Anomaly Detection and Analysis in Big Data

*A **Thesis***

submitted in partial fulfillment of the requirements for the award of degree of

Doctor of Philosophy

Submitted by

Sahil Garg

(901403026)

Under the guidance of

Dr. Shalini Batra

Associate Professor, Computer Science and Engineering Department,

Thapar Institute of Engineering and Technology, Patiala, India

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology

Patiala-147004, India

September 2018

Contents

List of Figures

List of Tables

List of Algorithms

Certificate

I hereby certify that the work which is being presented in this thesis entitled "**Anomaly Detection and Analysis in Big Data**", in partial fulfillment of the requirement for the award of degree of "Doctor of Philosophy" submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Shalini Batra and refers other research works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

(**Sahil Garg**)

Regn. No. 901403026

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(**Dr. Shalini Batra**)

Associate Professor

Computer Science & Engineering Department

Thapar Institute of Engineering and Technology

Patiala, 147004

Punjab, India.

Acknowledgements

With profound sense of gratitude and heartiest regard, I would like to express my sincere appreciation to my supervisor, **Dr. Shalini Batra**, for being a pillar of support and encouragement throughout my research work. Her experience, strength, generous and positive attitude, sincere effort and invaluable co-operation has taught me valuable lessons of life, which are going to be of immense help to me in taking decisions throughout my life.

I am grateful to **Dr. Maninder Singh**, Professor and Head, Computer Science and Engineering Department (CSED), Thapar Institute of Engineering and Technology, Patiala for providing all the necessary administrative assistance and making my study a relevant experience. I am thankful to my Ph.D. committee members, **Dr. Parteek Kumar**, Associate Professor, CSED, **Dr. Prashant Singh Rana**, Assistant Professor, CSED, and **Dr. Amit Kumar**, Assistant Professor, School of Mathematics Computer Applications for their constructive comments and regularly ensuring the progress of my research work. I am thankful to all the **faculty** and **staff members** of CSED for their support.

I would like to acknowledge the cooperation and motivation extended to me by **Dr. Neeraj Kumar**, Associate Professor, CSED. His supreme knowledge and invaluable guidance provided me a delightful ambiance for learning and making this thesis possible.

Most importantly, I would like to thank my **Family**, and **my wife Kuljeet** for showing me the right direction out of the blue and to help me stay calm in the oddest of the times. The chain of gratitude would be definitely incomplete without thanking the **Almighty**, the prime mover, for inspiring and guiding me to complete this task successfully.

Patiala

September, 2018

Garg

(Sahil Garg)

Abstract

With an exponential increase in the Internet traffic over the network, penetration of security threats in the underlying computer networks has witnessed a major increase. More recently, the severity of their impacts has undergone a considerable transformation from uncomplicated sniffing and spoofing attacks to the more complicated and critical attacks like denial of service. Due to the occurrence of such anomalies in the network, its normal operations get affected adversely in terms of traffic classification, resource allocation, and service management. Further, due to the rapid proliferation of emerging computing paradigms such as-Internet of Things (IoT), Edge/Fog Computing, Smart Grids, Software Defined Networks (SDN), etc., massive amount of data is being generated at an unprecedented rate. The unconventional 5V features (volume, velocity, variety, veracity and variability) of the data being generated have given birth to Big Data. The impact of this data abundance is further leading to several security threats and thus, its management and analysis requires schemes for Big Data analytics. Hence, it is essential to detect the notorious anomalies in the network on a real-time basis.

Most of the existing anomaly detection solutions reported in the literature are not so efficient for large-scale networks due to various reasons such as-curse of dimensionality, imbalance between classes, and variations in the types of anomalies. The efficiency of any model mainly depends on the selection of relevant features and the learning algorithms; which in turn play a vital role in classification of the network traffic patterns into benign and malicious. Keeping in view of the above challenges and proliferation of Big data, the problem of anomaly detection in network traffic data has been considered in this thesis. Consequently, two different ensemble based techniques for anomaly detection have been proposed particularly for network wide traffic.

The first technique, Ensemble based Anomaly Detection Technique (En-ADT), employs

the combination of Fuzzy K-means clustering algorithm, Extended Kalman Filter and Support Vector Machines for the detection of various anomalies in the network. Here, the first module is used in the identification of the optimal subset of features which are then refined by the second module. Using these features, the third module classifies the traffic to identify the malicious entities. Another technique, Fuzzified Cuckoo based Clustering Technique (F-CBCT), has been proposed for the proactive prediction of attacks in networked traffic data. It operates in two phases: a training phase, where the system is trained to recognize the anomalies, and a detection phase, where the system detects anomalies on the basis of employed algorithms and the input data. The results and analysis of the proposed anomaly detections schemes over the benchmark datasets are presented on the basis of standard evaluation parameters such as-detection rate, false positive rate, accuracy and F-score.

Chapter 1

Introduction

Anomaly detection is a branch of data mining that deals with the discovery of rare occurrences or patterns in the data that deviate substantially from their expected behavior. These unusual occurring patterns are interchangeably termed as anomalies, abnormalities, outliers, irregularities, deviations, exceptions or peculiarities in different application domains of literature [1]. Anomaly detection is one of the most evident but well established problems in data mining which have numerous applications. Its relevance in the modern era of Internet can be attributed to the fact that, anomalies in a computer or network can destabilize the working of data-overloaded modern society. The security of Internet, which is the medium of millions of interconnected computers and provider of transportation to all types of information, is of deep concern. Any sort of harm to such a network may disrupt its confidentiality, availability or integrity [2].

Anomalies are not always bad or indicative of a failure. It is true that these non-conforming patterns are often difficult to investigate but have a great significance. It is because the anomalous data instances sometimes contain useful information about the abnormal characteristics of the system that can be translated to significant actionable information for taking the prompt actions in a wide variety of critical application domains. For example,

fatal heart rates or unusual ECG artifacts may indicate the danger of heart-attack, anomalous transactions in credit card bill may be indicative of a credit card fraud, changes in shopper buying habits could signify a discovery of recent trend or anomalous stock fluctuations could mean that there is a instability in the stock market [3].

The problem to detect anomalies often becomes complicated due to the high variability in data. It is extremely valuable in many domains such as-IT security, finance, vehicle tracking, health-care, safety critical systems, e-commerce and all other domains where sensors exponentially produce large hierarchies of data. For all such applications, anomaly detection techniques consider that normal instances in dataset are much more than the number of anomalous instances [4]. Though anomalies are not always rare, for example, in case of computer worm detection, the worms (anomalies) are much more recurrent than normal instances. Thus, anomaly detection can be considered as deciphering the unknown with the help of known. It is basically a discovery process that forms a baseline behavioral model for data and any deviations from this baseline model is considered as anomalous.

Anomaly detection is vital for spotting rare events but many times these techniques fail to distinguish between anomaly and noise [5]. Anomaly detection is associated to, but distinct from noise removal. Noises are unwanted objects that cause hindrance in data analysis, while anomaly detection itself is a part of data analysis. Noise can be modeled as a weak form of anomaly but it does not mean that every noise will be anomalous. Though it is true that the separation between the anomaly and noise is not precise, but still its removal and identification is important for anomaly detection.

Similarly, there is confusion between change analysis and anomaly. These two areas are closely related but are not identical. In the first case, changes may occur slowly over time and can be detected by doing the detailed analysis over a long period of time [6]. But in the latter, the changes occur abruptly. So, it is not necessary that the change analysis corresponds

to anomalies but it can be a concept drift, though the second scenario resembles the same [7].

Anomalies are considered as the deviations from the normal behavior but it is still a question of interest that what is the sufficient deviation required for any point to be considered as anomalous. The data analysts are still working on this question and preparing such data models that can deal with similar situations. All such data models make distinct assumptions about the normal behavior of the data [8]. These models compute outlierness of a data point by evaluating the nature of fit between the data point and the data model. However, if the data model is not learnt with the suitable number of data samples or if the data that is to be evaluated does not fit well with the data model, then a model may not work properly. This in-turn may lead to incorrect results such as-normal data point may be erroneously evaluated as anomaly. Thus, a good intuitive understanding and careful evaluation of the data domain is needed to build an effective evaluation model [9]. Nevertheless, this is a case where the normal behavior of the data is known but the problem occurs in such situations where the behavior cannot be characterized by a data model. So, it is still a challenge to develop a data model that can detect anomalies without having a prior knowledge about the normal behavior of the data.

Different authors defined anomaly in numerous ways. Some of the most widely used definitions are given in Table 1.1.

1.1 Background

Anomaly detection has caught the interest of the researchers all around the world due to its vast utility in various domains like networking, financial fraud detection, tumor prediction, bug prediction, intrusion detection, spam filtering [16–18] *etc.* Network anomalies form an important area which requires robust anomaly detection techniques. In networks, various

Table 1.1: Several definitions of anomaly

Author(s)	Definition
Chandola *et al.* (2009) [3]	Those patterns in data that do not conform to a well defined notion of normal behavior
Jiang *et al.* (2010) [10]	The objects which behave in an unexpected way or have abnormal properties
Thottan *et al.* (2010) [11]	An event that deviates from the normal network behavior
Reilly *et al.* (2014) [12]	An anomaly or outlier is defined as an observation (or subset of observations) that appears to be inconsistent with the remainder data set
Daneshpazhouh *et al.* (2014) [13]	Outliers are generally viewed as observations that are far away from, or inconsistent with the main body of the data set
Kaur *et al.* (2015) [14]	An anomaly is defined as an unusual activity exhibiting a different behavior than others present in the same structure
Agrawal *et al.* (2015) [15]	The patterns in a dataset whose behavior is not normal on expected
Ahmed *et al.* (2016) [1]	Anomalies are referred to as patterns in the data that do not conform to a well-defined characteristic of normal behavior

types of threats and anomalies, from Denial of Service (DoS) attack to probing attack, lead to complex security challenges. These threats and anomalies take intentional actions against the data, design, monitoring ability and software or hardware to destroy, degrade or to make the network inaccessible [19,20]. So, the anomaly detection tools are increasingly becoming vital and valuable for all the organizations. Till date two dominant approaches have been used to handle attacks and intrusions, namely-signature based detection and anomaly detection.

The concept of signature based detection [21] relies on the learning of attributes of the network threats. More specifically, a network expert creates a signature for every attack such that the attack with the pre-known signature can be detected as soon as it is launched.

This technique detects only those attacks whose signatures have been provided earlier by the network expert and thus no novel attack can be detected by this technique. Moreover, signatures only work well against those attacks which have fixed behavioral pattern. So, it can be concluded that the systems which can detect only those attacks that are pre-known to it are highly fragile and vulnerable to the attacks.

On the contrary, anomaly detection techniques detect everything that does not conform to the expected behavior. These approaches build their knowledge base by extracting the information about normal instances from attack free training sets and consider those instances as anomalous that do not fall within the baseline of their normal activity profile [22]. Moreover, anomaly detection technique has an advantage over signature based technique that a new attack or intrusion for which a signature is not defined can be detected if it exhibits deviation from the baseline of its normal profile. However, if any malicious activity falls within the normal profile of anomaly detection algorithm then it will not be identified. Thus, it is quite expensive and time consuming to obtain attack free training sets and build normal activity profile. Moreover, it is also very complex to keep this normal activity profile updated once built.

Several machine learning and data-mining algorithms have been proposed by the researchers to solve the problem of anomaly detection but they suffer from certain dilemmas like forced assignments of data points to a cluster, class dominance, local convergence and sensitivity towards selection of cluster like problems [23]. Further, it has been observed that same technique when applied to different datasets produce dissimilar results in terms of performance. Such variations are observed because each dataset varies in terms of noise and complexity, so here selection of features and attributes becomes an important aspect of concern. Another issue which is frequently witnessed is the "*optimization of features*". Any algorithm or technique can give the best results only if the attributes provided for analy-

sis are the finest ones. Several anomaly detection techniques based on partitioning, fuzzy theory, density, artificial neural networks, multivariate analysis, *etc*. have been frequently considered as they have high detection rate (DR) and minimal false positive rate (FPR). Recent advances in this area clearly indicate that hybridization of multiple anomaly detection techniques will provide better results than individual anomaly detection methods [14, 24]. However, rapid growth of information technology has resulted in the enhancement of security threats in computer networks. Since any malicious activity on network may lead to serious consequences, the importance of information carried out in these networks makes the task of anomaly detection very crucial [16].

1.2 Classification of Anomalies

On the basis of nature of data instances, anomalies are grouped into three major categories which are as follows [3, 25]:

1.2.1 Point Anomalies

If an individual data instance is independent from rest of the data, then the instance is considered as point anomaly. For example, let the dataset correspond to an individual's tweeting habit. A day for which the tweets done is far more than the individual's normal range of tweeting per day, will be considered a point anomaly. Although this is the simplest kind of anomaly but the major problem associated with the detection of such anomalous events is the measurement of deviation that makes it different than others.

1.2.2 Contextual Anomalies

If an instance deviates significantly in a particular context, then it is considered as contextual anomaly. For example, the weather pattern is anomalous or not will depend upon the time and location. A -30°C temperature of Siachen Glacier is anomalous if considered in summers but, an equally high temperature during winters is normal. Defining a context for some cases is straightforward that makes contextual anomaly detection an easy task. However, it may not be easy to define a context for applying contextual anomalies. Each data instance has two set of attributes by which its context is defined:

- Contextual Attributes: These set of attributes are used to determine the context of data instance. For example, in weather example, the time and location are the contextual attributes.

- Behavioral Attributes: These attributes define the characteristics of data instances to identify their anomalous behavior in a particular context. For example, in weather example, temperature is a behavioural attribute.

1.2.3 Collective Anomalies

If multiple data instances are found to deviate simultaneously and depict different behavior with respect to the entire dataset, then it is called as collective anomaly. These types of anomalies often occur in those datasets in which data instances are closely related to each other. Here, the occurrence of individual data instances may be normal but their occurrence as a whole will be anomalous. For example, for a particular course only a group of students can register and no individual entry can be done. Now, if one student from the registered group leaves the course then it may be considered as normal. On the other hand, if multiple students leave the group then it will be marked as anomalous.

1.3 Modes of Machine Learning Algorithms

In the context of anomaly detection, two types of labels are associated with data instances, i.e., normal or anomalous [3]. However, the labeled data which represents all types of behaviors accurately is often difficult to obtain. Thus, anomaly detection techniques operate in three modes:

1.3.1 Supervised Anomaly Detection

These techniques assume the availability of a training dataset where data instances are labeled as normal and anomalous [26]. Here, the inducer generates a set of profiles corresponding to the class labels. Then, a class predictor module compares the test instances against the built model to determine which class it belongs to. There are certain drawbacks in supervised techniques such as-(i) less number of anomalous instances in comparison to normal instances for the training, (ii) obtaining correct labels for the anomalous class is challenging, and (iii) need of experts for labeling of classes is expensive and time consuming [27].

1.3.2 Unsupervised Anomaly Detection

These techniques do not require training data and groups data instances into categories on the basis of intrinsic properties of the objects, i.e., using a similarity measure or by computing the distance between any pair of instances [28]. In such techniques, it is assumed that normal instances in data are far more than the anomalous ones. If this assumption fails then such techniques often suffer from high false positives.

1.3.3 Semi-Supervised Anomaly Detection

A semi-supervised techniques require labeled instances for normal class in addition to the unlabeled ones for training purpose [29]. Thus, a model for the normal class is built which is then used to identify anomalies in the test data. Moreover, the anomalous behavior is often dynamic in nature and getting labeled set of anomalous data instances is a difficult task. Since, semi-supervised anomaly detection techniques do not require labeled data for the anomalous class, they are more often used in comparison to supervised approaches.

1.4 Types of Anomaly Detection Techniques

Anomaly detection techniques construct a region of normal profile and consider all the observations as anomalous which do not belong to this normal region. By adapting the concepts from diverse domains such as-machine learning, statistics, information theory, data mining, etc., several anomaly detection techniques have been proposed by considering the characteristics of the problem like nature of data, labels, type of anomaly, etc [1, 3, 4]. These are as demonstrated in Fig. 1.1. All these techniques are discussed in the subsequent sections.

1.4.1 Classification Techniques

It classifies the data as normal or anomalous based on some set of rules, patterns or some other distinguishing measure. Here, a profile of normal behaviour is built based on training data and categorization of new observations is done to determine the anomalous events. Classification based anomaly detection scheme operates in two steps [30].

- *Training Phase:* Trains a classifier by using the best available training data.
- *Testing Phase:* Classifier classifies a test instance as normal or anomalous.

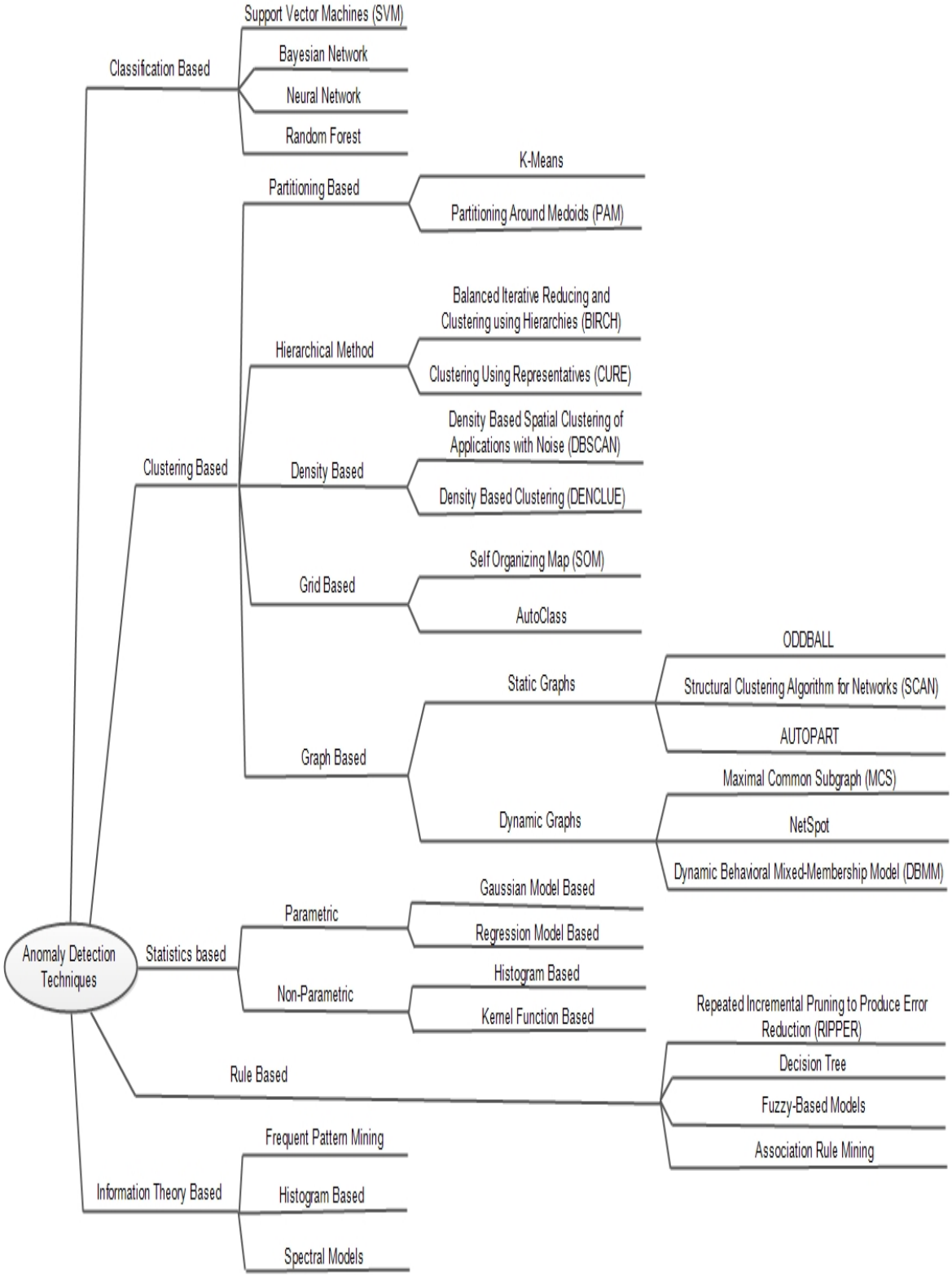


Figure 1.1: Overview of the anomaly detection techniques

On the basis of availability of labels, these techniques are classified into two categories:

One-class classification

If the classification of observed data is done in context to a single characteristic, then this type of classification is known as one-class classification. Here, the classifier uses a one-class classification algorithm such as one-class SVM to formulate a discriminative boundary around the normal instances. Any instance of dataset that does not fall within the formulated boundary is considered as anomalous.

Multi-class classification

If multiple characteristics are considered in order to classify the observed data, then it is known as multi-class classification. Here, anomaly detection techniques learn a classifier to distinguish between each normal class against the rest of the classes. A data instance is classified as anomaly if none of the classifiers are confident in considering that instance as normal.

1.4.2 Clustering Techniques

A cluster is an aggregation of entities that are alike and clustering is a technique that is used to partition similar observations into clusters. It is an unsupervised technique where labeled data is not required to discover the underlying patterns for partitioning, i.e., no prior knowledge about the labels is required [31]. This data analysis technique partitions the dataset into disjoint groups to derive more abstract structures of the dataset.

Clustering techniques have a great significance in the area of anomaly detection. They consider anomalies as dissimilarities and classify them by using similarity matrices. They assume that normal instances occur more frequently and hence follow a pattern; whereas

anomalies do not behave in this manner [32]. Clustering algorithms for anomaly detection are based on the following assumptions:

- *Assumption 1:* Normal data instances in the dataset follow the pattern and form a cluster based on similarity, while anomalies do not follow such patterns. A disadvantage of this assumption is that it is not focused on finding anomalies rather this approach mainly aims to form clusters.

- *Assumption 2:* Normal data instances lie close to the centroid cluster while anomalies lie far away from the centroid cluster. A disadvantage of this assumption is that if anomalies in the data form clusters by themselves then this technique will not be able to detect such anomalies.

- *Assumption 3:* Normal data instances belong to large and dense clusters whereas anomalies belong to sparse or small clusters. A disadvantage of this assumption is that it declares those data instances as anomalous whose size is small and below a threshold value.

Good clustering is characterized by evaluating a measure of similarity within a group and among the groups. Higher the similarity within a group and lower the similarity among the groups, better is the clustering. These techniques require a distance or similarity measure to detect the anomalies which are defined as follows [33]:

- *Distance:* It is defined as the quantitative degree of relationship among data. Some of the most commonly used distance measures are categorized in Table 1.2.

- *Similarity:* It is the qualitative measure of closeness among data. Some of the most frequently used similarity measures are categorized in Table 1.3.

If the distance is small there will be high degree of similarity and on the contrary, large distance indicates low degree of similarity. The analysis can be used in several ways, for example, in continuous attributes, distance is often used. For categorical attributes, simple matching coefficient is a popular choice. In the case of multivariate data instances, any of the two measures can be computed.

Clustering algorithms are categorized into distinct classes but the successful employment of clustering algorithm depends on (i) type of data, which may be uniform, non-uniform, skewed, etc., (ii) dimensionality of the data, i.e., number of attributes in the dataset and (iii) the application for which the clustering algorithm is to be used. Thus, these are broadly classified into two categories: *partitioning based techniques* and *hierarchy based techniques*. These are discussed as follows:

Partitioning based techniques

A partitioning based clustering approach divides a given set of data instances into *k* disjoint clusters where all the instances are iteratively rotated until optimal partitioning is achieved. It is defined as follows: "Given a dataset of n instances, a partitioning technique constructs k non-overlapping partitions of the dataset, where each cluster optimizes the clustering criterion such as-minimization of sum of squared error from mean within each cluster" [34]. These techniques are generally divided into two categories: centroid based and medoid based. In centroid based partitioning approach, each cluster is represented by the mean of the data instances while in medoid based approaches, each cluster is represented by the closest mean value. Basic partitioning techniques include K-means and Partitioning Around Medoids (PAM) in which the data points are relocated in clusters based on some distance measures as given in Table 1.2.

Table 1.2: Distance functions

Distance Measure	Formula	Explanation
Minkowski Distance	$d(x,y) = (\sum_{i=1}^{n} \lvert x_i - y_i \rvert^P)^{\frac{1}{P}}$	○ It is the generalization of Euclidean and Manhattan distances
Manhattan Distance	$d(x,y) = \sum_{i=1}^{n} (x_i - yi)$	○ It is equivalent to the sum of absolute difference ○ It is the special case of Minkowski Distance with P=1
Euclidean Distance	$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$	○ It is the distance between two points ○ It is the natural distance metric in geometric interpretation ○ It is the special case of Minkowski Distance with P=2
Canberra Distance	$d(x,y) = \sum_{i=1}^{n} \frac{\lvert x_i - y_i \rvert}{\lvert x_i \rvert + \lvert y_i \rvert}$	○ It is the weighted version of Manhattan Distance ○ It is often used for data that is scattered around the origin
Cosine Distance	$d(x,y) = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}}$	○ It is used to compute the angular distance between two vectors
Chebyshev Distance	$d(x,y) = \lim_{p \to \infty} (\sum_{i=1}^{n} \lvert x_i - y_i \rvert^P)^{\frac{1}{P}}$	○ It is a special case of the Minkowski distance where p goes to infinity ○ It is also known as Chessboard Distance
Mahalanobis distance	$d(x,y) = \sqrt{(x_i - x_j)^P S^{-1} (x_i - x_j)}$	○ It is the measure of distance with high computational complexity ○ Here S is the covariance matrix
Pearson's Distance	$d(x,y) = 1 - Corr(x,y)$	○ It is used to measure linear relationship between two vectors
Jaccard Distance	$d_J(u,v) = \frac{\lvert u \cup v \rvert - \lvert u \cap v \rvert}{\lvert u \cup v \rvert}$	○ It is used to measure the dissimilarity between two sets ○ Here u and v are two sets

Table 1.3: Similarity functions

Similarity Measure	Formula	Explanation
Jaccard Similarity	$J(u,v) = \frac{\lvert u \cap v \rvert}{u \cup v}$	○ It is the metric that measures similarity between two sets ○ Here u and v are two sets
Cosine Similarity	$d(u,v) = \frac{u.v}{\lvert u \rvert \lvert v \rvert}$	○ It calculates the normalized dot products of the two attributes
Standard Correlation	$S(u,v) = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{(\sum_{i=1}^{n} u_i^2)(\sum_{i=1}^{n} v_i^2)}}$	○ It measures the correlation between vectors ○ Here u and v are vectors
Pearson Correlation	$S(u,v) = \frac{\sum_{i=1}^{n} (u_i - \bar{u})(v_i - \bar{v})}{\sqrt{(\sum_{i=1}^{n} u_i - \bar{u})^2 (\sum_{i=1}^{n} v_i - \bar{v})^2}}$	○ It measures the correlation vector between u and v
Hamming Similarity	$S(u,v) = \frac{\sum_{i=1}^{min\{\lvert u \rvert, \lvert v \rvert\}} IdSim(u_i, v_i)}{max\{\lvert u \rvert, \lvert v \rvert\}}$	○ It is the measure of number of positions for which instances are same ○ IdSim is the identity similarity function where $S * S \to \{0,1\}$ and here S is the universal containing vectors u and v

Hierarchy based techniques

Hierarchical clustering is the method of cluster analysis which produces a sequence of nested clusters without even pre-specifying the number of clusters [35]. It performs clustering by creating a hierarchy of objects which are represented graphically with the help of a dendrogram. A dendrogram captures the process (order) by which the clusters in this clustering process are generated. The hierarchy produced by this clustering technique is more informative as compared to other clustering techniques. This approach explores the nested sequence of patterns on different levels of granularity. A tree in hierarchy based technique consists of a root node (cluster) and several child nodes (children clusters). Root of the tree is representative of the cluster containing all the nodes. A node that contains single object is referred to as singleton cluster. On the basis of strategies, hierarchy based clustering techniques are further categorized into two categories: agglomerative and divisive. Agglomerative approaches (bottom-up) starts with a single data object (singleton cluster) and iteratively merges it with nearby data objects until a single cluster is formed. Divisive approaches (top-down) initiates partitioning by splitting up a dataset containing all the objects (single cluster) into smaller clusters until each cluster contains only one object.

Density based techniques

These techniques measure the density of neighborhood of each data instance. An instance that lies in a dense neighborhood region is declared to be normal while an instance that lies in a neighburhood with low density is considered to be anomalous. It uses the concept of density reachability and density connectivity. The key idea in such techniques is that the core object of a cluster must contain minimum number of data objects (MinPts) in its ε (radius) neighborhood. These are as discussed below:

- *Density Reachability:* If a point 'x' is within ε distance from another point 'y' and 'y'

has sufficient ε neighborhoods then 'x' is said to be density reachable from a point 'y'.

- *Density Connectivity:* Points 'x' and 'y' are said to be density connected if there exists a point 'z' with sufficient neighbors and both the points 'x' and 'y' lie within the ε neighborhoods.

Grid based techniques

The grid based clustering approach uses a fix-up grid partitioning technology to partition the dataset. It quantizes every dimension of object space into a finite number of intervals and then clusters the cells in the quantized space [36]. Here, hyper-rectangles (cells) with certain number of data points are treated as dense. This approach operates as follows: Initially a grid is created, i.e., the data space is partitioned into equal number of non-intersecting cells. Secondly, the density for each interval is computed. Thirdly, sorting of the cells is performed in accordance with their densities. Then, cluster centres are identified. Finally, traversal operations are performed on neighboring cells of quantized space. The major benefit of this technique is its good processing speed which is independent of the number of data objects [37].

Graph based techniques

This technique tries to find out unusual structures within the data represented as graphs. Moreover, partitioning of graphs is done in distinct set of vertices and each of them is tested against one-another to depict the unusual structures. Most of the approaches require prior information for doing the comparisons but to have the data in advance is quite typical. The graph based schemes come to rescue where no prior information is available [25]. This approach provides mechanisms for handling the data which cannot be analyzed easily with traditional mining approaches. This scheme provides powerful representation that cap-

tures long-range correlations among various datasets and helps to monitor and reports abrupt changes in the data in an efficient way. So, if some data is represented in a form of graph then any anomalous activity should be easily identifiable. Based on the behavior of graphs, anomalies are classified into two categories which are as specified below:

- *Static Anomalies* - In a graph structure, a static anomaly can be any node, edge or subgraph that is different from the normal reference patterns existing in the graph. The current behavior of the node and/or edge is analyzed with respect to the remaining graph (network) to spot such anomalies. Further static graphs are of two types:
  - Plain Static Graphs: The graphs that consists of only nodes and edges with no feature associated with them are called plain static graphs. In these types of graphs, the structure of the graph is exploited to detect the patterns and spot the anomalous nodes or edges.
  - Attributed Static Graphs: The graphs where features are associated with the nodes and/or edges are called attributed static graphs. This type of graph exploits structure as well as attributes associated with structures so as to detect the anomalies.
- *Dynamic Anomalies* - The rare or non-conforming nodes and/or edges that occur in time series and data streams of plain or attributed graphs are termed as dynamic anomalies. To detect these types of anomalies, the time stamps or the patterns of top nodes and/or edges that correspond majorly to the change are exploited.

1.4.3 Statistical Techniques

These techniques work on a principle that normal data instances occur in high probability regions whereas anomalies occur in the low probability regions of the dataset. Here, all the

observations that are partially or wholly irrelevant to rest of the data and are not generated by the considered stochastic model, are considered as anomalous [38]. Initially, statistical techniques are applied to fit a statistical model to the given dataset and then a statistical test is applied to determine if an unseen instance belongs to the assumed stochastic model or not. These techniques are broadly classified into two categories [39]:

Parametric Techniques

These techniques assume that the normal data is generated by a parametric distribution with probability density function $f(x, \oslash)$, where $\oslash$ is any parameter which is estimated from the given data and x is an observation. Here, a statistical hypothesis test can also be used where if the statistical test rejects H_0 (null hypothesis), x is considered as anomaly. Examples of such techniques include Gaussian model, regression model, mixture model, etc.

Non-Parametric Techniques

These techniques use non-parametric statistical models which are not defined a priori and are determined from the given data. Examples of such techniques include histogram based, kernel function based techniques, etc. While parametric techniques assume the knowledge of underlying distribution and estimate the parameters from the given data, non-parametric techniques make fewer assumptions as compared to its contrary. It only assumes smoothness of the density function for determination of unseen instances.

1.4.4 Rule-based Techniques

These techniques use the rule learning algorithms, such as-Decision Trees, RIPPER, etc. to learn rules from the training data. Here, each of the formulated rules have an associated confidence value (a proportion of training instances which are correctly classified and the

total number of training instances covered by the rule) [40]. After predicting the behavior of a system, the set of rules that best capture the test instances are identified and the test instances that are not covered by any of the rules are considered as anomalous. These approaches often lead to very low false alarm rates because the rules are specifically designed for the detection of known anomalies. However, such approaches fail to detect novel anomalies so they do not have associated detection rules.

1.4.5 Information Theory based Techniques

In these techniques, several measures, such as-entropy, relative entropy, conditional entropy, and information gain are used to built an appropriate anomaly detection model. These are as defined below [1]:

- *Entropy:* It is used to measure the uncertainty of data items. For a given dataset D where each data instance $d \in D$ belongs to a class C_D, the entropy of D relative to C_D is defined as:

$$E(D) = \sum_{d \in D} P(d) log \frac{1}{P(d)} \tag{1.1}$$

  where $P(d)$ is the probability of d in D.

