Robin Babujee Jerome

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 05.03.2015

Thesis supervisor:

Dr. N. Asokan

Thesis advisor:

Dr. Kimmo Hätönen

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY

SCHOOL OF ELECTRICAL ENGINEERING Abstract of the Master's Thesis

Author: Robin Babujee Jerome

Title: Pre-processing techniques for anomaly detection in telecommunication networks

Date: 11th November 2014 Language: English Number of pages: 11+88

Department of Communications and Networking

Professorship: Networking Technology Code: S-38

Supervisor: Prof. N. Asokan

Instructor: Dr. Kimmo Hätönen

Anomalies in telecommunication networks can be signs of errors or malfunctions, which can originate from a wide variety of reasons. Huge amount of data collected from network elements in the form of counters, server logs, audit trail logs etc. can provide significant information about the normal state of the system as well as possible anomalies.

Unsupervised methods like '*Self-Organizing Maps'* (SOM) are often chosen for anomaly detection. They are useful for analyzing and categorizing high volume, high dimensional data. One of the major issues with using SOMs or other unsupervised methods for analyzing anomalies in *Telecommunication Management Networks* is that they are highly sensitive to pre-treatment of data.

The main objective of this thesis is to identify the right kind of pre-processing steps that can be applied to real mobile network traffic data measurements, so that the most interesting anomalies get detected in the anomaly detection stage. Two methods of evaluating the effectiveness of an anomaly detection technique for telecom network measurement data are also proposed.

Keywords: Anomaly Detection, Pre-processing, Self-Organizing Maps, SOM, Training set filtering, Scaling, Weighting, Outlier detection, Telecom Network Management, Quality of anomalies.

Acknowledgements

Firstly, I wish to thank my employer '*Nokia Networks*' for giving me the opportunity to work on this highly challenging and interesting thesis topic. This thesis has proved to be an excellent learning opportunity for me in the field of data mining and anomaly detection.

My interest in this field has multiplied over the past few months and I thank my instructor Dr. Kimmo Hätönen for guiding me throughout this period. I am grateful to my supervisor Professor N. Asokan for agreeing to supervise my thesis and providing me with valuable advice and encouraging comments. I also thank Professor Jörg Ott for his invaluable comments and suggestions in improving this thesis report.

I thank Leo Hippeläinen for building many software tools that I have used for my research work and also for his support. Thanks to my colleagues for providing an environment conducive for successfully completing my work.

Finally, I also thank my friend Hema for her valuable suggestions in formatting this document, and my parents for their love and support.

Table of Contents

Abbreviations and Acronyms

3GPP	Third Generation Partnership Project
AN	Access Network
BMU	Best Matching Unit
BSS	Base Station Sub System
BTS	Base Transceiver Station
CN	Core Network
DCN	Data Communication Network
DM	Data Mining
eNodeB	Evolved NodeB
EPC	Evolved Packet Core
EPS	Evolved Packet System
E-UTRAN	Evolved UMTS Terrestrial Radio Access Network
GSM	Global System for Mobile Communications
HSS	Home Subscriber Server:
IP	Internet Protocol
KDD	Knowledge Discovery in Databases
KPI	Key Performance Indicator
LogSig	Logarithmic Sigmoid
LTE	Long Term Evolution
MAD	Median Absolute Deviation
MME	Mobility Management Entity
NE	Network Element
NMS	Network Management System
PCRF	Policy Control and Charging Rules Function
PDN	Packet Data Network
P-GW	PDN Gateway
QoS	Quality of Service
RLog	Robust Logarithmic

RMAD	Robust Median Absolute Deviation about the median
SAE	System Architecture Evolution
S-GW	Serving Gateway
SOM	Self-Organizing Map
SON	Self-Organizing Network
TMN	Telecommunication Management Network
TRX	Transceiver
UE	User Equipment
UMTS	Universal Mobile Telecommunication System

List of Tables

List of Figures

1 Introduction

1.1 Background and motivation

Faults or anomalies in telecom networks have traditionally been detected using Fault Management or Performance Management components of *Network Management Systems* (NMS). These detected faults along with their severity are displayed to a network administrator, so that further action can be performed to rectify the faulty components. In a complex telecom network, there could be thousands of Network Elements (NEs) with each of them notifying the NMS of various activities. Managing thousands of alarms can be a cumbersome task for a network administrator and significant alarms can get missed due to human errors or even lost in the network due to network congestion or failures.

This is where *Knowledge Discovery in Databases* (KDD) and *Data Mining* (DM) techniques come into play. Huge amount of data collected from server logs, audit trail logs, traffic measurement data *etc.* can provide significant information about the normal state of a telecom system as well as possible anomalies. Anomaly detection forms a very important task in telecommunication network monitoring [1] and has been the topic of several research works in the past few years [2] [3] [4]. Since it is very difficult to obtain reference data with labeled anomalies from industrial processes, unsupervised methods are chosen for anomaly detection [5].

Among these unsupervised techniques, *Self-Organizing Maps* (SOMs) is a tool often used for analyzing telecommunication network data; characterized by its high volume and high dimensionality. The key idea of SOM is to map high-dimensional data into low-dimensional space by competitive learning and topological neighborhood [6] so that the topology is preserved [7]. Network traffic data obtained from several sources need to be pre-processed before they can be fed to SOMs or other anomaly detection mechanisms. These pre-processing steps could include numerization of log data, cleaning and filtering of training set, scaling and weighting of variables *etc.* depending on the type of data analyzed and goal of the anomaly detection experiment.

SOMs, as well as the other neural network models, follow the "*garbage in - garbage out*" principle. If poor quality data is fed to the SOM, the result will also be of poor quality. This is further emphasized in [5] which highlights that one of the major issues with using SOMs or other clustering methods for analyzing anomalies in TMNs is that they are highly sensitive to pre-processing of data. Thus pre-processing is of paramount importance for the success of anomaly detection using SOMs.

1.2 Scope of the Thesis

The goal of this thesis is to identify the appropriate pre-processing steps that are to be used for network traffic measurement data obtained from a Long Term Evolution (LTE) network. The data required for this research is extracted from *Nokia Serve atOnce Traffica* [8] which is one of the prominent Customer Experience Management products of *Nokia Networks*. *Nokia Serve atOnce Traffica* is a system which monitors real-time service quality and usage of a network down to telecom network cells, subscribers and devices [9].

The pre-processing steps are chosen by taking into account *a priori* knowledge about the application domain as well as the relative importance of the variables. Traffic data measurements from a live network are used for the experiments.

Different techniques of filtering the training dataset and their impact on the detection of anomalies are studied, and a novel method of filtering the training set using a newly introduced metric called the *failure significance metric* is proposed.

This thesis studies the classification of telecom network cells into groups based on their traffic levels, and its impact on the quantity and quality of anomalies detected. A comparative method of measuring the quality of anomalies is formulated.

Lastly, this thesis examines the impact of using different Key Performance Indicator (KPI) scaling techniques on the anomaly detection results and suggests an optimal scaling method for the data obtained for *Nokia Serve atOnce Traffica.*

1.3 Structure of the Thesis

The rest of the thesis is structured as follows:

Chapter 2 provides the necessary background information required to comprehend this research work. This section introduces the telecommunication network application domain and moves on to discuss the topic of anomaly detection and the various pre-processing techniques that are commonly used, along with their characteristics. This chapter also discusses briefly two clustering techniques that have been used for this research.

Chapter 3 describes in detail the specific problem this thesis tackles and formulates a suitable evaluation criteria for measuring the effectiveness of the solutions proposed.

Chapter 4 introduces the novel approaches that were developed as part of this research.

Chapter 5 proceeds to further evaluate the effectiveness of the novel approaches using the criteria defined in Chapter 3.

Finally, Chapter 6 concludes the thesis with the key outcomes of the various experiments performed and providing recommendations for future work.

2 Background

This chapter provides the necessary background required to assimilate this entire research work in three subsections. In first Section 2.1 the application domain i.e. telecommunication network and its management are described in sufficient detail. Section 2.2 introduces the concept of anomaly detection in general and describes related work in anomaly detection from telecom as well as other similar domains. This section also gives details about *Self-Organizing Maps* which is used in this research work for anomaly detection purposes. Section 2.3 concentrates on the role of pre-processing techniques in anomaly detection experiments and lists several related work in pre-processing techniques used in anomaly detection.

