratio. The algorithm C5.0 works by separating the sample into subsamples based on the result of a test on the value of a single feature. The specific test is selected by an information theoretic heuristic [14]. This procedure is iterated on each of the new subsamples and keeps on until a subsample cannot be separated or the partitioning tree has reached the threshold.

Information Gain

The training data is separated by using a well-define attributes. It is based on the entropy measure commonly used in information theory [31]. It is defined as the difference between the base entropy and the conditional entropy of the attribute.

Let T is the training dataset

X(c) is the class *I* where c=1, 2, 3…n

$I(T_1, T_2 \ldots T_n) = -\sqrt{p_c log_2 p_c}$ (4.7)

T_c is the number of samples in c

$Pc = \frac{T_C}{T}$ (4.8)

log_2 is binary logarithm let attribute A has v distant values Entropy=E(A) is

$\sum \{\frac{(T_{1j}+T_{2j}+\cdots.+T_{nj})}{T}\} * I(T_1, T_2, \ldots . T_n)$ j=1 (4.9)

Where$Tcj$is the sample in class c and subset j of attribute

$I(T_{1j}, T_{2j}, \ldots . T_{nj}) = -\sum p_{cj} log_2 p_{cj}$ (4.10)

$Gain(A) = I(T_1, T_2, \ldots . T_n) - E(A)$ (4.11)

The algorithm for C5.0 decision tree is [27]:

Step 1: The C5.0 generates a either a decision tree or a rule set

Step 2: Pick the most informative attribute

Step 3: Find the partition with the highest information gain using Eq (4.11)

Step 4: at each resulting node, repeat step1 and 2

4.7 Evaluation Measurement

Evaluation of classification algorithms is one of the key points in any process of data mining. The most commonly tools used in analyzing the results of classification algorithms applied are: confusion matrix, learning curves and receiver operating curves (ROC).

As stated earlier, performance improvement has been the major goal of this research work. To evaluate the results of classifier, standard metrics such as confusion matrix, true-positive rate, false positive rate, and classifier's accuracy have been used.

A confusion matrix contains information about actual and predicted classifications done by a classification system. Performance of such systems is commonly evaluated using the data in the matrix. The following table shows the confusion matrix for a two class classifier.

Table 4. 1 The confusion matrix

	Classes predicted	
Current classes	True class	False class
True class	True Positive	False Negative
False class	False Positive	True Negative

Several standard terms have been defined for the 2 class matrix:

- **Accuracy (AC):** It is the proportion of the total number of predictions that were correct. It is determined using the equation:

$$AC = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (4.12)$$

- **Specificity:** It is the proportion of true negative points to negative elements, as calculated using the equation:

$$Specificity = \frac{TN}{(TN+FP)} \quad (4.13)$$

- **False Alarm or False positive rate (FP):** It is the proportion of negatives cases that were incorrectly classified as positive, as calculated using the equation:

$$\text{FPR=1-Specifity} = \frac{\text{FP}}{(\text{TN+FP})} \tag{4.14}$$

- **Precision or Detection rate(DR)**: It is the proportion of the predicted positive cases that were correct, as calculated using the equation:

$$\text{DR} = \frac{\text{TP}}{(\text{TP+FP})} \tag{4.15}$$

- **Recall or True positive rate (TP):** It is the proportion of positive cases that were correctly identified, as calculated using the equation:

$$\text{TP} = \frac{\text{TP}}{(\text{TP+FN})} \tag{4.16}$$

- **False negative rate (FN):** It is the proportion of positives cases that were incorrectly classified as negative, as calculated using the equation:

$$\text{FN=1-R} = \frac{\text{FN}}{(\text{TP+FN})} \tag{4.17}$$

The F1 is a measure of a classification accuracy, which summarizes the measures Precision and Recall into single indicator. F1 measure is defined as follows:

$$\text{F\_Measure} = 2*\frac{\text{Precision}*\text{Recall}}{(\text{Precision+Recall})} \tag{4.18}$$

If F1-measure achieves high value it provides that both precision and recall are reasonably high. It is one of the measures of quality of a cluster algorithm using external criterion.

CHAPTER 5: IMPLEMENTATION AND RESULTS

This chapter discusses about the proposed work and implementation of the model. Here the discussion is about the architecture of the proposed model and different components used during the implementation. The proposed model was implemented and tested using MATLAB codes.

5.1 Implementation

The model was implemented and tested in MATLAB. The three stage implementation process consists: data processing, clustering, classification and testing as described in methodologies section. Input data processing is a stage for qualifying wireless sensor network data to be used for anomaly detection. Clustering and Classification are core process in the model. Clustering divides data into two similar structure using Inverse Weight Clustering algorithm. Classification uses C5.0 decision tree algorithm for classifying data in normal and abnormal group.

MALAB was used to run and test the unclassified data to get the results. It is relatively simple and easy to use compared to other high level programming language such as Java or C.

5.2 Experimental Results

The results of clustering and classification of data were obtained using MATLAB. To see the performance of proposed model; UNCG, IBRL and Bharatpur Airport WSN datasets have been used. The dataset have 11 attributes but only 2 attributes (i.e., humidity and temperature) have been taken, then IWC applied on the dataset for portioning the dataset into K clusters, here K=2, number of iteration=10. IWC Clustering alone cannot eliminate overlapping data anomaly, so C5.0 was used in each cluster built and partition into train and test set for classification of instance into normal and abnormal.

Clustering

IWC algorithm was run on input WSN data of Intel lab and Bharatpur Airport datasets, and then entire data was grouped into two classes, one in red belonging to class-1 and the other in green belonging to class-2 for labeling the data. These two values represent the cluster to which each reading (i.e. temperature and humidity) belongs.