- *Relative Entropy:* It is defined as the entropy between two probability distributions $p(d)$ and $q(d)$ defined over the same class $(d \in C_D)$. Mathematically, it is given as:

$$E^R(P|Q) = \sum_{d \in C_D} P(d) \frac{p(d)}{q(d)} \tag{1.2}$$

- *Conditional Entropy:* Given that X is the entropy of the probability distribution $(P(d_1|d_2))$, the probability of d_1 given d_2, conditional entropy is defined as the entropy of D where $P(d_1, d_2)$ is the joint probability of x and y. It is given as:

$$E(D|X) = \sum_{d_1,d_2 \in C_D,C_X} P(d_1,d_2) log \frac{1}{P(d_1|d_2)} \tag{1.3}$$

- *Information Gain:* It is defined as a gain of an attribute F in a dataset D. It is given as:

$$IG(D,F) = \sum_{f \in F} \frac{|D_f|}{D} E(D_f) \tag{1.4}$$

where f is the possible set of values of F and D_f is the subset of D.

1.5 Applications of Anomaly Detection

Detecting anomalies in various datasets is an important requirement in Big data, however little work has been done in terms of detecting anomalies in Big data. Anomaly detection techniques should be efficient enough to detect anomalies at an early stage as premature detection can provide high fidelity data leading to deduction in problems related to secure acquisition and processing of Big data [41, 42]. If detection and analysis is done effectively, it can provide security and privacy in various domains such as-telecommunication network, auction network, e-commerce, finance, forensic, healthcare, safety critical systems, vehicle tracking, *etc*. The following segment discusses the significance of anomaly detection in some of the application domains.

- *Internet of Things (IoT):* It is the next revolutionary paradigm that aims to connect every possible device on Earth with Internet using IPv6 technology. The connected devices in IoT are referred to as "Smart devices" having computational and communication capabilities without involving human-to-human or human-to- computer interactions [43]. This in turn has led to the evolution of limitless applications of IoT ranging from Smart Home, E-healthcare, Smart City, Smart Grid, Industrial IoT (IIoT), Inter-

net of Vehicles (IoV), etc [44]. Nevertheless, the information generated from these application domains is essentially heterogeneous and humongous in nature; which requires powerful data processing, storage and analysis to build smart systems and derive the best potential use of IoT [45]. According to recent estimations shared by Gartner [46], nearly 20+ billion devices are projected to be connected by 2020 which in turn would have imminent effect on the 3 V's of Big Data, *i.e.*, *Volume, Velocity & Variety* [47].

This in turn has led to an explosive increase in the data movement across different smart devices with respect to network activities such as-search requests, logs, location data, tweets, e-commerce, data footprint of individuals, *etc*. The amount of data being produced everyday has increased from terabytes to petabytes from different IoT-enabled sensors, actuators and devices. These objects are the largest sources of information flow across the Internet and it is projected that every person would have on an average 6-7 smart devices in the near future [48]. In a nutshell, it can be concluded that, in this age of in-stream data [49] and Internet of things (IoT) [50], there is no limit on the amount of data coming from varied sources. Moreover, the complexity of data and the amount of noise associated with the data is not predefined. Further, the security risks associated with the underlying network traffic have increased manifold in the last decade. Even a minor security risk can trigger consequential challenges ranging from network congestion and network downtime issues to severe data and financial losses. The statistics shared by Vectra Network in a post-intrusion report reveal significant rise in the number of network risks and vulnerabilities [51]. In another report [52], Hewlett Packard Enterprise (HPE) officials reported the emergence of security threats as a major concern for almost every domain; in which the end devices are interconnected with each other. In this context, anomaly detection has surfaced as a recent

trend to discover threats/risks at initial stages. In contrast of its traditional counterparts (i.e., signature-based schemes), anomaly detection measures help in proactive analysis of network streams to identify the unwanted security risks in the underlying network, thereby enhancing the reliability and safety of the systems.

- *Wireless Sensor Network (WSN):* It consists of a large number of spatially distributed sensing nodes that communicate with each other through wireless links to monitor physical and environmental conditions. The sensors deployed in this network are small and low-cost, having some resource constraints like low storage capacity, low computational power and short communication range. These networks have great potential applications in the field of transportation, agriculture, industry automation and process monitoring, military surveillance, environment monitoring, health-care, etc [53].
  The main motive of security in any network is to protect the network from various attacks but the increased variety and complexity of attacks always dominates in influencing the functionality of these networks [54, 55]. Due to the inherent limitations of sensing nodes and resource constraints of WSN, it is especially of major concern. Moreover, the requirement of reliable and robust monitoring of real-time events for diversified applications of WSN makes it more vulnerable, imperative and challenging [56]. Anomalies in WSN could arise due to the following conditions:

  - Since the sensors broadcast the readings taken by them to some main or central location so, adversaries may spoof or alter routing information to interrupt the communication.
  - Sensors used in WSN are operated by batteries so the performance starts to degrade with the consumption of power which can also become the cause of potential anomalies.

- Sensors in WSNs are used to monitor either physical or environmental conditions so the use of imperfect sensing devices may also lead to anomalous events.
- The limited power and resource constraints for WSN makes this network crucial from the anomaly point of view.
- Varying topology and self-organizing nature of WSN also makes it more vulnerable to attacks.

Anomalies in WSNs must be detected as they can affect the network communication, degrade the quality of collected data or can cause failures in the network. The key challenge for anomaly detection in this domain is the presence of large number of sensors and their deployment in the remote areas. Thus, anomaly detection should be computationally efficient to handle such huge volumes of data and able to detect unusual activity in timely manner to minimize the negative effect of such threats. The typical security threats are discussed in Table 1.4.

- *Intrusion Detection:* It refers to an act of monitoring the network traffic data for the detection of malicious activities. The presence of intrusions in a network can destabilize their performance by compromising security in terms of confidentiality, availability or integrity [57]. Thus, detection of these malicious activities is important for a secure and stable system. Key challenges of anomaly detection in this area are discussed below.
  - Huge volumes of data so anomaly detection schemes should be computationally efficient.
  - False alarm rate is very high.
  - Labelled data corresponding to anomalies is not easily available.

- Streaming data requires online analysis of data.
- *Infection:* These types of attacks disrupt the security of a victim network or computer system by tampering or installing evil ransomware executable files in the system. For example: Viruses, Worms, Trojans, etc.
- *Exploding:* These attacks aim to overflow the target system with bugs, for example: Buffer Overflow. Such attacks are often trivial, i.e., they do not require any training to mount.
- *Probe:* These attacks are among the most serious category of attacks which gather the information about targeted computer system through tools. For example: Security Scanning, Sniffing, Port Mapping, etc.
- *Traverse:* This category of attacks attempts to exploit the insufficient security validation for accessing restricted files and directories of targeted system. For example: Brute Force, Doorknob Attacks, Dictionary Attacks, etc.
- *Cheat:* This category of attacks adopts the fake identity either to exploit the vulnerabilities of the network or to make the resources unavailable for its intended users. For example: MAC Spoofing, IP Spoofing, Session Hijacking, DNS Spoofing, etc.
- *Concurrency:* These attacks attempt to interrupt the service of a victim network by sending a number of identical requests to exceed the service supply range of the network. For example: Distributed DoS, Flooding, etc.

A taxonomy of attacks aims to specify relationships among attacks, their classes and subclasses [58]. Intrusions are classified into six major categories which are illustrated below:

Table 1.4: Anomalies in wireless sensor networks

Anomaly	Description	Consequences
DOS Attack	It is an attempt to disrupt the normal functioning of a network or host by making the normal computing environment inaccessible for its intended users	It can effect the functionality or can take control over the network to reduce the overall performance of the network
Sinkhole Attack	It tries to attract network traffic by publicizing its own fake routing updates with the help of compromised node	It can spoof, drop or alter legitimate routing information to effect the communication
Sniffing Attack	It is an attempt to overhear valuable network information from the proximity nodes	It can eavesdrop information to corrupt or crash the network
Remote to Local (R2L)	In this attack, an attacker exploits some vulnerabilities of the network to gain local access as a user on the targeted machine. Most commonly the attacker uses the automated scripting method or brute force method to do the same	It can fracture the trust mechanism of a network or can paralyze the whole network
User to Root (U2R)	In this attack, an attacker abuses the vulnerabilities of the network to gain an illegal access as a super user (administrator)	It can manipulate the resources of a network
Probing	Probing is used when an attacker aims to gather the information about the vulnerabilities of the targeted host or network	Future attacks can be stages or reconnaissance purposes can be served
Wormhole Attack	In this attack, an attacking node captures the packet from one location and transmits them to another part of the network where a low latency out of bound channel replays the packets	It can confuse routing and normal functioning of network
Sybil Attack	It is a type of security threat where a node in a network claims multiple identities	Adversary can mislead the network by imposing false opinions during the communication

- *Social Network:* The Web expansion from Web of Things to Web of Thoughts has made social networking more popular. It is the largest, richest and most dynamic evidence base of human behavior that brings new opportunities for understanding individuals, groups and societies. It is a platform to connect people all around the world. It builds a cyber network amongst users enabling them to share their opinions, ideologies and stuff with each other regardless of their geographic locations. Today these sites have become one of the most popular sites on the Internet and this drastic increase of interest is the reason behind their upcoming on prime targets.

  Recent insights into the world of social media suggest that currently there are around 4 billion Internet users worldwide, out of which more than 3 billion are active social-media users [59]. Due to this proliferation of social networks, multimedia content in the form of Big Data is growing at an unprecedented rate [60]. However, the content-driven and object-oriented nature of social multimedia pose difficulties in gaining meaningful insights from this rich and ever-growing pool of Big data. Moreover, the enormous multimedia content offered on social networks includes sensitive and private information about users and their interactions. Such an abundance of readily available personal information makes it highly vulnerable to threats which often result in information and identity theft. Therefore, interoperability and security are the two biggest challenges in the underlying architecture [61]. Hence, a scalable and pervasive communication paradigm is required for data analytics and management of social multimedia data while maintaining adequate level of security.

  In this direction, anomaly detection plays an important role in this domain. In these networks, interaction among users need to be modeled to analyze the patterns hidden in the networks. Anomalies in social networks usually occur due to the presence of following situations [14].

  - When users hide their identity by publishing false information on a network.
  - Malicious user behavior such as-performing illegal or terrorist activities.
  - When particular user or group of users make sudden changes in their behavior of interaction.

  Challenges such as-lack of labeled datasets, lack of sufficient information, computational complexity and privacy in social networks make the task of anomaly detection very typical in this area of domain.

- *Fraud Detection:* It refers to the detection of criminal activities that usually occur in commercial organizations like stock market, banks, mobile companies, *etc*. It may involve either actual employees and customers or somebody who might pose as a customer. These malicious users consume the resources of an organization in an unauthorized way. Some of the specific applications are discussed below [3]:
  - *Credit Cards:* Here, the task is to detect fraudulent credit card transactions. It requires dynamic detection of frauds as soon as the unauthorized usage happens.
  - *Mobile Phone:* In this domain, the task is to detect the account that appears to be misused in terms of calling behaviour such as-high volume of calls or calls made to unlikely destinations.
  - *Insider Trading:* This application of anomaly detection is found in stock markets where the task is to prevent people/organizations from making illegal profits by using inside information.
  - *Insurance Claims:* The task of anomaly detection mechanisms in this application domain is to prevent the unauthorized and illegal claims that have been processed by the manipulation of claim processing system. Detection of such frauds is important to prevent financial losses of such organizations.

- *Financial Sector:* Anomalies in financial sector refer to the fraudulent activities that occur in organizations such as-banks, stock market, cell phone companies, insurance agencies, credit card organizations, etc [16]. Anomalies in such organizations occur when malicious users consume the resources of an organization in an unauthorized way. Since, financial losses can occur in this domain so, the organizations are interested to detect such anomalies immediately.

- *Image Processing:* Anomaly detection in image processing deals with the identification of changes in an image over time or the regions that look abnormal in a static image [62]. In this domain, anomalies are generally caused due to motion of foreign object or instrumentation errors. The major challenge in this domain is the huge data size with spatial as well as temporal characteristics.

- *Medical and Public Health Data:* It refers to the detection of abnormalities in patient records. In this domain, outlier detection is a very critical problem and requires high attention as well as a quick solution [63]. The reason of the abnormality can be anyone of the following:

  - Instrumentation error
  - Abnormal condition of patients
  - Disease outbreaks
  - Recording errors

  Since, the labeled data for healthy patients is available, most of the existing anomaly detection techniques adopt semi-supervised approaches to detect anomalous records. The major challenge in this application domain is the cost of classifying an anomaly as normal is very high.

- *Industrial Damage Protection:* Damage in industries usually occurs due to the continuous usage, wear and tear of normal components, or other unforeseen circumstances. Such damages are very crucial and are to be detected early to prevent further losses [64]. Here, anomaly detection techniques use data generated from multiple sensors to detect the industrial damages. Such damages can be further classified into two categories:
  - *Defects in Mechanical Components:* In this domain, anomaly detection techniques monitor the mechanical components such as-motors, engines, etc., to detect any damage and prevent further losses.
  - *Defects in Physical Structures:* Here, anomaly detection techniques deal with the detection of defects that are caused due to strains in airframes, cracks in beams, etc.
- *Text Data Anomaly Detection:* Detection of prime topics and news articles comes under the purview of text data anomaly detection [65]. Data in this domain is quite sparse and highly dimensional. In general, anomalies in this case are caused due to the occurrence of a new event. The major challenge in this domain is the handling of data which is high dimensional and sparse with a temporal aspect.

1.6 Thesis Organization

The organization of the thesis is as follows:

Chapter 1: Introduction: This chapter provides an overview of anomaly detection domain along with its related applications.

Chapter 2: Literature Review: This chapter provides a comprehensive literature review in the area of anomaly detection and highlights the various schemes proposed. A comparative

study of the existing anomaly detection approaches has also been provided. Additionally, the chapter concludes with objectives of the thesis.

Chapter 3: Ensemble based Anomaly Detection Technique: In this chapter, hybrid anomaly detection technique for networked traffic have been proposed which is referred as *Ensemble based Anomaly Detection Technique (En-ADT)*. It uses a combination of fuzzy K-means (FKM) clustering algorithm, extended Kalman filter (EKF), and support vector machines (SVM) to detect the anomalies.

Chapter 4: Fuzzified Cuckoo Based Clustering Technique: In this chapter, another anomaly detection technique, i.e., *Fuzzified Cuckoo based Clustering Technique (F-CBCT)*, is proposed which operates in two phases: training and detection.

Chapter 5: Experiments and Implementation Details: In this chapter, the experimental evaluation of the proposed anomaly detection techniques is reported. All the scripting for evaluation purpose has been done in MATLAB and Python.

Chapter 6: Conclusion and Future Scope: Thesis concludes with this chapter by highlighting the contributions made by the proposed research work. It also provides an insight into the future directions for working in this area.

Chapter 2

Literature Review

The relevance of anomaly detection and its analysis has attracted researchers to contribute in this field. The success rate of these techniques depend on their high detection accuracies along with low false positive rates. However, the existence of irrelevant and redundant features in the data set may degrade their performance. Motivated by these facts, most of the anomaly detection schemes utilize different approaches to improve their detection capability [66]. These approaches range from the dimensionality reduction to robust optimization, machine learning and deep learning approaches. To have a broader view of this domain, some of them are discussed in this chapter.

2.1 Dimensionality Reduction

From the past few years, the dimensionality of the data is increasing at a rapid pace which possess serious challenges for the existing learning methods. Data with large number of features leads to problems like over-fitting and under-fitting which results in degradation of the model's overall performance. To avoid such problems and improve learning performance in terms of accuracy, computational complexity, storage, and model interpretability;

dimensionality reduction techniques have been studied. These techniques are divided into two categories namely-feature extraction and feature selection. In the prior approach, dimensionality of the feature set is reduced by projecting features into a new space. Examples of such approaches include Linear Discriminant Analysis (LDA), Principle Component Analysis (PCA), Singular Value Decomposition (SVD), etc [67]. On the contrary, feature selection approaches select a small subset of features by minimizing redundancy and maximizing relevance to the target. For example: Relief, Information Gain, Fisher Score, Chi Squares, etc.

On the basis of outputs, feature selection techniques are classified into three categories: (i) subset selection, which returns subset of selected features, (ii) feature weighting, which returns weight corresponding to each feature, and (iii) ranking, which returns the ranked subset of features. All these feature selection techniques perform in four phases, i.e., subset creation, subset assessment, stopping criterion, and outcome validation. Initially, a candidate subset is chosen in subset creation phase. In second phase, the generated subset from the first phase is evaluated using certain evaluation criteria. All the candidate solutions are recursively iterated until the stopping criteria is met. Then, the solution that best fits the evaluation criteria is chosen which is then cross-validated using the validation set or domain knowledge in the final phase [68]. Apart from the above classification, feature selection techniques are also classified into the following four categories namely-filter models, wrapper models, hybrid models and embedded models (Table 2.1).

Table 2.1: Comparison of different feature selection techniques

	Filter Models	**Wrapper Models**	**Embedded Models**
Description	This approach evaluates the subset of features using statistical criteria, i.e., without involving a learning algorithm	The wrapper models uses a learning algorithm such as-SVM, KNN, etc., for subset evaluation	In this approach, the model interacts with learning algorithm to capture feature dependencies.
Advantages	Less computationally expensive and performs well for large-scale datasets	Selects features that has the most discriminative power thus they are superior in terms of classification accuracy	They achieves model fitting and feature selection simultaneously
Examples	Relief, CFS, Fisher score, and FCBF	FSSEM, l1SVM or any combination of search strategy and classifier	BlogReg, C4.5, and SBMLR

To bridge the gap between filter and wrapper models, hybrid models have been proposed which possess high efficiency (as of filter models) and accuracy (as of wrapper models). These models incorporate the statistical criteria to select features with given cardinality and highest classification accuracy. Dash and Liu [69] demonstrated the importance of feature selection in clustering. It has been shown that clustering algorithms are highly sensitive to the dimensionality of data. A relevant subset of features is thereby required for improving the performance of clustering algorithms. Thus, the authors employed RANK algorithm to order the features according to their importance in the cluster. Experimental evaluation of their work validated the effectiveness of the designed feature selection algorithm. Similarly, Fong *et al.* [70] proposed Accelerated PSO based feature selection algorithm for mining streams in big data. The related performance evaluation was done by collecting big data with high dimensionality. Duan *et al.* [71] also propounded clustering based technique which was facilitated by hierarchical feature selection scheme. The authors employed partial distance strategy to reduce the dimensionality of the dataset. The experimental analysis of the proposed technique along with various other state-of-the-art algorithms proved the effectiveness of the same.

It is evident from the above facts that a large number of such techniques are available in the literature. However, without knowing the relevance of features, it is quite complex to determine the effectiveness of such methods [72]. Therefore, new feature selection methods are being developed continuously to eliminate irrelevant and redundant features from the high dimensional datasets using different strategies: (i) using combination of several feature selection methods, (ii) ensemble methods, (iii) restructuring existing schemes, etc [73]. Table 2.2 provides the comparative analysis of some recently proposed feature selection schemes.

Table 2.2: Comparison of some recently proposed feature selection schemes

Approach	Technique(s)	Dataset Employed	Comparison	Contributions
Panday *et al.* [74]	Cluster-dependent feature-weighting mechanism	Synthetic datasets containing Gaussian clusters and Real-world data sets from UCI machine learning repository	Feature selection using feature similarity (FSFS) and Multi-cluster feature selection (MCFS)	An unsupervised feature selection algorithms has been proposed where features with relatively low weights are removed to improve running time and data visualisation
Yu *et al.* [75]	Markov blankets and Group alpha-investing (SGAI) algorithm	Datasets of UCI repository, biomedical, NIPS 2003 and microarray	Selection via representative set (SRS) algorithm and REpresentative Sets (RESs)	A new concept of representative sets had been used to perform Markov blanket feature selection in a Bayesian network
Xu *et al.* [76]	Semi-supervised learning method and Max-relevance and min-redundancy criterion	Binary data and multi-category data from UCI Repository and Face dataset	Fisher score, mRMR, Laplacian score, locality sensitive, and sSelect	A semi-supervised learning based feature selection scheme had been proposed to enhance the balance between classification performance and computational cost proposed
Zhou *et al.* [77]	K nearest neighbors and Neighborhood Rough Set theory	Seven high-dimensional and class imbalanced data sets, i.e., DNA microarray datasets	ReliefF, Fisher Score, S2N, Pearson Correlation Coefficient (PCC) and MI	A new feature selection algorithm had for high dimensional and class imbalanced data been proposed that works in an online manner
Das *et al.* [78]	Feature Association Map (FAM), Graph-theoretic principles and maximal independent set	High-dimensional publicly available datasets from UCI repository	Several benchmark feature selection algorithms like CFS, mRMR, DSCA, Laplacian, PFA and DSUB algorithms	A graph-based anomaly detection technique had been propounded that works for both supervised and unsupervised classification
Hou *et al.* [79]	Common graph Laplacian, Sparse $\ell_{2,p}$-norm constraint, and K-means(KM) clustering	Datasets like SensIT Vehicle, Caltech-101, NUS-WIDE-OBJ, MIRFLICKR, Animal, and MSRC-v1	Other feature selection approaches such as-LapScor, SPEC, MRSF, AMFS, and AUMFS	An efficient feature selection algorithm named as Multi-view Unsupervised Feature Selection with Adaptive Similarity and View Weight (ASVW) had been proposed to address the non-smooth minimization problem
Liu *et al.* [80]	Online group selection and online inter-group selection	Series of multi-label benchmark data sets from Mulan Library	Techniques such as-MLNB, MDDMspc, MDDMproj, NFNMI, PMU, and RF-ML	A multi-label feature selection scheme has been proposed that handles streaming features along by considering intrinsic group structures

Approach	Technique(s)	Dataset Employed	Comparison	Contributions
Tang *et al.* [81]	Design of experiments (DOE) and Sum of interaction information	Four UCI machine learning datasets, i.e., Spambase, Musk2, Madelon and Gisette	MI-based methods like mRMR and JMIM and non-MI based method ReliefF	A two-stage feature selection approach had been propounded to identify significant interactions among features
Hu *et al.* [82]	Candidate feature relevancy and Selected feature relevancy.	12 real-world data sets from different fields such as-face image, biology, and handwritten image and Two-tailed t-test	Benchmark feature selection techniques such as-DRGS, JMIM, mRMR, DISR and IG	A feature selection method named Dynamic Relevance and Joint Mutual Information Maximization (DRJMIM) had been proposed to deal with two significant problems: features relevancy and misinterpreted features
Izetta*et al.* [83]	Support Vector MachinesRecursive Feature Elimination (SVM-RFE) algorithm	Spectrometry measurements of food products, UCI repository and gene expression datasets from human tissues	OneVsOne (OVO) and OneVsAll (OVA) strategies along with 3 versions of MSVM-RFE	A new feature selection method based on binary decomposition had been proposed to yield improved selections
Zhu *et al.* [84]	$\ell_{2,p}$-norm regularization item and Label dependency to exploit label correlations	Benchmark datasets like Artificial, Bookmarks, Birds, Social, Reference, and Yeast	State-of-the-art feature selection algorithms like MDDM, PMU, SFUS and MDMR	An iterative reweighted least squares (IRLS) algorithm has been proposed to select the most discriminative features and remove noisy ones
Lin *et al.* [85]	Multilabel learning and Label Correlation	12 benchmark datasets from Mulan Library like CAL500, Birds, Emotions, Yeast, etc.	State-of-the-art feature selection schemes like MDDMspc, MDDMproj, RF-ML, PMU	An efficient algorithm for multilabel feature selection had been designed for analyzing relevance and redundancy fuzzy mutual information
Liu *et al.* [86]	Dependency margin approach and Greedy search algorithms	Synthetic datasets: CorrAL, CorrAL-46 and CorrAL-47, 15 UCI benchmark datasets and face recognition	ReliefF, dependency, consistency, information gain, FCBF, CFS-FS, and IRelief,	A subset selection algorithm for feature selection has been proposed which considers the relevance of both selected and remaining features for making the predictions
Qi *et al.* [87]	Matrix factorization based feature selection methods and Updating algorithm	5 face image databases, 3 biological databases and 1 object image database	Unsupervised methods like LS, MCFS, SPEC, UDFS, RSR and MFFS	A Regularized Matrix Factorization Feature Selection (RMFFS) had been propounded that takes correlation among features into consideration for making the selection
Prasad *et al.* [88]	Dual coordinate descent algorithm, SVMs and Sparse representation	7 datasets: Leukemia, RAOA, RAC, MNIST, EAL-SIM, Webspam and Kddb	State-of-the-art feature selection techniques like FCBF, QPFS, FGM, and GDM	A max-margin framework for feature selection had been proposed which describes the boundary for the region where the set of the features exists.

Approach	Technique(s)	Dataset Employed	Comparison	Contributions
Zhang *et al.* [89]	Firefly Algorithm (FA), Chaotic maps and Simulated Annealing (SA)	29 classification and 11 regression high dimensional benchmark data sets	Methods including 11 classical search methods and 10 advanced variants	A variant of FA algorithm has been propounded to identify optimal feature subsets in classification and regression models
Viegas *et al.* [90]	Genetic Programming approach and Gaussian Naive Bayes Classifier	K8 cancer-rescue mutants, 20 Newsgroups, 4 Universities, Reuters and ACL-BIN	By considering an entire spectrum of data skewness scenarios	A new feature selection approach had been designed which combines the most discriminative feature sets of highly dimensional skewed dataset
Kundu *et al.* [91]	Distance correlation and Pairwise similarity	Colon, Leukemia, Prostate, DLBCL, MLL, NSL-KDD, Isolet, COIL20 and Multiple Features (MFs) dataset	K-nearest neighbors, Naive Bayes, SVM, fsfs, mRMR, ranked t-test, and independence criterion HSIC	A new feature selection algorithm had been developed to select a subset of features that involves minimum redundancy and reduced parameter tuning
Zhu *et al.* [92]	Joint $\ell_{2,1}$-norm co-regularization and Feature selection matrices	Face image, microarray, digit image, multi-view image, and spoken letter dataset	Laplacian Score, MCFS, UDFS, SPEC, RUFS, and EUFS	A novel co-regularized unsupervised feature selection (CUFS) algorithm had been designed to ensure that the selected features preserves the data distribution
Mafarja *et al.* [93]	Crossover and mutation operators, Whale Optimization Algorithm (WOA) and Tournament and Roulette Wheel selection	Eighteen UCI repository benchmark datasets such as Breastcancer, Lymphography, Vote, Zoo, etc.	Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Ant Lion Optimizer (ALO), and Five standard filter feature selection methods	A new wrapper feature selection approach based on WOA had been proposed to reduce the number of features and improve the classification accuracy
Tang *et al.* [94]	Squared ℓ_2-norm and Conventional graph Laplacian	Speech signal, biomedical MicroArray, digit and face image	State-of-the-art unsupervised feature selection methods	A robust feature selection model based on feature self-representation and graph regularization had been proposed

2.2 Optimization Schemes

In recent years, several optimization schemes such as-A* search algorithm, linear programming, evolutionary algorithm, random trees, genetic algorithm (GA), particle swarm optimization (PSO), etc., have been used for detection of anomalies in large scale datasets [95]. Ghanem *et al.* [96] introduced a novel anomaly detection technique using meta-heuristic method and GA to escape from the problems of local optima and robust search. The proposed technique used a selection-based detector generation methodology to detect anomalies in large scale datasets. The work done by the authors to generate the detectors with such a high accuracy is quite convincing. However, in order to increase the adaptability and flexibility of the proposed model, the values of the parameters used by the multi-start searching method and GA should be decided dynamically instead of setting them apriori. Aburomman and Reaz [97] combined Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) along with PSO to propose a hybrid intrusion detection technique. In the proposed approach, KNN and SVM experts were trained using binary classifiers and these classifiers were then combined using PSO and Weighted Majority Algorithm (WMA). Ensembles are essentially created to combine the expert opinions to reach the final classification decision. Similarly, Shahreza *et al.* [98] combined Self-Organized Maps (SOM) and PSO for the purpose of anomaly detection. In order to validate the proposed unsupervised approach, a case study was also performed on forest fire detection.

Karami and Guerrero-Zapata [99] developed a hybrid technique for proactive prediction of DoS attacks in named data networking environment. In this approach, a combination of multi-objective optimization and PSO was used to resolve the hybrid learning problem of Radical Basis Function (RBF) Network. This RBF-based network classifier was then utilized to improve the prediction accuracy of DoS attacks. Another work by Karami and Guerrero-Zapata [100] combined PSO and K-means algorithms for anomaly detection in

content-centric networks. The proposed detection system operates in two phases: training and detection. The training phase employed two simultaneous cost functions, i.e., Davies-Bouldin Index (DBI) and Mean Square Error (MSE), whereas detection phase utilized two distance-based approaches, i.e., classification and outlier. In training phase, the hybridization of PSO and K-means determined the optimal number of clusters while the fuzzy approach employed in detection phase was used to detect the anomalies. Further, the performance was evaluated in terms of sensitivity, specificity and accuracy.

Li *et al.* [62] proposed a segmentation algorithm using hybridization of dynamic PSO and k-means clustering. The hybrid of two algorithms proposed in their work improved the global search capability of k-means clustering. Further, Inkaya *et al.* [101] addressed several challenges of clustering such as-solution evaluation, neighborhood construction, dataset reduction, *etc*. To solve these problems the authors proposed an Ant Colony Optimization (ACO) based clustering methodology by using two objective functions namely-adjusted compactness (to extract the local characteristics of datasets) and relative separation (for scalability). The results showed that the proposed method was capable of extracting the arbitrary shaped clusters with varying densities.

Apart from this, optimization based pattern recognition has also emerged as an important field in data mining. Several researchers have cascaded optimization techniques along with clustering algorithms to enhance the efficiency and accuracy of clustering algorithms. Alam *et al.* [102] reviewed PSO based clustering techniques. The comparison results indicated that PSO based hybrid clustering algorithms outperform other traditional clustering algorithms. It was also shown that this hybrid approach has the tendency to overcome local optima problems that generally occur in clustering. The automation, generalization and areas for applicability of such techniques are quite unexplored. Table 2.3 presents the comparative overview of the most widely used optimization schemes.

Table 2.3: Comparison of the existing optimization schemes

Approach	Inspiration	Behaviour	Optimization Steps
Ant Lion Optimizer (ALO) [103]	Models the interaction of antlions in nature	Digging capability of antlions	Random walk of ants, building traps, entrapment of ants in traps, catching preys, and re-building traps
Dragonfly Algorithm (DA) [104]	Static and dynamic swarming behaviours	Finding the location of Prey	Separation, alignment, cohesion, attraction to food source, distraction from enemies
Grey Wolf Optimizer (GWO) [105]	Hunting strategy of grey wolves	Compute shrunken circle for position	Encircling, hunting and attacking the prey
Bat Intelligence (BI) [106]	Prey hunting behaviours of bats	Bats can easily identify their surroundings and locate preys	Signal generation by constant absolute target direction (CATD) technique and Prey hunting process
Artificial Bee Colony (ABC) [107]	Intelligent foraging behaviour of honey bee swarm	Discover high nectar food sources	Explore, exploitation, recruitment, abandonment
Cat Swarm Optimization (CSO) [108]	Fascinating social behaviours of cats	Pouncing capability of cats	Seeking and tracing modes
Cuckoo Search Algorithm (CSA) [109]	Breeding behaviour of parasitic cuckoo species	Reduce the abandon probability of their eggs and increasing their reproductively	Local random walk with permutation and the global explorative random walk (Levy flight)
Firefly Algorithm (FA) [110]	Social (flashing) behaviour of fireflies	Global communication among the swarming particles	Variation of light intensity and Movement toward attractive firefly
Fish School Search (FSS) [111]	Social behaviour of biologic fish	Collective behaviour that increases mutual survivability and achieve synergy	Feeding operator, Swimming operator and Breeding operator
Particle Swarm Optimization (PSO) [112]	Inspired from the social behavior and dynamic movements of insects, birds and fish	Movement towards a promising area to get the global optimum	Intensification and Diversification with Separation, Alignment and Cohesion
Harmony Search (HS) [113]	Melodic themes	Interactions between harmonies	Preparation of harmony memory (HM), Improvisation of new harmony and Update of HM
Genetic Algorithm (GA) [114]	Inspired by biological evolution process	Natural selection and genetic inheritance	Mutation, crossover and selection

2.3 Machine Learning Approaches

Anomaly detection techniques have widespread applications in different domains such as-data classification, clustering, and prediction. Amongst these, clustering and classification techniques are the ones that are widely used [115]. The reason behind the wide-scale applicability of clustering techniques is that these techniques provide deep insight into the distribution of data without any prior knowledge of data labels; whereas classification techniques solve the purpose where labeled data is available. For clustering, distance based measures such as k-means and k-medoids; and density based measures such as-Density based Spatial Clustering of Applications with Noise (DBSCAN) and Ordering Points to Identify the Clustering Structure (OPTICS) are generally used [116]. On the contrary, SVM, Naive Bayes Classifier and Decision Tree are generally employed for classification [117]. Table 2.4 provides the comparison of some popular machine learning techniques.

The K-means technique is the simplest and the most intuitive partitioning technique but it suffers from local convergence and initialization of clusters like problems. Inspired from K-means technique, a number of partitioning algorithms have been developed by researchers like K-medoid, fuzzy C-means (FCM), etc. K-medoids algorithm divides a dataset into number of partitions by minimizing the absolute distance between the points and the centroid rather than minimizing the square distance. K-medoid is found to be more robust than K-means algorithm, but it is more expensive as it involves pairwise distance computation. Both K-means and K-medoids tend to give inaccurate results during cluster overlapping scenarios. In such cases, better accuracy can be achieved using FCM, in which each element is assigned to the cluster having highest membership grade. Thus, the nature of the patterns associated with the datasets should be evaluated carefully before selecting the appropriate clustering technique [32].