2.1 Application domain: telecommunication network

Mobile technologies have become an integral part of our daily life. The quality and reliability expected by customers from mobile operators is high. To live up to this high expectation, operators have to invest in new infrastructure as well as maintenance of existing infrastructure. The keys to reliability, quality and dependability of telecom networks are in the management and operations of the network [10].

2.1.1 Telecommunication network overview

Telecom networks provide mobile services for a large geographical area like a state or a country. A mobile phone user using this service can move within areas of service without losing his/her connection to the network. *Base Transceiver Stations* (BTS) placed at various locations throughout the geographical area help in giving continuous coverage to the customers. A BTS has one or more transmitter-receiver pairs which are called as *Transceivers* (TRX). The area covered by a transceiver which is beamed through BTS antennas is called as a *cell* [11]. Fig 2-1 shows how cells are organized around the BTSs in the network.

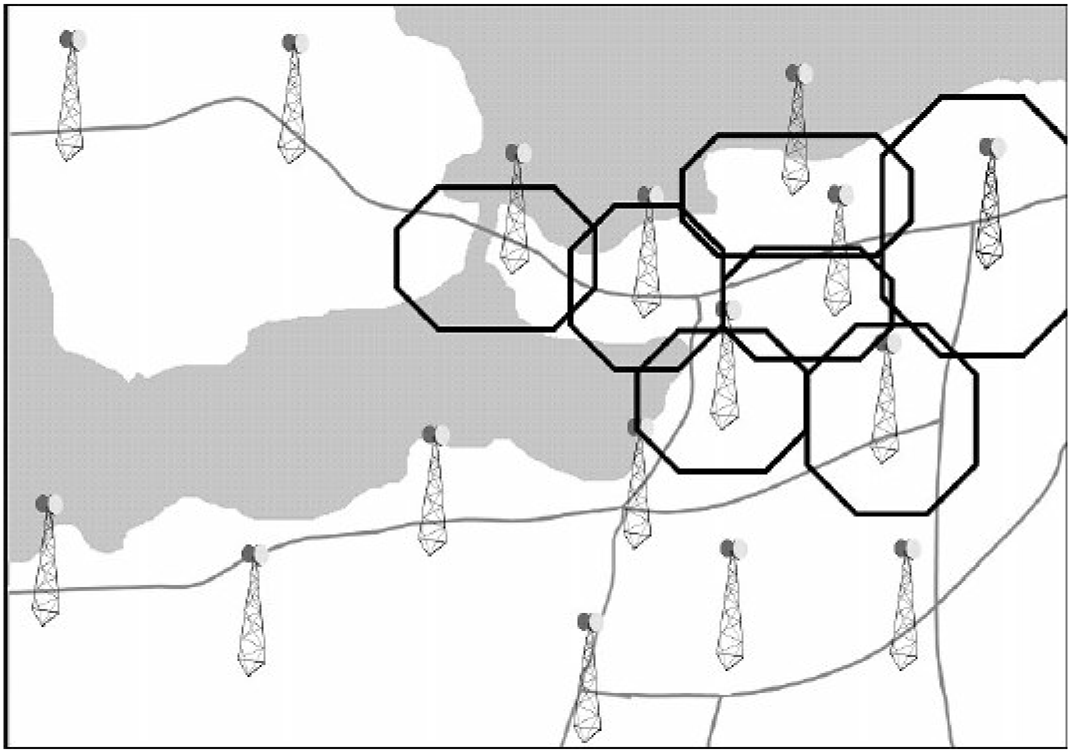


Fig 2-1 Arrangement of cells around BTS in a network **[11]**

When a call is made by a subscriber, a radio connection is created from the subscriber to the BTS. The BTS forwards the call to a transmission network which connects the BTSs in the network to each other as well as other networks [12]. Since the data used for the experiments in this thesis are from a live *Long Term Evolution* (LTE) network, a short description of the LTE network and its architecture, summarized from Palat and Godin [13] is provided in the subsequent paragraphs.

LTE is a project started by the telecommunication body known as *Third Generation Partnership Project* (3GPP) in 2004. LTE evolved from an earlier 3GPP system known as the *Universal Mobile Telecommunication System* (UMTS). The term "LTE" encompasses the evolution of the UMTS radio system through the Evolved UMTS Terrestrial Radio Access Network (E-UTRAN). On the other hand, the term "*System Architecture Evolution*" (SAE) encompasses the evolution of the non-radio aspects also known as the Evolved Packet Core (EPC) network. These two systems together comprise the Evolved Packet System (EPS).

The LTE network, which has been designed to support only packet-switched services, provides seamless IP connectivity between the User Equipment (UE) and the Packet Data Network (PDN) as shown in Fig 2-2. The EPS uses the concept of 'bearers' to route traffic from the PDN to the UE. These bearers are just IP packer flows with a defined Quality of Service between the UE and the gateway.

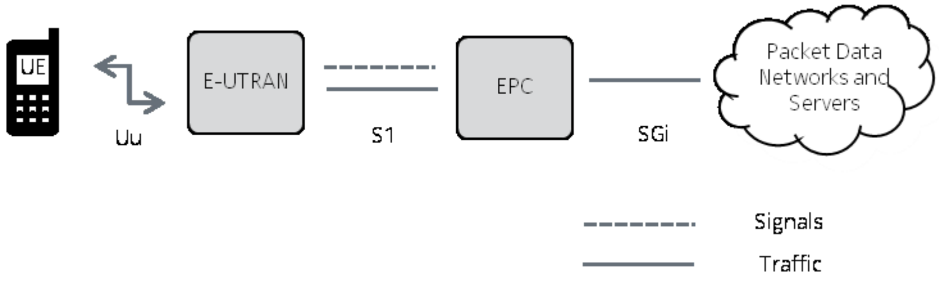


Fig 2-2 Simplified LTE network architecture

A more detailed representation of the network elements in an EPS network including the network elements and the standard interfaces is given in Fig 2-3. The whole network comprises of two parts: the Core Network (CN) and the Access Network (AN). The Access Network is made up of only one kind of node: the evolved NodeB (eNodeB) which connects to the UE through the U_u radio interface. All interfaces are standardized to facilitate multi-vendor operability.

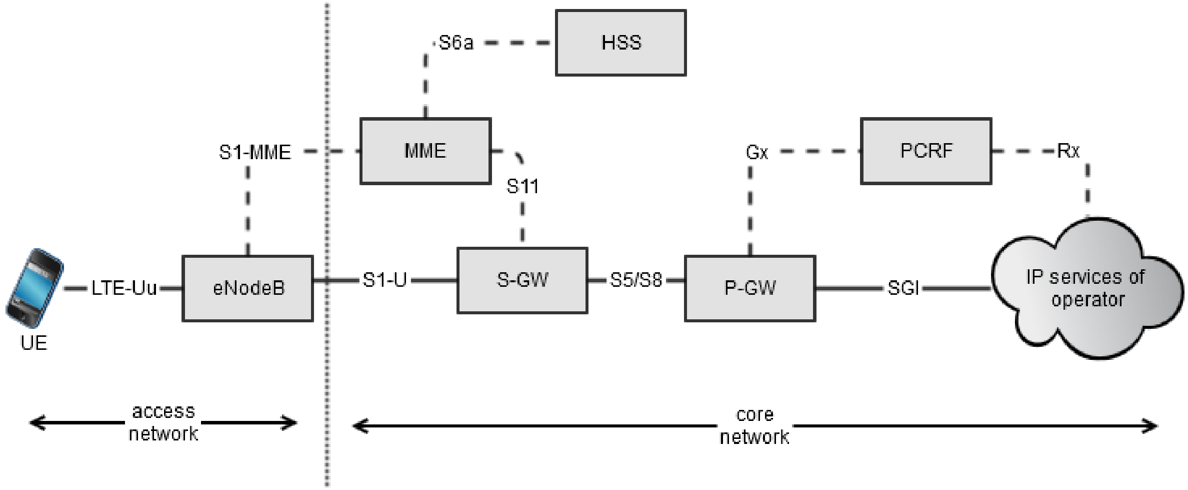


Fig 2-3 Evolved Packet System network elements **[13]**

The Access Network of LTE comprises of a network of eNodeBs as shown in Fig 2-4. This architecture is considered to a flat architecture as there is no centralized controller for the eNodeBs. The X2 interface connects the eNodeBs with each other and the S1 interface connects the eNodeBs to the Core Network.