On IBRL datasets

The new IWC+C5.0 method was evaluated with real IBRL WSNs datasets. The datasets includes two attributes i.e. humidity and temperature. Temperature is in degrees Celsius. Humidity is temperature corrected relative humidity, ranging from 0-100%. Here, a portion of datasets with 7526 reading instances were taken for the experimental purpose.

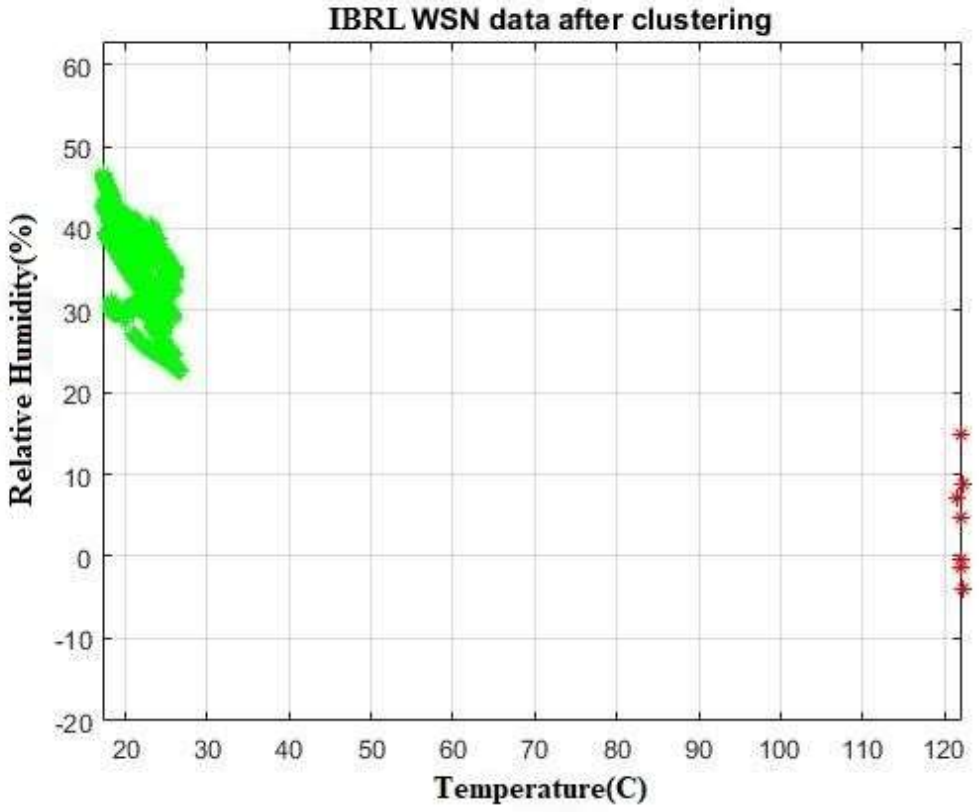

Figure 5. 1 *Intel Berkeley Research Lab WSN data after clustering*

On Bharatpur Airport WSN datasets

Similarly, the new IWC+C5.0 method was evaluated on the Bharatpur Airport WSN datasets. The datasets consist of humidity and temperature measurements of 3044 records. Here also, a portion of WSN datasets was utilized for experimental purpose. Here 164 (i.e. 5.6% of total records) anomalies were manually added to 2880 readings of Bharatpur Airport WSN; thus making the input datasets 3044 instances.

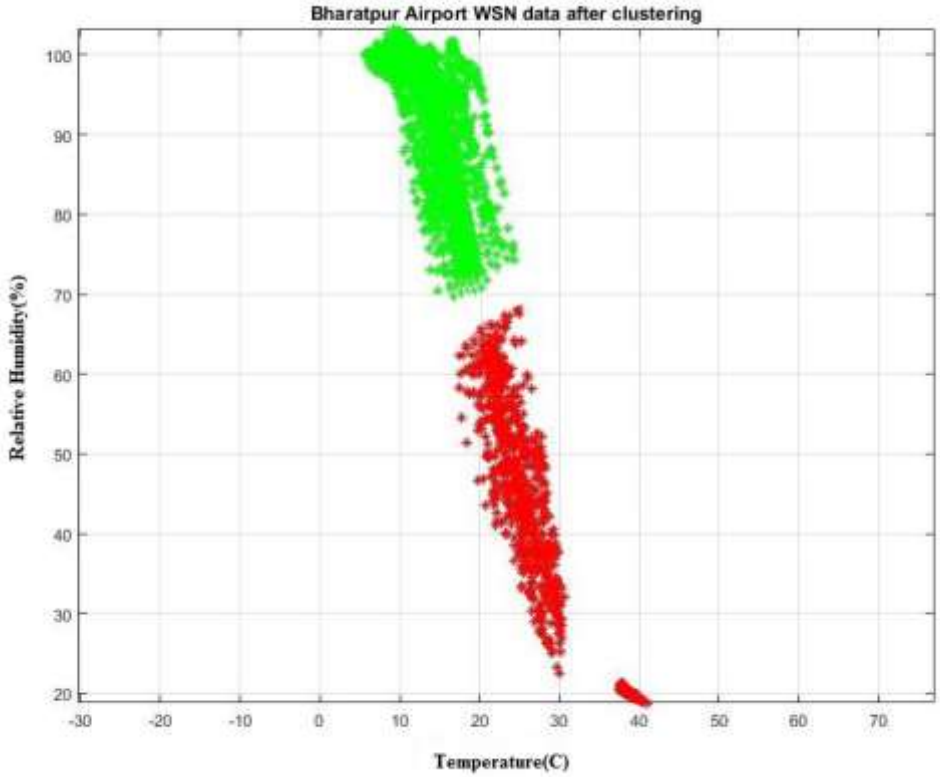

Figure 5. 2 *Bharatpur Airport WSN data after clustering*

Figures 5.1 and Figure 5.2 show the results of the data after clustering – humidity and temperature readings.