Table 2.4: Overview of popular machine learning techniques

Approach	Description	When to use?	Applications	Advantages
Nave Bayes Classifier [118]	Works on Bayes Theorem of Probability to build learning models	Large training set with several attributes	Sentiment Analysis, Email Spam Filtering, Document Categorization, etc.	Performs well for categorical variables and multi class predictions.
K Means Clustering [119]	A non-deterministic and iterative method popularly used for cluster analysis	Data without defined categories or groups	Search engines like Yahoo and Google to cluster web pages	Produce dense clusters and computes faster compared to hierarchical approach
Support Vector Machine [120]	Computes a hyperplane which maximizes the separation between classes	Non-linear classification and Huge number of features	Stock market forecasting for managing investments and making decisions	Best classification performance in terms of accuracy, Avoids over-fitting, etc.
Apriori Algorithm [121]	Unsupervised machine learning algorithm which generates association rules for partitioning	Large set of properties about dataset is given	Detecting Adverse Drug Reactions, Auto-Complete Applications, Market Basket Analysis, etc.	Easy to implement and parallelize
Linear Regression [122]	To predict the relationship between two variables	Independent variables are of numeric type	Risk Assessment, Estimating Sales, etc.	Requires minimal tuning, Fast execution and most interpretable
Logistic Regression [38]	Applies a logistic function to a linear combination of features to predict the outcome	To model the probabilities of the response variable as well as to predict the probabilities of categorical variable	Bakeries, Snowfall prediction, etc.	Robust, Less complex, Handle non-linear effects and controls confounding and tests interaction
Random Forests [123]	Uses a bagging approach to create a bunch of decision trees, the outputs of which are combined to make the predictions	Missing values, Presence of Numerical, binary and categorical features	In Banks to predict loan risks, Automobile industry to predict the failure of a mechanical part, healthcare industry to predict disease risk, regression tasks, etc.	Less Over-fitting, Fast execution, More robust to noise, Requires less tuning, Runs efficiently on large databases, etc.
Decision Trees [124]	Use of branching methodology to extract all possible decision values	If the training data contains errors, Where instances are represented by attribute value pairs, target function has discrete output values, etc.	In finance for option pricing, Remote sensing,banks to classify loan applicants, etc.	Handle both categorical and numerical variables, useful in data exploration, save data preparation time, etc.

Park & Jun [125] explored both k-means and k-medoid partitioning techniques to propose a novel algorithm for clustering. Firstly, it was analyzed that k-means algorithm may fail to converge if used with distances that are inconsistent with mean. Secondly, it was noticed that despite of its efficiency in terms of computational time, k-means clustering is less favorable due to its sensitiveness to noise. On the contrary, k-medoid algorithm exploited medians instead of centroids thus making it less sensitive to noise. After analyzing both the techniques, the authors concluded that k-medoid algorithm is more robust as compared to k-means partitioning technique. Zadegan *et al.* [126] proposed Ranked k-medoids algorithm for clustering of large datasets. The proposed algorithm was capable of detecting all Gaussian shaped clusters by computing the similarity between pairs of objects only once. The experimental evaluation on large datasets clearly indicated that the proposed algorithm did not get trap in local optima and provided efficient results in terms of speed and accuracy. Similar work by Lai & Fu [127] utilized variance measure to propose variance enhanced k-medoid clustering algorithm. Further, a web based clustering system was employed to validate the clustering results of the proposed model.

Apart from this, DBSCAN has also been widely used in data modeling due to its ability of detecting clusters of different sizes and shapes. However, it suffers from several problems like evaluation of parameters, nearest neighbor search, border point detection for adjacent clusters, *etc.* To overcome these problems, several alternatives have been proposed. Kumar & Reddy [128] proposed a graph based technique to accelerate the nearest neighbor operation of DBSCAN. Since the stability of Groups method over index based structures is quite high, it was employed to compute the nearest neighbors. The proposed technique also pruned the noise points at the initial stage to eliminate the unnecessary distance computations. The clustering results obtained were quite impressive but the efficiency of the proposed technique was not verified on larger datasets. On the other hand, Lv *et al.* [129] addressed the prob-

lem of nearest neighbor search in DBSCAN by applying randomness based p-stable locality sensitive hashing technique. Moreover, the concept of influence space was proposed to evaluate the parameters of DBSCAN. Tran *et al.* [130] proposed a revised DBSCAN algorithm to improve the clustering performance of the traditional DBSCAN algorithm by solving the problem of border points detection of adjacent clusters. This was done by altering the core-density-reachable process of the algorithm. Several simulated datasets were employed to compute the performance of the proposed algorithm.

Kwedlo *et al.* [131] presented a clustering method which used a combination of differential evolution and k-means. Here, k-means algorithm was utilized to tune the solutions obtained by mutation and crossover operators of differential evolution. The results showed that the proposed algorithm obtained solutions with lower Sum of Squared Errors (SSE) when compared with its counterparts.

Warrender *et al.* [132] compared several anomaly detection techniques and it was found that Hidden Markov Model (HMM) outperformed all other anomaly detection techniques with regard to average anomaly detection rate and false positive rate. Forrest *et al.* [133] also developed intrusion detection system for modeling normal sequences using contiguous sequences and look ahead pairs. Ghosh *et al.* [134] proposed Artificial Neural Network (ANN) based anomaly detection technique. This technique exploited several variations in Neural Network for classifying system-call sequences. Experiments conducted on 1998 and 1999 DARPA datasets revealed the improvement in the performance of the detection rate. Liao *et al.* [135] presented analogy between a text document and the sequences of system calls. In the proposed technique KNN classifier along with Robust Support Vector Machine (RSVM) was used to classify new programs as normal or intrusive. Dromard *et al.* [136] proposed an unsupervised anomaly detection technique that uses discrete time-sliding window and incremental grid clustering to detect the network anomalies. Shin *et al.* [137] explored

interdependent structure of failures for an early detection of anomalies in an interconnected power grid and communication network. Yuan *et al.* [138] propounded a graph based method for anomaly detection in real-world hyper-spectral images.

The presence of anomalies in operational networks often leads to network disruptions. Thus, several techniques have been proposed to analyze the abnormal changes in network traffic. Jiang *et al.* [139] analyzed the network wide traffic for detecting the abnormalities in high dimensional and large scale datasets. In [63], the problem of anomaly detection was analyzed for large scale communication networks. In this work, the wavelet transform method in integration with empirical mode decomposition approach was employed to diagnose the anomalies present in multimedia medical devices. The wavelet transform was utilized to capture the multi-scale characteristics of anomalies in high speed network traffic [140]. In [141], the network traffic estimation was done by employing time frequency analysis mechanism. Similarly, another work by Jiang *et al.* [142] approximated the network traffic flow in large scale networks. In this work, the Generalized Regression Neural Network (GRNN) technique was utilized to propose a traffic matrix estimation method. The simulation results of the proposed technique on Abilene network real data showed the robustness of the proposed model.

Tian *et al.* [143] developed a technique by utilizing a functional approach of fastly converging radial basis neural network. The proposed approach detected the intrusion characteristics rapidly and effectively which were then utilized to detect the network anomalies. Yong *et al.* [144] used SVM for developing an intrusion detection system. In this approach, learning vector samples were selected to effectively reduce the training samples. This summarized training set was supplied as an input to the intrusion detection system to reduce computational cost overhead. Sharma *et al.* [145] also proposed an anomaly detection algorithm using K-means clustering and Naive Bayes concepts. In the proposed ensemble

approach, a higher detection rate was achieved as compared to single clustering or classification algorithm. However, the proposed technique had huge false positive rate.

K-medoid clustering and Naive Bayes techniques were combined by Chitrakar *et al.* [146] for anomaly detection. The comparison of the proposed technique was done independently with k-Means and Naive Bayes classification algorithms. For validating the proposed technique, KDD dataset was utilized and 2% of improvement was seen in terms of accuracy. Elbasiony *et al.* [147] proposed a data-mining based hybrid technique for network intrusion detection where importance of features was computed by random forest algorithm. In this approach, the patterns of intrusions were built to classify the captured network connections. Based on this, Sotiris *et al.* [148] investigated the use of one-class SVM to detect the system anomalies. Since user-defined threshold was needed as input to one-class SVM, the classification accuracy of algorithm was important to the training data.

To address the problem of data analytics, Forestiero [149] proposed a self organizing multi-agent algorithm where data items were associated with bio-inspired agents. These agents were disseminated on a virtual space where they formed groups on the basis of similarity of their associated objects. Finally, the objects associated with isolated agents were considered as anomalous. Moshtaghi *et al.* [150] described an approach for anomaly detection in data streams where the ability of fuzzy rule-base methods was used to learn the incoming samples. In an another work, Xu *et al.* [151] proposed a dynamic extreme learning machine for classification of patterns present in continuous data stream. It was observed that their proposed technique was faster than decision tree algorithm but its accuracy was slightly less. With reference to this problem, Sindhu *et al.* [152] proposed an intrusion detection system for multi-class categorization using decision tree. Their proposed technique showcased good performance for datasets with multiple classes which confirmed that decision tree is a promising strategy for intrusion detection.

To have a broader view of this domain, comparison of such techniques along with their application areas and types are provided in Table 2.5. Further, some existing machine learning based anomaly detection techniques are discussed in Table 2.6 on the basis of distinct characteristics.

2.4 Deep Learning Approaches

In recent years, Cybersecurity researchers have designed a number of anomaly detection models to protect the network against attacks perpetrated by malicious users against different applications such as-remote video-on demand, video conferencing, real-time content delivery, online gaming, etc. In this direction, another trend that has grabbed the attention of researchers for network anomaly detection is deep-learning (DL). It is a widely-accepted machine learning approach that plays a significant role in detecting the most relevant features from huge datasets using back propagation. Ever since its inception, different architectures have been proposed in the literature such as-Deep Belief Network (DBN), Deep Neural Networks (DNN), Restricted Boltzmann Machine (RBM), Stacked AutoEncoders, Recurrent Neural Networks (RNN), and Convolutional Neural Networks (CNN) [153].

For example, Chu *et al.* [154] devised an abnormal event detection scheme for videos where they used 3-dimensional CNN to extract the spatio-temporal information of the inputs. Xu *et al.* [155] proposed a method for detection of unusual events in videos via Stacked Sparse Coding and intra-frame classification strategies based on the probabilistic outputs of SVM. Sabokrou *et al.* [156] used a fully-connected CNN to detect anomalies in crowded activities. In [157], Shone *et al.* devised an deep and shallow learning model for detection network intrusions by analysing the network streams. More specifically, in this approach Non-symmetric Deep Auto-Encoder was coupled with the fast and accurate Random Forest

to detect network anomalies. Kim *et al.* [158] devised a DNN based technique for detecting network attacks. In the proposed technique, the data was initially pre-processed using different techniques such as-data transformation and normalization. The pre-processed data was then served as an input to the designed DNN for learning the model and verifying it on KDD'99 dataset. In [159], Tang *et al.* employed DL model to detecting flow-based anomalies in SDN setup. In order to attain this objective, a DNN model was built and trained on NSL KDD dataset.

In addition to this, Kanarachos *et al.* proposed a novel DNN architecture for detecting anomalies in time series data. Their proposed scheme combines wavelets, neural networks, and Hilbert transform for learning of pattern interdependencies and detecting anomalies in the Seismic Electrical Signal [160]. AL-Hawawreh *et al.* [161] devised a DL approach for malicious activity detection in industrial IoT. More specifically, their proposed scheme used deep auto-encoder and deep feed-forward neural network architecture for detecting existing and new attacks which can learn using information collected from TCP/IP packets. Feng *et al.* [162] presented a deep Gaussian mixture model (GMM) based intrusion detection scheme for modeling abnormal events in videos.

2.5 Comparative Analysis

Some of the most comprehensive contributions in this field are provided in this section. Several techniques have been surveyed and it was found that data instances are often interdependent in nature and they exhibit long range co-relations whereas anomaly detection problems are often relational in nature. The unbalanced and relational nature of data, asymmetric errors, novel and diverse nature of anomalies, noise of labels, cost of anomaly detection, real time discontinuity detection and size of the search space are some of the challenges.

This section provides a comprehensive review of some recently proposed anomaly detection schemes on the basis of distinctive characteristics. Table 2.7 provides a comprehensive review of some recently proposed anomaly detection schemes based on the distinctive characteristics.

2.6 Various Sources of Datasets

This section illustrates the sources of the datasets that are popularly employed for evaluation of the anomaly detection model. The related description is summarized in Table 2.8.

Table 2.5: Comparison of anomaly detection techniques on the basis of data labels

	Supervised	**Unsupervised**	**Semi-supervised**
Description	A learning approach that analyzes the training data and produces an inferred function, which can be used for mapping new examples.	It is a task of inferring a function to compute the hidden structures or relationships between different inputs.	It is a sub-class of supervised learning technique that makes use of labelled data (small amount) in conjunction with unlabelled data (large amount) for training.
Application Areas	Pattern recognition, Speech recognition, Spam detection, Bioinformatics, *etc.*	Anomaly/Intrusion Detection, Network Operations, Optimization and Analytics, Dimensionality reduction and visualization, *etc.*	Speech recognition, Webpage classification, *etc.*
Examples	Decision tree, Random forest, K-nearest neighbors, Naive bayes classifier, Support vector machines, *etc.*	K-means clustering, Hierarchical clustering, Spectral clustering, DBSCAN, *etc.*	Transductive SVMs, Graph-Based Methods, *etc.*

Table 2.6: Comparison of some existing machine learning based anomaly detection techniques

Technique	Type	Feature Selection	Hybrid Model	Performance Evaluation	Dataset
Xu *et al.* [163]	Supervised	✓	×	✓	RTD
Tu *et al.* [164]	Semi-Supervised	×	×	✓	RTD, SD
Gaddam *et al.* [165]	Supervised	×	✓	✓	ORD
Ashfaq *et al.* [166]	Semi-Supervised	✓	×	✓	ORD
Jager *et al.* [167]	Supervised	✓	✓	✓	RTD
Peng *et al.* [168]	Semi-Supervised	×	✓	✓	ORD
Li *et al.* [169]	Semi-Supervised	✓	×	✓	RTD
Song *et al.* [170]	Unsupervised	✓	×	✓	RTD
Almalawi *et al.* [171]	Unsupervised	×	×	✓	RTD, SD
Li *et al.* [172]	Unsupervised	✓	✓	✓	RTD

RTD: Real-time dataset; SD: Synthetic dataset; ORD: Online repository dataset

Table 2.7: Comparison of some existing anomaly detection schemes based on distinctive characteristics

Approach	Technique(s)	Dataset Employed	Parameters	Contributions
Muniyandi *et al.* [173]	K-Means Clustering and C4.5 Decision Tree Algorithm	KDD Cup 1999 from UCI Repository	TPR, FPR, Accuracy, F-Measure, ROC Curve and AUCs	A cascading algorithm for supervised anomaly detection using the hybridization of k-Means and C4.5 Decision Tree.
Lv *et al.* [129]	Density Based Spatial Clustering of Applications with Noise (DBSCAN) Algorithm	Three UCI Repository datasets: Habermans Survival dataset, Iris dataset and Letter Recognition dataset	Correct Rate	An effective, efficient and extensible density-based clustering algorithm for complex structured datasets. Results shows that the proposed algorithm is better than the traditional DBSCAN algorithm and the improved algorithm IS-DBSCAN.
Miller *et al.* [65]	Density Based Spatial Clustering of Applications with Noise (DBSCAN), K-Means clustering algorithm along with DenStream and StreamKM++	3239 user accounts with a sample tweet from each account.	Accuracy, Balanced Accuracy, Precision, F-Measure, Recall, Specificity and FPR	Two stream clustering algorithms, StreamKM++ and DenStream were modified to facilitate spam identification.
Ahmed *et al.* [115]	Clustering Based algorithms, i.e., K-Means Algorithm and X-Means Algorithm	Three datasets namely: DARPA 1998, KDD Cup 99 & Kyoto.	Accuracy and Sensitivity	Network Knowledge Independent Collective Anomaly Detection Technique (NKICAD) to detect anomalies that are based upon an empirical analysis of a characteristic of an attack.
Karami *et al.* [174]	K-Means Clustering Algorithm and Particle Swarm Optimization (PSO) Algorithm	Five benchmark datasets: Iris dataset, Glass dataset, Wine dataset, Ionosphere dataset and Zoo dataset	DR, FPR, Accuracy, F Score, Specificity, Sensitivity	A fuzzy based anomaly detection algorithm that detects the anomalies in two phases, i.e., training and detection. Well separated clusters, high detection rate and low false positive rate are achieved by the proposed method.
Khoa *et al.* [175]	Principal Component Analysis (PCA), Distance based approaches and Density based approaches for data mining.	Two real world datasets: NICTA wireless mesh network and Abilene backbone network	FPR and FNR	A commute distance based technique for network anomaly detection. Distance based technique using Euclidean distance and commute distance can handle the anomalies well and is more stable and less sensitive to the parameters than PCA and density based techniques.

Approach	Technique(s)	Dataset Employed	Parameters	Contributions
Erfani *et al.* [176]	Deep belief network and One Class SVM	Four Real-time datasets from machine learning repository namely: GAS, OAR, DSA and HAR Two synthetic datasets: Banana and Smiley	Time and Memory complexities	A hybrid unsupervised anomaly detection technique using DBN and one class SVM for high dimensional unlabeled datasets. Proposed algorithms reduces training time and testing time
Laxhammar and Falkman [177]	Sequential Hausdorff Nearest Neighbor Conformal Anomaly Detector and Discords Algorithm	Four Labeled Trajectory Datasets are generated using trajectory generator software named Piciarelli	Sensitivity, Precision and Detection Delay	A parameter-light algorithm for online learning and anomaly detection in trajectories. The proposed algorithm achieves competitive classification rate when applied on large set of synthetic trajectories
Fiore *et al.* [178]	Restricted Boltzmann Machine	KDD 99 Dataset from UCI Machine learning repository	Accuracy, Speed, Comprehensibility and Time to learn	Discriminative Restricted Boltzmann Machine has been applied to network anomaly detection to infer the aspects for normal traffic
Bhuyan *et al.* [179]	MIGE-FS: Entropy based feature selection technique, TCLUS: Tree based clustering technique and an outlier score function	DR, FPR, Precision, Recall and F-measure	UCI Machine Learning Repository datasets, Real life TUIDS packets, KDD Cup 99 and NSL-KDD dataset	An efficient multi-step technique to detect anomalies in network wide traffic. The proposed technique performs well as compared to its counterparts in terms of different evaluation parameters employed
Kim *et al.* [142]	C4.5 Decision Tree Algorithm and One Class SVM Models	A refined version fo KDD99 dataset namely NSL-KDD Dataset	DR, FPR and Detection Time	A hybrid intrusion detection technique that hierarchically integrates misuse detection and anomaly detection for the detection of known as well as unknown attacks.
Sun *et al.* [180]	Support Vector Machine (SVM) and Moving Average and Entropy	CloudWatch data from AWS and log-data from servers	Accuracy, Precision, Recall and FPR	This approach makes use of log information to detect anomalies in public cloud, AWS during DevOps operation
Bosman *et al.* [181]	Correlation Map and Collection Tree Protocol, Recursive least squares and Extreme learning machine	Real-world network deployments and datasets	Neighborhood size and Spatio-temporal correlation	The neighborhood information was consumed to detect anomalies at sensor nodes

Approach	Technique(s)	Dataset Employed	Parameters	Contributions
Zhou *et al.* [182]	Convolutional Neural Networks (CNN), Stochastic gradient descent based back propagation technique and Spatial-temporal CNN model	UCSD dataset, UMN dataset, Subway dataset and U-turn dataset	Equal Error Rate, DR, TPR and FPR	A spatial temporal CNN model to detect abnormal behavior in crowded scenes of video sequences
Ding *et al.* [183]	Level-set boundary description methods, Kernel density estimation and Support Vector Machines (SVM)	Synthetic datasets, Ripley dataset, Vowel dataset and UCI-ML datasets	ROC curves, AUC and Relative performance comparisons	A level set boundary description approach for novelty detection in input space
Karami *et al.* [99]	Particle Swarm Optimization (PSO), Radial Basis Function (RBF)	Benchmark datasets, Packet traces of monitored traffic	Accuracy, Specificity and Sensitivity	A hybrid multi-objective technique for pro-active prediction of DoS attacks in named data networking environment
Forestiero *et al.* [149]	Density based Spatial Clustering of Applications with Noise (DBSCAN), Swarm Intelligence and Self organizing technique and Density-based clustering	Gauss, STREAM, Forest Covertype, DARPA, Fishers Iris and BUPA	Percentage of purity, Precision, Recall and Mean number of comparisons	A multi-agent algorithm was proposed to detect anomalies in distributed data streams
Aburomman and Reaz [184]	Support Vector Machine, K-Nearest Neighbor and Particle Swarm Optimization	Five random subsets from KDD'99 intrusion detection dataset	DR, Convergence Rate and Accuracy	A novel method that uses ensemble of classifiers for intrusion detection was proposed. Here, Local unimodal sampling (LUS) method is used to estimate the parameters of PSO
Fernandes *et al.* [185]	Principal Component Analysis and Ant Colony Optimization	Real Network Environment	TPR and FPR	Network anomaly detection model was proposed for the analysis of IP flows
Hsiao *et al.* [186]	Pareto depth analysis and Pareto Front	Real dataset with thousands of pedestrian trajectories and Synthetic dyads	Accuracy, Computation Time and Dissimilarities between trajectories	A similarity-based anomaly detection model was proposed which uses multi-criteria dissimilarity measure to exhibit anomalous behavior in a dataset

Approach	Technique(s)	Dataset Employed	Parameters	Contributions
Peng *et al.* [187]	Gaussian RBF kernel and Support Vector Data description	Hyper-spectral images and Multivariate datasets	Computation time, Probability of detection and FPR	Sparse kernel learning has been modelled as a mixed integer programming problem for the purpose of anomaly detection
Chorppath *et al.* [188]	Support Vector Machines, Naive Bayes Classifier and K-Nearest Neighbor	Numerical studies and Botnets dataset	Variation of price, Variation of utilities and Probability mass function	A novel detection method that uses machine learning methods for the determination of malicious users in a wireless network was proposed
Pascoal *et al.* [189]	Mutual Information Metric and Robust Principal Component Analysis	Network scenario of real traffic conditions	Recall, False Positive Rate and Precision	An anomaly detection model which uses a combination of feature selection and robust statistics was proposed for the classification of Internet traffic

FPR: False Positive Rate, FNR: False Negative Rate, DR: Detection Rate, TPR: True Positive Rate, and TNR: True Negative Rate

Table 2.8: Sources of datasets for anomaly detection models

Type	Source	Data set details	Data set retrieval
Data Repositories	Weboscope [190]	This library is created by Yahoo Labs with the aim to provide interesting and scientifically useful datasets to researchers for non-commercial use	User can get the data by sending a data request to the source
	Numenta Anomaly Benchmark (NAB) [191]	The NAB dataset contains streaming data samples for the evaluation of real-time anomaly detection algorithms	The complete NAB repository is available at github.
	UCI Repository [192]	This repository provides dataset as a service to the machine learning communities for analyzing the performance of machine learning algorithms	The datasets from this repository can be downloaded online by visiting their web portal
	Kaggle [193]	It is a modeling and analytic platform that contains a collection of widely used publically available datasets	Datasets can be downloaded directly by visiting Kaggle online portal
Data Marketplaces	Amazon Web Services (AWS) Public Datasets [194]	It is a centralized repository of public datasets where users have to pay for the services like computation and storage	User have to create an AWS account for getting the services
	Microsoft Azure Marketplace [195]	It is an infrastructure that provides services for buying and selling premium datasets through their global network of managed data centers	User can buy a wide variety of data from this source
	Big Data Exchange (BDEX) [196]	It is a real-time data marketplace that provides buying and selling modules for big data	It provides paid data service
	AggData [197]	They develops their own datasets to provide data driven services to the users	User can get the data either from free dataset directory of the source or can buy the data
Governmental	Open Government Data (OGD) Platform India [198]	This platform contains data published by Ministries/ Organizations of India to increase the transparency in the functioning of Government	Data is available online at data.gov.in
	U.S. Government Open Data [199]	Data is provided to serve millions of people for research purpose	Data can be accessed from www.data.gov
	DataBC (British Columbia) [200]	It is the platform to share data among the government and the public	Data is available at data.gov.bc.ca
	U.K. Government Data [201]	Data is provided under the initiative of Opening Up Government	Data can be accessed from data.gov.uk

Type	Source	About	How to get the dataset
Real-Time Dataset	Wired or wireless Sensors	Sensors generates the data by detecting some types of inputs from the physical environment	Firstly, choose the application domain for which the data is required. Secondly, identify the best available sensor for chosen application domain. Thirdly, identify the location where the sensor can give the best results. Finally, deploy the sensor to get the data.
Synthetic Data	Data generation tools. For Example, the open source R packages like synthpop and simPop, IBM Quest, geopeaks in MATLAB, etc.	It is generated to meet some specific needs that are not met with the real data or it is generated to validate mathematical models by comparing the behavior of real data with that of synthetic one	Firstly, analyze the original data sample to discover the model that best suites to its behavior. Secondly, identify the parameters for the detection of anomalous events. Thirdly, create the normal and anomalous profile of the data. Finally, simulate the normal and anomalous activity profiles to generate the synthetic data.

2.7 Crucial Aspects of Anomaly Detection

Anomaly detection is a well researched area but still the task of detecting anomalies is one of the important issues. Anomalies are generally identified by analyzing deviations from the normal patterns but this is a non-trivial solution because there is no generalized model to specify the normal behavior of the patterns. One of the methods which can be considered to identify normal behavior is by taking an attack-free training set but obtaining the clean data for training is again a major bottleneck. Apart from distinguishing a normal and abnormal behavior of the data, there are several other factors that make the task of identifying anomalies more challenging. Some of the key challenges are specified below. For example, in firewalls, it can be controlled that what kind of network traffic is considered as normal and acceptable.

Apart from distinguishing a normal and abnormal part of the data, there are many challenges of finding anomalies in big data [3, 5, 25] and some of them are illustrated below.

- *Data Analysis:* To make sure that whether the data itself is accurate, complete and consistent or not is again a typical task. It doesn't matter how good the developed model is, it will not perform well until the quality of data supplied to it is not reliable.

- *Computational Efficiency:* One of the key challenges in this domain is the large size of the input. With the evolving nature of the data, the need of analyzing data-streams has also grown. Traditional anomaly detection algorithms require the storage of data to a particular location for processing, but now, i.e., in the present scenario, with such a high rate of incoming observations, it can not be even thought of to store the data at a single location. So, making the traditional anomaly detection techniques memory efficient or discovering new memory efficient techniques is again quite challenging. Hence, computational efficiency of the anomaly detection technique is an important

issue.

- *Unbalanced Data:* Anomaly detection mechanisms explore the unseen space but due to the unbalanced nature of the data it becomes quite complex to distinguish between the actual datasets and the anomalous datasets. Any outlying observation which lies close to the outlying boundary can actually be normal. So, it becomes quite complex to define a normal region in such situations.

- *Feature Selection:* The best set of features are required for any algorithm to give the best results in terms of accuracy as well as computational efficiency. But the selection of feature vectors that are able to reflect the anomalous behavior of the data patterns is a major concern.

- *Ensemble Techniques:* Selection of ensemble techniques for anomaly detection is a challenging task. Here, accuracy and detection rate are the paramount considerations taken into account while designing an ensemble technique.

- *Frequently changing anomalies:* Another challenge is associated with the nature of the anomalies which keeps on changing with time. Thus, a single anomaly detection algorithm cannot work well in all cases.

- *Noise in incoming data:* In big data technology, data might be collected in distributed fashion. The presence of noise in the data collected from various sources makes it difficult to distinguish between anomalies and noise and thus, makes anomaly detection even more challenging. In other words, the presence of noise sometimes leads to false detections as well.

- *Non-availability of training data:* Lack of availability of data for training purpose is another major challenge. If the data model is not trained with the suitable number of

data samples or if the data to be evaluated does not fit well with the data model; then a case may occur wherein a model may not fit in and lead to incorrect results. For instance, normal data point may be incorrectly evaluated as anomaly and vice-versa.

- *Optimality of Technique:* Anomaly detection techniques that produce too many false positives or allow some issues to go undetected are not feasible at all. Finding a robust anomaly detection technique with high accuracy rate and minimal false positive rate is very difficult.

- *Scalability:* Anomaly detection technique developed for one domain may not work well for other domains. This is due to the reason that both normal and abnormal behaviors vary from one domain to other. Scalability is required to provide a system with an ability to detect anomalies in large datasets in an efficient way. Hence, to develop scalable approaches for real time detection in big data is yet another challenge. Thus, developing a generalized anomaly detection technique is still a challenge.

2.8 Motivation

Network anomalies disrupt the normal functionalities of the network by inducing high traffic flow within short intervals of time. Thus, critical information regarding intrusions, faults, and system failures is required before it induces widespread damage. However, issues like availability of attack-free training set, imbalance between types of anomalies and presence of categorical features make it difficult to tune such techniques for high dimensional network datasets [1, 3, 5]. Some of these significant challenges are illustrated as under.

First and foremost is the unavailability of the clean dataset, i.e., attack-free training set. Hence, majority of the existing approaches are based on the assumption that the dataset being utilized for the training is correct with respect to its labels. However, this is not true in

practice as it is quite difficult to obtain clean data. The uncertainty in the boundaries between the normal and anomalous instances are the reasons behind it. Moreover, in some cases attacks occur during the training phase. So, if such attacks remain undetected during this phase, then they will be assumed as part of the training model thereby, causing a significant effect on the developed model.

Ideally, an anomaly detection model should achieve 100% detection accuracy rate and 0% false alarm rate, i.e., all the instances should be correctly identified by the model [202]. However, achieving this idealized situation often pose number of challenges to anomaly detection techniques. This is due to the reason that the model of normal traffic over-fits the training data. The presence of unknown anomalous instances in the training data renders the quality of classifier. Due to this, the model results in false negatives and the normal traffic that is not present during the training phase may be regarded as anomalous whereas the instances that do not belong to the true normal model may be regarded as normal. It also makes the anomaly detection model incapable of detecting closely related instances. Thus, to enhance the true detection capabilities of anomaly detection model, training data should be sanitized by removing both non-regular and unknown anomalous instances.

Further, network traffic data is characterized by high dimensionality and noise which makes it difficult to extract the most optimal and relevant set of features. These datasets often contain too many variables which make them unsuitable for the existing learning models. Due to this, a learning model tends to over-fit and the correlation between features leads to poor classification results. Removal of such features provides better learning performance in terms of searching speed, learning accuracy, computational cost, and model interoperability [68, 73]. However, accidental elimination of important features might decrease the accuracy rate of classification. In such situations, unsupervised techniques perform better as they enhance the separability and likelihood of features.

On the contrary, unsupervised techniques make two assumptions while working with the network traffic data [203]. The primary assumption is that the number of normal instances of data are more than the number of anomalous ones. Second assumption is that the attacks or anomalies will significantly differ from normal network traffic. There exists a possibility where the above mentioned assumptions may fail or may lead to degradation in performance of anomaly detection algorithms. An example of such intrusion where such a situation occurs is DoS attack. The reason behind the cause of this situation is occurrence of DoS attack in similar number(s) as of normal instances. So, algorithms mostly fail to label these instances as anomalies because the regions of occurrence of these anomalies are highly dense and quite similar to the region of normal instances [204].

Moreover, optimization of feature-set is desirable to improve the detection speed and accuracy in network traffic data. Most of the traditional optimization approaches consider only linear correlation coefficient between attributes but non-linear correlation analysis is also required to provide more computational power to the classifier [66]. Further, existing classification approaches for network anomaly detection are unsuitable for real-time use. Although significant research has been done by the researchers for intrusion detection but still the importance of network based techniques for large scale datasets has not been explored to its full potential [34]. Thus, the main challenge is to design an efficient anomaly detection model that can analyze real-time network traffic for the identification of anomalies.

2.9 Need for Anomaly Detection in Big Data

The fundamental characteristics of anomaly detection in context of Big Data is heavily dependent on the application area, type of anomaly and availability of labeled data. For instance, with respect to anomaly detection in streaming data, the items occurring more than a

given fraction of time are considered anomalous. It is one of the most highly studied problems in data stream mining, dating back to the 1980s. Many applications rely directly or indirectly on finding the frequent items, and implementations are used in large scale industrial systems across different domains. In electrical and water sensors, it is essential to detect the faulty sensors, and raise alerts when abnormal amount of water is being used. In commercial organizations such as-banks, insurance companies, stock market, etc., fraudulent activities involving financial thefts are considered anomalous. On the contrary, poor instrumental readings, abnormal patient condition and human errors are regarded as anomalies in healthcare domain. In context of image processing, particularly satellite imagery, motion detection is deemed as an important anomaly detection problem. In Industrial domain, faults in mechanical equipment(s) and structural defects are considered anomalous. Likewise, anomaly can be defined or redefined in accordance with the application domain and availability of data.

2.10 Machine Learning for Anomaly Detection

Different anomalies showcase different characteristics under varying network scenarios. To detect these anomalies, numerous techniques have been exploited ranging from statistics, Machine Learning (ML), data mining, information theory, etc. However, designing a generic model for network anomaly detection often possess challenges in identifying various security attack vectors. In this context, model-based approaches are found to be less portable and inappropriate for different application domains since they are susceptible to even minor alterations in the attributes of the underlying network traffic. Thus, in order to cater these challenges, non-parametric approaches have emerged as possible solutions due to their capability to learn, adapt, recognise and optimize the decisions from the available data inputs by themselves. In this direction, McKinsey Global Institute asserted on the notion that ML, data

mining and predictive analytics would be the major drivers of the next-generation paradigm of innovation and creativity [205].