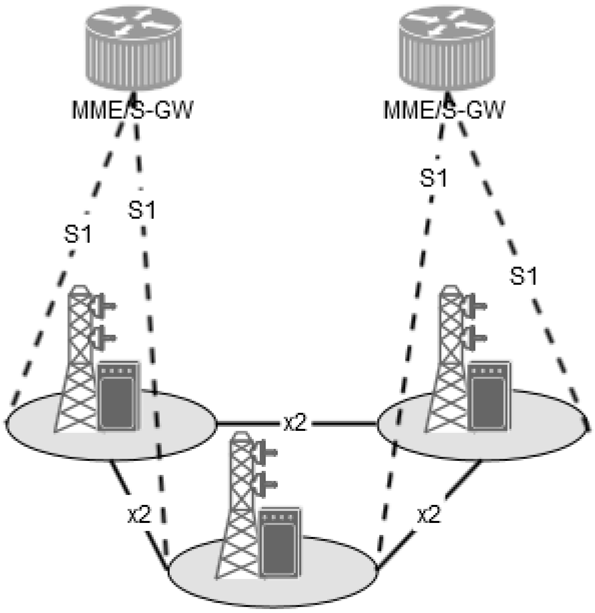


Fig 2-4 Overall E-UTRAN architecture **[13]**

The Core Network, also known as Evolved Packet Core (EPC), is responsible for the overall control of the UE as well as the EPS bearer establishment. The most important functionalities of some of the key components of the CN are:

- Mobility Management Entity (MME): As the control node that processes signaling between UE and the CN, the main functions related to this component are related to bearer, connection and session management.
- Serving Gateway (S-GW): This component acts as the local mobility anchor for data bearers when the UE moves between different eNodeBs.
- PDN Gateway (P-GW): This component is responsible for IP address allocation and QoS enforcement.
- Home Subscriber Server (HSS): This entity contains the users' subscription data such as QoS profile, roaming access restrictions and related content.

- Policy Control and Charging Rules Function (PCRF): This module is responsible for decision making, controlling policies and flows based on charging functionalities in the Policy Control Enforcement Function.

There are several other functionalities as well for all the components mentioned earlier, however, those are not within the scope of this thesis.

2.1.2 Telecommunication network management

The purpose of network management is to optimize the operational capabilities of a telecommunication network [14]. This includes ensuring peak performance for the network components, informing the operator of degradation, and finding possible reasons for problems using fault diagnosing tools [3]. Telecommunication Management Network (TMN) provides functions for management and communication between parts of the network and the OSS [15]. The transfer of information between the TMN and the NEs is carried out by the Data Communication Network (DCN). Examples of data carried out by the DCN include the NE configuration parameters which are sent to the NE from the Network Management System (NMS), performance report data sent from the NE to the NMS, alarm information sent from the NE to the NMS etc.

TMNs comprise of five key management functionality areas [14] [15] which are,

- Fault Management: The main management functionality of this component is to detect, log, notify users and if possible fix network issues.
- Configuration Management: This component monitors the system and network configuration information (software as well as hardware) so that the impact of various network operations performed on a day to day basis can be tracked.
- Accounting Management: The management functionality of this component is to measure network utilization parameters so that individual users or groups of users on a network can be regulated, billed or charged [16].

- Performance Management: This component is responsible for measuring the network performance data and making it available to the network managing entity so that an acceptable threshold of performance can be maintained.
- Security Management: This component controls the access to network resources based or pre-established organizational security guidelines.

A simplistic view of a network management topology is given in Fig 2-5. The agents (depicted as triangular shaped objects), also called as network management agents are software modules, that compile information about the devices in which they are located. This information is passed on to the Network Management System (NMS) via a network management protocol such as SNMP. The network manager controls these network management agents and ensures that, the appropriate information required for an efficient network management are collected. Once the required data reaches the network management servers, the data is correlated to provide meaningful information for analysis to the network administrator.

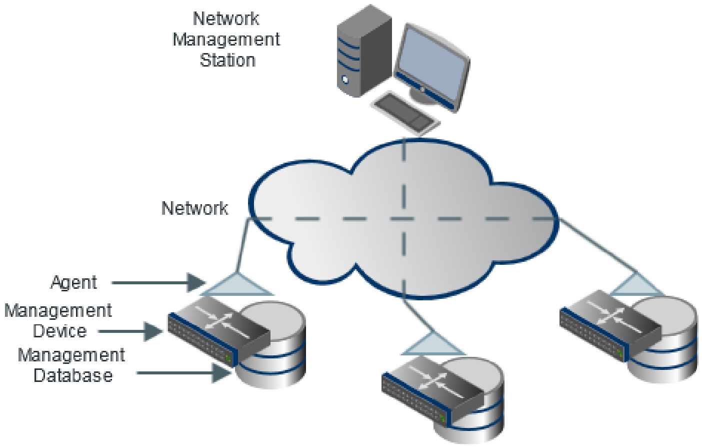


Fig 2-5 Simplistic view of network management topology

2.1.1 Key Performance Indicators

Data collected from the telecommunication network can be used primarily for two purposes [11]: support of operational decisions and control and accumulate knowledge of the application domain. The important events that happen in various NEs during the operation of the network are counted, which forms the raw low level data called counters. The time frames for the counting purposes are chosen depending upon different management purposes. Since the number of these low level data counters is too large and often unmanageable, several of these counters are aggregated to form high level counter variables called *Key Performance Indicators* [17]. Since these KPIs are used to represent understandable and easily interpretable functional factors, they are often given very descriptive names.

There are no standard formulas for calculating these KPIs as they are often kept confidential by telecom operators and equipment vendors. Since a KPI with the same name can be calculated in multiple ways using multiple formulas, it is not meaningful to compare two KPIs that are obtained from two different networks operated by two different operators. In some cases, these KPIs can be calculated using different formulas even in the same network [3].

2.1.2 Self-Organizing Networks (SON) and Self-Healing

Self-organizing networks (SON) aim at automation of network functions and operations through a closed loop control mechanism. In telecommunication network management systems employing the SON mechanism, the states of NEs are constantly monitored and depending on the state, certain actions are applied to the system. The logic for choosing the action is pre-fed into the system using a set of rules and conditions [18]. The emergence of new type of failures, and the need to address them with quickly, underlines the importance of reducing the extent of human intervention in the general network management process and fault management in particular.

Three main areas of SON include,

- Self-Configuration: This area of SON is applied during the initial phases of network deployment. Automatic configuration helps in shorter times for provisioning new network equipment.
- Self-Optimization: Self-optimization refers to managing a normally behaving system to improve its performance.

- Self-Healing: Self-healing refers to automatic detection, diagnosis as well as compensation of problems in network operation. The area of anomaly detection and classification described in Chapters 4 and 5 respectively, are relevant to this area of self-organizing networks.

2.2 Anomaly Detection

Anomaly detection is one of the four important tasks in data mining [19] and is also a crucial part of several industries dealing with process monitoring. Since the amount of data produced by industrial applications is usually of high dimensions and high volume, it is a cumbersome task for process operators to browse all the data manually. Hence, arises the need for automated applications, which find the most critical information from the whole lot of the data to support operators or other automatic systems in the decision making process. Automatic anomaly detection application can be described as a tool, that helps in filtering out a large part of the normal behavior and expose the abnormal behavior to the end user or the system [3].

In the context of mobile telecommunication networks, the volumes of data produced by a typical GSM network could be in the order of gigabytes [11] [1]. This data could be in several forms like KPI counters, audit logs, security logs etc. collected from various parts of the network. It is impossible for network operators to analyze the entire data manually. Traditionally, univariate tools have been used by operators which analyze a time series of observations and search for observations which have values above a certain predefined threshold. This resulted in storing, maintaining and periodically updating these thresholds to keep the system running efficiently.

These thresholds were set based on known errors, past observations and human judgment. The main drawback of this approach is that, only the most well-known errors, as well as the errors that have previously occurred, could be detected. An entirely new phenomenon could very well go unnoticed. This justifies the need for employing advanced anomaly detection mechanisms by telecom operators in their networks.