Classification

The two classes extracted as a result of data clustering was split into two parts. Using the decision tree, data was split into two groups constituting 66% of the training group and 34% of the testing data of the clustered data.

Training datasets: it contains 66% records that are classified to normal and abnormal.

Testing datasets: it contains 34% records without labels that can be used for testing using C5.0.

Decision tree was trained using labeled training dataset consisting of 66% of input dataset records and then tested against 34% of testing dataset, but without giving it the cluster ID that shows to which cluster each record in the testing dataset belongs.

The classification phase was performed using C5.0 decision tree, after the entire preprocessing step. The resulting predictions were compared with the cluster ID of each record in the labeled testing dataset. C5.0 Decision tree classification divides the network behavior to normal and abnormal and assigns the abnormal behavior to its specific category.

Results and Discussion

The confusion matrix was realized, after running the classifier on test data from various instances for the classification of the proposed model on real Intel Berkeley Research Laboratory (IBRL) and Bharatpur Airport WSN datasets. Table 5.1, 5.3 and 5.5 show the confusion matrix obtained in testing the proposed approach.

Table 5. 1 Confusion Matrix on labeled UNCG datasets with C5.0

X= 1407	**Predicted anomalies**	**Predicted normality**
Actual anomalies	TP = 16	FN = 1
Actual normality	FP = 2	TN = 1388

Table 5. 2 Result of Performance Evaluation on UNCG datasets

Metric	**Formula**	**Value**
Accuracy	(TP+TN)/(TP+TN+FP+FN)	99.79%
Detection Rate	(TP)/(TP+FP)	88.89%
False Alarm Rate	(FP)/(FP+TN)	0.14%

Table 5.2 shows the results of C5.0 on labeled UNCG datasets. The result shows that on labeled datasets C5.0 achieves accuracy reaching 99.79%, detection rate 88.89% at very low false alarm rate 0.14% nearly.

Table 5. 3 Confusion Matrix on Intel lab WSN dataset with IWC+ C5.0

X= 2535	**Predicted anomalies**	**Predicted normality**
Actual anomalies	TP = 583	FN = 1
Actual normality	FP = 6	TN = 1945

Table 5. 4 Result of Performance Evaluation on IBRL WSN datasets

Metric	**Formula**	**Value**
Accuracy	(TP+TN)/(TP+TN+FP+FN)	99.72%
Detection Rate	(TP)/(TP+FP)	98.98%
False Alarm Rate	(FP)/(FP+TN)	0.31%

The confusion matrix for classification of the proposed approach was calculated as in Tables 5.4 and it shows that, detection rate is 98.98% at same time false alarm rate is 0.31%. The confusion matrix is tested on proposed IWC +C5.0 method.

Table 5. 5 Confusion Matrix on Bharatpur Airport WSN dataset with IWC+C5.0

X= 1035	**Predicted anomalies**	**Predicted normality**
Actual anomalies	TP = 462	FN = 1
Actual normality	FP = 2	TN = 570

Table 5. 6 Performance Evaluation on Bharatpur Airport WSN datasets

Metric	**Formula**	**Value**
Accuracy	(TP+TN)/(TP+TN+FP+FN)	99.71%
Detection Rate	(TP)/(TP+FP)	99.57%
False Alarm Rate	(FP)/(FP+TN)	0.35%

Table 5.6 presents the results of accuracy, detection rate, and false alarm rate; clearly, the result indicated a high rate of detection 99.57 %, at the same time low false alarm rate of 0.35%.

The performance evaluation of the proposed approach consists of two phases. First, a mathematical equation was applied and the second phase was carried out by

comparing the result of the proposed approach and different hybrid intelligent approaches.

Table 5. 7 Performance Evaluation averaged over 5 trials for 2 attributes.

Classifier Algorithms	**Datasets**	**Performance Measures in %**			
		Accuracy	**Detection rate**	**False Alarm**	**F–Measure**
C5.0	UNCG	0.9979	0.8889	0.0014	0.9143
IWC+C5.0	IBRL	0.9972	0.9898	0.0031	0.9940
	Bharatpur Airport	0.9971	0.9957	0.0035	0.9966

As in Table 5.7, anomaly detection in two attributes (i.e. temperature and humidity) with C5.0 on labeled UNCG WSN datasets with 4690 reading records, and with combined IWC+C5.0 on both IBRL and Bharatpur Airport WSN data whose detection rate and false alarm results show that when two of the best classifier is combined, the detection rate exceeds 98% with very low false alarm rate up to 0.35%.