ML is a method of data analytics which drives the field of Artificial Intelligence (AI) into the future while providing advanced model building capabilities. The concept of ML has been around for a long time, however they have gained momentum over the last several years. With the constant evolution of this field, they are currently being used in a wide variety of domains that are advancing the underlying security measures into a new realm. This can be credited to their ability to learn from the data and make the related predictions based on the past learning experience. Additionally, ML-based techniques have the ability to amplify the search process; thereby enhancing the effectiveness and the speed of identification. Essentially, their key objective is identifying the fraudulent activities, recognizing the unusual and strange patterns, and connecting the dots. In summary, ML-based techniques act as "Detectors to identify the security risks".

Some of the prominent applications of ML in context of anomaly detection are detailed as follows. In connection to complex signals, anomaly detection relates to the identification of data items that cross the normal range within a predefined threshold. In context of event streams associated with Web traffic, historical data can be exploited by ML to detect anomalous activities such as-brute-force attempts, data tempering attacks, etc. On the other hand, ML can also be employed to analyse the source of data (namely-network, server, logs, devices, etc.) with respect to response time, incoming traffic/data, rouge users, unknown security attacks, zero-day attacks, etc. In a nutshell, ML can be thought of a combination of following features and characteristics:

- Employ probabilistic models to classify the normal and contrasting activities.
- Fix an adaptive threshold in order to find out the data items which do not belong to the normal range

- Leverage the historical data for identifying the anomalies in sporadic event streams
- Study the pattern of deviations in order to detect the fraudulent activities.

2.11 Objectives

The key objectives undertaken in this thesis are summarized as follows:

1. To study, analyze and explore the existing anomaly detection and analysis techniques for Big data and identify important aspects that have not been addressed till yet.
2. To propose a novel technique for anomaly detection and analysis in massive datasets.
3. To test and validate the proposed technique using various existing anomaly detection and analysis parameters.

2.12 Concluding Remarks

The existing anomaly detection techniques employee different strategies such as-dimensionality reduction, optimization methods and machine learning models for enhancing the detection rate and reducing the false positives. These strategies are discussed in detail in this chapter along with the comparative review of the existing state-of-the-art techniques. Additionally, motivation for carrying out this work is also highlighted, followed by the objectives of this thesis work. In the next chapter, an ensemble based network anomaly detection technique has been proposed.

Chapter 3

Ensemble based Anomaly Detection Technique

In this chapter, a hybrid variant of anomaly detection technique have been proposed which is referred as *"Ensemble based Anomaly Detection Technique (En-ADT)"*. The detailed working of the proposed technique is mentioned as follows.

3.1 Working of En-ADT

The proposed technique employs Fuzzy K-means clustering (FKM) algorithm as a feature extraction algorithm. FKM extracts the important features which are then optimized by the Extended Kalman Filter (EKF). The optimized set of features are then utilized to train the SVM classifier. Once the Support Vector Machines (SVM) classifier is trained, it is then used to label the samples present in the testing dataset.The flow of the proposed technique is demonstrated in Fig. 3.1.

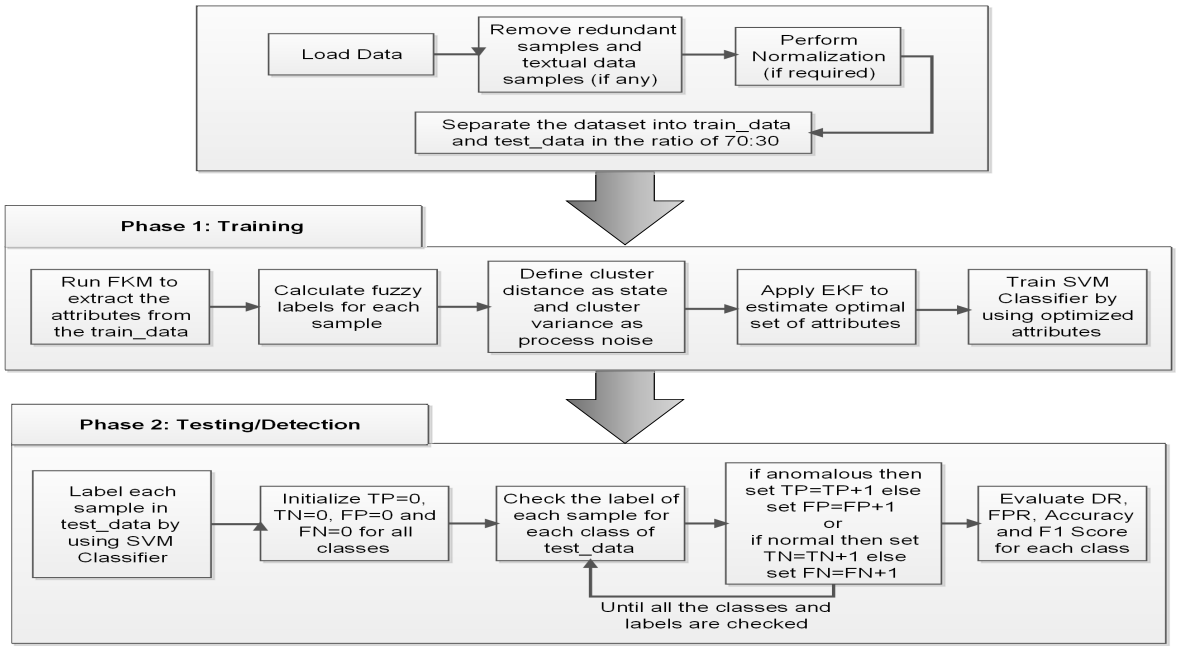


Figure 3.1: Flow of the proposed technique

3.1.1 Feature Selection

The anomaly detection problem is a challenging problem as compared to normal classification problems due to the major difference in the size of the classes. The number of features have a crucial role in computing the accuracy and predicting the timing of the algorithm. Publicly available well known network traffic datasets: DARPA 1998 and KDD CUP 1999 have several set of features. Of them, features that have the ability to reflect the malicious behaviours of the traffic instances should be identified to intensify the performance of the detection technique. This scheme uses FKM as a feature selection technique for extracting the meaningful features out of the large dimensional dataset. This technique is discussed in detail in the following subsection.

Fuzzy K-Means Algorithm:

K-means is a frequently used technique for clustering data without any labels. In k-means algorithm, data points are precisely allocated to predefined clusters according to some distortion measure. The most extensively used distortion measure is Euclidean distance in which a short distance signifies a huge resemblance whereas big distance signifies a low resemblance.

Since, the anomaly detection problem involves several attributes so building models directly on this data involves high rate of errors. Further the regions where normal and abnormal points lies is not clearly defined in such type of data. Traditional K-means generates crisp partitions and assignment of each object is done only to one cluster. But as each point has a probability of belonging to various cluster rather then completely belonging to a single cluster so fuzzy set theory can be used to overcome these issues [206].

In classical set theory, the boundaries of the objects are precisely defined and the objects in the classes are mutually exclusive. It works on a bivalent truth and no partial membership of objects within the classes is allowed. Each object can lie within a class with a membership grade of 1 (fully belong to a class) or with 0 (not a member of class). The object will have a membership grade 1 for a class to which it belongs, whereas 0 for rest of the classes. Thus, to overcome this sharp bounded condition of classical sets, fuzzy set theory is used which allows the objects to belong to several classes at the same time. In fuzzy theory, objects are considered as a fuzzy set of goals and for each goal the degree of membership in terms of numeric weights between 0 and 1 is associated [109]. This degree of membership expresses the extent of belongingness of objects to the different classes.

Let S be the sample subspace of objects O in the dataset. A class is considered as a feature of the dataset whereas a cluster is considered as a feature of FKM clustering algorithm. Now, an object is represented using fuzzy set theory as:

Definition 1: A fuzzy set in subspace S is characterized by a membership function F(S) such that a real number in the interval of [0,1] is associated to each point in S. Here, F(S) represents the grade of membership (μ) of O in S to the classes. The close the value of F(S) to unity, the higher is the grade of membership.

The membership functions that are used to represent objects in fuzzy set theory can be of different shapes. Different membership functions means different objects. In the current context of research, three-part crisp objects are considered to define a separation between internal, external and boundary points of clusters. Internal objects are the points that definitely belongs to the cluster and have a membership $\mu = 1$, boundary points are those points that possibly belongs to a cluster, i.e., for such points $0 < \mu < 1$ and the points that does not belongs to a cluster or lie outside the boundary of cluster are the external points with $\mu = 0$. Thus, depending upon the relative location of data points, a certain grade of membership is allocated to them. These values of membership functions are thus utilized to do the optimal assignment of objects to clusters.

As the fuzzy logic provides the ability of having partial belongingness to objects thus it is a better way to reflect the reality where everything is not just true or false but often conditions like partial truth or partial false also exists. The transition from classic sets to fuzzy sets is the best suitable way to adequately imprecision the human thinking and enhance the subjectivity. Thus, FKM algorithm is employed in the current context of research to solve the problem of feature selection.

Definition 2 (FKM): It is a clustering algorithm that groups a dataset into clusters and each data-point contained in the dataset belongs to each cluster with certain belonging probability [207].

In the FKM approach, n elements $x_i (i = 1, 2, \cdots, n)$ are partitioned into k clusters $c_j (j =$

$1,2,\cdots,k$) and relationship between x_i and c_j is fuzzy which means a feature vector x_i can have a degree of membership in each cluster j. A degree of belongingness between data point X_i and cluster j is represented by $u_{ij} \in [0,1]$ where probability function 1 means that data point lies near to the centroid and 0 otherwise. It also tries to minimize the trailing objective function.

$$F_m = \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij}^m |x_i - c_j|^2, 1 \leq m < \infty \tag{3.1}$$

where $|x_i - c_j|^2$ represents the distance between the data object x_i and the cluster center c_j. The squared error is used as a notion of similarity to measure the weighted sum of distances between the data objects and fuzzy clusters. The smaller the distance between objects, more familiar are the objects. Further, the variable m reflects the influence of degree of membership on clustering. The more the value of m, the more fuzzified is the clustering. The conditions that are necessary to make Eq. (3.1) reach its minimal are:

$$u_{ij} = \left(\sum_{l=1}^{k} \left(\frac{|x_i - c_j|}{|x_i - c_l|} \right)^{2/(m-1)} \right)^{-1} \tag{3.2}$$

$$and \;\; c_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m} \tag{3.3}$$

Here matrix u_{ij} from Eq. (3.2) and the associated cluster centers from Eq. (3.3) are computed followed by computation of squared error by Eq. (3.1). The algorithm is terminated when either of the two conditions are met, i.e., either a computed squared error reaches below tolerance level or the computed error in next iteration over previous iteration is below a certain defined threshold.

In FKM, process of initializing the parameters, iteration of loops, and termination of the algorithm are the same as those considered in k-means algorithm. The only difference is that the resulting clusters in this approach are analyzed probabilistically rather than doing a hard

assignment of labels. The FKM algorithm is summarized in Algorithm 3.1. This technique utilizes fuzzy set theory to represent objects in a cluster. FKM is used to select the features out of the input support vectors in a particular cluster.

Algorithm 3.1 Feature extraction by FKM

input:Data samples $x_i \in S_{tr}$ where S_{tr} represents train_data set.
output:Extracted attributes a_e from train_data.

1: **for** each attribute a_t **do**
2: Calculate maximum likelihood estimate (MLE) of a_t to label l_j using $f(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp^{\frac{-(x-\mu)^2}{2\sigma^2}}$
3: **end for**
4: Initialize k number of clusters.
5: Assign random centroids $(c_1, c_2, \cdots, c_k)$ to all k clusters using uniform distribution.
6: **while** there are changes in the mean: **do**
7: Compute the degree of belongingness u_{ij} $\forall$ data point x_i in cluster j by using MLE of a_t
8: **for** j from 1 to k **do**
9: Replace m_j with the fuzzy means of all the data points of cluster c_j as
10: $m_j = \frac{\sum_{x_i \in k} u_{ij}^m x_i}{\sum_{x_i \in k} u_{ij}^m}$
11: **end for**
12: **end while**
13: $a_e = 0$
14: **for** each cluster c_j **do**
15: max(c_j)=label of dominant samples
16: **if** max(c_j)$\geq$ threshold **then**
17: $a_e = c_j$
18: **end if**
19: **end for**
20: Return extracted attributes a_e where m<t.

Complexity Analysis

The overall complexity of FKM as detailed in Algorithm 3.1 is found to be $O(ntk^2i)$; wherein the variable n denotes the number of elements, t represents the number of attributes, k denotes the number of clusters and i represents the number of iterations, respectively.

3.1.2 Extended Kalman Filter

Kalman Filter is a linear quadratic estimation algorithm that provides an computationally efficient means to evaluate the state estimates in a way that minimizes mean-square error. It addresses state estimation problem to implement a predictor-corrector type estimator that is optimal when the precise nature of the modeled system is unknown. The process is discretized and this form can be expressed as:

$$\vec{x}_n = F_{n-1} + \vec{x}_n + \vec{w}_{n-1} \tag{3.4}$$

with a measurement $z \in \Re^n$, i.e.,

$$\vec{z}_n = H\vec{x}_n + \vec{v}_n \tag{3.5}$$

where $x \in \Re^n$ is a state of a discretized process, $\vec{x}_n$ in difference Eq. (3.4) denotes a state vector, $\vec{w}_n$ is a discrete white noise for process and F is the state transition matrix relating the state of the previous time step (n-1) to the current time step n. Eq. (3.5) is a measurement equation with $\vec{v}_n$ as a measurement noise. H relates the state to the measurement $\vec{z}_n$. Both the noises are independent and have the normal probability distributions as:

$$P(\vec{w}) \sim N(0, Q) \tag{3.6}$$

$$P(\vec{v}) \sim N(0, R) \tag{3.7}$$

where Q and R are the covariances for process noise and measurement noise respectively. Further, the state transition matrix F is given by :

$$
\begin{aligned}
F &= e^{A\triangle n} \\
&= \begin{bmatrix} 1 & \triangle n & \frac{\triangle n^2}{2} \\ 0 & 1 & \triangle n \\ 0 & 0 & 1 \end{bmatrix} \cdot
\end{aligned} \tag{3.8}
$$

which is computed by taking the Laplace transform of the state vector $\vec{x}_n$. Here A relates to the previous time step n and $\triangle n$ denotes the sample time interval between (n-1) and n.

Here terms upto second order are considered and higher order terms are neglected for the expansion of the state transition matrix. By using the process model, the states are propagated (predicted) in time and the measurements are used to correct the predicted estimates. The equations utilized to solve this problem in predictor-corrector estimator are:

Predictor step

$$
\begin{aligned}
\vec{\hat{x}}_n^- &= F\vec{\hat{x}}_{n-1}^+ + \vec{\hat{x}}_n \\
P_n &= F_{n-1}P_{n-1}F_{n-1}^T + Q_{n-1}
\end{aligned}
$$

Corrector step

$$
\begin{aligned}
K_n &= P_n H_n^T (H_n P_n H_n^T + R_n)^{-1} \\
\vec{y}_n{}^- &= H_n \vec{\hat{x}}_n^- \\
\vec{\hat{x}}_n^+ &= \vec{\hat{x}}_n^- + K_n(\vec{y}_n - \vec{y}_n{}^-) \\
P_n^+ &= (I - K_n H_n)P_n^-(I - K_n H_n)^T + K_n R_n K_n^T
\end{aligned} \tag{3.9}
$$

where $\vec{\hat{x}}_n^-$ is the apriori state vector, the state transition matrix F is computed from Eq. (3.8), P denotes the covariance matrix, H represents the measurement model from Eq. (3.5), covariance for process noise Q and covariance for measurement noise R are determined from

Eq. (3.6) and Eq. (3.7) respectively, K_n is the Kalman Gain, $\vec{y}_n$ is the measurement vector, $\vec{y}_n^-$ is the apriori measurement vector, $\vec{\hat{x}}_n^+$ is the aposteriori state vector, $\vec{\hat{x}}_n^-$ is the apriori state vector, P_n^- is the apriori covariance and P_n^- is the aposteriori covariance. For each predictor and corrector step, previous posteriori estimates are used to predict new apriori estimates.

To design a non-linear filter, EKF based on first-order local linearization is used. In this Bayesian approach, Taylor series expansion is used to linearize this non-linear system and the nominal value for doing this linearization is taken from estimation value of the last time step. Now, the process is described by the non-linear stochastic differential equation with discrete measurement values as:

$$\dot{x}(t) = f(x(t), \tau_c, w(t)) \tag{3.10}$$

with a measurement $x \in \Re^n$, i.e.,

$$z_n = h(x_n + v_n.) \tag{3.11}$$

The non-linear function f in differential Eq. (3.10) relates the states at time t and non-linear function h in Eq. (3.11) relates the state x_n to the measurement z_n. The variables $w(t)$ and v_n. represent the process and measurement noise as represented in Eq. (3.4) and Eq. (3.5), $z_n \in D_x \subset \Re^p$ represents the p-dimensional system measurement vector, non-linear states to inputs mapping of system is represented by $f(.) : D_x \to \Re^n$, $h(.) : D_x \subset \Re^n \to \Re^p$ denotes the non-linear states to output mapping of system, continuous process noise is denoted by $\tau_c \in \Re^{nxw}$, w-dimensional random process noise is represented by $w(t) \in D_w \subset \Re^w$ and $v_n \in D_v \subset \Re^v$ represents the v-dimensional measurement noise. Both of the random process noise and random measurement noise are considered to have zero mean. Mathematically, the

process of Additive White Gaussian Noise (AWGN) is specified by (Eq. (3.12)).

$$E[w(t)w(t-\tau)^T] = Q\delta(t-\tau) = \begin{cases} Q & \text{if } t = \tau \\ 0 & \text{if } t \neq \tau \end{cases}$$
$$E[v_n v_j^T] = R_n \delta_{nj} = \begin{cases} R_n & \text{if } n = j \\ 0 & \text{if } n \neq j. \end{cases} \}$$
(3.12)

where the covariance of continuous process noise and the covariance of discrete measurement noise are denoted by Q and R_n as shown in Eq. (3.6) and Eq. (3.7) respectively, the uncorrelated random variables $w(t)$ and v_n follows the normal distribution as $w(t) \sim \mathcal{N}(0,Q)$ and $v_n \sim \mathcal{N}(0,R_n)$ and the Dirac delta function and Kronecker delta function are given by $\delta(.)$ and δ_{nj}.

Now, let n be the discrete index, $\bar{x_0}$ is the mean and P_0 is the covariance for the initial state of the system. Then mathematically Extended Kalman Filter is given by:

Predictor step

$$\begin{aligned} \hat{x}(t) &= \hat{x}_{n-1} \\ P(t) &= P_{n-1} \\ \dot{\hat{x}}(t) &= f(\hat{x}(t)) \\ \dot{P}(t) &= F(t)P(t) + P(t)F^T(t) + \tau_c Q \tau_c^T \\ \text{where } F(t) &= \left.\frac{\partial f}{\partial x(t)}\right|_{x(t)=\hat{x}(t)} \end{aligned}$$
(3.13)

Corrector step

$$\hat{x}_n = \hat{x}(t) + \dot{\hat{x}}(t)\Delta t$$

$$P_n^- = P(t) + \dot{P}(t)\Delta t$$

$$z_n^- = h(\hat{x}_n^-)$$

$$\hat{x}_n = \hat{x}_n^- + K_n(z_n - z_n^-)$$

$$P_n = (I - K_n H_n)P_n^-$$

$$\text{where } H_n = \left.\frac{\partial h}{\partial x_n}\right|_{x_n = \hat{x}_n^-}$$

here the Jacobian H_n in the corrector step of EKF evaluates the relevant components of measurement. An algorithm depicting the whole process of Kalman Filter is summarized in Algorithm 3.2.

Algorithm 3.2 Feature optimization by EKF

input:Extracted attributes a_e by FKM.
output:Optimized attributes a_o.

1: Initialize weight w_m of fuzzy clusters attributes a_e as $\vec{\hat{x}}_0^+$ and cluster variance as $\mathbf{P}_0^+$.
2: n=0
3: **while** true **do**
4: $\vec{\hat{x}}_n^- = \vec{\hat{x}}_{n-1}^+$ $/*apriori\ state\ vector*/$
5: $P_n = F_{n-1}P_{n-1}F_{n-1}^T + Q_{n-1}$ $/*apriori\ covariance\ of\ estimation\ error*/$
6: $K_n = P_n H_n^T (H_n P_n H_n^T + R_n)^{-1}$ $/*kalman\ gain*/$
7: $\vec{y}_n^- = H_n \vec{\hat{x}}_n^-$ $/*apriori\ measurement\ vector*/$
8: $\vec{\hat{x}}_n^+ = \vec{\hat{x}}_n^- + K_n(\vec{y}_n - \vec{y}_n^-)$ $/*posteriori\ state\ vector*/$
9: $P_n^+ = (I - K_n H_n)P_n^-(I - K_n H_n)^T + K_n R_n K_n^T$
$/*apriori\ covariance\ of\ estimation\ error*/$
10: n=n+1
11: **if** $P_n^+ < \varepsilon$ **then**
12: break
13: **end if**
14: **end while**
15: Return a_o given by $\vec{\hat{x}}_n^+$

Complexity Analysis

The complexity of Algorithm 3.2 can be defined with respect to the individual complexity of the Predictor and Corrector steps. In the former step, the complexity is found to be $O(n^3)$; while in the later step it is found to be $O(n^2)$. Therefore, the overall complexity of Algorithm 3.2 is essentially $O(n^3)$; with n denoting the number of time steps.

3.1.3 Support Vector Machines

It is a supervised learning technique that classifies the data into distinct categories by constructing a hyperplane and extends this hyper plane to non-linear boundaries [208]. By doing so it tries to maximize the margin of separation to obtain best classification results. Mathematically:

$$\text{If } Z_{ix} = +1; \eta_{xi} + b \geq 1 \tag{3.14}$$

$$\text{If } Z_{ix} = -1; \eta_{xi} + b \leq 1 \tag{3.15}$$

$$\text{For all i}; Z_i(\eta_{xib}) \geq 1 \tag{3.16}$$

where Z_{ix} is a vector point, η is weight vector and b is some constant. To find the solution for non-linear generalization, a dual problem is constructed which is much simpler than primal. Now, the task is to find η and b such that $\phi(\eta) = \frac{1}{2}|\eta||b|$ is minimized and for all (x_i, y_i) :

$$y_i(\eta * x_i + b) \geq 1 \tag{3.17}$$

Suppose, (x_1, y_1),..., (x_n, y_n) are the n labeled samples with labels $y_i \in \{1, -1\}$, now the problem is to compute the hyperplane given by $< \eta, x > +b = 0$ (parametrized by (η, b)) such that:

1. A canonical plane is in position due to fixed scale of (η, b) corresponding to $\{x_1, x_2 \ldots, x_n\}$, *i.e.*,

$$\min_{i \leq n} | < \eta, x_i > + b| = 1$$

2. The $+1$'s are separated from the -1's by the plane (η, b). *i.e.*,

$$y_i(< \eta, x_i > + b) \geq 0, \forall i \leq n \tag{3.18}$$

3. Maximum margin of the plane is $\rho = 1/|\eta|$, *i.e.*, minimum $|\eta|^2$.

The observed data might not have a separating plane so the support vectors are used to create the same which are discussed in the following sub-section.

The Support Vectors of SVM

The Karush-Kuhn-Tucker (KKT) conditions are applied to the problem of finding the boundary hyperplane having maximum margin. These conditions provide the optimal solutions under such conditions where each equality constraint is an affine function and all inequality constraints are continuously differentiable. Secondly, in the case of SVM, primal objective and hyperplane constraints are convex thus such conditions are required that can hold at saddle point and solves the problem of convex optimization. Further, the dual form has simpler constraints as compared to primal and it also allows the usage of kernel trick, thus conversion from primal to the dual is required to convert the problem from saddle point to simple maximum. Thirdly, affine constraint functions in SVM results in zero duality gap. Thus complementary slackness condition related to the value of Lagrangian multiplier needs to be satisfied. This condition states that either the primal constraint is satisfied with equality or its corresponding Lagrangian multiplier is zero. The complementary slackness condition present in KKT can be directly fed to SVM which makes KKT conditions convenient to

use [209, 210].

As these conditions satisfies certain requirements as demanded in SVM so in the current context of research such conditions are applied to determine the boundary hyperplane having maximum margin. The Lagrangian is given by:

$$\mathscr{L}(\eta, b, \lambda) = \frac{1}{2}\sum_{i=1}^{d}\eta_i^2 - \sum_{j=1}^{n}\lambda_j\left\{y_j(<\eta, x_j> +b) - 1\right\}, \tag{3.19}$$

and the optimality is found such that

$$\begin{aligned} \nabla_\eta L &= 0, \text{i.e., } \eta = \sum_{j=1}^{n}\lambda_j y_j x_j & (3.20)\\ \nabla_b L &= 0, \text{i.e., } \sum_{j=1}^{n}\lambda_j y_j = 0 & (3.21)\\ \lambda_j\left\{y_j(<\eta, x_j> +b) - 1\right\} &= 0, \text{ for all } j \leq n. & (3.22) \end{aligned}$$

These provide a complete characterization of the optimal plane where a linear combination of x_j forms a normal η in Eq. (3.20). The coefficients of the linear combination provided by Eq. (3.20) adds upto zero in Eq. (3.21). Eq. (3.22) expresses the complementarity conditions where the only non-zero Lagrange multipliers λ_j are related to the vectors x_j present on the margin such that:

$$y_j(<\eta, x_j> +b) = 1 \tag{3.23}$$

They are known as *support vectors* and

$$\eta = \sum_{j\in J_0}\lambda_j y_j x_j \tag{3.24}$$

where $J_0 = \{j : x_j \text{ is a support vector}\}$. The observations x_j at the exact distance $\rho = 1/|\eta|$ from the separating plane are the support vectors. Usually such vectors are very small and

large number of such vectors must be considered with x_j having many coordinates. The following functions are computed to maximize the margin of SVM as follows.

$$W(VM) = \sum_{i=1}^{l} VM_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} VM_i VM_j y_i y_j (x_i x_j) \tag{3.25}$$

$$W(VM) = \sum_{i=1}^{l} VM_i + \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} VM_i VM_j y_i y_j (x_i x_j) \tag{3.26}$$

$$\text{subject to } \forall i : 0 \leq VM_i \leq C, and \sum_{i=1}^{l} VM_i y_i = 0,$$

where the number of training dataset instances is denoted by l, VM denotes the set of l variables, VM_i correlates to the training data x_i and the parameter of the margin C prevents the effect of anomalies in the training data. The above function calculates vector VM in which each entity indicates the weight of each sample point. The data points whose VM_i values are greater than 0 will be considered as support vectors.

Kernel Functions in SVM

In SVM, there are several kernel functions and any of these kernels can be used to achieve the boundary function [211].

$$k(x_i, x_j) = \begin{cases} (x_i \centerdot x_j) & \text{Linear Kernel} \\ (\gamma x_i \centerdot x_j + b)^d & \text{Polynomial Kernel} \\ \exp(-\gamma |x_i - x_j|^2) & \text{Radical Basis Function (RBF) Kernel} \\ \tanh(\gamma x_i \centerdot x_j + b) & \text{Sigmoid Kernel} \end{cases}$$

where $k(x_i, x_j)$ represents the mapping of input data point into high dimensional feature space and γ here is an adjustable parameter.

Algorithm 3.3 Label detection by SVM

input:Optimized attributes a_o from EKF and test_data
output:Detected labels TL_i of test_data.

```
Initialize weights η of hyperplane h.
for each sample point x_i do
    Compute the Lagrangian by (Eq. (3.19))
    if (VM_i{y_i(< η, x_i > +b) − 1})<0 then
        Update weights η using (Eq. (3.25))
    else if (VM_i{y_i(< η, x_i > +b) − 1})>0 then
        Update weights η using (Eq. (3.26))
  else
        Do preserve the weights
    end if
    Calculate support vectors SV_i for sample point x_i to hyperplane h
    VM=∞
    for f=1 to i do
        if SV_f <VM then
            VM=SV_f
        end if
    end for
end for
for each test sample TS_i do
    Predict TL_i = (VM_i{y_i(< η, x_i > +b) − 1})
end for
Return computed labels TL_i of test_data
```

In SVM, training complexity of the classifier highly depends upon the size of the dataset and since the datasets used in this study involves huge number of data points so this dataset cannot be fed directly to the SVM. Therefore, a hybridization of FKM and EKF is used to extract the optimized set of attributes from the dataset. These attributes are then utilized as the input support vectors of the SVM to train the SVM classifier. Once the SVM classifier is trained, it classifies the labels of the test dataset as normal and anomalous. The whole process of this classification mechanism is summarized in Algorithm 3.3.

Complexity Analysis

The overall complexity of label detection by SVM (as detailed in Algorithm 3.3) is defined as follows: $O(\text{Number of samples}^2 \times \text{Number of Optimized features})$, *i.e.*, $O(n^2 f)$.

3.1.4 Ensembled Anomaly Detection Technique

The proposed technique solves the anomaly detection problem by using SVM with the aide of FKM and EKF. The ensembling of FKM and EKF is done to reduce the input support vectors of SVM. FKM selects the important features from training set by clustering the similar features in same clusters. The fuzzy membership functions generated by FKM are used instead of the crisp clusters to extract the relevant information from data objects lying between clusters. These features are then optimized by an EKF which treats this problem as a state estimation problem.

The performance of the FKM algorithm depends on the characteristics of the membership functions. So, for a given rule base, the performance of a fuzzy system is enhanced by optimizing the membership functions by EKF. The reason behind the selection of this Bayesian approach over evolutionary approaches is the convergence time. The problem of optimizing the fuzzy membership functions is actually a weighted least-squares minimization problem. In this problem, the difference between the output of the fuzzy system along with the targets is calculated and the mean square of these values is treated as the error function which needs to be minimized. In the proposed approach, this problem is converted to a nonlinear state estimation problem where the states of the system itself are the membership functions. Now, for this conversion EKF is used because in the estimation theory it is mentioned that EKF is the non-linear version of Kalman Filter. Kalman Filter estimates the state of a discrete time controlled process that is governed by a linear stochastic differential equation. It assumes linear relationships in both the cases, i.e., propagation of the state and projection onto the

Algorithm 3.4 Anomaly detection by the proposed algorithm

input:Dataset.
output:Detected Anomalies in data and performance measures of the proposed algorithm.

Read dataset.
n=number_of_rows(dataset) /* *size of the dataset* */
$\forall$ n check for redundancy
if Redundant samples are present **then**
    Remove redundant samples
  else
    Check whether the data is normally distributed by plotting normal distribution curve
end if
if data is normally distributed **then**
    train=70
    test=30
    train_data=dataset[n* train/100] /* *70% of data as training samples* */
    test_data=dataset[n* test/100] /* *30% of data as testing samples* */
  else
    dataset=shuffle(dataset) /* *reshuffling the dataset to get normal distribution* */
end if
call(Algorithm 3.1) /* *feature extraction from dataset by FKM* */
call(Algorithm 3.2) /* *optimization of features by EKF* */
call (Algorithm 3.3) /* *label detection by SVM* */
n_{te}=sample.count(test_data) /* *no. of samples in testing dataset* */
for each class of test_data **do**
    Set counters TP=0, FP=0, TN=0, FN=0;
end for
for i=1 to n_{te} **do**
    c=class_TL_i /* *determining the class of the label* */
    if (TL_i==Anomalous) **then**
        $TP_c = TP_c + 1$
  else
        $FP_c = FP_c + 1$
    end if
    if (TL_i==Normal) **then**
        $TN_c = TN_c + 1$
  else
        $FN_c = FN_c + 1$
    end if
end for
Evaluate the DR.
Calculate FPR.
Compute Precision. /* *calculation of performance matrices* */
Calculate Accuracy.
Evaluate F-Score.
End.

measurement space. Here, optimal Bayesian filtering recursions cannot be solved exactly. Thus, these filters relies on the computation of expected values but due to the presence of non-linear transformed random vectors, the computation of expected values goes beyond the filtering context. In EKF, the requirement of linear equations for the measurement and state transition matrix is relaxed thus it transforms non-linearities into linear models so that Kalman Filter can be applied. The optimized features by EKF are then utilized by the SVM to detect the anomalies.

The ensembled anomaly detection algorithm is summarized in Algorithm 4. In the proposed algorithm, firstly, the sample dataset is input which is to be used to build the detection model. Then from line 3 to line 8, pre-handling of the dataset is done and the condition for normality is checked. The handled dataset is then splitted into training set and testing set (lines 10-13). In line 17, FKM algorithm is built according to Algorithm3.1. The training part of the dataset is then fed to FKM for the purpose of selecting the relevant features. After lowering the intrinsic dimensionality of datasets by FKM, nature of data objects are uncovered to find the best optimal solution. For this, EKF approach is embedded in the detection model (line 18). This phase is established by Algorithm 3.2. The resulting optimal solutions are then supplied to SVM in line 19 which performs the process of classification using Algorithm3.3. After the accomplishment of detection task, the counters for TP, TN, FP and FN are set (line 22) to obtain the detection results. Finally, after the determination of labels for both normal and anomalous events, performance measures for the proposed anomaly detection technique are computed from line 37 to line 41 of algorithm. The results obtained from the proposed approach are discussed as follows.

Complexity Analysis

Algorithm 3.4 illustrates the overall process of anomaly detection using FKM, EKM and SVM. The complexity in general can be expressed as follows: $O(n) + O(ntk^2i) + O(n^3) + O(n^2f)$. The same can be reduce to $O(ntk^2i + n^3)$.