2.2.1 Application Domains in Anomaly Detection

Anomaly detection has found use in wide variety of applications. Examples of application of anomaly detection in network intrusion detection can be found in [20] and [21]. The use of anomaly detection in fault diagnosis can be found in [22], [23]. Several examples of anomaly detection in mobile network management can be found in [24], [25], [26], [27], [2] and [28]. Examples of usage of anomaly detection for fraud detection can be found in [29], [30], [31], [32]. An extensive list of application areas using outlier detection is described in [33] and [34].

2.2.2 Anomalies and Outliers

An anomaly is defined in the Oxford Dictionary as "*something that deviates from what is standard, normal or expected*" [35]. As such, the word anomaly has a wide scope which covers all abstract phenomena that are not expected [3]. An outlier on the other hand is defined as "*a data point on a graph or in a set of results that is very much bigger or smaller than the next nearest data point*" [35]. This definition of outlier is different from the definition of anomaly in the sense that it refers to a measured dataset.

Even though the definitions have different meanings, they both are used interchangeably, depending upon the point of view of the study. This is because, measuring an anomalous event in a process will result in an outlier in the recorded dataset [3]. The term *anomaly* is preferred over *outlier* in this thesis, and has been used even though both words mean the same in most contexts.

Fig 2-6 represents the scatter plot of an artificial two dimensional dataset. There are 4 visually distinguishable clusters in the dataset. A group of data points O1, which are located towards the center of the scatter plot and away from the 4 clusters can be considered as an outlier group. O2 and O3 are two data points that are located considerably away from the 4 clusters, as well as the outlier group O1, and can be considered as outliers.

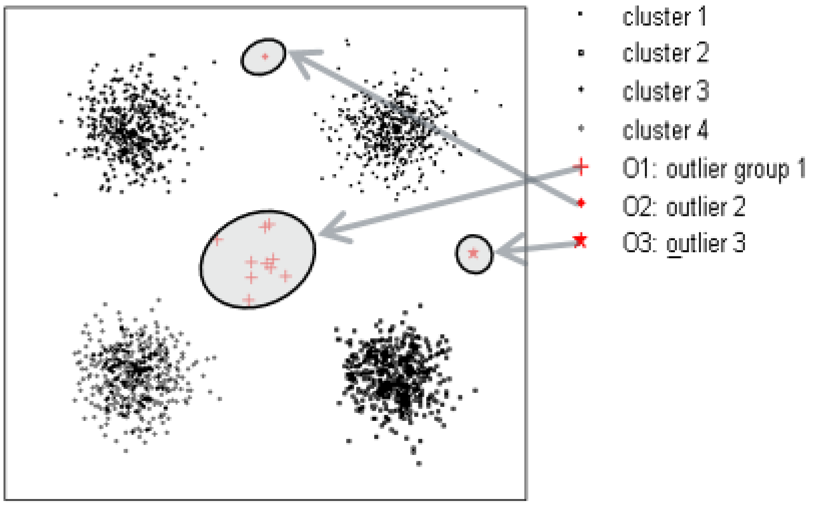


Fig 2-6 Simple example of two outliers and one outlier group in a two dimensional dataset

Anomalies in the data are often signs of errors or malfunctions in a process which could include errors in measurement devices, data transfer, data storage or even unauthorized usage of the system or its components [3] [1]. In the telecom network domain, anomalies could be the result of various problems in the network such as cell outages, traffic congestion, poor network coverage etc. Anomalies can also be result of unauthorized activities carried out by subscribers or third parties intentionally or unintentionally.

2.2.3 Anomaly Detection Methods

Anomaly detection methods can be classified on a number of different ways. One such classification is given in [21] where anomaly detection methods are classified into distribution based, depth-based, distance-based and cluster-based techniques. Another classification technique is based on dividing into model-based, proximity-based and density-based techniques [19]. Perhaps the most common and accepted division is based on the characteristics of the available data [33], [19] and [22] which classifies into anomaly detection techniques into three categories which are explained in the following sections.

2.2.3.1 Unsupervised Techniques

This kind of anomaly detection assumes that the normal state of the process is far more common than the anomalous states. Hence, the data points corresponding to the normal behavior are supposed to appear more frequently in the dataset than anomalous data points [1] [3]. To identify what the normal behavior corresponds to, a large dataset which represents the normal behavior of the system is first recorded.

However, in real life industrial applications, identifying the normal behavior of the system from the recorded dataset is not a straightforward task. The recorded dataset from applications can already contain anomalous data points which can fool the anomaly detection system into believing that the anomalous data points are also part of normal behavior. Yet another challenge with this approach is that, the normal as well as the expected behavior of the system, is not a constant vector. There could be multitude of factors which define what is normal. For example, traffic fluctuations in networks depending upon geographic, seasonal variation etc. can make this a rather complicated affair. Unsupervised anomaly detection techniques are the most commonly used [3] and has been adopted in this thesis.

2.2.3.2 Semi-Supervised Techniques

This kind of anomaly detection technique assumes prior knowledge of one of the classes, normal or anomalous. Prior knowledge in such a case would mean that, the normal class data points are usually labeled since all kinds or possible anomalies are not known beforehand in real life applications [3]. Since the normal behavior is identified from the model, observations that deviate markedly from the known model are considered as anomalous.

2.2.3.3 Supervised Techniques

Supervised anomaly detection methods require labeled reference data set for identification of both normal and anomalous classes. Observations in the reference data set have a label identifying them as normal or anomalous. With this data in hand, new analysis data is classified into either of the two classes based on a measure of distance or a similar metric [3]. These kinds of anomaly detection techniques can be considered as classification problems [19].

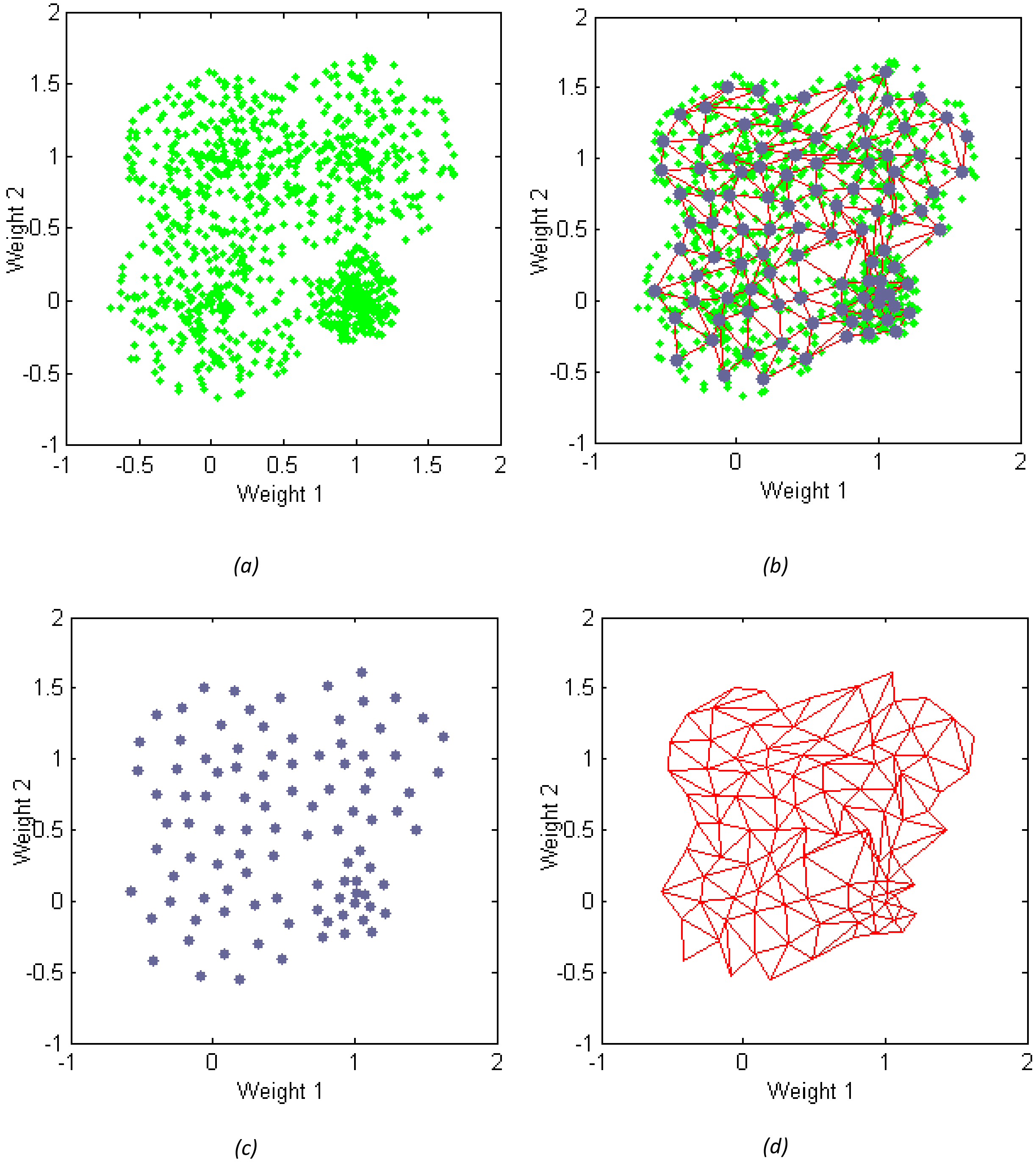
The single, most important, reason for adopting unsupervised techniques of anomaly detection in this research work is due to the absence of labeled datasets which say '*what is normal and what is not*'.