Result Analysis

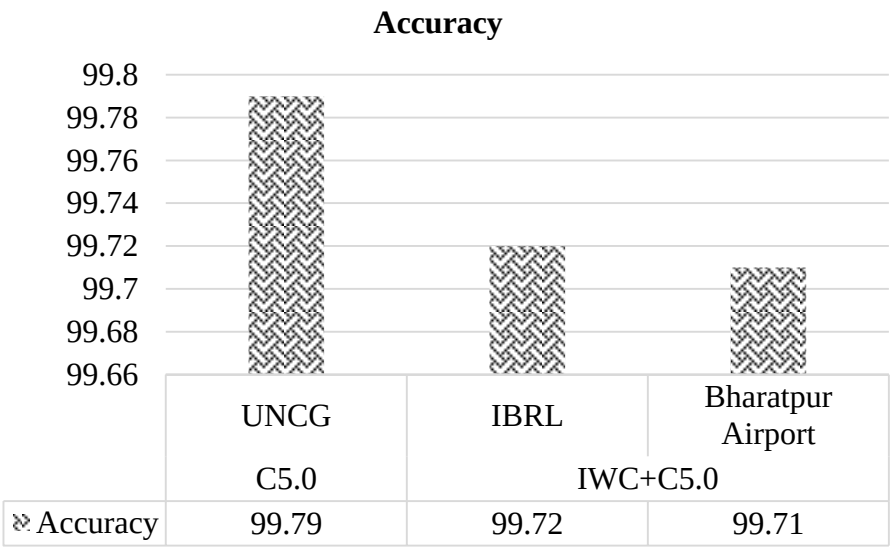

Figure 5. 3 *Accuracy on different data*

As in Figure 5.3, we can see that the accuracy with C5.0 on labeled UNCG datasets is 99.79%, accuracy of integrated IWC+C5.0 on real Intel Berkeley Research Lab dataset is 99.72% and on Bharatpur airport WSN datasets is 99.71% respectively.

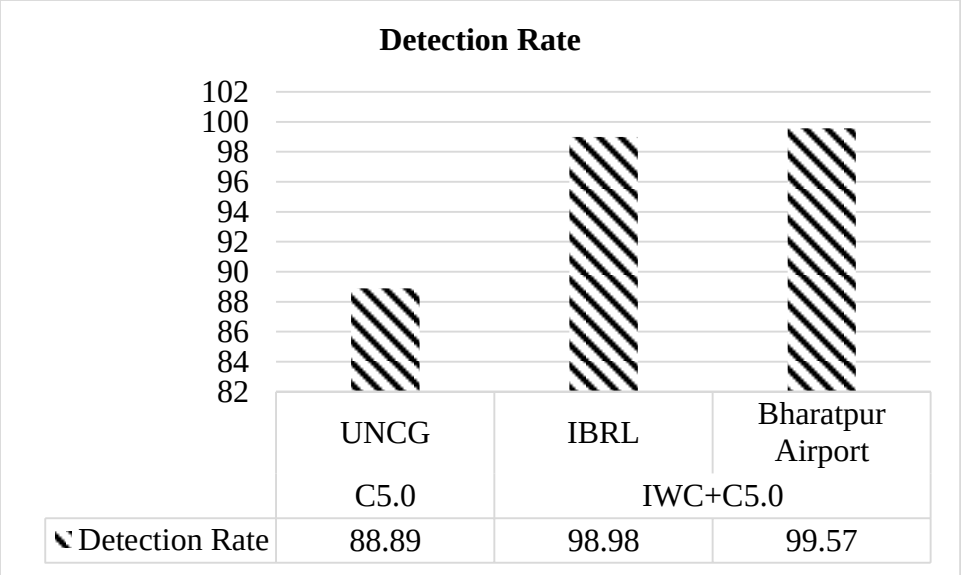

Figure 5. 4 *Detection rate on different data*

In Figure 5.4, we can see the detection rate with C5.0 on labeled UNCG dataset is 88.89%, detection rate with integrated IWC+C5.0 on real IBRL datasets is 98.98% and on Bharatpur Airport WSN datasets is 99.57% respectively.

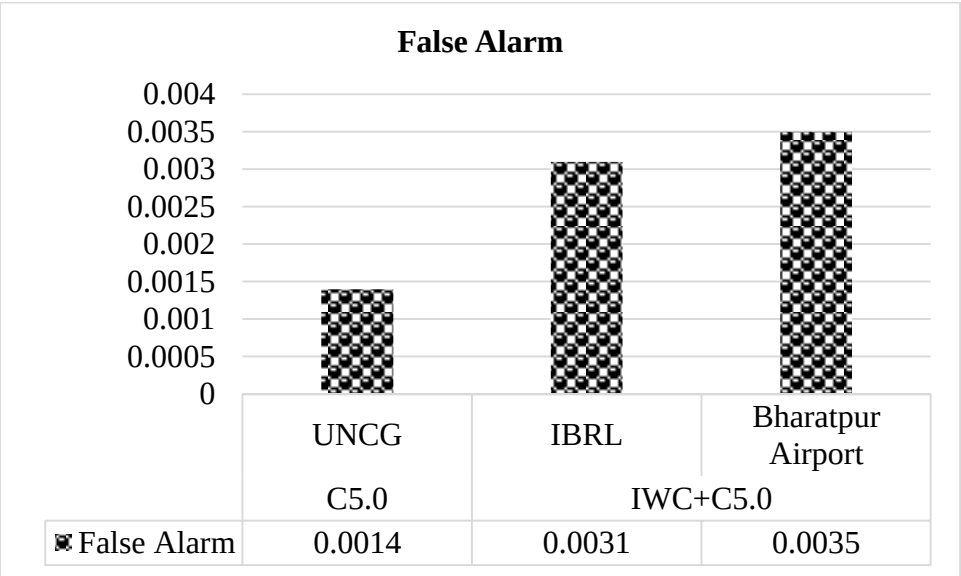

Figure 5. 5 *False Alarm results on different data*

In Figure 5.5, we can see the false alarm rate with C5.0 on labeled UNCG data is 0.14% and with integrated IWC+C5.0 on IBRL is 0.31% and on Bharatpur Airport WSN is 0.35% respectively.