3.2 Concluding Remarks

Ensemble based anomaly detection schemes, i.e, En-ADT is proposed for the detection of network anomalies. The proposed scheme is adduced specifically for increasing the efficiency of networks by securing them from various types of threats that do not comply with the network traffic. In the next chapter, another variant for network anomaly detection has been proposed.

Chapter 4

Fuzzified Cuckoo Based Clustering Technique

In this chapter, another anomaly detection technique, i.e., *Fuzzified Cuckoo based Clustering Technique (F-CBCT)*, is proposed. The detailed working of the proposed technique is mentioned as follows.

4.1 Working of F-CBCT

Figure 4.1 shows the step by step execution flow of F-CBCT. The proposed anomaly detection technique operates in two phases: training and detection. The training phase is supported using Decision Tree (DT) followed by an algorithm based on hybridization of Cuckoo Search Optimization (CSO) and K-means clustering. In the detection phase, a fuzzy decisive approach is used to detect anomalies on the basis of input data and distance functions computed in the previous phase. The detailed description of these phases is discussed as follows.

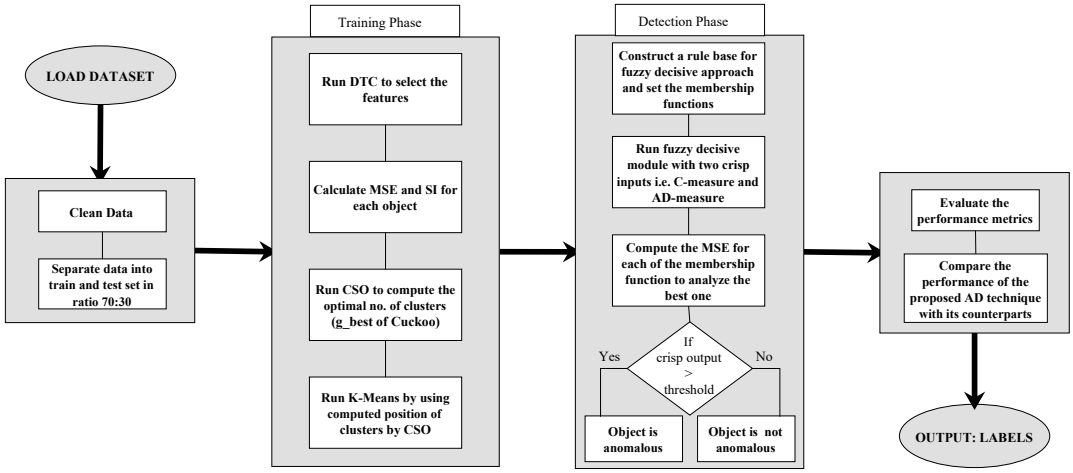


Figure 4.1: Framework of the proposed F-CBCT

4.1.1 Training Phase

The modules that are used to establish the training phase are discussed below:

Decision Tree Criterion (DTC):

The datasets with high dimensions always pose serious concerns to anomaly detection techniques. Due to large number of features, over-fitted results are generated which generally lead to performance degradation. Thus in this scheme, a post-pruning based DTC is used as a feature selection technique to fetch a subset of relevant features from the actual feature set of the dataset.

A decision tree is a classification technique that is used to solve the decision-based problems under uncertainty. It recursively partitions the attributes into classes-based on their respective features. Like any ordinary tree, decision tree consists of three types of nodes: Root Node, Internal Nodes and Leaf Nodes. Root node has no incoming edges, internal nodes have exactly one incoming edge, whereas leaf nodes are regarded as the decision nodes. Each internal node in the decision tree, splits the object space into multiple (two or more) subspaces according to a certain discrete function of the input attribute value [206].

The decision trees are known for estimating the capability of features for separability of objects under different classes. This characteristic of decision trees can be exploited for the purpose of feature selection. So, in F-CBCT, DTC is used to identify the best set for features in the dataset. The identified set of features are then provided as an input to the CSO algorithm, which thereby performs the task of feature optimization and evaluates the position of cluster centroids for clustering. Now, to identify the best set of features, DTC invokes the concept of entropy reduction and information gain. Entropy (E) measures the homogeneity of the attributes in the dataset, whereas Information Gain (IG) measures the degree of separation of training samples with respect to the splitting criterion.

Let S^T be the samples in the training dataset D^{Tr} and j be the total number of distinct classes in S^T. So, E(S^T) is calculated by Eq. (4.1) as:

$$E(S^T) = -\sum_{i=1}^{j}(P_i \log_2 P_i) \tag{4.1}$$

where, P_i is the proportion of S^T belonging to the i^{th} class. IG for all the attributes present in feature set F_T is calculated and the feature with maximum IG is selected as the root node of DT which is further used to partition the S^T. IG can be computed using the below mentioned equation.

$$IG(S^T, F) = E(S^T) - \sum_{k=1}^{m}\left(\frac{|S_k^T|}{S^T}E(S_k^T)\right) \tag{4.2}$$

where, k is the set of all possible values for feature F such that $F \in F_T$ and S_k^T is the subset of S^T for which the feature F has value k. This feature F is considered as the base for tree construction.

Algorithm 4.1 uses entropy-based *Information Gain* as a heuristic for the selection of attributes. This splitting criterion segregates the training dataset into classes. Here, IG of every individual attribute is determined and the attribute with the highest possible value is

selected as the split point. To evaluate the best split point, unique values of each attribute in the dataset are considered for the computation of IG.

Algorithm 4.1 Decision tree formation

Input: D^{Tr}: Training Dataset, n_f: Number of Attributes in F_T of D^{Tr}, Node N_L
Output: Decision Tree DT

Fetch samples S^T from D^{Tr}
if D^{Tr} is empty **then**
    return NULL
else
    Extract the feature set F_S from D^{Tr}
    if F_S is empty **then**
        return $N_{Root} \leftarrow N_L$
    else
        Initialize IG_{max}=0 $/*initializing\ maximum\ information\ gain\ IG_{max}*/$
        for l=1;l$\leq n_f$; l++ **do**
            Compute Information Gain IG_l using Eq. (4.2)
            if $IG_l \geq IG_{max}$ **then**
                $IG_{max} \leftarrow IG_l$
                N_P=l $/*node\ with\ maximum\ IG*/$
            end if
        end for
        Set Parent_Node: N_P
        Compute Entropy $E(D^{Tr})$ using Eq. (4.1)
        Split D^{Tr} on N_P using $E(D^{Tr})$
    end if
end if

The formulated Decision Tree (DT) has been applied to the current problem of interest to compute the priorities of the respective attributes. The attributes that show at the top three levels of the DT are the most relevant ones. It is hypothesized, that if the feature in the deeper levels is relevant enough, then it will automatically get raised up to any of the upper levels during the execution steps. The features that are less discriminating are discarded and union of the relevant features from each run is considered to get the best set of features. Algorithm 4.2 summaries the prioritization scheme of DT, using which the prediction of best set of features is performed. In this algorithm, a top down induction approach is followed to divide the current set of attributes in the dataset.

To further enhance the accuracy of the selected features, post pruning of DT is done before considering the top three levels as the best features. Pruning a DT is the process of replacing some of its subtrees by leaves. This algorithm uses the property of DT that the nodes closer to the root are more relevant. In order to overcome over-fitting, this approach shortens the tree by removing some nodes or some sub-trees from the original DT. Since the pruning technique works to reduce the overfitting of nodes in the tree, predictive accuracy of the features is enhanced using this technique.

• **Complexity Analysis:** Algorithm 4.1 describes the process of DT formation in order to determine the priorities of the respective attributes. It essentially comprises of the sorting process which is carried out for n_f number of optimized features. Therefore, the overall computational complexity is equivalent to $O(n_f nlogn)$; where the variable n denotes the number of data instances in D^{Tr}.

On the other hand, the overall complexity of Algorithm 4.2 can be expressed as $O(N(S^T) \times l_{max} + N(S^T) \times C_D \times FS(S^T_{best}))$. Here, the variables $N(S^T), l_{max}, C_D$ and $FS(S^T_{best})$ denote the number of nodes in S^T, number of nodes in S^T_{best}, number of classes in dataset D and number of features extracted from S^T_{best}, respectively.

Multi-objective Cuckoo-Search Optimization Algorithm (CSO):

Civicioglu and Besdok [212] did a conceptual comparison between Cuckoo-search and PSO which demonstrated that CSO algorithm provides more robust and precise results as compared to PSO with respect to performance and accuracy. Thus, in this research work, CSO algorithm is utilized in the training phase of F-CBCT instead of PSO to determine the optimal positions of cluster centroids for K-means algorithm.

Cuckoo Search is one of the nature inspired algorithms that was firstly introduced by Xin-She Yang and Suash Deb in 2009 [109]. It was stimulated by the procreation behavior

Algorithm 4.2 Post-Pruning of Decision Tree (DT)

Input: Built Decision Tree DT from Algorithm 4.1
Output: Selected features F_{sel} from the dataset D

Split DT in a sequence of sub-trees S^T where T=$\{1, 2, \cdots, n\}$
Determine $N(S^T)$ from S^T /* *number of nodes in* S^T */
Initialize E_{best}=0; /* *initializing entropy for best subtree* S^T_{best} */
for (t=1;t$\leq N(S^T)$; t++) **do**
    Compute overall Entropy for each S^T using Eq. (4.1)
    if $E(S^T_t) \geq E_{best}$ **then**
        $E_{best} \leftarrow E(S^T_t)$ /* *computing entropy of best subtree* S^T */
        $S^T_{best} \leftarrow S^T(E_{best})$ /* *determining the best subtree* S^T_{best} */
    end if
end for
$dep(S^T_{best})$=0 /* *initializing depth of subtree* S^T_{best} */
while $dep(S^T_{best}) \leq 3$ **do**
    $left(S^T_{best})$=0 /* *initializing depth of left of* S^T_{best} */
    $right(S^T_{best})$=0 /* *initializing depth of right of* S^T_{best} */
    Determine l_{max} from S^T_{best} /* *number of nodes in* S^T_{best} */
    for (j=0;j$\leq l_{max}$;j++) **do**
        if ($left(S^T_{best}) \neq$ NULL) **then**
            $left(S^T_{best})$+=1 /* *traversing left of subtree* S^T_{best} */
            $left(S^T_{best}$[j])=$S^T_{best}[j]$ /* *extracting features from left of* S^T_{best} */
        else if ($right(S^T_{best}) \neq$ NULL) **then**
            $right(S^T_{best})$+=1 /* *traversing right of subtree* S^T_{best} */
            $right(S^T_{best}$[j])=$S^T_{best}[j]$ /* *extracting features from right of* S^T_{best} */
        end if
    end for
    $dep(S^T_{best})$+=1 /* *traversing subtree* S^T_{best} */
    $FS(S^T_{best})$=$left(S^T_{best})$+$right(S^T_{best})$ /* *fetching features from complete* S^T_{best} */
    for each class C_D in dataset D **do**
        for each feature in $FS(S^T_{best})$ **do**
            if $FS(S^T_{best}) \in C_D$ **then**
                $FS(S^T_{best}) \rightarrow C_D$ /* *fetching features for each individual class* C_D */
            end if
        end for
    end for
    $F_{sel} \leftarrow FS(S^T_{best})$
end while
return F_{sel} /* *returning selected features from DT* */

(to lay their eggs in the nest of the host birds) of certain species of cuckoos present in nature. It is a population-based stochastic global search algorithm that optimizes a problem by

initializing a search space where a pattern corresponds to a "nest" and Cuckoo's individual attribute is referred as "Cuckoo-egg". Each egg present in the nest indicates a present solution so a Cuckoo search algorithm considers egg laid by Cukoo as a new solution. The aim is to exploit the new and probably better solutions (cuckoos) to substitute an old and not so-good solution in the nests. For this, Cuckoo-Search algorithm makes three assumptions: (i) each cuckoo can lay one egg at a time and choose a random nest in the environment to dump it, (ii) the nests that are having a good quality eggs (solutions) will only be carried over to the next generation, (iii) the availability of host nests are fixed and the probability by which a host bird can identify cuckoo egg is $p_e \in \{0,1\}$. In such cases, decision is made by the host bird, i.e., either to throw out a cuckoo egg from the nest or to leave that nest and build a new one. A fresh solution n_c^{t+1} for a cuckoo c is generated by performing a Levy flight as:

$$n_c^{t+1} = n_c^t + \alpha \oplus \lambda(Levy) \tag{4.3}$$

Eq. (4.3) represents the random walk stochastic equation, where $\alpha>0$ denotes the step size and it may be selected according to the characteristics of the problem. It is usually assumed to be one. The symbol $\oplus$ in Eq. (4.3) represents (Exclusive OR) entry wise multiplications. Further, Levy Flight is a random walk having the random step size. Here, the random steps of Levy flight are inferred from the Levy distribution as shown in Eq. (4.4).

$$Levy \sim u = t^{-\lambda}, (1<\lambda \leq 3) \tag{4.4}$$

Here, the random steps of cuckoos will form a random walk process. Random walk process of CSO is almost similar to other optimization algorithms like Ant Colony Optimization, PSO, etc. The only difference is that, in CSO, Levy Flight is used to perform random walks. Due to the provision of random step sizes, random walks performed via Levy flight becomes

much more competent in probing the search space.

Algorithm 4.3 Multi-objective CSO algorithm

Input: n_host_nests: Number of Solutions, nd: Size of the F_{sel} determined by DTC, toll: Tolerance of Solution,i_max= Maximum number of iterations

Output: Optimal positions of cluster centroids (n_cen)

Define parameter constraints, lb: lower bound of position vector of cuckoo and ub: upper bound of position vector of cuckoo, toll: absolute error rate
cuckoo.position: a n_cen×nd matrix with lower (lb) and upper (ub) parameter constraints
cuckoo.cost: Compute the Silhouette for each cuckoo on the basis of generated cuckoo.position from Eq. (4.14)
cuckoo.sol:[] (sol is a fitness value of two simultaneous objective functions: SI and MSE)
cuckoo.global_best.cost=[] (global best of cost)
cuckoo.global_best.sol=[] (global best of sol)
while t<toll **do**
    Get a Cuckoo (c) randomly by Levy Flight using Eq. (4.3)
    Choose best nest, n_b=1,2,....,n_host_nests by using the steps shown below:
    Evaluate Objective Functions, MSE by Eq. (4.5) and SI by Eq. (4.15)
    Generate new Cuckoo position using SI and MSE (but keep the current best)
    Update cuckoo's current best:
    if (cuckoo(c).cost==cuckoo(c).best.cost) && (cuckoo(c).sol.MSE<cuckoo(c).best.sol.MSE) **then**
        cuckoo(c).best.position=cuckoo(c).position /∗ *position of best cuckoo* ∗/
        cuckoo(c).best.sol=cuckoo(c).sol
    else if cuckoo(c).cost<cuckoo(c).best.cost **then**
        cuckoo(c).best.position=cuckoo(c).position
        cuckoo(c).best.sol=cuckoo(c).sol /∗ *current best of solution* ∗/
        cuckoo(c).best.cost=cuckoo(c).cost /∗ *current best of cost* ∗/
    end if
    Update global best:
    if ((cuckoo(c).best.cost==cuckoo.global_best.cost) && (cuckoo(c).best.sol.MSE< cuckoo.global_best.sol.MSE)) OR (cuckoo(c).best.cost < cuckoo.global_best.cost) **then**
        cuckoo.global_best=cuckoo(c).best
    end if
    if i> i_max **then**
        EXIT
    else
        i=i+1
    end if
end while
return n_cen /∗ *optimal positions of cluster centroids* ∗/

Petrovic [213] has shown that the approach using Silhouette Index (SI) produces more accurate results as compared to the approach that uses DBI. This is due to the fact that SI takes into account both cohesion (closeness of objects within a cluster) and separation criteria (inter-cluster diversity). Thus, to evaluate the optimal number of clusters, SI and Mean Square Error (MSE) are employed as two validation criterion.

Given a set of data points $D_i = \{d_1, \cdots, d_N\}$, the set of cluster centroids $C_j = \{c_1, \cdots, c_K\}$, the average pairwise distance between data points and their corresponding centroids, MSE, is calculated as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N} D_{dc}^2 \qquad (4.5)$$
$$\text{where } D_{dc} = d(d_i, c_j)$$

here, D_{dc} is the distance between data points and their corresponding centroids.
In the current partitioning process, MSE is used to compute the intra-cluster diversity whereas SI is employed to evaluate the optimal number of clusters. To evaluate the closeness of one data object with respect to its neighboring objects, the value of SI varies in the range of [-1,1]. Here, negative value indicates an undesirable case, i.e., the allotment of data objects to a wrong cluster, 0 indicates that the sample data objects is on or very near to the boundary between neighboring clusters whereas, positive value indicates the data object is far away from the neighboring clusters.

The average distance $dist_i^x$ between the i^{th} data object in the cluster C_x and other data objects is given by Eq. (4.6).

$$dist_i^x = \frac{1}{N_x - 1}\sum_{l=1}^{N_x} dist(d_i^x, d_k^x),\ C_x = \{d_1^x, \cdots, d_N^x\} \qquad (4.6)$$

where, N_x is the number of points in the cluster C_x. Symbols d_i^x and d_k^x are the i^{th} and k^{th}

data-objects in cluster C_x respectively. Now, $dist_i^x$ can be expressed in terms of MSE as follows:

$$dist_i^x = MSE(1)_i^x \text{ (Using Eq. (4.5))} \tag{4.7}$$

where, $N = N_x - 1$, i=1 to N_x and $D_{dc} = dist(d_i^x, d_k^x)$

The minimum average distance ($madist_i^x$) from i^{th} data object in C_x to all other clustered data objects D_i in C_j, $j = \{1, \cdots, K\}$ is given by

$$madist_i^x = \min_{j=1,\cdots,K;\ j\neq x} \left\{ \frac{1}{N_D} \sum_{m=1}^{N_D} dist(d_i^x, d_m^j) \right\} \tag{4.8}$$

where, N_D is the number of data objects in the dataset D and d_m^j is the set of data objects D_i present in clusters C_j. On the similar lines, $dist_i^x$ and $madist_i^x$ can also be expressed in terms of MSE as:

$$madist_i^x = MSE(2)_i^x \text{ (Using Eq. (4.5))} \tag{4.9}$$

here, $N = N_D$, i=1 to N_D and $D_{dc} = dist(d_i^x, d_m^j)$

Using Eq. (4.6) and Eq. (4.8), the Silhouette width of the i^{th} data object in the cluster C_x is defined as:

$$s_i^x = \frac{madist_i^x - dist_i^x}{\max\{dist_i^x, madist_i^x\}} \tag{4.10}$$

From Eq. (4.7) and Eq. (4.9), Silhouette width, s_i^x can also be expressed in terms of MSE which is as follows:

$$s_i^x = \frac{MSE(2)_i^x - MSE(1)_i^x}{\max\{MSE(1)_i^x, MSE(2)_i^x\}} \tag{4.11}$$

The value of s_i lies between [-1,1] such that $-1 \leq s_i^x \leq 1$ and to verify the cluster goodness, diversity within the cluster is computed by Eq. (4.11). Further, Silhouette of the cluster C_x is

defined as:

$$S_x = \frac{1}{N_x} \sum_{i=1}^{N_x} s_i^x \tag{4.12}$$

and global Silhouette of the clustering is defined by

$$S = \frac{1}{K} \sum_{j=1}^{K} S_j \tag{4.13}$$

In the above equation, S_j is the Silhouette of the cluster C_j and is calculated by using Eq. (4.12). To calculate the Silhouette of the cluster in terms of diversity of the clusters, the highest value of silhouette width from Eq. (4.11) is substituted to Eq. (4.13):

$$S = \frac{1}{K} \sum_{j=1}^{K} diversity_j \tag{4.14}$$

The obtained optimal centroid positions from Algorithm 4.3 of CSO is then supplied to K-means clustering algorithm to perform the clustering process.

• **Complexity Analysis:** The complexity of Multi-objective CSO as detailed in Algorithm 4.3 is expressed as $O(\text{number of solutions} \times \text{number of iterations})$, *i.e.*, $O(ni)$.

K-Means Clustering Algorithm:

The K-means algorithm is one of the simplest partitioning algorithm that is used to solve the clustering problems. It partitions a set of data objects into predefined number of clusters such that the similarity between intra cluster data objects are more than that of inter cluster data objects. For this, it minimizes the sum of squared distances within a cluster. The most commonly used distance measure to compute similarity is Euclidean distance. The squared Euclidean distance from each data object to each cluster is computed to allocate the data objects to their closest clusters. Eq. (4.15) shows the evaluation of Euclidean distance

between point p_i and cluster c_j with N data objects in space:

$$Distance(d,c) = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{K}(d_i - c_j)^2} \tag{4.15}$$

All the objects $p_i \in$ N, where i=1,2,$\cdots$,N are arranged into clusters $c_j \in$ K, where j=1,2,$\cdots$,K by the standard process of K-means clustering algorithm as illustrated in Algorithm 4.4 which operates in two phases: initialization phase and iterative. The first phase selects K most centrally located objects as initial centroids using Algorithm 4.3. In the second phase, each data object of cluster is examined to determine those data objects whose average dissimilarity to all other objects is minimal. This is done by computing Euclidean distance using Eq. (4.15). Here, the relationship between p_i and c_j is crisp which means that an object p_i can lie only in a single cluster. The algorithm iterates its working until there is no change in the values of centroids.

Algorithm 4.4 K-Means clustering algorithm

Input: n_{cen}: Position of cluster centroids from Cuckoo-Search Algorithm (Algorithm 4.3), thresh: threshold value for number of iterations

Output: Clustered data objects C_O

1: Initialize K pseudo-centroids by n_{cen}
2: Compute the Euclidean distance metric between data objects and pseudo cluster centroids using Eq. (4.15)
3: Assign each data object to the cluster with nearest centroid
4: Recompute the new cluster centroid vector as, $v_j = \frac{1}{d_j}\sum_{j=1}^{d_j} p_j$, where v_j is the centroid vector of cluster j, d_j represents the total number of data objects in the j^{th} cluster and p_j denotes the j^{th} data object vector
5: **while** change in centroids **do**
6: **if** m_iter<thresh **then**
7: goto Step 2
8: **else**
9: exit
10: **end if**
11: **end while**
12: **return** C_O

In F-CBCT, K-means clustering algorithm receives optimal position values of cluster centroids from CSO to perform clustering.

• **Complexity Analysis:** The complexity of K-means clustering algorithm is dependent on three parameters, i.e., n, k and t; which denote the number of objects, number of clusters and number of iterations, respectively. The complexity is thus represented as $O(nkt)$.

Classification and Anomaly Detection:

After the optimal placement of cluster centroids by CSO algorithm, the clustered data objects from K-means clustering technique are deployed for anomaly detection in testing data. In this phase, two distance-based measures, i.e., Classification measure (C-Measure) and Anomaly Detection measure (AD-Measure) are utilized to solve the problem of current interest. Anomaly detection can be modeled into a distance function $F : d_i \longrightarrow d_o$. Here, the function F depends upon the distance between data object d_i and other data objects d_o in the dataset D.

Definition 1: (C-Measure)
Given a dataset $D_i \in D$; $i = \{1, \cdots, d\}$, a set of labels $L_e \in L$; $e = \{1, \cdots, l\}$, a classification is defined as a task of providing labels L_e to each data object D_i using a learnt classification model that is trained a-priori with the help of labelled data objects.

In F-CBCT, C-Measure is employed to detect the normal or anomalous instances of known types whose characteristics match with the instances of those present in its training set. To detect the anomalous traffic using this measure, Weighted Euclidean Distance (WED) is used as a linkage criteria and purity function (PF) is used as an external clustering evaluation

criteria. The below mentioned equation expresses the WED function.

$$C-Measure = WED = dist(t^e, t^r) = \sqrt{\sum_{i=1}^{n} w_i^{tr}(t_i^e - t_i^r)^2} \tag{4.16}$$

where, t_i^e is the i^{th} data object of testing dataset, t_i^r is the i^{th} data object of training dataset and w_i^{tr} is the value corresponding to the weight of the cluster in which t_i^r is contained such that $0 \leq w_i^{tr} \leq 1$ and $\sum_{i=1}^{n} w_i^{tr} = 1$. The value of weights are assigned to every generated cluster, and is computed by purity function as:

$$PF = \frac{1}{N^{tr}} \sum_{j=1}^{k} \max N_j^{tr}(c) \; where \; 1 \leq c \leq C \tag{4.17}$$

where, N^{tr} is the number of data objects in training dataset. On the other hand, $N_j^{tr}(c)$ depicts the number of training instances in cluster j of class $c \in C$. Here, if all the data instances present in cluster j are of class c, then the value of purity function will be set to 1 and the cluster is called pure cluster.

Once C-measure for each testing data instance is computed, classification of data objects is done. The object which is closer to the normal cluster is classified as normal and the data object which is closer to the anomalous cluster is classified as anomalous.

Definition 2: (AD-Measure)

Given a training dataset $D^{Tr} \in D$ and a testing dataset $D^{Te} \in D$, an anomaly in testing dataset is defined as the object $D_i^{Te} \in D^{Te}$ that differs significantly from most of the objects contained in D^{Te} or the object $D_i^{Te} \in D^{Te}$ that lies close to anomalous cluster boundary of D^{Tr} is considered as anomalous.

This measure helps in detection of anomalies that did not appear in the training phase. Here, Chebyshev distance (CD) measure is employed as a distortion measure because this measure

uses maximum value distance approach to compute the distance. In this approach, distance between two vectors is computed by examining the absolute magnitude of their differences along any coordinate of objects. It is computed as follows:

$$CD(d_i, c_j) = max|d_i^s - c_j^s| \tag{4.18}$$

where, d_i is the data object, c_j is the centroid point of normal cluster and d_i^s and c_j^s are the standard coordinates.

As compared to the C-Measure, AD-Measure does not make use of the anomalous cluster centroids for computation. Thus, the goodness value (GV) of the normal cluster GV_{norm} is used to measure this metric. It is computed using Eq. (4.17). The value of goodness is influenced by the value of purity in such a way that higher is the purity of the cluster, higher is the goodness value and vice-versa. The distance to the normal cluster is computed using AD measure as follows:

$$AD\ Measure = GV_{norm} * CD \tag{4.19}$$

Thus, in the proposed F-CBCT approach, C-measure and AD-measure are combined to overcome the inherent limitations of each measure. C-measure can detect only the known types of anomalies that are present in its knowledge base, whereas AD-measure does not make use of the anomalous cluster centroid to make the detections. Thus, a hybrid of both measures is considered such that the object that lies closer to the anomalous cluster boundary compared to the normal one is considered as anomalous. Further, if the distance of the data object to the normal cluster is greater than CD then also it is considered as anomalous.

The computed distance measures from Algorithm 4.5 are thus utilized by the detection phase to measure the level of anomalousness.

• **Complexity Analysis:** Algorithm 4.5 illustrates the detailed steps of computing the values

Algorithm 4.5 Computation of C-Measure and AD-Measure

Input: C_O: Clusters formed by k-means algorithm, C_K: Number of clusters generated by k-means algorithm, D^{Tr}: Training dataset, D^{Te}: Testing dataset
Output: C-Measure, AD-Measure

for j=1;j$\leq$ K;j++ **do**
    Compute PF[j] using Eq. (4.18) $/*computing\ purity\ function*/$
end for
P_{SORTED}=Sort($C_O[j]$,PF,asc) $/*sorting\ clusters\ according\ to\ purity\ function*/$
N $\leftarrow$ P_{SORTED}[1,$\cdots$,K]
for l=1;l$\leq$ K;l++ **do**
    while $P_{SORTED} \geq 1$ **do**
        w_l^{tr}:Set(w[l],des) $/*assigning\ weights*/$
        GV_l: Set(GV[l],$cluster_{sim}$) $/*computing\ goodness\ value\ GV*/$
    end while
end for
Determine N^{te} from D^{Te} $/*determining\ number\ of\ training\ instances*/$
while i$\leq$ N^{te} **do**
    $D_i^{Te}sim \leftarrow$ WED (D[i], closest.cluster.center[D^{Tr}[i]])
    Compute C-Measure using Eq. (4.16) $/*computing\ classification\ measure*/$
    Compute AD- Measure using Eq. (4.19) $/*computing\ anomaly\ detection\ measure*/$
    i=i+1
    return C-Measure and AD-Measure
end while

of C-Measure. The related complexity is essentially $O(K+Kt+N^{te})$; which can be reduced to $O(Kt)$; wherein K denotes the number of clusters and t represents the number of iterations, respectively. The variable N^{te} denotes the number of training instances.

4.1.2 Detection Phase

The fuzzy decisive module is used to establish the detection phase of F-CBCT. The detailed explanation of this phase is as below:

Fuzzy Detection Phase:

For detecting the potential anomalies in dataset, F-CBCT employees MATLAB fuzzy logic toolbox that consists of modules mentioned in Table 4.1.

The fuzzy detection module consists of four phases: fuzzification phase, rule-base construction, fuzzy inference engine and defuzzification phase. In fuzzification phase, crisp set of inputs are transformed into non-crisp (fuzzy) sets. To achieve this, membership functions are used to associate each object of dataset with some grade of membership. In rule-base construction phase, fuzzy rules are defined by using IF-THEN relationship. Further, in phase 3, an inference is built that uses fuzzy rules to construct a mapping from input to fuzzy output which is further defuzzified into crisp set of outputs by using membership functions in the defuzzification phase.

If the crisp output generated by this approach exceeds the threshold value, ε, then it will be considered anomalous otherwise it will be considered a normal instance.

Table 4.1: Components of fuzzy inference system

1. Fuzzy Inference System (FIS) Type: Mamdani
2. Fuzzy set of inputs (Num_Inputs)=2; In_Labels: C-Measure and AD-Measure
3. C-Measure membership: Very Near, Near, Average, Far, Very Far
4. AD-Measure membership: Near, Average, Far
5. Fuzzy output (Num_Outputs)=1; Out_Label: Alert
6. Alert membership: Normal, Low Vulnerable, High Vulnerable, Abnormal
7. Num_Rules: 15 (refer Table 4.2 and Table 4.3)
8. Mem_Functions: (refer Table 4.4)
9. Fuzzy_Oper: Fuzzy set operations: min for AND and Implication both; max for OR and Aggregation both
10. Defuzz_Method: Centroid of Gravity $(COG) = \frac{\int_{min}^{max} z\mu(z)dz}{\int_{min}^{max} \mu(z)dz}$ where z is the output variable, μ denotes the membership function, min and max are the minimum and maximum limits for defuzzification respectively.

The rule-matrix and sample rule-set (considering all the types of alerts) for F-CBCT are provided in Table 4.2 and Table 4.3 respectively.

Further, five membership functions including Triangular shaped (trimf), Trapezoidal (trapmf), Gaussian (gaussmf), Generalized bell shaped (gbellmf) and Product of two sigmoidal are employed to find the most suitable one. Table 4.4 illustrates the same. The

mapping from C-measure and AD-measure to the Alert for all the pre-discussed membership functions is represented by the 3-D curves, i.e., surface viewers in Figure 4.2. A trimf rule viewer for fuzzy inference system of F-CBCT is shown in Figure 4.3. This rule viewer projects a roadmap of the complete fuzzy inference process. The first two columns of the rule viewer display the membership functions referenced by C-measure and AD-measure respectively (i.e., the IF part of the rules) and the third column shows the membership functions referenced by the Alert (i.e., the THEN part of the rules). The aggregated decision inferred from the rules is shown by the sixteenth plot of the third column (i.e., Alert) and the bold red line on this plot signifies the defuzzified output.