2.2.4 Self-Organizing Map (SOM)

Self-Organizing Maps are a type of neural network developed by *Dr. Teuvo Kohonen*, Emeritus Professor of the Academy of Finland, 1982. SOMs are a class of unsupervised systems based on competitive learning. The development of SOMs was motivated by a distinct feature in the human brain. Most parts of the brain are organized in such a way that different sensory inputs are represented by topologically ordered computational maps [6] [24]. SOMs are "Self-Organizing" because no supervision is required for them. They are "Maps" because they try to map the weights of the neurons to conform to the provided training data.

SOMs are very effective tools for visualization of high-dimensional data. They convert complex statistical relationships into simple geometric relationships on a low dimensional space. This is done in such a way that the most important topological and metric relationships of the primary data elements are preserved [6], [24]. SOMs have several important features, which make it a useful tool in data mining and exploration [36].

At first, neurons are placed along the nodes of a one or two dimensional lattice. It is possible to have higher dimensional lattices as well; however they are not so common. The weights of the nodes are initialized randomly (other ways of initialization also exist). A vector is chosen at random from the training set and presented to the network of neurons. The nodes in the network compete among themselves to find out which one's weight is most similar to the input vector and a winning neuron (also known as Best Matching Unit or BMU) is selected. A radius of neighborhood of the BMU is calculated. This radius starts with a high value and diminishes gradually for each time-step. Nodes within the radius of the BMU are then adjusted to resemble the input vector. This is done in such a way that the nodes closest to the BMU are adjusted more when compared to a node located farther from the BMU. This process continues on, till the positions of neurons provide an accurate statistical quantification of the input space [6].



(a)

(b)

(c)

(d)

Fig 2-7 Demonstration of SOM algorithm an example dataset of 1000 samples of 2 elements: (a) input data (b) overlapping plot of input, neuron positions and SOM structure (c) neuron positions (d) SOM structure

An example of the way a two dimensional SOM learns the structure of the input data is provided in Fig 2-7. This is demonstrated using the *Neural Net Clustering App* of *MATLAB 2014a.* Fig 2-7(a) represents the position of the input data points and Fig 2-7(c) represents the position of the neurons after 200 iterations of training.

Very detailed descriptions about the SOM algorithm including equations can be found in numerous text books [6], [37], [38], [39], [40], [41] and has been excluded from this chapter for the sake of brevity.

2.2.5 The Anomaly Detection System using SOMs

The anomaly detection system used in this thesis is based on the anomaly detection method by Höglund et al [24]. This method uses quantization errors of a one-dimensional SOM for anomaly detection. One-dimensional SOMs are used here because they are considered to be more flexible when compared to two-dimensional SOMs [3] [42]. The basic steps in the algorithm are listed step by step.

1. A SOM is fitted to the reference data of n samples. Nodes without any hits are dropped out.
2. For each of the n samples in the reference data, the distance from the sample to its BMU is calculated. These distances, denoted by $D_1 \ldots . D_n$ are also called as quantization errors.
3. A threshold is defined for the quantization error which is a predefined quantile of all the quantization errors in the reference data.
4. For each of the samples in the analysis data set, the quantization errors are calculated based on the distance D_{n+1} from its BMU.
5. A sample is considered as an anomaly if its quantization error is larger than the threshold calculated in step 3.

2.2.6 Reasons for choosing SOM

A comparative study of parameter sensitivity and overall robustness of four anomaly detection techniques Gaussian Mixture Model (GMM), Two Layer Clustering (2-LC), Local SOM and Clustering (L-SOM-C) and One-Class Support Vector Machine (OC-SVM) is provided in [3]. This experiment dealt with anomaly detection applied to data extracted from log files produced by network management servers.

Results of this experiment showed that, L-SOM-C and OC-SVM were the most sensitive in detecting novel behavior in the test data. Further, these results also showed L-SOM-C was one of the most robust method among the lot, in the sense that it was significantly less sensitive to changes in parameter values than the other three methods [3]. Since the data used in this experiment is obtained from the database of network management server's database, the benefits of robustness of SOMs as well as the decreased parameter sensitivity are expected to hold.

SOMs have also been successfully applied to several industrial process monitoring tasks [43] and also in the classification of mobile cells [44]. Anomaly detection experiments have been done with performance data from GSM network in the form of KPIs, and were found to be highly beneficial [36]. They have also been used in anomaly detection from server log data as well as radio interface data [42].

SOMs have been around for a long time and there are very good tools for experimenting with them and visualizing the results. The *Neural Net Clustering App* of *MATLAB* uses SOM for clustering. This has proved to be a good tool in understanding the impact of several pre-processing techniques on the structure of SOMs and their properties, as can be observed in the following sections. Moreover, several enhancements have been done to the initial *Kohonen*'s SOM algorithm by various research works to improve the speed, scalability and performance [45] [46] [47].

2.2.7 Relevance of the training set

SOMs, as well as the other neural network models, follow the "*garbage in - garbage out*" principle. If poor quality data is fed to the SOM, the result will also be of poor quality [5]. For any meaningful anomaly detection technique, the results heavily depend on the quality and completeness of the training data set.

Fig 2-8 and Fig 2-9 depict hourly traffic variations: number of total voice calls and number of active cells during a 24 hour period monitored from a group of *12134* cells in a mobile network. As can be seen clearly, traffic is lowest at night and rises gradually throughout the day when it reaches its highest point around 16:30 hours and then reduces during the rest of the day.

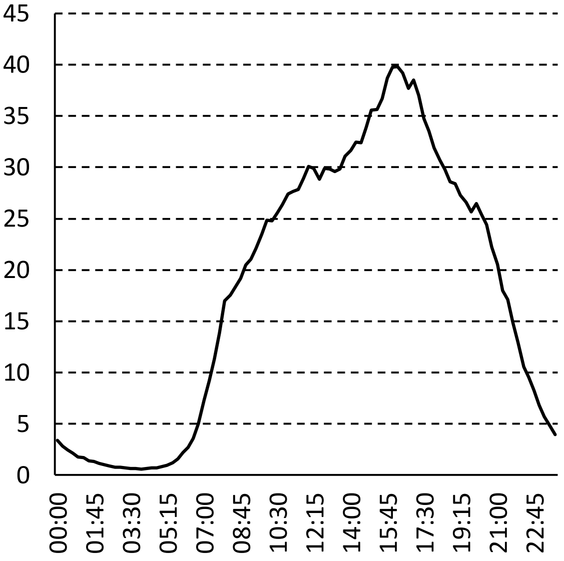


Fig 2-8 Hourly traffic variation (number of calls) in thousands

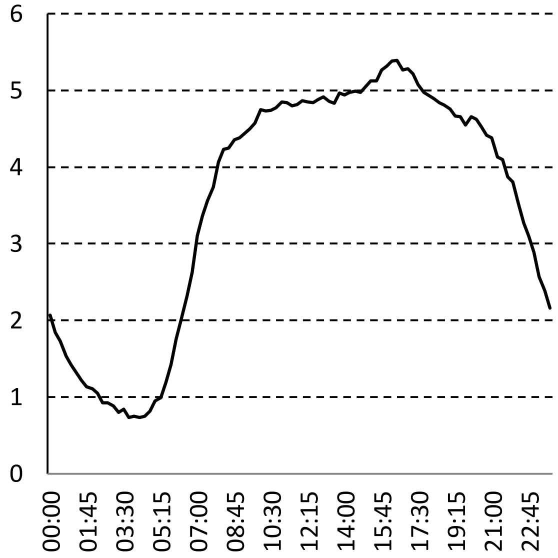


Fig 2-9 Hourly traffic variation (number of cells with calls) in thousands

Putting this in the context of anomaly detection, it is not optimal to use a sample of 1 hour time window's data taken during 16:00 – 17:00 hours for training the SOM and then analyze another 1 hour time window's data (e.g. 00:00 – 01:00 hours) with entirely different traffic patterns and volume. Thus the training and analysis datasets should be identical.