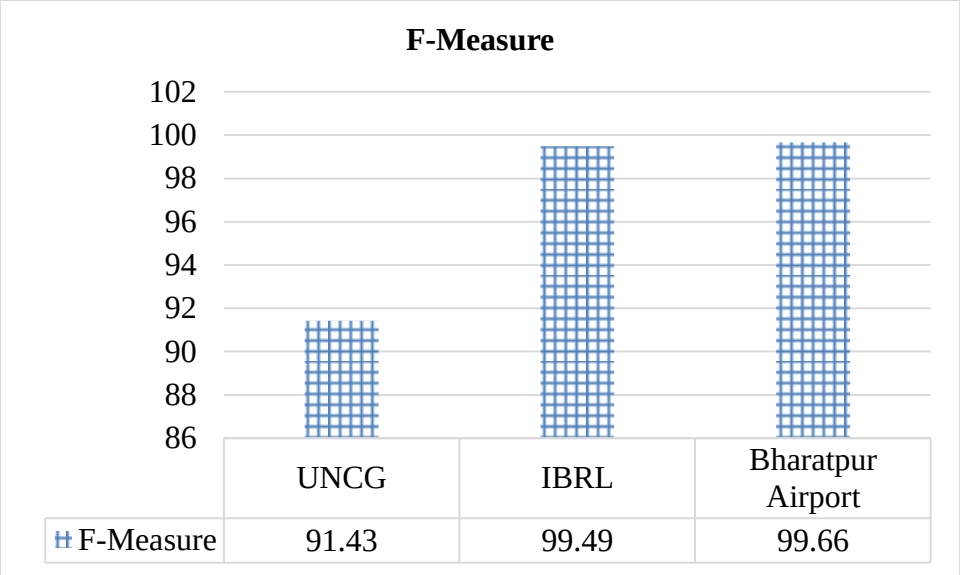

Figure 5. 6 F-*Measures on Different data*

In Figure 5.6, we can see the F-Measures with C5.0 on labeled UNCG data is 91.43% and with integrated IWC+C5.0 on IBRL is 99.49% and on Bharatpur Airport WSN is 99.66% respectively.

Comparisons

The final phase was the evaluation process to enable correlation with some intelligent approaches for WSN anomaly detection to verify that, the proposed approach has improved the detection rate and decrease the false alarm rate. Based on the foregoing, the proposed approach was compared with four some of the hybrid intelligent approaches for WSN anomaly detection on IBRL and Bharatpur Airport datasets. Table 5.8 shows the comparison and differences between these approaches in detection rate.

Table 5. 8 Different Techniques Vs. Proposed IWC+C5.0 Approach comparison

Datasets	Techniques	Accuracy	Detection Rate	Alarm Rate	F-Measure
IBRL	IWC+Naïve Bayes	98.20	98.2	1.79	98.2
	IWC+KNN	98.64	98.6	1.35	98.6
	IWC+SVM	99.37	99.4	0.62	99.4
	IWC+C5.0	**99.72**	**98.9**	**0.31**	**99.4**
Bharatpur Airport	IWC+Naïve Bayes	98.74	98.8	1.25	98.7
	IWC+KNN	99.61	99.6	0.38	99.6
	IWC+SVM	99.42	99.4	0.57	99.4
	IWC+C5.0	**99.71**	**99.5**	**0.35**	**99.6**

The Table 5.8 gives the percentage of accuracy, detection rate, false alarm rate and f-measure. The accuracy for proposed algorithm on IBRL is 99.72 and is greater than Naïve Bayes (98.2), KNN (98.64) and SVM (99.37). Similarly the detection rate is 98.9 which is greater than IWC+Naïve Bayes (98.2), IWC+KNN (98.6) except IWC+SVM (99.4). Also false alarm rate is 0.31 which is lesser than that of Naïve Bayes (1.79), KNN (1.35) and SVM (0.62). On Bharatpur Airport dataset accuracy is 99.71 and is greater than Naïve Bayes (98.7), KNN (99.61) and SVM (99.42). Similarly the detection rate is 99.5 which is greater than IWC+Naïve Bayes (98.8), IWC+SVM (99.4), except the IWC+KNN (99.6). Also false alarm rate is 0.35 which is lesser than that of Naïve Bayes (1.25), KNN (0.38) and SVM (0.57).