Table 4.2: Rule-matrix for the proposed fuzzy system

AD-Measure	C-Measure				
	Very Near	Near	Average	Far	Very Far
Near	Normal	Normal	Normal	LV	LV
Average	LV	LV	HV	HV	HV
Far	HV	HV	Anomalous	Anomalous	Anomalous

LW: Low Vulnerable; HV: High Vulnerable

Table 4.3: Rule-set for the proposed fuzzy system

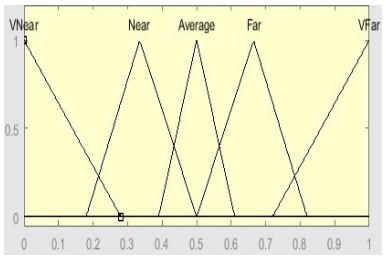
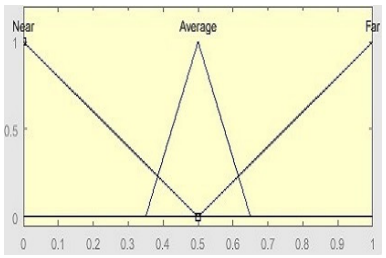
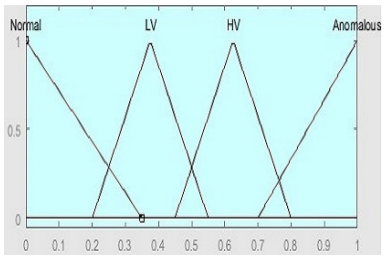
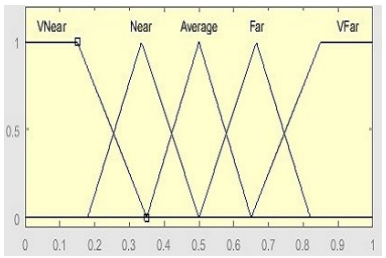
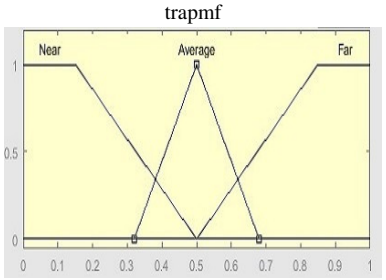
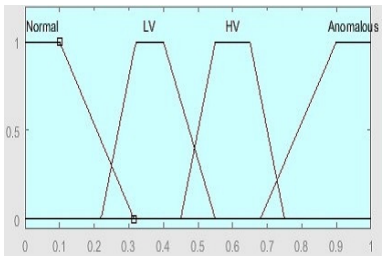
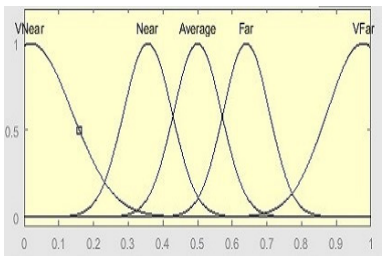
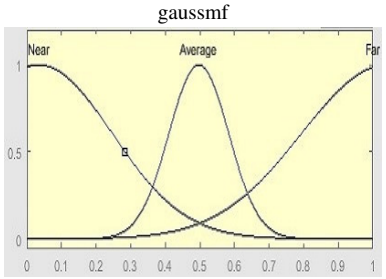
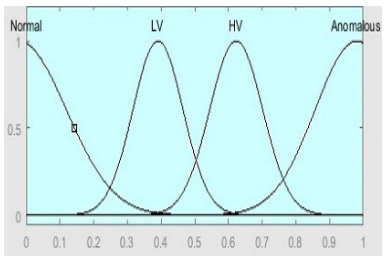
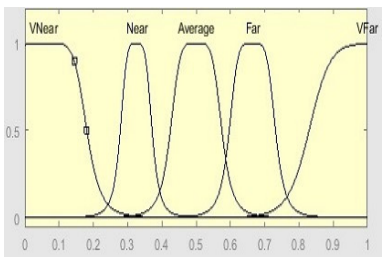
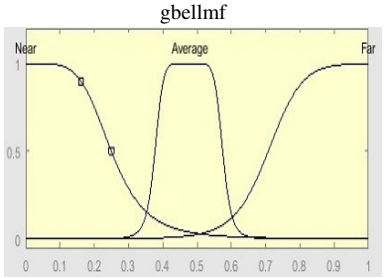
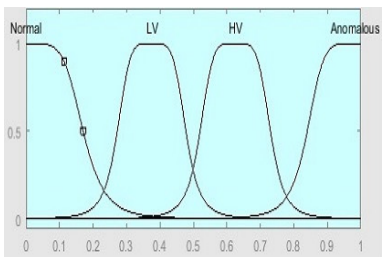
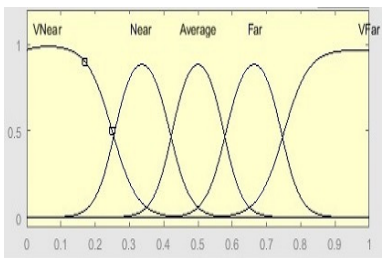
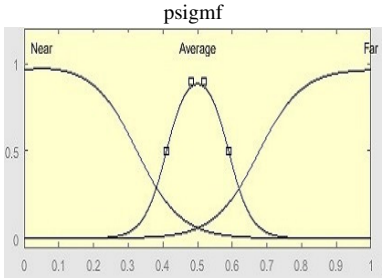
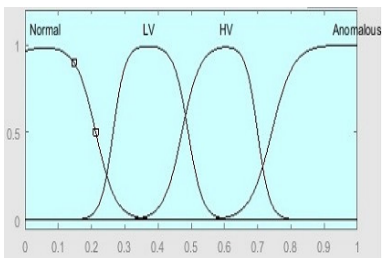
R1: if C-Measure ← Average **AND** AD-Measure ← Near **THEN** Alert ← Normal

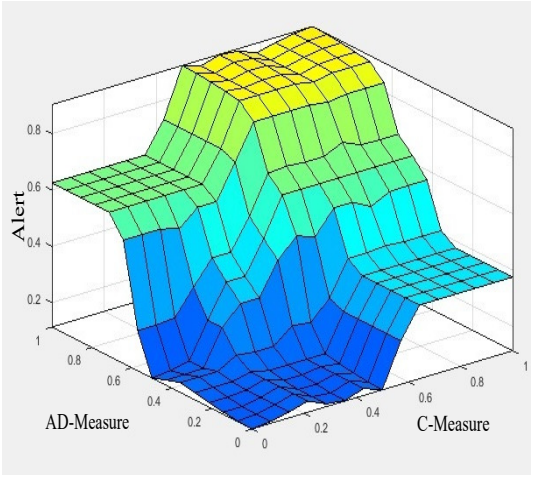
R2: if C-Measure ← Near **AND** AD-Measure ← Average **THEN** Alert ← Low Vulnerable

R3: if C-Measure ← Very Near **AND** AD-Measure ← Far **THEN** Alert ← High Vulnerable

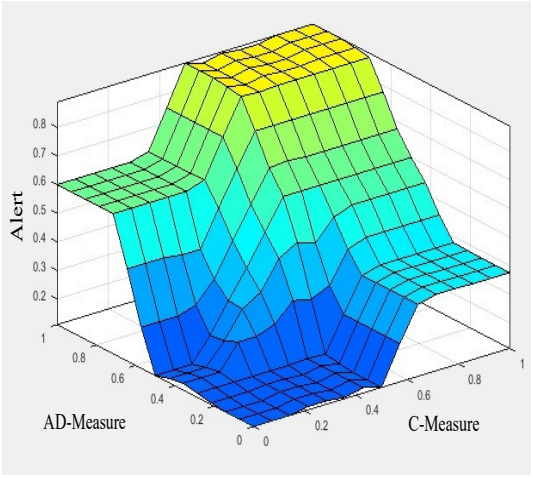
R4: if C-Measure ← Average **AND** AD-Measure ← Far **THEN** Alert ← Anomalous

Table 4.4: Membership functions with two inputs & one output

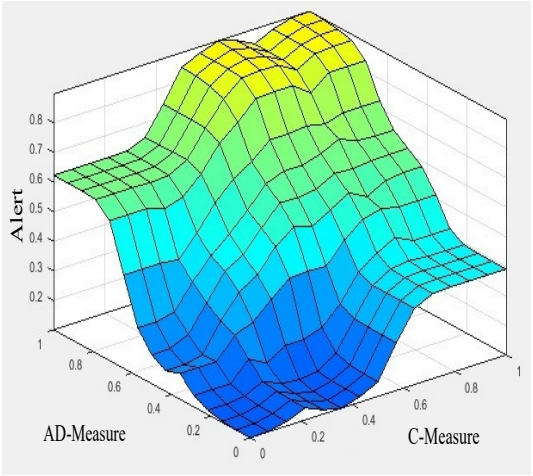
Input		Output
C-Measure	AD-Measure	Alert
	trimf	
		
	trapmf	
		
	gaussmf	
		
	gbellmf	
		
	psigmf	
		



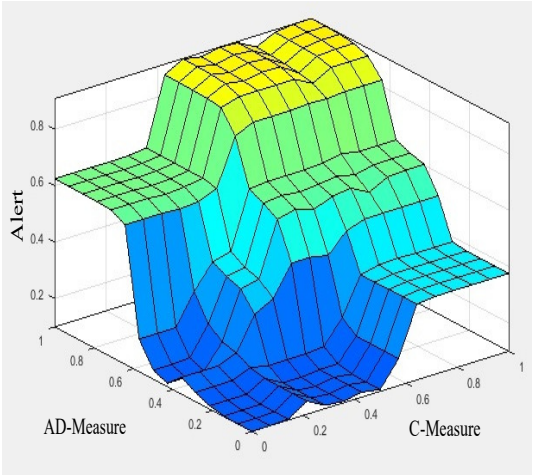
(a) trimf.



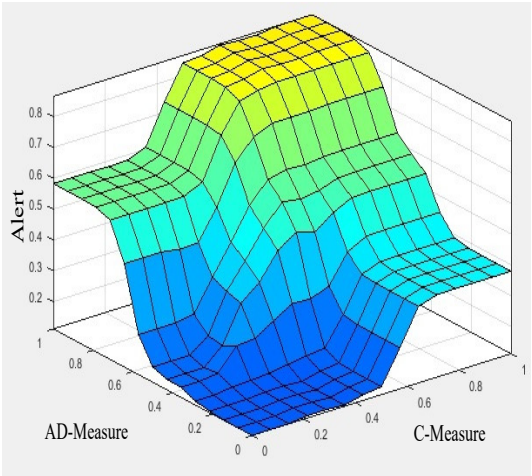
(b) trapmf.



(c) gaussmf.



(d) gbellmf.



(e) psigmf.

Figure 4.2: Surface plots for different membership functions

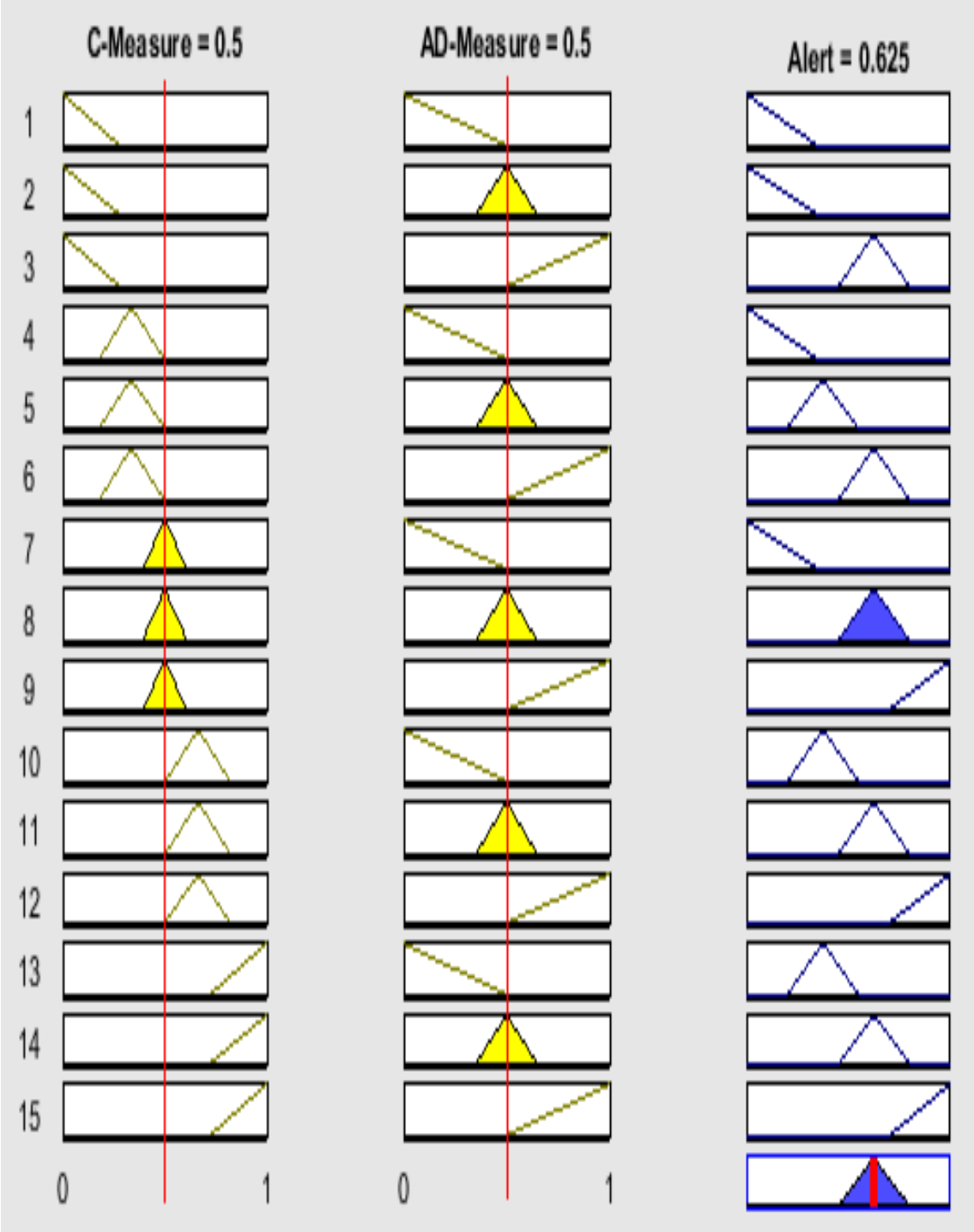


Figure 4.3: Trimf rule viewer for fuzzy inference system (FIS) of F-CBCT

4.2 Concluding Remarks

A hybrid anomaly detection scheme, F-CBCT, is designed which operates in two phases: training and detection. In the first phase-the training phase, DTC, CSO and K-means clustering algorithms are combined to bifurcate the irrelevant features and obtain optimal clustering results. In the second phase (detection phase), a fuzzy decisive approach is employed to infer the labels corresponding to normal and anomalous classes. In the next chapter, experimental evaluation of the proposed schemes has been presented.

Chapter 5

Experiments and Implementation Details

This chapter discusses the simulation setup employed to validate the performances of the proposed anomaly detection schemes on different datasets. Along with this, the results obtained are also illustrated in this chapter.

5.1 En-ADT

All the scripting for validating En-ADT has been done in Python language. Further, RBF kernel has been applied in SVM to achieve the boundary function. The performance matrices and datasets that were used to identify the capability of the proposed technique are also illustrated in this section.

5.1.1 Datasets

In this section the robustness and accuracy of the proposed En-ADT are validated. For this research work, two benchmark datasets namely-DARPA 1998 and KDD CUP 1999 were selected from MIT Lincoln Library and UCI machine learning repository respectively. Table 5.1 describes these datasets in detail. DARPA'98 dataset consists of 5 weeks of Basic Se-

curity Module (BSM) assessment of all processes when run on a Solaris machine. KDD'99 dataset is selected from UCI whereas DARPA 1998 dataset is selected from MIT Lincoln library. This dataset incorporates a huge range of intrusions simulated in a military networked environment.

Table 5.1: Summary of datasets

Dataset	Instances (approx. 10%)	Features	Attribute Type	Label	# Attacks
DARPA'98	5,00,002	22	float	4	57
KDD CUP'99	4,94,021	41	float	4	22

Both the datasets have normal and anomalous behavioral patterns corresponding to the network traffic so individually both of these datasets are employed to train and evaluate the performance of the proposed anomaly detection technique. Experiments are conducted by taking 70% of dataset as the training data and the remaining part as testing data. As DARPA dataset is a collection of tcp_dump data which consists of packets passing from the network, it is almost impossible to distinguish attacks from normal packets just by analyzing this tcp_dump data. This is a reason that all the attacks that are present in this dataset are considered as a single class and the whole problem of anomaly detection in DARPA'98 dataset is considered as a binary-class classification problem. On the contrary, the presence of multiple classes of attacks in KDD'99 dataset makes the problem of anomaly detection in KDD'99 as multi-class classification problem. The performance metrics such as-accuracy, rate of detection, false positive rate and F-Score are calculated for each class of dataset.

5.1.2 Performance Metrics

For evaluating the performance of the proposed technique, performance metrics such as-DR, FPR, Accuracy and F-Score are considered. DR corresponds to the number of anomalies detected by the anomaly detection technique and is computed by (Eq. 5.1), FPR refers to

the number of normal instances inaccurately classified as anomalies and is computed by Eq. (5.2). Accuracy is the number of instances that are accurately classified as anomalies. It is the most intuitive performance measure and is computed by Eq. (5.4). The F-score signifies the weighted mean of precision and recall; this performance measure is highly useful for classes that have an uneven distribution. It is computed by Eq. (5.5). These are discussed below:

$$DR(Recall) = \frac{TruePositive}{TruePositive + FalseNegative} \tag{5.1}$$

$$FPR = \frac{FalsePositive}{FalsePositive + TrueNegative} \tag{5.2}$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{5.3}$$

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative} \tag{5.4}$$

$$F - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5.5}$$

where true positive (TP) indicates the number of anomalies correctly diagnosed as anomalies whereas true negative (TN) denotes the number of normal instances correctly diagnosed. Similarly, False positive (FP) indicates the number of normal instances incorrectly diagnosed as anomalous and false negative (FN) indicates the number of anomalies incorrectly diagnosed as normal.

Binary-class classification problem:

For DARPA'98 dataset, the problem at hand is treated as binary-class classification problem and labels corresponding to both the classes of dataset, i.e., normal class and anomalous class are detected. Fig. 5.1 shows the performance of the proposed technique on DARPA'98 dataset with respect to DR, FPR, accuracy and F-score.

The results of binary class classification shows that the proposed ensemble based tech-

nique achieves very high DR reaching more than 95% for normal class and more than 98% for anomalous class with a very low FPR, i.e., less than 2% for normal class and less than 4% for anomalous class. It can also be noticed that the proposed technique achieves high accuracy (>95%) and F-score values (>97%) for both the classes.

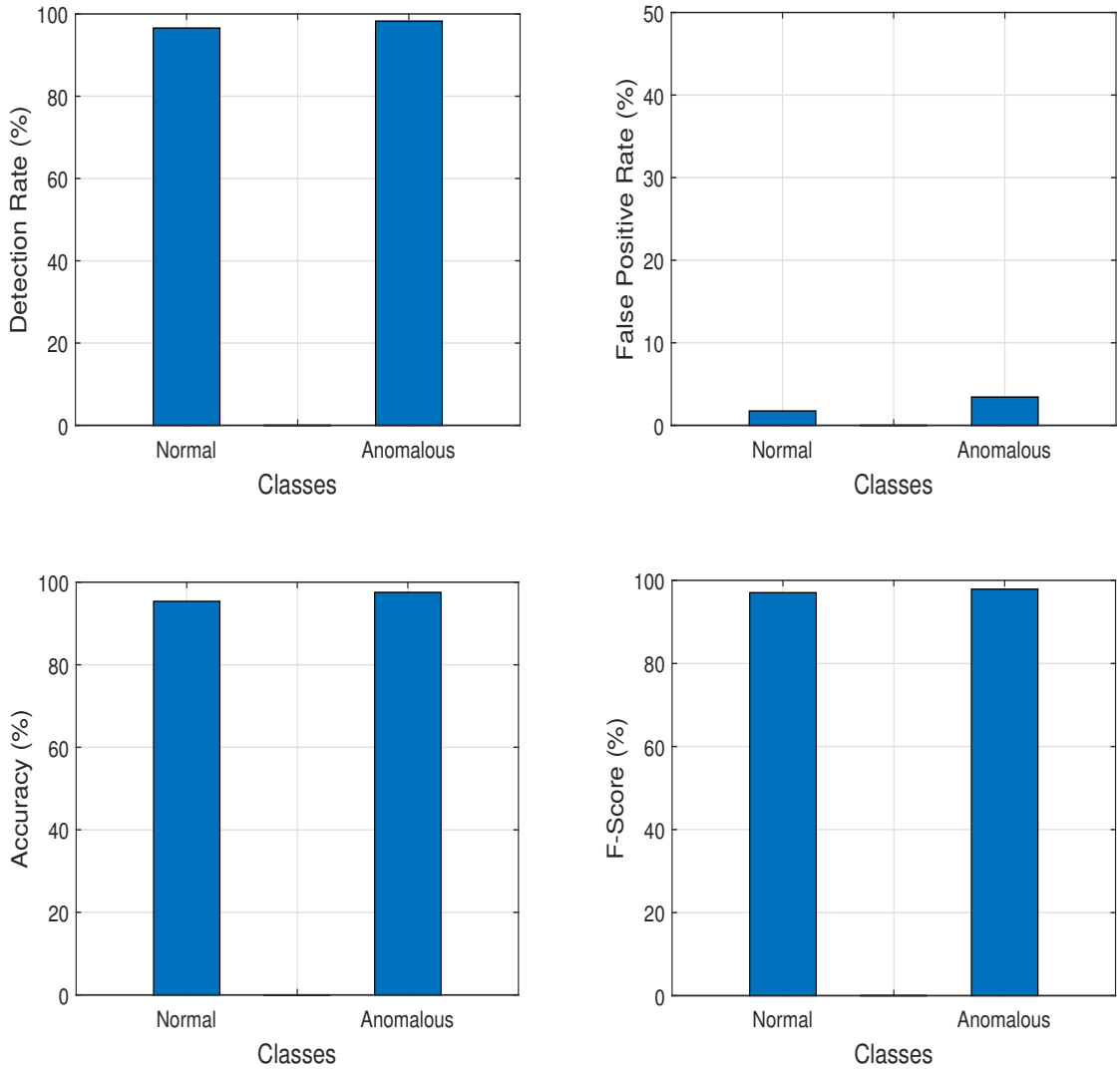


Figure 5.1: Evaluation of proposed technique on DARPA'98 dataset

Multi-class classification problem:

For KDD'99 dataset, the problem at hand is treated as multi-class classification problem and labels corresponding to each class of dataset, i.e., normal class, DOS, U2R, R2L and

Probe are detected. Attacks corresponding to all the classes of this dataset are categorized in Table 5.2. Fig. 5.2 depicts the performance of the proposed technique on KDD'99 dataset corresponding to DR, FPR, accuracy and F-score.

The results of multi class classification shows that the proposed ensembled technique achieves very high DR reaching more than 95% for normal, DOS, R2L and Probe class except U2R class but its is not a major problem as it is a low frequent occurring attack and the proposed algorithm is performing quite well in terms other high frequently occurring attacks. Further, it was able to achieve a low FPR, i.e., less than 6% for normal class as well as less than 2% for all other classes. It can also be noticed that the proposed technique achieves high accuracy, i.e., more than 96% for all classes which reaches more than 99% for U2R class. The proposed approach also achieves high F-score values, i.e., more than 90% for normal, DOS and R2L class, approximately 85% for Probe class and more than 70 % for U2R class.

Results in Figs. 5.1 and 5.2 shows that the proposed anomaly detection technique achieves high accuracy rate for almost all the classes indicating efficiency of the proposed algorithm in terms of detection accuracy. Similarly, results obtained by computing DR, FPR and F-score are also found to be quite convincing.

Table 5.2: Summary of anomalous classes in KDD'99 Dataset

Class	No. of Attacks	No. of Instances	Attacks
dos	6	3,91,458	back, smurf, land, pod, teardrop, neptune
u2r	4	1126	rootkit, buffer_overflow, perl, load_module
r2l	8	52	warezclient, ftp_write, multihop, guess_passwd, spy, imap, phf, warezmaster
probe	4	4107	satan, nmap, ipsweep, portsweep

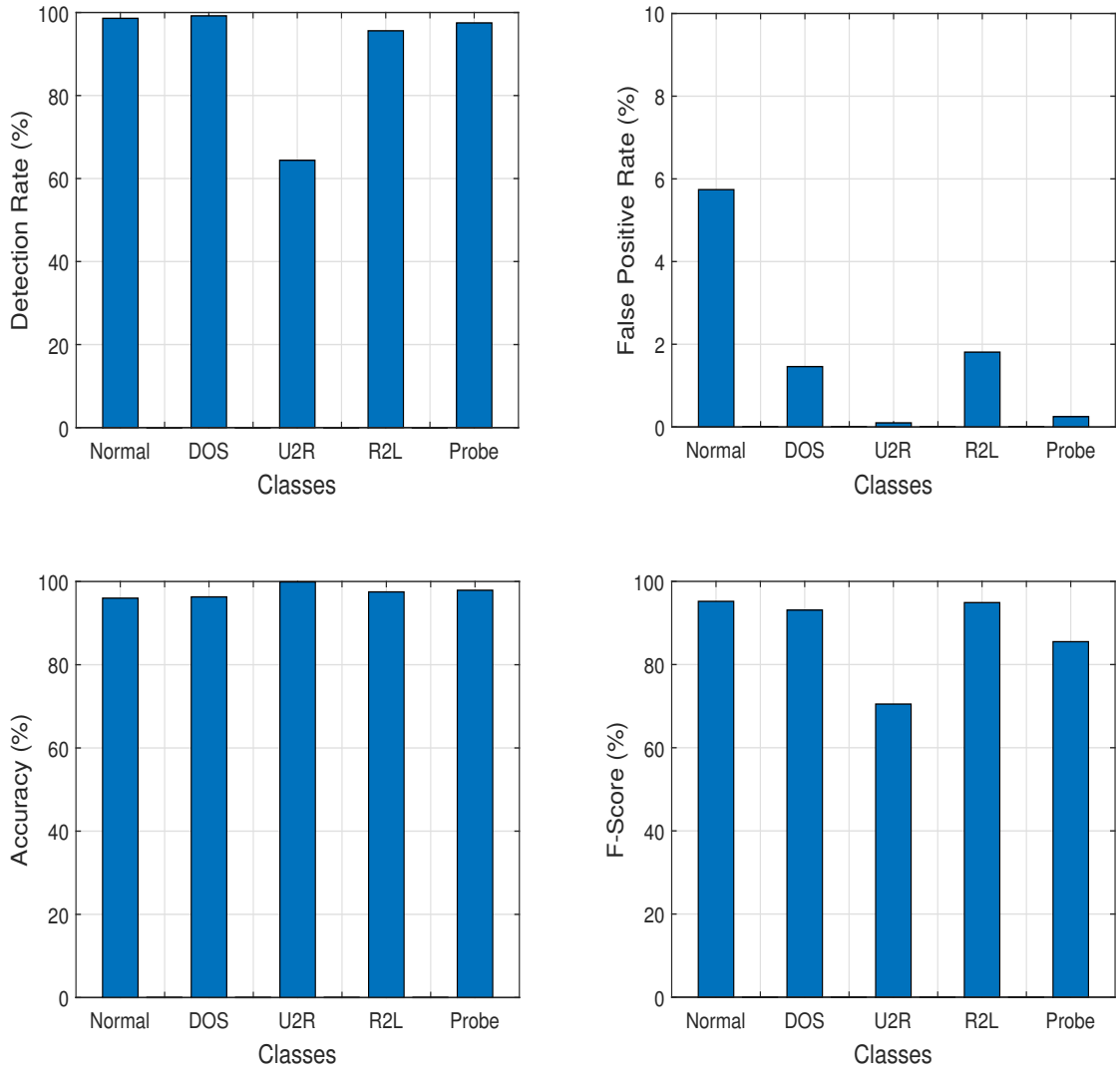


Figure 5.2: Evaluation of proposed technique on KDD'99 dataset

5.1.3 Comparison of the Proposed Technique with its counterparts

In this section, the proposed ensemble based technique is evaluated against the existing techniques. Table 5.3 demonstrates the comparison results of [16,40,214–217] for the DARPA'98 dataset and Table 5.4 demonstrates the comparison results of [145,218–229] for the KDD'99 dataset. In addition to the comparison result tables, graph demonstrating the performance of the proposed technique over wider spectrum is included. Fig. 5.3 shows the overall performance of the proposed technique on DARPA'98 and KDD'99 dataset under different evalu-

ation criterion. In both the datasets, the DR of the proposed approach is quite high (>96%) whereas the FPR is quite low (<3%). Similarly, the values of accuracy, precision and F-score are quite high as it crosses more than 92% rate.

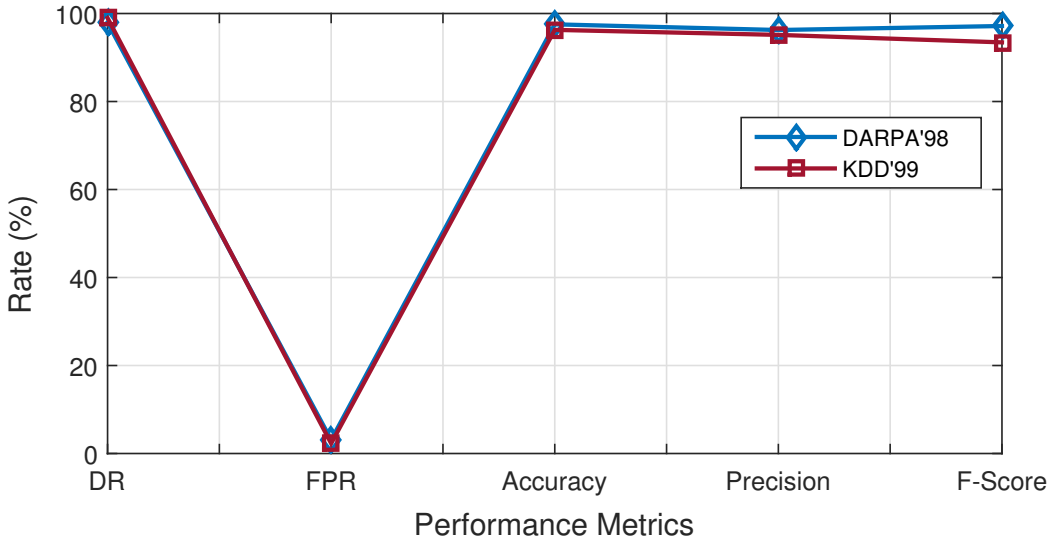


Figure 5.3: Performance evaluation of En-ADT on DARPA'98 and KDD'99 dataset

Although some of the techniques mentioned in the literature have DR and FPR better than the proposed technique but the accuracy and F-score of the proposed technique outperform all the existing anomaly detection techniques referred for comparison in Table 5.3.

Likewise in Table 5.2, although DR and FPR of some techniques are better than the proposed technique but if we consider all four parameters, i.e., DR, FPR, Accuracy and F-Score, the proposed technique provides efficient results for all the above mentioned parameters. Thus the comparison results shows in Table 5.3 and 5.2 proves the effectiveness of the proposed technique in terms of anomaly detection.

Table 5.3: Comparison of existing anomaly detection techniques on DARPA'98 dataset

Technique	DR (in %)	FPR (in %)	Accuracy (in %)	F-Score	Additional Parameters (if any)
Hu *et al.* (2003) [214]	81.8	<3	–	–	–
Zhanchun *et al.* (2006) [215]	92.2	2.8	–	–	–
Catania *et al.* (2012) [216]	92.5	5.0	–	–	–
Elfeshawy *et al.* (2013) [217]	98.43	4.6	95.39	–	Error Rate=1.56%
Breier *et al.* (2015) [40]	80.29	0.077	–	–	Error Rate = 0.098
Ahmed *et al.* (2016) [16]	99.23	–	92.82	96	Precision=92.36%, Attack Cluster Purity=92.36% and Normal Cluster Purity=95.6%
Proposed Technique	97.91	3.09	97.54	97.16	Precision=96.25%

DR, detection rate; FPR, false positive rate.

Table 5.4: Comparison of existing anomaly detection techniques on KDD'99 dataset

Technique	DR (in %)	FPR (in %)	Accuracy (in %)	F-Score	Additional Parameters (if any)
Kuang *et al.* (2007) [218]	94.6	3.0	95.1	–	–
Hwang *et al.* (2007) [219]	94.71	3.8	93.52	–	–
Zhang *et al.* (2008) [220]	94.7	2.0	–	–	–
Mukherjee *et al.* (2012) [221]	90.92	–	97.78	–	Root Mean Squared Error=0.083
Sharma *et al.* (2012) [145]	93.41	0.275	99.05	93	Precision=93.17%
Panda *et al.* (2012) [222]	99.5	0.1	–	99.7	Precision=99.9% and Root Mean Squared Error=0.045
Bhat *et al.* (2013) [223]	99.1	2.0	99	99	Precision=99%
Elbasiony *et al.* (2013) [224]	98	1.5	–	–	–
Nadiammai *et al.* (2014) [225]	96.86	0.5	98.12	–	Specificity(TNR)=92.36 %
Mohammadi *et al.* (2014) [226]	98	3.0	–	–	–
Ghanem *et al.* (2015) [227]	–	0.033	96.1	–	–
Duque *et al.* (2015) [228]	70.75	0.74	–	–	FNR=99.82%
Pandeeswari *et al.* (2015) [229]	98	3.05	–	83.20	Precision=85.22%
Proposed Technique	98.98	2.28	96.28	93.44	Precision=95.14%

DR, detection rate; FPR, false positive rate.

5.2 F-CBCT

In this section, the experimental evaluation for F-CBCT is reported. All the scripting for evaluation purpose is done in MATLAB.

In order to evaluate the performance of F-CBCT, firstly, datasets that are employed in the current context of research are discussed. Secondly, evaluation criterion which is adopted to access the performance of the proposed technique is explored. Finally, the results obtained for F-CBCT are compared with other existing techniques.

5.2.1 Datasets

Six benchmark datasets from UCI-ML repository have been employed to evaluate the effectiveness of F-CBCT. This includes Zoo, Diabetes, Iris, Vehicle, Wine and Glass [192]. Their detailed descriptions are provided in Table 5.5.

Table 5.5: Characteristics of datasets from UCI ML repository

S.N.	Dataset	#Features	# Classes	#Instances
1	Zoo	18	7	101
2	Diabetes	8	2	768
3	Iris	4	3	150
4	Vehicle	18	4	946
5	Wine	13	3	178
6	Glass	10	6	214

NSL-KDD Dataset:

In addition to the UCI ML datasets, NSL-KDD dataset [230] is also utilized as another set of experiment for evaluating the performance of F-CBCT. This dataset is the modified version of the most widely used intrusion detection dataset, i.e., KDD Cup'99. This dataset consists of 41 features along with 23 attack types which are classified into 4 different classes: Denial

of Service (DoS), Probe, User to Root (U2R) and Remote to Local (R2L). The characteristics of this dataset are provided in Table 5.6.

Table 5.6: NSL-KDD dataset description

Class	Training Set	Attacks	
Normal	67,343	Type	Total
DoS	45,927	back, teardrop, pod, neptune, smurf, land	6
Probe	11,656	back, ipsweep, portsweep, nmap, satan	5
U2R	995	buffer_overflow, rootkit, perl, loadmodule	4
R2L	52	ftp_write, imap, spy, guess_passwd, warezclient, phf, warezmaster, multihop	8
Total	1,25,973		23

All the features present in this dataset may not have the ability to reflect the anomalous behavior of all classes. Thus, it is required to choose relevant set of features from individual classes to enhance the capability of detection mechanism. In F-CBCT, post-pruning based DTC is applied to NSL-KDD dataset to select the relevant features corresponding to each class. The results for this feature selection process is provided in Table 5.7 and Table 5.8.