Extending this reasoning to a longer duration of 1 week, the traffic patterns observed during weekdays are different from that of weekends as shown in Fig 2-10 and Fig 2-11. Hence training the SOM with data measured during weekdays and analyzing the traffic during weekends might lead to unexpected results.

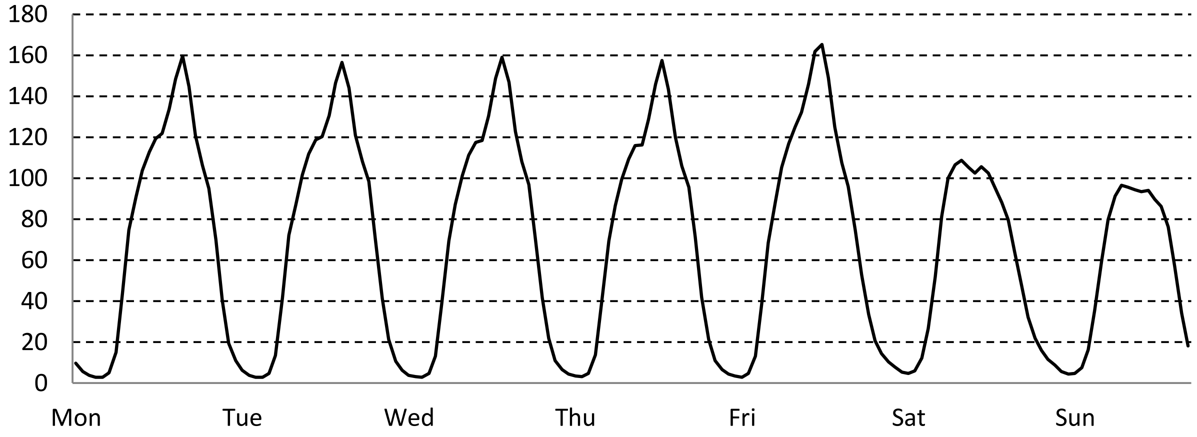


Fig 2-10 Weekly traffic variation (number of calls) in thousands

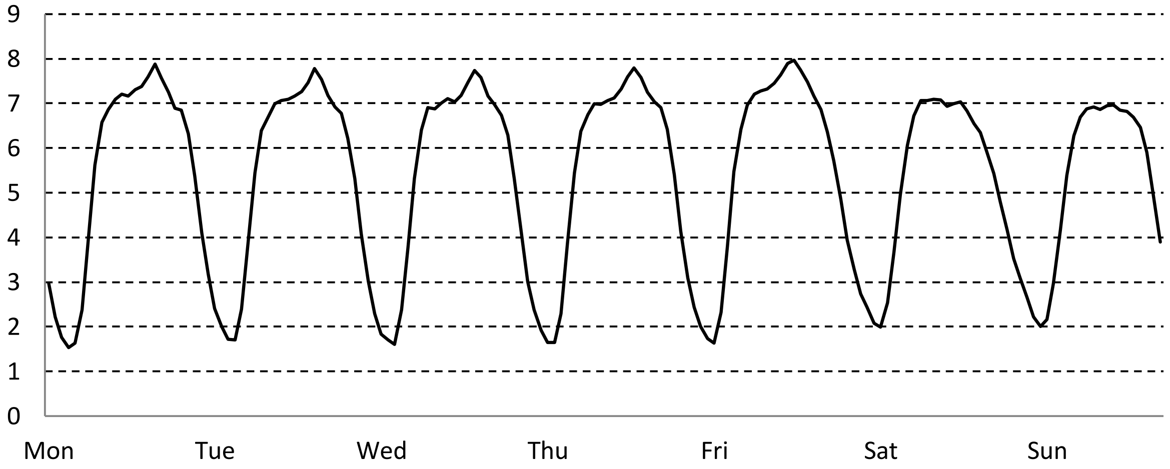


Fig 2-11 Weekly traffic variation (number of cells with calls) in thousands

In this thesis, special care has been taken to make sure that the training and analysis datasets are similar. If the analysis data set is of longer duration: for e.g. 3 weeks, the SOM is trained with data of at least 1 week duration. Detailed analysis of the variation in traffic patterns and anomaly detection based on the variation of traffic patterns is documented in [48] and [42].

2.3 Pre-processing techniques

Pre-processing techniques are a critical part of all anomaly detection processes. The main purpose of the pre-processing phase is to ensure that during the analysis stage, the required and proper information are extracted [49]. Pre-processing dictates what levels of variance are significant for each variable in the dataset and also what data points are suitable for analysis. It is unfortunate that despite of its importance to data analysis applications like anomaly detection, issues of pre-processing are commonly passed without discussion in scientific literature [50]. In several cases, the usage of simple distance-based methods in anomaly detection can provide sufficient performance if the variables have been pre-processed properly [5].

There are a number of important pre-processing techniques as defined in [51] (see Fig 2-12). Some of the most important ones are

- data cleaning
- data integration
- data transformation
- data reduction

Data cleaning is the process of cleaning the data to remove noise and fix inconsistencies. This consists of populating missing values, identifying and removing outlier data from the training set, smoothing noisy data and resolving inconsistencies. Data integration, on the other hand, is the process of merging data from multiple sources into one coherent data set.

Data transformation is the process by which data are transformed or consolidated into forms appropriate for data mining. Scaling and weighting of variables by their importance etc. come under the broader area of data transformation. Finally, data reduction techniques are applied to obtain a reduced representation of the entire data set that is much smaller in volume and closely maintains the integrity of the original dataset [51]. This thesis concentrates on data cleaning and data transformation.