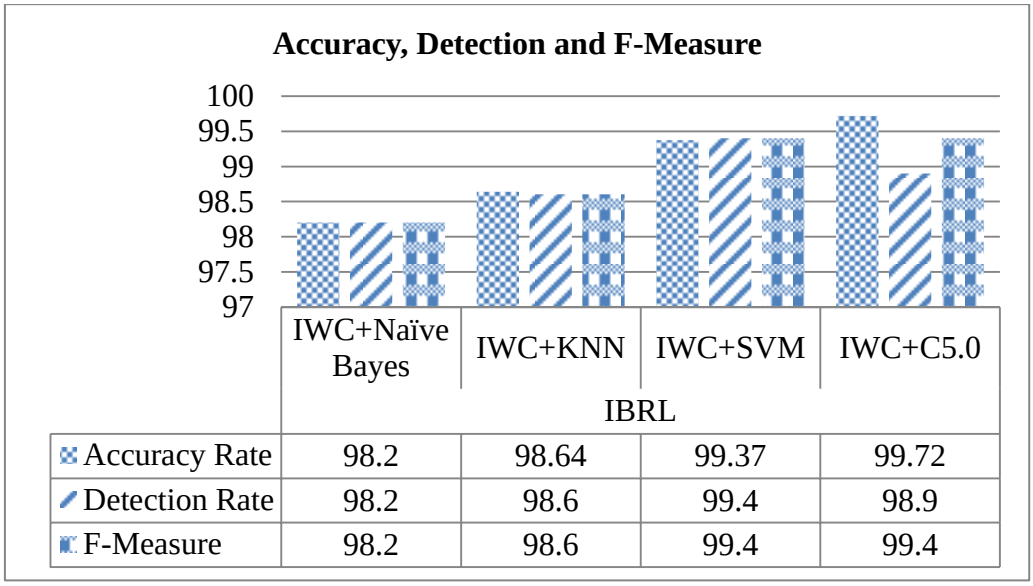

Figure 5. 7 *Accuracy, Detection rate and F-Measure on IBRL data*

The Figure 5.7 chart compares the results of proposed method IWC+C5.0 for accuracy, detection rates and f-measure with different integrated techniques such as IWC+Naïve Bayes, IWC+KNN, IWC+SVM on IBRL datasets.

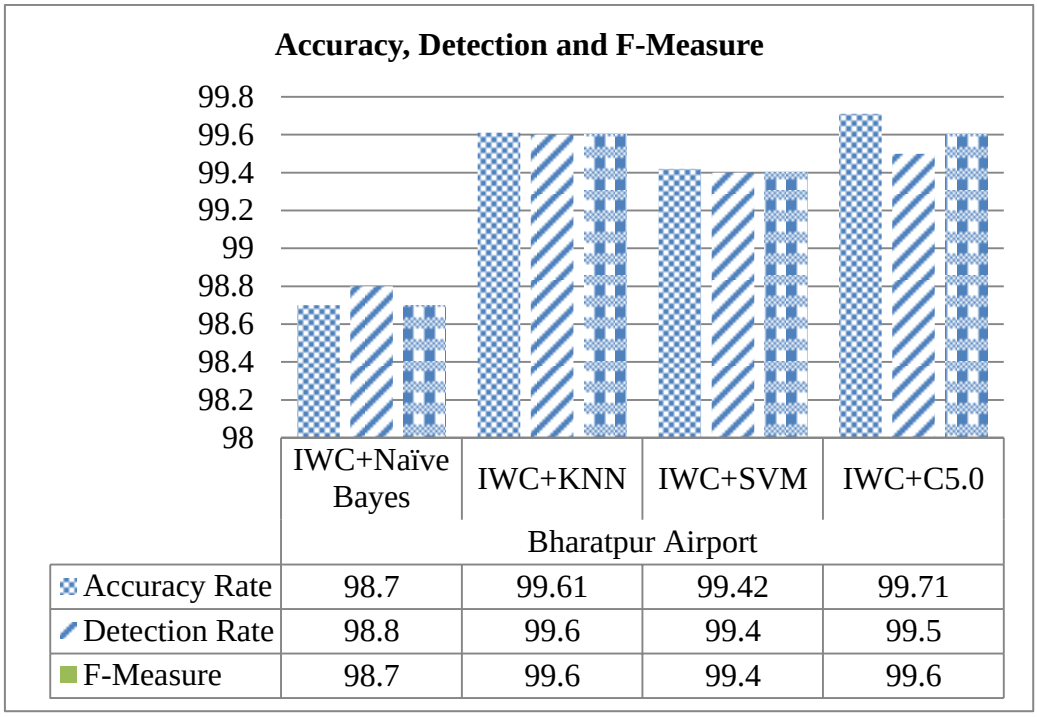

Figure 5. 8 *Accuracy, Detection rates and F-Measure on Bharatpur Airport data*

The Figure 5.8 chart compares the results of proposed method IWC+C5.0 for accuracy, detection rates and f-measure with different techniques such as IWC+Naïve Bayes, IWC+KNN, IWC+SVM on Bharatpur Airport datasets.

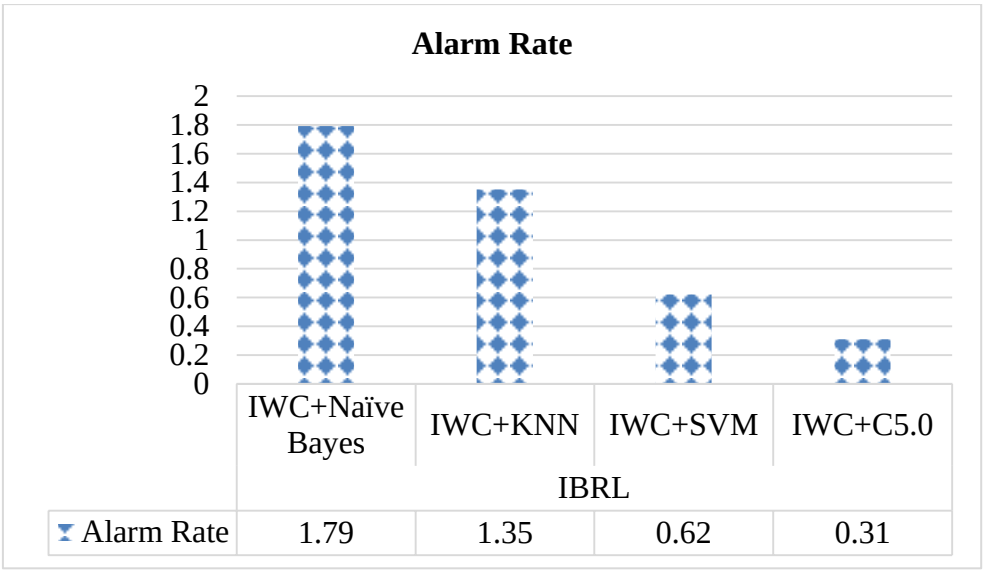

Figure 5. 9 *False Alarm rate with different techniques on IBRL*

The Figure 5.9 chart compares the results of proposed method IWC+C5.0 for false alarm rates with different integrated techniques such as IWC+Naïve Bayes, IWC+KNN, IWC+SVM on IBRL datasets.