Table 5.7: Description of selected features from NSL-KDD dataset using decision tree

S.N.	Feature	Type	Selected For	S.N.	Feature	Type	Selected For
1	duration	continuous	U2R	22	is_guest_login	continuous	Probe, R2L
2	flag	symbolic	Normal, Probe, R2L	23	serror_rate	continuous	DoS
3	service	symbolic	Normal, R2L, U2R	24	srv_count	continuous	DoS, R2L, U2R
4	num_access_files	continuous	–	25	count	continuous	DoS, Probe, U2R
5	src_bytes	continuous	Normal, DoS, Probe, U2R	26	num_failed_logins	continuous	Probe
6	protocol_type	symbolic	DoS	27	same_srv_rate	continuous	Normal, Probe, U2R
7	hot	continuous	–	28	su_attempted	continuous	–
8	dst_host_serror_rate	continuous	Normal, Probe	29	rerror_rate	continuous	–
9	logged_in	continuous	Normal	30	diff_srv_rate	continuous	Normal, Probe
10	land	continuous	U2R	31	dst_host_srv_count	continuous	Normal, Probe, R2L, U2R
11	srv_serror_rate	continuous	–	32	num_outbound_cmds	continuous	U2R
12	urgent	continuous	Normal, DoS	33	srv_diff_host_rate	continuous	–
13	num_compromised	continuous	–	34	dst_host_same_src_port_rate	continuous	DoS, Probe, U2R
14	num_file_creations	continuous	–	35	dst_host_rerror_rate	continuous	Probe
15	srv_rerror_rate	continuous	–	36	dst_host_same_srv_rate	continuous	Normal, DoS
16	num_root	continuous	U2R	37	dst_host_srv_serror_rate	continuous	DoS, R2L
17	root_shell	continuous	U2R	38	wrong_fragment	continuous	U2R
18	dst_host_count	continuous	–	39	dst_host_srv_diff_host_rate	continuous	R2L, U2R
19	dst_bytes	continuous	Normal, DoS, Probe, R2L	40	dst_host_srv_rerror_rate	continuous	–
20	num_shells	continuous	–	41	dst_host_diff_srv_rate	continuous	Normal, Probe, R2L
21	is_host_login	continuous	U2R				

Table 5.8: Number of selected features for all classes from NSL-KDD dataset

	Classes				
	Normal	DoS	Probe	U2R	R2L
#Features Selected	12	10	13	15	9

5.2.2 Performance Evaluation Metrics

To compute the performance of F-CBCT, following metrics are employed.

Varying C-Measure and AD-Measure:

The values of two distance functions, i.e., C-measure and AD-measures are varied to compute the effect of change on the membership functions. A specific set of levels in the range of [0,1], i.e., $\{0, 0.25, 0.5, 0.75, 1\}$ are used to identify an ideal membership function. Post-computation, it has been concluded that Triangular membership function (trimf) gives the optimum results (highest average alert) for the current problem as evident from Table 5.9.

Root Mean Square Error:

It is quite difficult to identify the membership function that best suits the problem apriori thus, several membership functions (as shown in Table 4.4) are utilized to compute the efficiency of F-CBCT. The RMSE measure for different techniques is computed by employing seven different datasets as illustrated in Table 5.10. This is evident from the results, that in most of the cases, F-CBCT performs quite well as compared to other techniques. Particularly, the results obtained for '$trimf$' are quite convincing as shown in Figure 5.4.

Other Metrics:

For evaluating the performance of F-CBCT, several performance metrics such as-DR, FPR, Accuracy and F-measure are considered. Results are reported by applying these metrics to aforementioned datasets. Table 5.11 shows the corresponding results using K-means, Decision Tree, PSO, CSO (MSE), CSO (MSE, SI), FCOAC [231], SADA [232], SSAD [233], TVCPSO [234] and F-CBCT. Further, it can be noticed from Figure 5.5 that TVCPSO gives comparable performance in most of the cases but FPR of the proposed F-CBCT is quite less

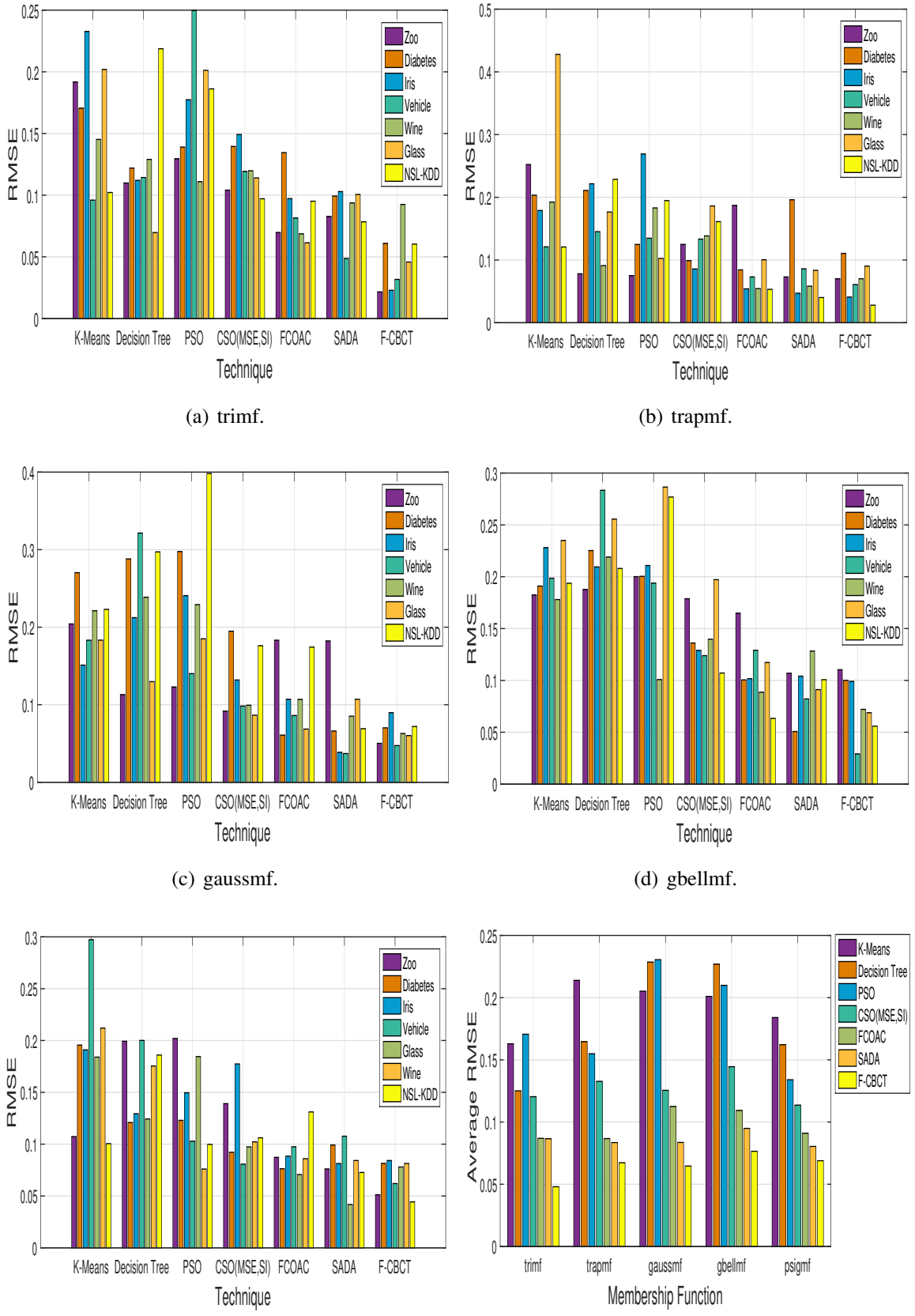
Table 5.9: Evaluation of membership functions for anomaly detection

Parameters		Alert				
C-Measure	AD-Measure	trimf	trapmf	gaussmf	gbellmf	psigmf
0.00	0.00	0.113	0.110	0.101	0.099	0.112
0.00	0.25	0.133	0.121	0.126	0.120	0.120
0.00	0.50	0.113	0.110	0.198	0.132	0.161
0.00	0.75	0.625	0.600	0.608	0.622	0.576
0.00	1.00	0.625	0.600	0.623	0.624	0.583
0.25	0.00	0.137	0.130	0.137	0.164	0.127
0.25	0.25	0.137	0.130	0.162	0.173	0.135
0.25	0.50	0.332	0.249	0.374	0.367	0.271
0.25	1.00	0.625	0.600	0.623	0.622	0.579
0.50	0.00	0.113	0.110	0.150	0.106	0.137
0.50	0.25	0.133	0.121	0.189	0.130	0.148
0.50	0.50	0.625	0.600	0.573	0.601	0.564
0.50	0.75	0.886	0.878	0.800	0.892	0.830
0.50	1.00	0.903	0.888	0.848	0.903	0.841
0.75	0.00	0.375	0.379	0.390	0.376	0.380
0.75	0.25	0.375	0.379	0.411	0.377	0.385
0.75	0.50	0.625	0.600	0.610	0.603	0.585
0.75	0.75	0.883	0.868	0.836	0.849	0.841
0.75	1.00	0.883	0.868	0.857	0.872	0.848
1.00	0.00	0.375	0.375	0.391	0.375	0.379
1.00	0.25	0.375	0.377	0.403	0.376	0.383
1.00	0.50	0.625	0.600	0.616	0.617	0.587
1.00	0.75	0.886	0.878	0.863	0.883	0.854
1.00	1.00	0.903	0.888	0.897	0.909	0.866
Average Alert		0.4918	0.4774	0.4910	0.4913	0.4705

(in the considered datasets) as compared to the proposed one. As noticed from the results, the combination of MSE and SI considerably enhances the performance when integrated with CSO. Also, the support of DTC and fuzzy decisive module with the cuckoo-based clustering took the performance of F-CBCT to an upper level with DR=96.86%, FPR=1.297%, Accuracy=97.77%, and F-measure=98.30%. This proves that the proposed F-CBCT technique is efficient in terms of anomaly detection.

Table 5.10: Comparison of RMSE for different membership functions

Dataset	Membership Functions				
	trimf	trapmf	gaussmf	gbellmf	psigmf
Technique: K-Means					
Zoo	0.1917	0.2523	0.2042	0.1824	0.1073
Diabetes	0.1705	0.2034	0.2703	0.1910	0.1956
Iris	0.2327	0.1790	0.1511	0.2280	0.1910
Vehicle	0.0960	0.1211	0.1832	0.1986	0.2973
Wine	0.1452	0.1924	0.2212	0.1780	0.1840
Glass	0.2019	0.4280	0.1833	0.2350	0.2120
NSL-KDD	0.1022	0.1208	0.2230	0.1937	0.1005
Technique: Decision Tree					
Zoo	0.1099	0.0781	0.1131	0.1877	0.1993
Diabetes	0.1220	0.2110	0.2881	0.2253	0.1209
Iris	0.1120	0.2216	0.2123	0.2095	0.1294
Vehicle	0.1143	0.1452	0.3214	0.2834	0.2001
Wine	0.1290	0.0912	0.2387	0.2190	0.1243
Glass	0.0697	0.1766	0.1298	0.2556	0.1754
NSL-KDD	0.2187	0.2287	0.2971	0.2080	0.1860
Technique: PSO					
Zoo	0.1295	0.0752	0.1229	0.2001	0.2020
Diabetes	0.1390	0.1249	0.2976	0.2004	0.1230
Iris	0.1774	0.2692	0.2407	0.2108	0.1496
Vehicle	0.2497	0.1348	0.1401	0.1939	0.1029
Wine	0.1109	0.1831	0.2291	0.1008	0.1845
Glass	0.2012	0.1026	0.1850	0.2865	0.0760
NSL-KDD	0.1862	0.1948	0.3981	0.2769	0.0998
Technique: CSO(MSE,SI)					
Zoo	0.1041	0.1250	0.0918	0.1788	0.1391
Diabetes	0.1395	0.0991	0.1948	0.1361	0.0922
Iris	0.1492	0.0858	0.1319	0.1289	0.1774
Vehicle	0.1192	0.1334	0.0983	0.1239	0.0807
Wine	0.1198	0.1385	0.0993	0.1398	0.0974
Glass	0.1140	0.1862	0.0865	0.1972	0.1022
NSL-KDD	0.0971	0.1612	0.1761	0.1071	0.1061
Technique: FCOAC [231]					
Zoo	0.0698	0.1872	0.1832	0.1648	0.0873
Diabetes	0.1345	0.0843	0.0608	0.1005	0.0763
Iris	0.0972	0.0541	0.1071	0.1017	0.0884
Vehicle	0.0815	0.0731	0.0861	0.1291	0.0975
Wine	0.0686	0.0545	0.1070	0.0886	0.0707
Glass	0.0614	0.1004	0.0686	0.1173	0.0860
NSL-KDD	0.0951	0.0533	0.1744	0.0634	0.1311
Technique: SADA [232]					
Zoo	0.0827	0.0731	0.1823	0.1070	0.0761
Diabetes	0.0993	0.1961	0.0661	0.0507	0.0992
Iris	0.1029	0.0471	0.0387	0.1041	0.0813
Vehicle	0.0487	0.0860	0.0370	0.0821	0.1077
Wine	0.0937	0.0583	0.0852	0.1283	0.0418
Glass	0.1006	0.0836	0.1071	0.0911	0.0842
NSL-KDD	0.0784	0.0402	0.0691	0.1007	0.0727
Technique: F-CBCT					
Zoo	0.0216	0.0702	0.0502	0.1103	0.0512
Diabetes	0.0609	0.1106	0.0702	0.1001	0.0815
Iris	0.0898	0.0410	0.0229	0.0991	0.0842
Vehicle	0.0317	0.0609	0.0473	0.0291	0.0620
Wine	0.0925	0.0702	0.0629	0.0721	0.0780
Glass	0.0458	0.0903	0.0602	0.0688	0.0814
NSL-KDD	0.0603	0.0280	0.0719	0.0559	0.0443



(a) trimf. (b) trapmf.

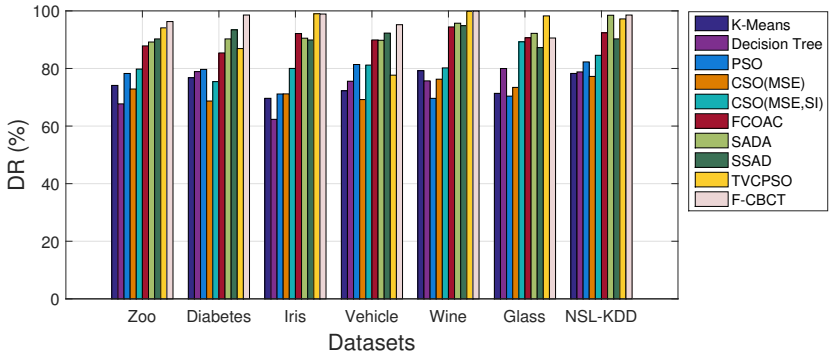
(c) gaussmf. (d) gbellmf.

(e) psigmf. (f) Computation of best MF.

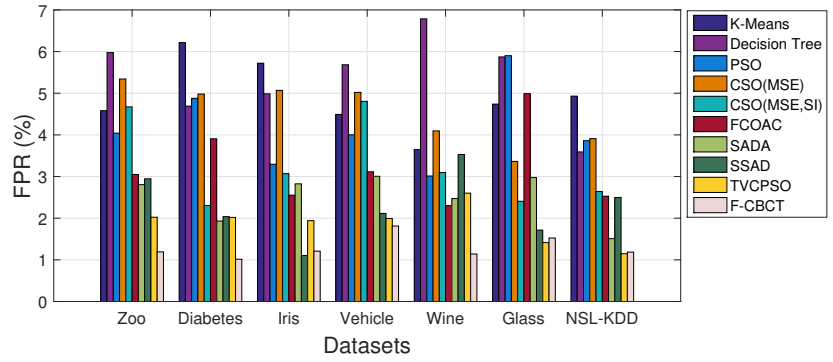
Figure 5.4: Performance evaluation of the membership function in terms of RMSE

Table 5.11: Comparison of the proposed technique with its variants

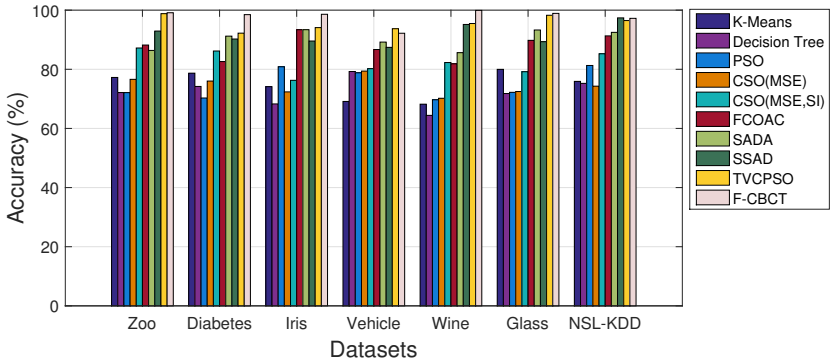
Metric	Datasets						
	Zoo	Diabetes	Iris	Vehicle	Wine	Glass	NSL-KDD
Technique: K-means							
DR	74.12	76.78	69.64	72.25	79.23	71.36	78.28
FPR	4.582	6.216	5.721	4.492	3.647	4.739	4.931
Accuracy	77.24	78.69	74.12	69.13	68.20	79.98	75.91
F-Measure	78.51	75.45	76.29	70.28	69.95	79.32	78.49
Technique: Decision Tree							
DR	67.71	78.90	62.34	75.57	75.70	79.98	78.81
FPR	5.976	4.690	4.987	5.684	6.786	5.872	3.592
Accuracy	72.13	74.20	68.29	79.24	64.42	71.80	75.24
F-Measure	73.18	76.82	67.24	80.28	68.81	82.96	75.70
Technique: PSO							
DR	78.27	79.68	71.14	81.38	69.63	70.40	82.28
FPR	4.042	4.878	3.294	4.003	3.013	5.902	3.861
Accuracy	72.11	70.29	80.86	78.82	69.70	72.21	81.29
F-Measure	70.28	72.66	87.20	77.19	70.76	74.11	84.10
Technique: CSO(MSE)							
DR	72.87	68.70	71.18	69.20	76.28	73.43	77.25
FPR	5.342	4.980	5.068	5.020	4.096	3.364	3.910
Accuracy	76.60	76.02	72.36	79.39	70.20	72.48	74.29
F-Measure	82.24	79.10	71.28	80.28	76.69	81.07	73.54
Technique: CSO(MSE,SI)							
DR	79.80	75.43	80.02	81.18	80.20	89.26	84.58
FPR	4.672	2.304	3.070	4.805	3.096	2.406	2.640
Accuracy	87.20	86.18	76.27	80.21	82.28	79.18	85.25
F-Measure	88.02	87.93	80.27	84.24	78.97	84.03	88.11
Technique: FCOAC [231]							
DR	87.84	85.40	92.12	89.91	94.43	90.66	92.38
FPR	3.051	3.907	2.553	3.115	2.303	4.990	2.528
Accuracy	88.20	82.61	93.38	86.69	81.85	89.79	91.30
F-Measure	87.59	84.18	83.92	88.53	86.12	83.27	93.04
Technique: SADA [232]							
DR	89.19	90.27	90.52	89.80	95.72	92.20	98.49
FPR	2.806	1.932	2.825	3.006	2.471	2.975	1.509
Accuracy	86.38	91.21	93.39	89.20	85.62	93.29	92.49
F-Measure	86.64	89.28	95.42	89.55	94.85	95.50	95.62
Technique: SSAD [233]							
DR	90.27	93.42	89.90	92.29	94.86	87.25	90.28
FPR	2.946	2.038	1.105	2.114	3.530	1.714	2.497
Accuracy	92.93	90.24	89.56	87.42	95.16	89.33	97.39
F-Measure	92.97	93.05	90.27	86.31	96.29	91.82	97.90
Technique: TVCPSO [234]							
DR	94.11	86.91	99.00	77.68	99.86	98.26	97.21
FPR	2.023	2.019	1.941	1.993	2.601	1.414	1.148
Accuracy	98.81	92.22	94.09	93.72	95.48	98.27	96.48
F-Measure	98.92	95.38	91.30	91.15	97.82	98.74	97.24
Technique: F-CBCT							
DR	96.29	98.56	98.91	95.20	99.92	90.58	98.56
FPR	1.192	1.015	1.209	1.812	1.141	1.524	1.186
Accuracy	99.12	98.48	98.60	92.19	99.92	98.91	97.23
F-Measure	98.83	99.27	97.90	94.99	98.59	99.53	99.05



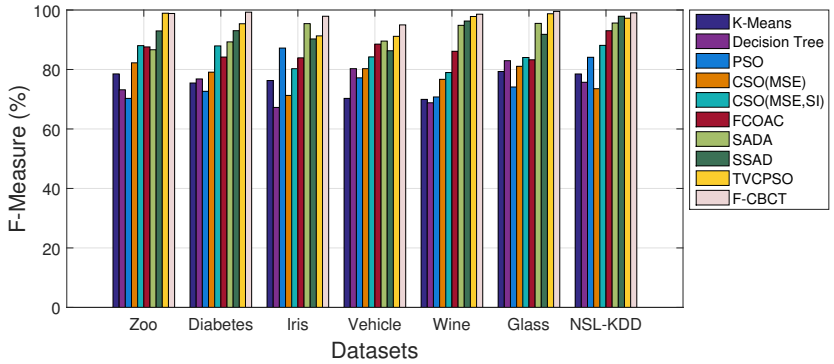
(a) Detection Rate (DR).



(b) False Positive Rate (FPR).



(c) Accuracy.



(d) F-Measure.

Figure 5.5: Performance evaluation of F-CBCT

5.3 Concluding Remarks

The results achieved on various benchmark datasets have shown major performance improvements in terms of network anomaly detection. Further, the comparison with the existing anomaly detection frameworks on benchmark parameters like DR, FPR, accuracy and F-measure assures the efficacy of the proposed anomaly detection schemes, i.e., *En-ADT* and *F-CBCT*. The next chapter concludes the work carried out in this thesis along with future directions.

Chapter 6

Conclusion and Future Scope

Rapidly increasing data and evolving Internet have raised the significance of analyzing the network traffic efficiently. Thus, different network anomaly detection models have been proposed in this thesis. The proposed models demonstrated promising results which authorize their future investigations and incorporation in real-time intrusion detection systems involving big data. The following segment sheds some lights on the fundamental thesis contributions and the conclusions drawn from the proposed work.

6.1 Thesis Contributions

The objective of this thesis is to study and design effective network anomaly detection schemes in Big Data. This segment outlines the major contributions of this thesis.

- In this research work, the state-of-the-art techniques for network anomaly detection have been reviewed in detail.
  - In particular, the literature review presents the detailed discussion about dimensionality reduction, optimization strategies, machine learning and deep learning approaches in this direction.

  - The thesis also summarizes different anomaly detection techniques on various application domains; which in turn assisted in understanding why these techniques obtain a certain level of classification trade-off.

- In this thesis, curse-of-dimensionality problem has been identified as a significant challenge to anomaly detection; which has empirically shown poor detection rates with respect to certain anomalous classes. Thus, this research work demonstrates that FKM and post-pruning DT are capable of learning from imbalanced data and therefore, selecting more relevant features.

- The research work demonstrated in this thesis proposed new methods that take into account both selection of relevant features and classification. This trade-off has not been considered in the current classifier combinations. Offering a different approach to training, this work proposed two different anomaly detection techniques for the networks transmitting Big data:

  - The first technique blends FKM clustering algorithm, EKF and SVM together for mitigating the problem of classifying data according to normal and anomalous behaviors in a networked environment. The proposed technique is referred to as *En-ADT*. In the proposed technique, FKM, a variant of K-means clustering algorithm is used for feature selection. The hybrid algorithm then optimizes the membership functions of the fuzzy clusters with a non-linear Bayesian approach known as EKF. Further, SVM is employed that utilizes the selected features generated in previous step for the detection of network anomalies.

  - The second technique, *i.e.*, *F-CBCT*, operates in two phases: training and detection. In the training phase, Cuckoo-Search optimization (CSO) algorithm, Decision Tree Criterion (DTC) and K-Means Clustering are cascaded to compute

two simultaneous distance functions, *i.e.*, Classification measure (C-measure) and Anomaly detection measure (AD-measure). For the computation of these measures two internal evaluation criterion are employed, *i.e.*, Mean Square Error (MSE) and Silhouette Index (SI). In the detection phase, the previously computed distance functions are employed by the fuzzy detection module to classify the instances into normal and anomalous classes.

- The proposed anomaly detection schemes have been implemented using MATLAB and Python with major performance improvements compared to the other schemes.
  - *En-ADT* has been evaluated using standard network traffic datasets, *i.e.*, DARPA 1998 and KDD Cup 1999 dataset on the basis of accuracy, DR, FPR, and F-score.
  - In order to validate the proposed *F-CBCT*, six benchmark datasets from UCI-ML repository and NSL-KDD dataset, a standard dataset for anomaly detection has been used. To evaluate the effectiveness and reliability of the proposed technique, several performance evaluation criterion such as-Root Mean Square Error (RMSE), DR, FPR, Accuracy, and F-Measure have been employed.
- The proposed anomaly detection schemes allow deep analysis of network-wide traffic systems; specifying their influence on the performance of real-time applications such as-time series anomaly detection in Cloud Computing Systems, cyber-threat detection in Intelligent Transportation System, theft detection in Smart Grid, malicious network traffic detection in IoT, suspicious flow detection in Software Defined Networks, identifying key events in Social Networks, etc.

6.2 Future Scope

Though, the proposed anomaly detection schemes demonstrated good performance relative to the current state-of-the-art techniques, but there is always a scope for improvement. This section elaborates some of the future directions in context of the proposed work.

In the near future, the optimization of the fuzzy membership problems in *En-ADT* can be done using advanced Bayesian filters such as-particle filters. Similarly, the proposed *F-CBCT* scheme can be extended using the recently proposed multi-objective optimization strategies such as-Grey Wolf Optimization, Moth-flame Optimization, Multi-Verse Optimization, etc. Moreover, weight vectors in the fitness function can be considered as future enhancements in this work. Further, the proposed scheme can also be implemented on real-time networks to validate its effectiveness and obtain real-time analysis of the designed detection models.

Bibliography

[1] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19 – 31, 2016.

[2] I. Yaqoob, E. Ahmed, M. H. ur Rehman, A. I. A. Ahmed, M. A. Al-garadi, M. Imran, and M. Guizani, "The rise of ransomware and emerging security challenges in the internet of things," *Computer Networks*, vol. 129, no. Part 2, pp. 444 – 458, 2017, Special Issue on 5G Wireless Networks for IoT and Body Sensors. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128617303468

[3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[4] D. K. Bhattacharyya and J. K. Kalita, *Network anomaly detection: A machine learning perspective*. CRC Press, 2013.

[5] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection for discrete sequences: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 823–839, 2012.

[6] D. Carrera and G. Boracchi, "Generating high-dimensional datastreams for change detection," *Big Data Research*, 2017, doi: https://doi.org/10.1016/j.bdr.2017.09.001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2214579617300163

[7] T. S. Sethi and M. Kantardzic, "Handling adversarial concept drift in streaming data," *Expert Systems with Applications*, vol. 97, pp. 18 – 40, 2018.

[8] A. Heydari, M. ali Tavakoli, N. Salim, and Z. Heydari, "Detection of review spam: A survey," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3634–3642, 2015.

[9] R. Gow, F. A. Rabhi, and S. Venugopal, "Anomaly detection in complex real world application systems," *IEEE Transactions on Network and Service Management*, 2017, doi: 10.1109/TNSM.2017.2771403.

[10] F. Jiang, Y. Sui, and C. Cao, "An information entropy-based approach to outlier detection in rough sets," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6338–6344, 2010.

[11] M. Thottan, G. Liu, and C. Ji, "Anomaly detection approaches for communication networks," in *Algorithms for Next Generation Networks*. Springer, 2010, pp. 239–261.

[12] C. O'Reilly, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Anomaly detection in wireless sensor networks in a non-stationary environment," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1413–1432, 2014.

[13] A. Daneshpazhouh and A. Sami, "Entropy-based outlier detection using semi-supervised approach with few positive examples," *Pattern Recognition Letters*, vol. 49, pp. 77–84, 2014.

[14] R. Kaur and S. Singh, "A survey of data mining and social network analysis based anomaly detection techniques," *Egyptian Informatics Journal*, vol. 17, no. 2, pp. 199–216, 2016.

[15] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," *Procedia Computer Science*, vol. 60, pp. 708–713, 2015.

[16] M. Ahmed, A. N. Mahmood, and M. R. Islam, "A survey of anomaly detection techniques in financial domain," *Future Generation Computer Systems*, vol. 55, pp. 278–288, 2016.

[17] C. Laorden, X. Ugarte-Pedrero, I. Santos, B. Sanz, J. Nieves, and P. G. Bringas, "Study on the effectiveness of anomaly detection for spam filtering," *Information Sciences*, vol. 277, pp. 421–444, 2014.

[18] M. Xie, S. Han, B. Tian, and S. Parvin, "Anomaly detection in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1302–1325, 2011.

[19] C. Sample and K. Schaffer, "An overview of anomaly detection," *IT Professional*, vol. 15, no. 1, pp. 8–11, 2013.

[20] D. Jiang, Z. Xu, and H. Xu, "A novel hybrid prediction algorithm to network traffic," *annals of telecommunications-annales des télécommunications*, vol. 70, no. 9-10, pp. 427–439, 2015.

[21] M. Roesch, "Snort: Lightweight intrusion detection for networks," in *Lisa*, vol. 99, no. 1, 1999, pp. 229–238.

[22] N. Duffield, P. Haffner, B. Krishnamurthy, and H. Ringberg, "Rule-based anomaly detection on IP flows," in *IEEE INFOCOM 2009*. IEEE, 2009, pp. 424–432.

[23] A. Qayyum, M. Islam, and M. Jamil, "Taxonomy of statistical based anomaly detection techniques for intrusion detection," in *Proceedings of the IEEE Symposium on Emerging Technologies, 2005.* IEEE, 2005, pp. 270–276.

[24] L. V. Allen and D. M. Tilbury, "Anomaly detection using model generation for event-based systems without a preexisting formal model," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 42, no. 3, pp. 654–668, 2012.

[25] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[26] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," 2007.

[27] S. Garcia, J. Luengo, J. A. Sáez, V. Lopez, and F. Herrera, "A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 4, pp. 734–750, 2013.

[28] J. Wang, Y. Miao, A. Khamis, F. Karray, and J. Liang, "Adaptation approaches in unsupervised learning: A survey of the state-of-the-art and future directions," in *International Conference Image Analysis and Recognition*. Springer, 2016, pp. 3–11.

[29] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[30] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[31] L. Xu, R. Collier, and G. M. OHare, "A survey of clustering techniques in wsns and consideration of the challenges of applying such to 5g iot scenarios," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1229–1249, 2017.

[32] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, no. 2, pp. 165–193, 2015.

[33] D. J. Weller-Fahy, B. J. Borghetti, and A. A. Sodemann, "A survey of distance and similarity measures used within network intrusion anomaly detection," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 70–91, 2015.

[34] S. J. Nanda and G. Panda, "A survey on nature inspired metaheuristic algorithms for partitional clustering," *Swarm and Evolutionary computation*, vol. 16, pp. 1–18, 2014.

[35] K. Maheswari and M. Ramakrishnan, "Hierarchical Clustering on Massive Datasets," *International Journal for Research in Emerging Science and Technology*, vol. 2, no. 1, pp. 56–59, 2016.

[36] S. Kisilevich, F. Mansmann, and D. Keim, "P-DBSCAN: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos," in *Proceedings of the 1st international conference and exhibition on computing for geospatial research & application (COM.Geo '10), Washington, D.C., USA*. ACM, 2010, pp. 38:1–38:4.

[37] B. Wu and B. M. Wilamowski, "A fast density and grid based clustering method for data with arbitrary shapes and noise," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1620 – 1628, 2017.

[38] D. A. Freedman, *Statistical models: theory and practice*. cambridge university press, 2009.

[39] D. C. Montgomery, G. C. Runger, and N. F. Hubele, *Engineering statistics*. John Wiley & Sons, 2009.

[40] J. Breier and J. Branišová, "A dynamic rule creation based anomaly detection method for identifying security breaches in log records," *Wireless Personal Communications*, vol. 94, no. 3, pp. 497–511, 2017.

[41] F. Xia, W. Wang, T. M. Bekele, and H. Liu, "Big Scholarly Data: A Survey," *IEEE Transactions on Big Data*, vol. 3, no. 1, pp. 18–35, 2017.

[42] X. Fu, Y. Gao, B. Luo, X. Du, and M. Guizani, "Security Threats to Hadoop: Data Leakage Attacks and Investigation," *IEEE Network*, vol. 31, no. 2, pp. 67–71, 2017.

[43] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in internet-of-things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, 2017.

[44] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[45] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, L. T. Yang *et al.*, "Data mining for internet of things: A survey." *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 77–97, 2014.

[46] Gartner, "Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016, Up 30 Percent From 2015," Nov. 2015, [Accessed on: Jun. 2018]. [Online]. Available: https://www.gartner.com/newsroom/id/3165317

[47] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep Learning for IoT Big Data and Streaming Analytics: A Survey," *IEEE Communications Surveys & Tutorials*, 2018.

[48] M. A. Chatti, A. Muslim, and U. Schroeder, "Toward an Open Learning Analytics Ecosystem," in *Big Data and Learning Analytics in Higher Education*. Springer, 2017, pp. 195–219.