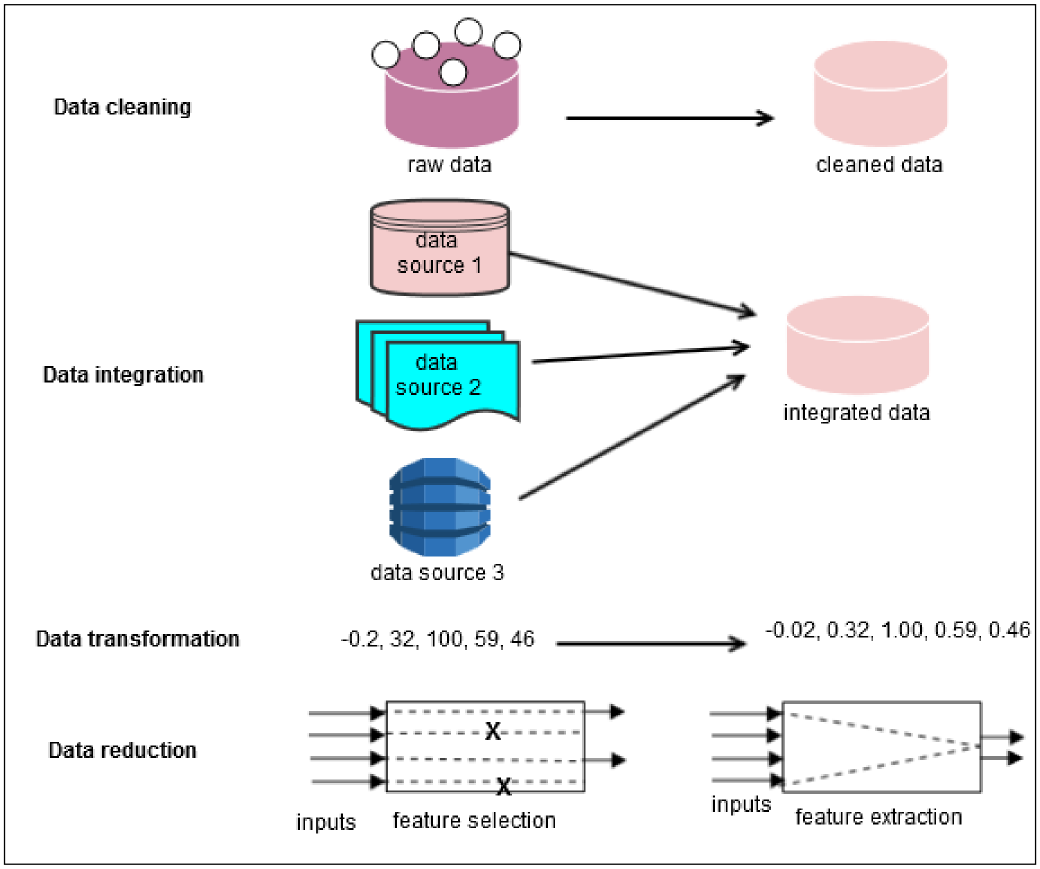


Fig 2-12 Different techniques of data pre-processing **[51]**

2.3.1 Filtering the training set

This section of the thesis focuses on some techniques of cleaning up the training set before the training set is actually used in the anomaly detection process. The presence of outliers in the training set can dominate the analysis results and thus hide essential information. The detection and removal of these outliers from the training set results in improved reliability of the analysis process [36]. This task of removal of outliers from the training set is referred to as '*training set filtering*' in this thesis.

The objective for *training set filtering* is to remove data points that have a particular pattern associated with them, from the training set, so that similar behavior in the analysis dataset is detected as anomalies. The *training set filtering* techniques evaluated as part of thesis are listed in the following sections.

2.3.1.1 SOM smoothening

This is one of the widely used training set filtering techniques. This technique is a generic one and does not use any application domain knowledge in filtering the training set. The steps in this process are:

Step 1: The entire training set is used in the training to learn the model of the training data.

Step 2: Anomaly Detection is carried out on the training set to filter out the non-anomalous points using the model generated in *Step 1*.

Step 3: The non-anomalous points from the training set obtained in *Step 2* are used once again to generate a refined model of the training set.

Step 4: This new model is used in the analysis of anomalies in the analysis dataset

Fig 2-13 represents a view of entire process.

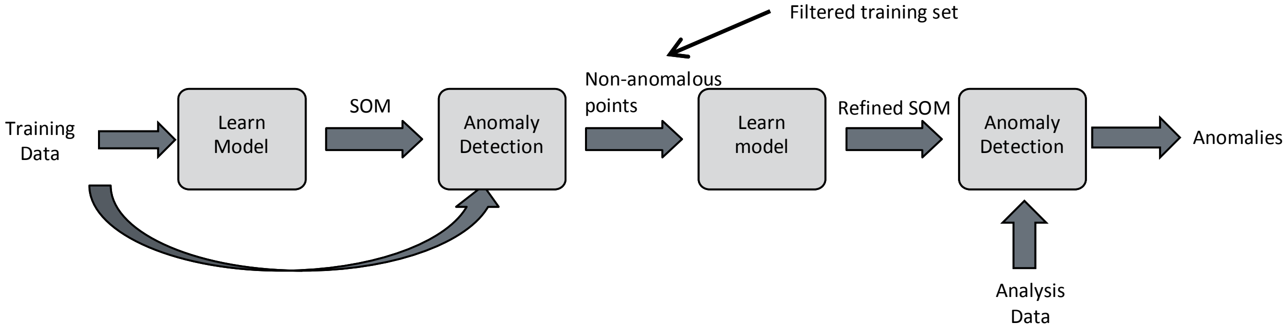


Fig 2-13 SOM smoothening technique of training set filtering

2.3.1.2 Statistical Filtering techniques

The general assumption behind this kind of techniques is that extreme values in either side of distribution are representative of anomalies. Hence statistical filtering techniques try to eliminate extreme values from the training set to reduce their impact on scaling variables. Two statistical filtering techniques are evaluated as part of this thesis: one which uses application domain knowledge and another which does not.

Fig 2-14 represents the process flow of statistical filtering techniques

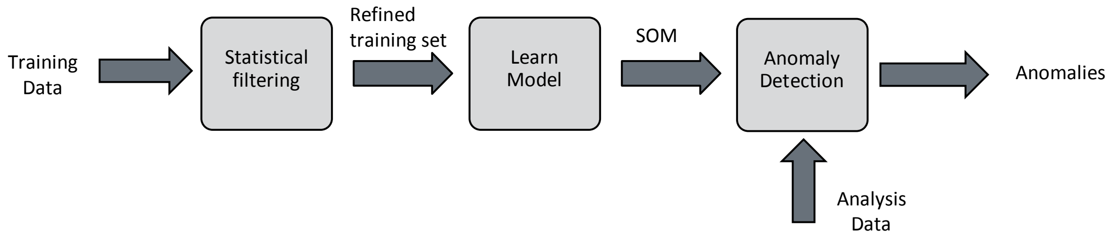


Fig 2-14 Statistical filtering techniques

Percentile based filtering: This kind of filtering removes $\boldsymbol{k}$ % of the observations of each of the KPIs from either side of the distribution.

Failure ratio based filtering: This kind of filtering removes $\boldsymbol{k}$ % of the observations from the training set which correspond to the highest failure ratio of a relative KPI. The failure ratio based technique of filtering from the training set can be considered to be a technique that uses the application domain knowledge in filtering the training set as high failure ratios are considered as anomalies in networks.

2.3.2 Scaling

Distance-based methods are very often used in unsupervised anomaly detection methods using SOM. The results of the anomaly detection experiment are dependent on the distance metrics and the scaling of the variables. "*In many cases very simple methods can provide sufficient performance if the variables have been scaled properly*" [5]. Unfortunately scaling and weighting of variables is a neglected and underrated part in research [5] [3] [50]

Proper scaling methods should make all the variables of equal importance within the problem in which they are used [5]. In cases where sufficient application domain and process knowledge is available to know beforehand the relative importance of variables, weighing of variables by their importance etc. can be used as well in anomaly detection.

There are numerous methods of scaling and explaining all of them in detail is not in scope of this thesis. Some of the most important ones that have been used for anomaly detection techniques in related fields are described briefly in the following sections.

2.3.2.1 Mean center scaling

As the name of this scaling method suggests, the mean of the variable is subtracted from each of the variables to obtain mean centered data. Mathematically this kind of scaling is found out using the formula in equation 2-1

$$x_s = x - \bar{x} \qquad 2\text{-}1$$

where,
x : un-scaled value
$\bar{x}$: mean
x_s : mean center scaled value

This scaling method takes the relative change of the variable into consideration and does not consider the magnitude.

2.3.2.2 Linear scaling

This kind of scaling divides the variable by its standard deviation as shown in equation 2-2

$$x_s = \frac{x}{s_x} \qquad 2\text{-}2$$

where,
x : un-scaled value
s_x : standard deviation
x_s : linear scaled value

2.3.2.3 Z-score scaling

This kind of scaling is one of the most commonly adopted scaling techniques in anomaly detection experiments [3] [50]. In this process, each of the scaled variables will have zero mean and unit variance which is achieved by subtracting the mean and dividing by the standard deviation as shown in equation 2-3

$$x_s = \frac{x - \bar{x}}{s_x} \qquad 2\text{-}3$$

where,
x : un-scaled value

$\bar{x}$: mean
s_x : standard deviation
x_s : z-score scaled value

This process is also known as *normalization, standardization* and *auto-scaling* and the scaled variables are referred to as *standard scores* or *z-scores.* This scaling technique can be considered as a combination of *mean center scaling* and *linear scaling.* Much of the research work in the area of anomaly detection does not cite reasons for the use of *z-score* scaling in pre-processing stage. This process is accepted as a standard procedure without further investigations whether it is the most appropriate method for the specific case or not [5] [3].

2.3.2.4 Scaling by range

In this kind of scaling, the range of the variable is used for scaling so that the scaled variable x_s will always fall between 0 and 1 as shown in equation 2-4.

$$x_s = \frac{x - min(x)}{max(x) - min(x)} \quad \text{2-4}$$

where,
x : un-scaled value
$min(x)$: minimum value of variable in the distribution
$max(x)$: maximum value of variable in the distribution
x_s : range scaled value

This method has been shown to give best results in clustering of artificially generated data [52] [53], however it was outperformed by *z-score* scaling in anomaly detection [54]. The main drawback of this method of scaling is that it is very sensitive to the presence of outliers in the data. As such this kind of scaling is rarely used in telecommunication applications which require more robust scaling methods [3].