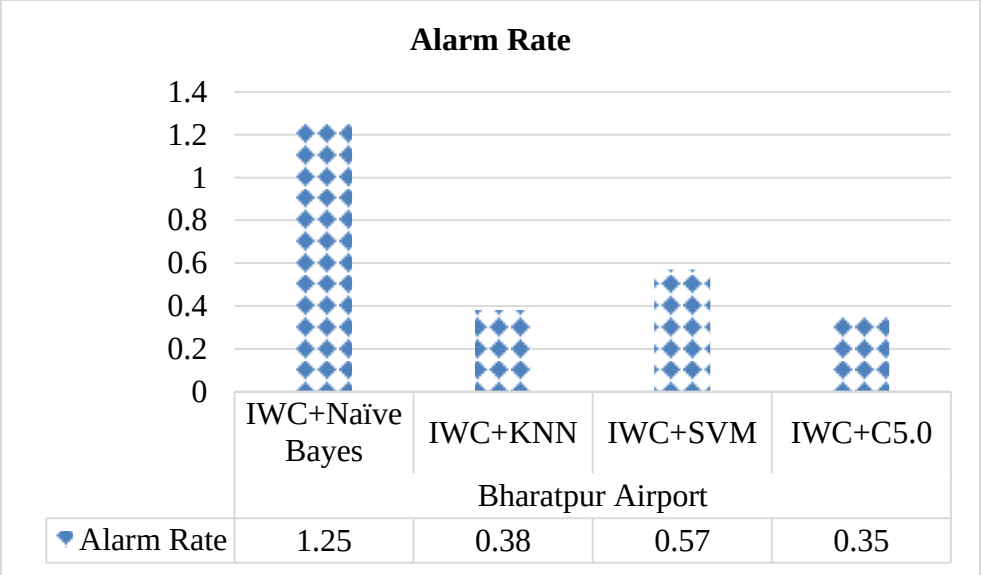

Figure 5. 10 *False Alarm rate with different techniques on Bharatpur Airport*

The Figure 5.10 chart compares the results of proposed method IWC+C5.0 for false alarm rates with different integrated techniques such as IWC+Naïve Bayes, IWC+KNN, IWC+SVM on Bharatpur Airport datasets.

The result findings show that the proposed IWC+C5.0 model is the efficient technique for detecting anomaly averaged over 5 trials for Intel Berkeley Research lab wireless sensor network, with a higher rate of detection (98.98 %), and lower false alarm rate (0.31%). Similarly for Bharatpur Airport WSN readings, higher rate of detection (99.57%) and false alarm rate is (0.35%).

CHAPTER 6: CONCLUSION AND RECOMMENDATION

This chapter describes the conclusions derived from the results and the future enhancements and recommendation the research work.

6.1 Conclusion

The main goal of this research was to implement anomaly detection using two machine learning techniques in WSN. It was implemented through usage of two algorithms, Inverse Weighted K-Means and Decision tree algorithms. The Inverse Weighted K-means was first applied to partition the dataset into K clusters and then C5.0 decision tree was built on each cluster for better classification of instances, the C5.O decision tree and cluster labels were used to classify the instances as normal and anomaly. The results gained from implementing these two algorithms were then compared in order to see which of them is best suited to perform anomaly detection in a Wireless Sensor Network environment.

The Experimental results were performed on WSN Dataset, and it was shown that overall performance of the proposed approach improved in terms of detection rate and low false alarms rate. The proposed model integrated IWC algorithm for clustering jobs and C5.0 decision tree for prediction jobs for anomaly detection in WSNs and tested in terms of accuracy, True positive (TP), True negative (TN), False positive (FP), False negative (FN), Recall, Precision and F-Measure. The experiment has been performed using Intel core 5 Processor with 4 GB of RAM and MATLAB 2017.

To evaluate the performance of proposed technique Confusion matrix was used, it contains data about actual and predicted classifications. The proposed anomaly detection model could reach high detection rate exceeding 98.98% with a very small false alarm rate 0.31% on IBRL WSN datasets; and detection rate of 99.57% at low false alarm of 0.35% on Bharatpur Airport data.

To evaluate the proposed model with a set of experiments with both real life dataset obtained from Intel Berkeley research lab, Bharatpur Airport WSN and synthetic dataset from UNCG was performed. Experimental results and comparison with recent

existing work indicate that the new model is promising in terms of achieving high detection effectiveness while efficiently utilizing the limited resources.

6.2 Recommendations and Future Work

There are so many different algorithms that could be used to do classification and in this research two of the most common (i.e. they are widely used in research) clustering and classification algorithms were chosen to be compared. The limitation to detect application level anomaly will be enhanced to detect network layer anomaly too.

For further improvement in the detection accuracy, future work will be to combine different clustering algorithms such as Hierarchical clustering, Adaptive resonance (ART) Neural network and Kohonen's self _organizing maps with decision tree C5.0. Furthermore, other partitioning algorithms could be used instead of IWC to find out whether better results can be achieved.

REFERENCES

[1] S. S. Bhojannawar, C. M. Bulla and e. al., "Anomaly Detection Techniques for Wireless Sensor Networks - A Survey," International Journal of Advanced Research in Compuetr and Communication Engineering, vol. 2, no. 10, Oct 2013.