[49] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2002, pp. 1–16.

[50] R. Alur, E. Berger, A. W. Drobnis, L. Fix, K. Fu, G. D. Hager, D. Lopresti, K. Nahrstedt, E. Mynatt, S. Patel *et al.*, "Systems computing challenges in the internet of things," *A Computing Community Consortium (CCC), Computers and Society*, pp. 1–15, 2016, doi:arXiv:1604.02980.

[51] (2016) Post-Intrusion Report. Vectra Networks. [Online]. Available: https://info.vectranetworks.com/hubfs/2016APRIL13_Post-Intrusion_Report.pdf?t=1477354055879

[52] (2016) HPE Security Research Cyber Risk Report. Hewlett Packard Enterprise. [Online]. Available: https://www.thehaguesecuritydelta.com/media/com_hsd/report/57/document/4aa6-3786enw.pdf

[53] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Anomaly detection in wireless sensor networks," *IEEE Wireless Communications*, vol. 15, no. 4, pp. 34 – 40, 2008.

[54] A. Rachedi and A. Benslimane, "Multi-objective optimization for security and QoS adaptation in Wireless Sensor Networks," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–7.

[55] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1086–1090, 2009.

[56] A. Benslimane and H. Nguyen-Minh, "Jamming Attack Model and Detection Method for Beacons Under Multichannel Operation in Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 6475–6488, July 2017.

[57] A. Abduvaliyev, A.-S. K. Pathan, J. Zhou, R. Roman, and W.-C. Wong, "On the vital areas of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1223–1237, 2013.

[58] R. Singh and T. P. Sharma, "On the IEEE 802.11 i security: a denial-of-service perspective," *Security and Communication Networks*, vol. 8, no. 7, pp. 1378–1407, 2015.

[59] D. Chaffey, "Global Social Media Research Summary 2018," Smart Insights, Tech. Rep., 2018. [Online]. Available: https://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/

[60] S. A. Alvi, B. Afzal, G. A. Shah, L. Atzori, and W. Mahmood, "Internet of multimedia things: Vision and challenges," *Ad Hoc Networks*, vol. 33, pp. 87 – 111, 2015.

[61] M. Fire, R. Goldschmidt, and Y. Elovici, "Online social networks: threats and solutions," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2019–2036, 2014.

[62] H. Li, H. He, and Y. Wen, "Dynamic particle swarm optimization and K-means clustering algorithm for image segmentation," *Optik-International Journal for Light and Electron Optics*, vol. 126, no. 24, pp. 4817–4822, 2015.

[63] D. Jiang, Z. Yuan, P. Zhang, L. Miao, and T. Zhu, "A traffic anomaly detection approach in communication networks for applications of multimedia medical devices," *Multimedia Tools and Applications*, vol. 75, no. 22, pp. 14 281–14 305, 2016.

[64] S. Maleki, C. Bingham, and Y. Zhang, "Development and realization of changepoint analysis for the detection of emerging faults on industrial systems," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 1180–1187, 2016.

[65] Z. Miller, B. Dickinson, W. Deitrick, W. Hu, and A. H. Wang, "Twitter spammer detection using data stream clustering," *Information Sciences*, vol. 260, pp. 64–73, 2014.

[66] X.-S. Yang, *Nature-inspired optimization algorithms*. Elsevier, 2014.

[67] P. Benner, V. Mehrmann, and D. C. Sorensen, *Dimension reduction of large-scale systems*. Springer, 2005, vol. 45.

[68] S. Alelyani, J. Tang, and H. Liu, "Feature Selection for Clustering: A Review." *Data Clustering: Algorithms and Applications*, vol. 29, pp. 110–121, 2013.

[69] M. Dash and H. Liu, "Feature selection for clustering," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2000, pp. 110–121.

[70] S. Fong, R. Wong, and A. V. Vasilakos, "Accelerated PSO swarm search feature selection for data stream mining big data," *IEEE Transactions on Services Computing*, vol. 9, no. 1, pp. 33–45, 2016.

[71] G. Duan, W. Hu, and Z. Zhang, "A Novel Multilayer Data Clustering Framework based on Feature Selection and Modified K-Means Algorithm," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 4, pp. 81–90, 2016.

[72] R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki, "A Survey on semi-supervised feature selection methods," *Pattern Recognition*, vol. 64, pp. 141 – 158, 2017.

[73] V. Kumar and S. Minz, "Feature selection: A literature review," *Smart CR*, vol. 4, pp. 211–229, 2014.

[74] D. Panday, R. C. de Amorim, and P. Lane, "Feature weighting as a tool for unsupervised feature selection," *Information Processing Letters*, vol. 129, pp. 44–52, 2018.

[75] K. Yu, X. Wu, W. Ding, Y. Mu, and H. Wang, "Markov blanket feature selection using representative sets," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 11, pp. 2775 – 2788, 2017.

[76] J. Xu, B. Tang, H. He, and H. Man, "Semisupervised feature selection based on relevance and redundancy criteria," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 9, pp. 1974 – 1984, 2017.

[77] P. Zhou, X. Hu, P. Li, and X. Wu, "Online feature selection for high-dimensional class-imbalanced data," *Knowledge-Based Systems*, vol. 136, pp. 187–199, 2017.

[78] A. K. Das, S. Goswami, A. Chakrabarti, and B. Chakraborty, "A new hybrid feature selection approach using feature association map for supervised and unsupervised classification," *Expert Systems with Applications*, vol. 88, pp. 81–94, 2017.

[79] C. Hou, F. Nie, H. Tao, and D. Yi, "Multi-view unsupervised feature selection with adaptive similarity and view weight," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 9, pp. 1998 – 2011, 2017.

[80] J. Liu, Y. Lin, S. Wu, and C. Wang, "Online multi-label group feature selection," *Knowledge-Based Systems*, 2017, doi: https://doi.org/10.1016/j.knosys.2017.12.008.

[81] X. Tang, Y. Dai, P. Sun, and S. Meng, "Interaction-based feature selection using factorial design," *Neurocomputing*, 2017, doi: https://doi.org/10.1016/j.neucom.2017.11.058.

[82] L. Hu, W. Gao, K. Zhao, P. Zhang, and F. Wang, "Feature selection considering two types of feature relevancy and feature interdependency," *Expert Systems with Applications*, vol. 93, pp. 423–434, 2018.

[83] J. Izetta, P. F. Verdes, and P. M. Granitto, "Improved multiclass feature selection via list combination," *Expert Systems with Applications*, vol. 88, pp. 205–216, 2017.

[84] P. Zhu, Q. Xu, Q. Hu, C. Zhang, and H. Zhao, "Multi-label feature selection with missing labels," *Pattern Recognition*, vol. 74, pp. 488–502, 2018.

[85] Y. Lin, Q. Hu, J. Liu, J. Li, and X. Wu, "Streaming feature selection for multi-label learning based on fuzzy mutual information," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 6, pp. 1491 – 1507, 2017.

[86] Y. Liu, F. Tang, and Z. Zeng, "Feature selection based on dependency margin," *IEEE Transactions on Cybernetics*, vol. 45, no. 6, pp. 1209–1221, 2015.

[87] M. Qi, T. Wang, F. Liu, B. Zhang, J. Wang, and Y. Yi, "Unsupervised feature selection by regularized matrix factorization," *Neurocomputing*, vol. 273, pp. 593–610, 2018.

[88] Y. Prasad, D. Khandelwal, and K. Biswas, "Max-margin feature selection," *Pattern Recognition Letters*, vol. 95, pp. 51–57, 2017.

[89] L. Zhang, K. Mistry, C. P. Lim, and S. C. Neoh, "Feature selection using firefly optimization for classification and regression models," *Decision Support Systems*, 2017, doi: https://doi.org/10.1016/j.dss.2017.12.001.

[90] F. Viegas, L. Rocha, M. Gonçalves, F. Mourão, G. Sá, T. Salles, G. Andrade, and I. Sandin, "A genetic programming approach for feature selection in highly dimensional skewed data," *Neurocomputing*, vol. 273, pp. 554–569, 2018.

[91] P. P. Kundu and S. Mitra, "Feature selection through message passing," *IEEE transactions on cybernetics*, vol. 47, no. 12, pp. 4356 – 4366, 2016.

[92] P. Zhu, Q. Xu, Q. Hu, and C. Zhang, "Co-regularized unsupervised feature selection," *Neurocomputing*, vol. 275, pp. 2855–2863, 2017.

[93] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Applied Soft Computing*, vol. 62, pp. 441–453, 2018.

[94] C. Tang, X. Zhu, J. Chen, P. Wang, and X. Liu, "Robust graph regularized unsupervised feature selection," *Expert Systems with Applications*, 2017, doi: https://doi.org/10.1016/j.eswa.2017.11.053.

[95] Y. Shao, K. Wang, L. Shu, S. Deng, and D.-J. Deng, "Heuristic Optimization for Reliable Data Congestion Analytics in Crowdsourced eHealth Networks," *IEEE Access*, vol. 4, pp. 9174–9183, 2016.

[96] T. F. Ghanem, W. S. Elkilani, and H. M. Abdul-Kader, "A hybrid approach for efficient anomaly detection using metaheuristic methods," *Journal of advanced research*, vol. 6, no. 4, pp. 609–619, 2015.

[97] A. A. Aburomman and M. B. I. Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," *Applied Soft Computing*, vol. 38, pp. 360–372, 2016.

[98] M. L. Shahreza, D. Moazzami, B. Moshiri, and M. Delavar, "Anomaly detection using a self-organizing map and particle swarm optimization," *Scientia Iranica*, vol. 18, no. 6, pp. 1460–1468, 2011.

[99] A. Karami and M. Guerrero-Zapata, "A hybrid multiobjective RBF-PSO method for mitigating dos attacks in named data networking," *Neurocomputing*, vol. 151, pp. 1262–1282, 2015.

[100] ——, "A fuzzy anomaly detection system based on hybrid PSO-Kmeans algorithm in content-centric networks," *Neurocomputing*, vol. 149, pp. 1253–1269, 2015.

[101] T. İnkaya, S. Kayalıgil, and N. E. Özdemirel, "Ant Colony Optimization based clustering methodology," *Applied Soft Computing*, vol. 28, pp. 301–311, 2015.

[102] S. Alam, G. Dobbie, Y. S. Koh, P. Riddle, and S. U. Rehman, "Research on particle swarm optimization based clustering: a systematic review of literature and techniques," *Swarm and Evolutionary Computation*, vol. 17, pp. 1–13, 2014.

[103] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.

[104] ——, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053–1073, 2016.

[105] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[106] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature inspired cooperative strategies for optimization (NICSO 2010)*, pp. 65–74, 2010.

[107] D. Karaboga and B. Basturk, "Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems," *Foundations of fuzzy logic and soft computing*, pp. 789–798, 2007.

[108] S.-C. Chu, P.-W. Tsai, and J.-S. Pan, "Cat swarm optimization," in *Pacific Rim International Conference on Artificial Intelligence*. Springer, 2006, pp. 854–858.

[109] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *World Congress on Nature & Biologically Inspired Computing, NaBIC'09*. IEEE, 2009, pp. 210–214.

[110] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *International symposium on stochastic algorithms*. Springer, 2009, pp. 169–178.

[111] C. J. Bastos Filho, F. B. de Lima Neto, A. J. Lins, A. I. Nascimento, and M. P. Lima, "Fish school search," in *Nature-inspired algorithms for optimisation*. Springer, 2009, pp. 261–277.

[112] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of machine learning*. Springer, 2011, pp. 760–766.

[113] Z. W. Geem, J. H. Kim, and G. Loganathan, "A new heuristic optimization algorithm: harmony search," *simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[114] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[115] M. Ahmed and A. N. Mahmood, "Novel approach for network traffic pattern analysis using clustering-based collective anomaly detection," *Annals of Data Science*, vol. 2, no. 1, pp. 111–130, 2015.

[116] D. J. Weller-Fahy, B. J. Borghetti, and A. A. Sodemann, "A survey of distance and similarity measures used within network intrusion anomaly detection," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 70–91, 2015.

[117] A. K. Tiwari and R. Mishra, "Protein Function Prediction Using Support Vector Machine," *International Journal of Computational Bioinformatics and In Silico Modeling*, vol. 2, no. 5, pp. 239–244, 2013.

[118] K. P. Murphy, "Naive bayes classifiers," *University of British Columbia*, 2006.

[119] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[120] C. Cortes and V. Vapnik, "Support vector machine," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[121] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. 20th int. conf. very large data bases, VLDB*, vol. 1215, 1994, pp. 487–499.

[122] J. Cohen, P. Cohen, S. G. West, and L. S. Aiken, *Applied multiple regression/correlation analysis for the behavioral sciences*. Routledge, 2013.

[123] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[124] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[125] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[126] S. M. R. Zadegan, M. Mirzaie, and F. Sadoughi, "Ranked k-medoids: A fast and accurate rank-based partitioning algorithm for clustering large datasets," *Knowledge-Based Systems*, vol. 39, pp. 133–143, 2013.

[127] S. Lai and H.-C. Fu, "Variance enhanced K-medoid clustering," *Expert Systems with Applications*, vol. 38, no. 1, pp. 764–775, 2011.

[128] K. M. Kumar and A. R. M. Reddy, "A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method," *Pattern Recognition*, vol. 58, pp. 39–48, 2016.

[129] Y. Lv, T. Ma, M. Tang, J. Cao, Y. Tian, A. Al-Dhelaan, and M. Al-Rodhaan, "An efficient and scalable density-based clustering algorithm for datasets with complex structures," *Neurocomputing*, vol. 171, pp. 9–22, 2016.

[130] T. N. Tran, K. Drab, and M. Daszykowski, "Revised DBSCAN algorithm to cluster data with dense adjacent clusters," *Chemometrics and Intelligent Laboratory Systems*, vol. 120, pp. 92–96, 2013.

[131] W. Kwedlo, "A clustering method combining differential evolution with the K-means algorithm," *Pattern Recognition Letters*, vol. 32, no. 12, pp. 1613–1621, 2011.

[132] C. Warrender, S. Forrest, and B. Pearlmutter, "Detecting intrusions using system calls: Alternative data models," in *IEEE Symposium on Security and Privacy*. IEEE, 1999, pp. 133–145.

[133] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for unix processes," in *IEEE Symposium on Security and Privacy*. IEEE, 1996, pp. 120–128.

[134] A. K. Ghosh and A. Schwartzbard, "A Study in Using Neural Networks for Anomaly and Misuse Detection," in *USENIX security symposium*, vol. 99, 1999, p. 12.

[135] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Computers & security*, vol. 21, no. 5, pp. 439–448, 2002.

[136] J. Dromard, G. Roudiere, and P. Owezarski, "Online and scalable unsupervised network anomaly detection method," *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 34–47, 2017.

[137] D.-H. Shin and J. Zhang, "Early Anomaly Detection in an Interconnected Power Grid and Communication Network: Exploiting Interdependent Structure of Failures," in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.

[138] Y. Yuan, D. Ma, and Q. Wang, "Hyperspectral anomaly detection by graph pixel selection," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 3123–3134, 2016.

[139] D. Jiang, Z. Xu, Z. Chen, Y. Han, and H. Xu, "Joint time–frequency sparse estimation of large-scale network traffic," *Computer Networks*, vol. 55, no. 15, pp. 3533–3547, 2011.

[140] D. Jiang, C. Yao, Z. Xu, and W. Qin, "Multi-scale anomaly detection for high-speed network traffic," *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 3, pp. 308–317, 2015.

[141] D. Jiang, Z. Zhao, Z. Xu, C. Yao, and H. Xu, "How to reconstruct end-to-end traffic based on time-frequency analysis and artificial neural network," *AEU-International Journal of Electronics and Communications*, vol. 68, no. 10, pp. 915–925, 2014.

[142] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1690–1700, 2014.

[143] J. Tian, M. Gao, and F. Zhang, "Network intrusion detection method based on radial basic function neural network," in *E-Business and Information System Security, 2009. EBISS'09. International Conference on*. IEEE, 2009, pp. 1–4.

[144] Y. Xie and Y. Zhang, "An intelligent anomaly analysis for intrusion detection based on SVM," in *Computer Science and Information Processing (CSIP), 2012 International Conference on*. IEEE, 2012, pp. 739–742.

[145] N. Sharma and S. Mukherjee, "A novel multi-classifier layered approach to improve minority attack detection in IDS," *Procedia Technology*, vol. 6, pp. 913–921, 2012.

[146] R. Chitrakar and H. Chuanhe, "Anomaly detection using Support Vector Machine classification with k-Medoids clustering," in *Internet (AH-ICI), 2012 Third Asian Himalayas International Conference on*. IEEE, 2012, pp. 1–5.

[147] R. M. Elbasiony, E. A. Sallam, T. E. Eltobely, and M. M. Fahmy, "A hybrid network intrusion detection framework based on random forests and weighted k-means," *Ain Shams Engineering Journal*, vol. 4, no. 4, pp. 753–762, 2013.

[148] V. A. Sotiris, W. T. Peter, and M. G. Pecht, "Anomaly detection through a bayesian support vector machine," *IEEE Transactions on Reliability*, vol. 59, no. 2, pp. 277–286, 2010.

[149] A. Forestiero, "Self-organizing anomaly detection in data streams," *Information Sciences*, vol. 373, pp. 321–336, 2016.

[150] M. Moshtaghi, J. C. Bezdek, C. Leckie, S. Karunasekera, and M. Palaniswami, "Evolving fuzzy rules for anomaly detection in data streams," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 3, pp. 688–700, 2015.

[151] S. Xu and J. Wang, "Dynamic extreme learning machine for data stream classification," *Neurocomputing*, vol. 238, pp. 433–449, 2017.

[152] S. S. S. Sindhu, S. Geetha, and A. Kannan, "Decision tree based light weight intrusion detection using a wrapper approach," *Expert Systems with applications*, vol. 39, no. 1, pp. 129–141, 2012.

[153] Z. X. Yang, X. L. Qin, W. R. Li, and Y. J. Yang, "A DDoS detection approach based on CNN in cloud computing," in *Applied Mechanics and Materials*, vol. 513. Trans Tech Publ, 2014, pp. 579–584.

[154] W. Chu, H. Xue, C. Yao, and D. Cai, "Sparse Coding Guided Spatiotemporal Feature Learning for Abnormal Event Detection in Large Videos," *IEEE Transactions on Multimedia*, 2018, DOI: 10.1109/TMM.2018.2846411.

[155] K. Xu, X. Jiang, and T. Sun, "Anomaly Detection Based on Stacked Sparse Coding With Intraframe Classification Strategy," *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1062–1074, 2018.

[156] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes," *Computer Vision and Image Understanding*, 2018, DOI: https://doi.org/10.1016/j.cviu.2018.02.006.

[157] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[158] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, "Method of intrusion detection using deep neural network," in *Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on*. IEEE, 2017, pp. 313–316.

[159] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in *Wireless Networks and Mobile Communications (WINCOM), 2016 International Conference on*. IEEE, 2016, pp. 258–263.

[160] S. Kanarachos, S.-R. G. Christopoulos, A. Chroneos, and M. E. Fitzpatrick, "Detecting anomalies in time series data via a deep learning algorithm combining wavelets, neural networks and hilbert transform," *Expert Systems with Applications*, vol. 85, pp. 292–304, 2017.

[161] A.-H. Muna, N. Moustafa, and E. Sitnikova, "Identification of malicious activities in industrial internet of things based on deep learning models," *Journal of Information Security and Applications*, vol. 41, pp. 1–11, 2018.

[162] Y. Feng, Y. Yuan, and X. Lu, "Learning deep event models for crowd anomaly detection," *Neurocomputing*, vol. 219, pp. 548–556, 2017.

[163] J. Xu, S. Denman, C. Fookes, and S. Sridharan, "Detecting rare events using Kullback–Leibler divergence: A weakly supervised approach," *Expert Systems with Applications*, vol. 54, pp. 13–28, 2016.

[164] E. Tu, Y. Zhang, L. Zhu, J. Yang, and N. Kasabov, "A graph-based semi-supervised k nearest-neighbor method for nonlinear manifold distributed data classification," *Information Sciences*, vol. 367, pp. 673–688, 2016.

[165] S. R. Gaddam, V. V. Phoha, and K. S. Balagani, "K-Means+ ID3: A novel method for supervised anomaly detection by cascading K-Means clustering and ID3 decision tree learning methods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 345–354, 2007.

[166] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Sciences*, vol. 378, pp. 484–497, 2017.

[167] M. Jager, C. Knoll, and F. A. Hamprecht, "Weakly supervised learning of a classifier for unusual event detection," *IEEE Transactions on Image Processing*, vol. 17, no. 9, pp. 1700–1708, 2008.

[168] L. Peng, W. Chen, W. Zhou, F. Li, J. Yang, and J. Zhang, "An immune-inspired semi-supervised algorithm for breast cancer diagnosis," *Computer Methods and Programs in Biomedicine*, vol. 134, pp. 259–265, 2016.

[169] Y. Li, R. Xiao, J. Feng, and L. Zhao, "A semi-supervised learning approach for detection of phishing webpages," *Optik-International Journal for Light and Electron Optics*, vol. 124, no. 23, pp. 6027–6033, 2013.

[170] J. Song, H. Takakura, Y. Okabe, and K. Nakao, "Toward a more practical unsupervised anomaly detection system," *Information Sciences*, vol. 231, pp. 4–14, 2013.

[171] A. Almalawi, X. Yu, Z. Tari, A. Fahad, and I. Khalil, "An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems," *Computers & Security*, vol. 46, pp. 94–110, 2014.

[172] H. Li, A. Achim, and D. Bull, "Unsupervised video anomaly detection using feature clustering," *IET signal processing*, vol. 6, no. 5, pp. 521–533, 2012.

[173] A. P. Muniyandi, R. Rajeswari, and R. Rajaram, "Network anomaly detection by cascading k-Means clustering and C4. 5 decision tree algorithm," *Procedia Engineering*, vol. 30, pp. 174–182, 2012.

[174] A. Karami and M. Guerrero-Zapata, "A fuzzy anomaly detection system based on hybrid pso-kmeans algorithm in content-centric networks," *Neurocomputing*, vol. 149, pp. 1253–1269, 2015.

[175] N. L. D. Khoa, T. Babaie, S. Chawla, and Z. Zaidi, "Network anomaly detection using a commute distance based approach," in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*. IEEE, 2010, pp. 943–950.

[176] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.

[177] R. Laxhammar and G. Falkman, "Online learning and sequential anomaly detection in trajectories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 6, pp. 1158–1173, 2014.

[178] U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, "Network anomaly detection with the restricted Boltzmann machine," *Neurocomputing*, vol. 122, pp. 13–23, 2013.

[179] M. H. Bhuyan, D. Bhattacharyya, and J. Kalita, "A multi-step outlier-based anomaly detection approach to network-wide traffic," *Information Sciences*, vol. 348, pp. 243–271, 2016.

[180] D. Sun, M. Fu, L. Zhu, G. Li, and Q. Lu, "Non-Intrusive Anomaly Detection With Streaming Performance Metrics and Logs for DevOps in Public Clouds: A Case Study in AWS," *IEEE Transactions on Emerging Topics in Computing*, vol. 4, no. 2, pp. 278–289, April 2016.

[181] H. H. Bosman, G. Iacca, A. Tejada, H. J. Wörtche, and A. Liotta, "Spatial anomaly detection in sensor networks using neighborhood information," *Information Fusion*, vol. 33, pp. 41–56, 2017.

[182] S. Zhou, W. Shen, D. Zeng, M. Fang, Y. Wei, and Z. Zhang, "Spatial–temporal convolutional neural networks for anomaly detection and localization in crowded scenes," *Signal Processing: Image Communication*, vol. 47, pp. 358–368, 2016.

[183] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "Novelty detection using level set methods," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 3, pp. 576–588, 2015.

[184] A. A. Aburomman and M. B. I. Reaz, "A novel svm-knn-pso ensemble method for intrusion detection system," *Applied Soft Computing*, vol. 38, pp. 360–372, 2016.

[185] G. Fernandes, L. F. Carvalho, J. J. Rodrigues, and M. L. Proença, "Network anomaly detection using IP flows with Principal Component Analysis and Ant Colony Optimization," *Journal of Network and Computer Applications*, vol. 64, pp. 1–11, 2016.

[186] K.-J. Hsiao, K. S. Xu, J. Calder, and A. O. Hero, "Multicriteria Similarity-Based Anomaly Detection Using Pareto Depth Analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 6, pp. 1307–1321, 2016.

[187] Z. Peng, P. Gurram, H. Kwon, and W. Yin, "Sparse kernel learning-based feature selection for anomaly detection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 1698–1716, 2015.

[188] A. K. Chorppath, T. Alpcan, and H. Boche, "Bayesian Mechanisms and Detection Methods for Wireless Network with Malicious Users," *IEEE Transactions on Mobile Computing*, vol. 15, no. 10, pp. 2452–2465, 2016.

[189] C. Pascoal, M. R. de Oliveira, R. Valadas, P. Filzmoser, P. Salvador, and A. Pacheco, "Robust feature selection and robust PCA for Internet traffic anomaly detection," in *Proceedings of IEEE INFOCOM, Orlando, Florida, USA*. IEEE, 2012, pp. 1755–1763.

[190] Webscope datasets. Yahoo. [Accessed on: Oct. 2015]. [Online]. Available: https://webscope.sandbox.yahoo.com/

[191] "NAB: Numenta Anomaly Benchmark." 2015, [Accessed on: Oct. 2015]. [Online]. Available: https://github.com/numenta/NAB

[192] M. Lichman, "UCI Machine Learning Repository," University of California, Irvine, School of Information and Computer Sciences, 2013, [Accessed on: Oct. 2015]. [Online]. Available: http://archive.ics.uci.edu/ml

[193] The home of data science & machine learning. Kaggle Inc. [Accessed on: Oct. 2015]. [Online]. Available: https://www.kaggle.com/

[194] Aws public datasets. Amazon Web Services. [Accessed on: Oct. 2015]. [Online]. Available: https://aws.amazon.com/public-datasets/

[195] Datestream data. Microsoft. [Accessed on: Oct. 2015]. [Online]. Available: http://datamarket.azure.com/dataset/boyanpenev/datestream

[196] Stop looking for people to sell to - sell to the people who need you. BDEX. [Accessed on: Oct. 2015]. [Online]. Available: http://www.bigdataexchange.com/

[197] Aggdata. AggData. [Accessed on: Oct. 2015]. [Online]. Available: https://www.aggdata.com/

[198] Open government data (ogd) platform india. GOVERNMENT OF INDIA. [Accessed on: Oct. 2015]. [Online]. Available: https://data.gov.in/

[199] The home of the u.s. governments open data. [Accessed on: Oct. 2015]. [Online]. Available: https://www.data.gov/

[200] Bristsh columbia databc. [Accessed on: Oct. 2015]. [Online]. Available: https://data.gov.bc.ca/

[201] data.gov.uk. [Accessed on: Oct. 2015]. [Online]. Available: https://data.gov.uk/

[202] E. Lundin and E. Jonsson, "Anomaly-based intrusion detection: privacy concerns and other problems," *Computer networks*, vol. 34, no. 4, pp. 623–640, 2000.

[203] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on neural networks*, vol. 16, no. 3, pp. 645–678, 2005.

[204] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[205] N. Henke, J. Bughin, M. Chui, J. Manyika, T. Saleh, B. Wiseman, and G. Sethupathy, "The age of analytics: Competing in a data-driven world," *McKinsey Global Institute*, vol. 4, 2016.

[206] R. C. Barros, M. P. Basgalupp, A. C. De Carvalho, and A. A. Freitas, "A survey of evolutionary algorithms for decision-tree induction," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 3, pp. 291–312, 2012.

[207] F. d. A. De Carvalho and C. P. Tenório, "Fuzzy K-means clustering algorithms for interval-valued data based on adaptive quadratic distances," *Fuzzy Sets and Systems*, vol. 161, no. 23, pp. 2978–2999, 2010.

[208] A. Jindal, A. Dua, K. Kaur, M. Singh, N. Kumar, and S. Mishra, "Decision tree and SVM-based data analytics for theft detection in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 1005–1016, 2016.

[209] T. Eitrich and B. Lang, "Efficient optimization of support vector machine learning parameters for unbalanced datasets," *Journal of computational and applied mathematics*, vol. 196, no. 2, pp. 425–436, 2006.

[210] H.-C. Wu, "The Karush–Kuhn–Tucker optimality conditions in an optimization problem with interval-valued objective function," *European Journal of Operational Research*, vol. 176, no. 1, pp. 46–59, 2007.

[211] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," 2003.

[212] P. Civicioglu and E. Besdok, "A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms," *Artificial Intelligence Review*, vol. 39, no. 4, pp. 315–346, 2013.

[213] S. Petrovic, "A comparison between the silhouette index and the davies-bouldin index in labelling ids clusters," in *Proceedings of the 11th Nordic Workshop of Secure IT Systems*, 2006, pp. 53–64.

[214] W. Hu, Y. Liao, and V. R. Vemuri, "Robust anomaly detection using support vector machines," in *Proceedings of the international conference on machine learning*, 2003, pp. 282–289.

[215] L. Zhanchun, L. Zhitang, and L. Bin, "Anomaly detection system based on principal component analysis and support vector machine," *WuHan University Journal of Natural Sciences*, vol. 11, no. 6, pp. 1769–1772, 2006.

[216] C. A. Catania, F. Bromberg, and C. G. Garino, "An autonomous labeling approach to support vector machines algorithms for network traffic anomaly detection," *Expert Systems with Applications*, vol. 39, no. 2, pp. 1822–1829, 2012.

[217] N. A. Elfeshawy and O. S. Faragallah, "Divided two-part adaptive intrusion detection system," *Wireless networks*, vol. 19, no. 3, pp. 301–321, 2013.

[218] L. Kuang, "DNIDS: a dependable network intrusion detection system using the CSI-KNN algorithm," Ph.D. dissertation, Queen's Univ, Kingston, Ontario, Canada, 2007, Master's Thesis.

[219] T. S. Hwang, T.-J. Lee, and Y.-J. Lee, "A three-tier IDS via data mining approach," in *Proceedings of the 3rd annual ACM workshop on Mining network data*. ACM, 2007, pp. 1–6.

[220] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 5, pp. 649–659, 2008.

[221] S. Mukherjee and N. Sharma, "Intrusion detection using naive Bayes classifier with feature reduction," *Procedia Technology*, vol. 4, pp. 119–128, 2012.

[222] M. Panda, A. Abraham, and M. R. Patra, "A hybrid intelligent approach for network intrusion detection," *Procedia Engineering*, vol. 30, pp. 1–9, 2012.

[223] A. H. Bhat, S. Patra, and D. Jena, "Machine learning approach for intrusion detection on cloud virtual machines," *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, vol. 2, no. 6, pp. 56–66, 2013.

[224] R. M. Elbasiony, E. A. Sallam, T. E. Eltobely, and M. M. Fahmy, "A hybrid network intrusion detection framework based on random forests and weighted k-means," *Ain Shams Engineering Journal*, vol. 4, no. 4, pp. 753–762, 2013.

[225] G. Nadiammai and M. Hemalatha, "Effective approach toward Intrusion Detection System using data mining techniques," *Egyptian Informatics Journal*, vol. 15, no. 1, pp. 37–50, 2014.

[226] M. Mohammadi, A. Akbari, B. Raahemi, B. Nassersharif, and H. Asgharian, "A fast anomaly detection system using probabilistic artificial immune algorithm capable of learning new attacks," *Evolutionary Intelligence*, vol. 6, no. 3, pp. 135–156, 2014.

[227] T. F. Ghanem, W. S. Elkilani, and H. M. Abdul-Kader, "A hybrid approach for efficient anomaly detection using metaheuristic methods," *Journal of advanced research*, vol. 6, no. 4, pp. 609–619, 2015.

[228] S. Duque and M. N. bin Omar, "Using data mining algorithms for developing a model for intrusion detection system (IDS)," *Procedia Computer Science*, vol. 61, pp. 46–51, 2015.

[229] N. Pandeeswari and G. Kumar, "Anomaly detection system in cloud environment using fuzzy clustering based ANN," *Mobile Networks and Applications*, vol. 21, no. 3, pp. 494–505, 2016.

[230] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "NSL-KDD dataset," 2012, [Accessed on: Oct. 2015]. [Online]. Available: http://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html

[231] E. Amiri and S. Mahmoudi, "Efficient protocol for data clustering by fuzzy Cuckoo Optimization Algorithm," *Applied Soft Computing*, vol. 41, pp. 15–21, 2016.

[232] D. Zheng, F. Li, and T. Zhao, "Self-adaptive statistical process control for anomaly detection in time series," *Expert Systems with Applications*, vol. 57, pp. 324–336, 2016.

[233] N. B. Aissa and M. Guerroumi, "Semi-supervised Statistical Approach for Network Anomaly Detection," *Procedia Computer Science*, vol. 83, pp. 1090–1095, 2016.

[234] S. M. H. Bamakan, H. Wang, T. Yingjie, and Y. Shi, "An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization," *Neurocomputing*, vol. 199, pp. 90–102, 2016.

List of Publications

Refereed Journals

1) Sahil Garg and Shalini Batra, "A novel ensembled technique for anomaly detection," *International Journal of Communication Systems, Wiley*- SCIE Indexed, 2016, doi:10.1002/dac.3248. (IF: 1.717)

2) Sahil Garg and Shalini Batra, "Fuzzified cuckoo based clustering technique for network anomaly detection," *Computers & Electrical Engineering, Elsevier*- SCIE Indexed, 2017, doi: http://dx.doi.org/10.1016/j.compeleceng.2017.07.008. (IF: 1.747)