[2] Z. Feng, J. Fu and et al., "A new approach of anomaly detection in wireless sensor networks using support vector data description," International Journal of distributed Sensor Networks, vol. 13, 2017.

[3] C. O'Reilly, S. Rajasegarar and e. al., "Anomaly Detection in Wireless Sensor Networks in a Non-Stationary Environment," ieee communications surveys & tutorials, vol. 16, no. 3, Third Quarter 2014.

[4] R. Sharma and V. A. Athavale, "Survey of Intrusion Detection Techniques and Architecture in Wireless Sensor Networks," Int. J. Advanced Networking and Applications, vol. 10, no. 04, pp. 3925-3937, 2019.

[5] M. E. Elhamahmy, H. N. Elmahdy and et al., "A New Approach for Evaluating Intrusion Detection System," International Journal of Artificial Intelligent Systems and Machine Learning, vol. 2, no. 11, Nov 2010.

[6] N. Reka, "Wireless Sensor Network Architecture(WSN)," (IJCSIT) International Journal of Computer Science and Information Technologies, vol. 6, no. 4, pp. 3706-3708, 2015.

[7] [Online]. Available: http://article.sapub.org/10.5923.j.jwnc.20150501.03.html.

[8] [Online]. Available: https://www.elprocus.com/architecture-of-wireless-sensor-network-and-applications.

[9] W. Barbakh and C. Fyfe, "Inverse Weighted Clustering Algorithm," The University of Paisley, Scotland.

[10] S. R. Gaddam, V. V. Phoha and et al., "K-Means+ID3: A Novel Method for Supervised Anomaly Detection by Cascading K-Means Clustering and ID3

Decision Tree Learning Methods," IEE Transactions on Knowledge and Data Engineering, vol. 19, no. 3, March 2007.

[11] R. M. Elbasiony, T. E. Eltobely and et al., "A hybrid network intrusion detection framework based on random forests and weighted k-means," Ain Shams Engineering, p. 753–762, Dec 2013.

[12] M. Wazid and A. K. Das, "An Efficient Hybrid Anomaly Detection Scheme using K-Means Clustering for Wireless Sensor Networks," Wireless Personal Communications, vol. 90, no. 4, pp. 1971-2000, Oct 2016.

[13] Y. Li and J. Xia, "An Efficient Intrusion Detection System based on SVM and Gradually Feature Removal Method," in Expert Systems with Applications, Elsevier, 2012.

[14] V. Golmah, "An Efficient Hybrid Intrusion Detection System based on C5.0 and SVM," International Journal of Database Theory and Application, vol. 7, no. 2, pp. 59-70, 2014.

[15] W. Yassin, N. I. Udzir and et al., "Anomaly - Based Intrusion Detection through K-Means Clustering and Naive Bayes Classification," in Procedings of the 4th International Conference on Computing and Informatics, ICOCI, Sarawak, Malaysia, 2013.

[16] H. M. Tahir, W. Hasan and et al., "Hybrid Machine Learning Technique for Intrusion Detection System," vol. 209, p. 464–472, 2015.

[17] K. H. Rao, G. Srinivas and et al., "Implementation of anomaly detection technique using machine Learning Algorithms," International journal of computer science and and Telecommunications, vol. 2, no. 3, June 2011.

[18] P. C. Yong , C. Xiang and et al., "Design Of Multiple-Level Hybrid Classifier For Intrusion Detection System Using Bayesian Clustering And Decision Trees," in Pattern Recognition Letters, Singapore, 2008.

[19] G. Kim and S. Lee, "A Novel Hybrid Intrusion Detection Method Integrating

Anomaly Detection With Misuse Detection," in Expert Systems with Applications, Daejeon, South Korea, 2014.

[20] J. Wang, Q. Yang and et al., "An intrusion detection algorithm based on decision tree technology," in 2009 Asia-Pacific Conference on Information Processing, 2009.

[21] A. P. Muniyandi, R. Rajeswari and et al., "Network Anomaly Detection by Cascading K-Means clustering and C4.5 Decision tree algorithm," in International Conference on Communication Technology and System Design, 2012.

[22] M. A. Alsheikh, S. Lin and et al., "Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications," in Sense and Senseabilities Programme, Institute for Infocomm Research, Singapore 138632, 2015.

[23] S. Omar, A. Ngadi and et al., "Machine Learning Techniques for Anomaly Detection: An Overview," International Journal of Computer Applications, vol. 79, no. 2, p. 0975 –8887, Oct 2013.

[24] P. Bansal and D. Garg, "A Hybrid Approach to improve the Anomaly Detection rate using Data Mining Techniques," Patiala, 2015.

[25] T. S. Madhulatha, "An Overview on Clustering Methods," IOSR Journal of Engineering, vol. 2(4), pp. 719-725, Apr. 2012.

[26] J. R. Quinlan, "C4.5: programs for machine learning," Baltimore, Morgan Kaufmann publishers, 1993, pp. 235-240.

[27] L. Rokach and O. Maimon, "Clustering Methods," Department of Industrial Engineering, Tel-Aviv University.

[28] N. Patil, R. Lathi and et al., "Comparison of C5.0 & CART Classification algorithms using pruning technique," International Journal of Engineering Research & Technology (IJERT), vol. 1, no. 4, 2012.

[29] [Online]. Available: http://www.uncg.edu/cmp/downloads/.

[30] [Online]. Available: http://db.csail.mit.edu/labdata/data.txt.gz.

[31] S. R. Meesala and S. B. Xaviar, "A Hybrid Intrusion Detection System Based on C5.0 Decision tree and One-Class SVM," International journal of current Engineering and Technology, vol. 5, no. 3, June 2015.