2.3.2.5 SoftMax-scaling

The *SoftMax* scaling technique is a method of squashing data that is unevenly distributed into an appropriate range. A variable in the dataset is squashed to a degree depending on two factors: (i) distance from the mean and (ii) standard deviation of the dataset. A variable located far from the mean is squashed more when compared to a variable which is located closer. A larger degree of scaling is required for a dataset which has larger standard deviation [55].

The *SoftMax* scaling function scales values which fall outside the designated range to values close to one or zero and thus forms an *S*-shaped function. This is based on the logistic function given by equation 2-5

$$x_s = \frac{x - \bar{x}}{\lambda\, s_x / 2\pi} \qquad 2\text{-}5$$

where,
x : un-scaled value
$\bar{x}$: mean
s_x : standard deviation
λ : size of the designated range
x_s : SoftMax-scaled value

2.3.2.6 Logarithmic sigmoid (LogSig scaling)

The Logarithmic sigmoid scaling (*LogSig-scaling*) attempts to integrate expert knowledge into the data analysis process. Here the scaling is based on a nominal range defined by the user for each variable. This approach resembles *SoftMax*-scaling (refer section 2.3.2.5) as both use an *S*-shaped function to scale the data. The main advantage of this type of scaling is that it eliminates the influence of outliers. Though the *LogSig*-scaling is more laborious to the user, it operates more robustly when compared to *SoftMax*-scaling because the latter takes the variance of the variable into account whereas the former does not [50]. The *LogSig*-scaling is defined by equation 2-6

$$x_s = \frac{1}{1 + e^{-2(x-c)/r}} \qquad 2\text{-}6$$

where,
x : un-scaled value
c : center of the nominal range
r : radius of the nominal range
x_s : scaled value

This function serves two purposes: firstly, it scales the outliers close to the main body of the data thereby reducing its effect on the anomaly detection process; secondly the values in the dataset are transformed to comparable ranges thereby allowing algorithmic analysis [50].

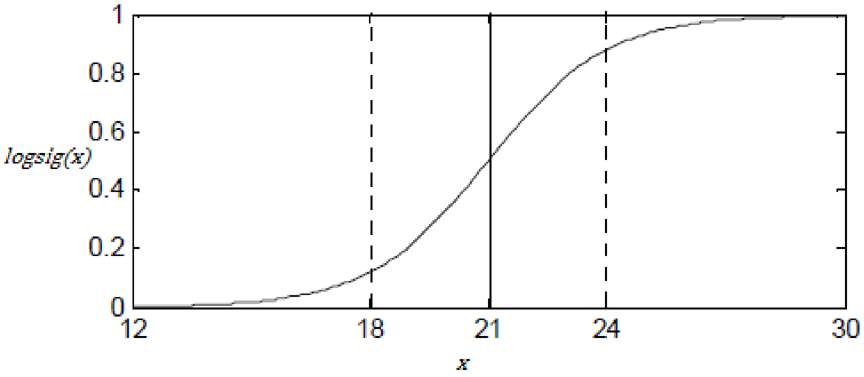


Fig 2-15 LogSig-scaling of variable x with nominal value = 21, variation = 3, range = (18 – 24) **[50]**

The main advantage of this process is that it gives valuable information about the main body of the data which corresponds to the normal mode of operation of the system. However, the disadvantage of this approach is that the scaling information is needed before any of the further steps could be performed. Gaining information about this scaling information could be a difficult task because the user needs to have sound understanding of the process before choosing the nominal ranges [50].

2.3.2.7 Robust scaling techniques using median

The central motivation behind several robust scaling techniques using median as well as the robust logarithmic scaling is to reduce the effect of anomalous data points on the anomaly detection process. The presence of anomalous data points in the training set can completely distort the estimates of mean and standard deviation used in z-scores [3] [56] [54] [57]. The values of Median Absolute Deviation (MAD) about the median is considered as a robust way to describe the variation of the data [56]. This values $MAD\ (x)$ is defined using equation 2-7

$$MAD\ (x) = Med\{|x - Med(x)|\} \qquad 2\text{-}7$$

where,

x : un-scaled value

$Med(x)$: median of the distribution
$MAD(x)$: median absolute deviation about median for x

This value of $MAD\ (x)$ can be used in a robust scaling utilizing the median which is represented by equation 2-8

$$x_s = \frac{x - Med(x)}{MAD(x)} \qquad 2\text{-}8$$

where,
x_s : robust MAD scaled value of c

A normalized variation of $MAD\ (x)$ represented by $MADN\ (x)$, is used which is referred to as "normalized MAD" $MADN\ (x)$, which is scaled in such a way that $MADN$ for the normally distributed variables is equal to the standard deviation [56].

A trimmed version of the mean also has been used wherein a predefined portion α of the highest and lowest vales are excluded from the calculation. Another example of a similar case can be found in [26], where the authors use trimmed mean for the calculation of standard deviation. This technique also called as the '*Interquartile range (IQR) technique*' of estimating the deviation, eliminated one quarter of the observations from both ends of the ordered set.

2.3.2.8 Robust Logarithmic scaling (RLog scaling)

Robust logarithmic scaling as defined in [26] is a technique that attempts to preserve the importance of the variables used in the analysis. *Z-scores* as defined in Section 0 suffers from the inherent problem that variables with small variance are amplified and hence could be over-valued in the analysis. Similarly, variables with high variance are attenuated and could be under-valued in the analysis. Robust logarithmic scaling attempts to solve this problem using the scaling method proposed in equation 2-9.

$$x_s = \frac{\ln(x+1)}{s_x} \qquad 2\text{-}9$$

In the above equation $s_x = \ std\ \{\ ln\ (x+1)|x > 0, x < q_{99}\ \}$ and q_{99} refers to 0.99 quantile of variable x. Adding of 1 in the equation is to eliminate the need to separately handle zeroes in the data. The standard deviation ignores the zeroes in the distribution along with 1% of values from upper tail. The scaled value is further derived by equation 2-10.

$$x_s = x_{logs} - mean\{x_{logs}\} \qquad 2\text{-}10$$

An example of the effect of the *RLog scaling* technique is shown in the Fig 2-16. The auto-scaling approach as shown in the center is highly affected by the presence of outliers (3 of them as can be observed towards the right of the scatter plots). This hides important information regarding other possible variations in the system behavior. The robust logarithm approach as shown in the right shows meaningful variation in the general body of the data and at the same time keeps the outliers away from the rest of the data distribution [26]. This technique could prove handy in cases when detecting outliers in a variable that does not have a high variance.

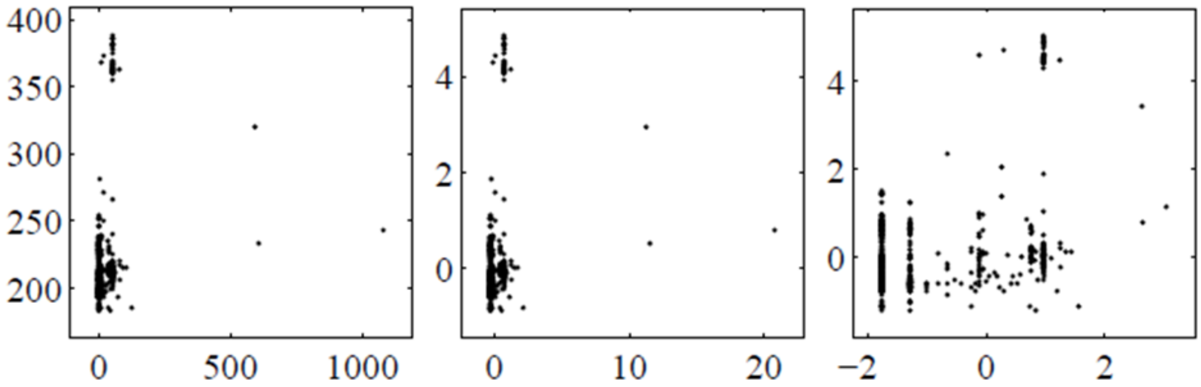


Fig 2-16 Scatter plot of two variables on three scales, no scaling (left), z-score scaling (center) and robust logarithm scaling (right) **[26]**

2.3.3 Weighing variables by importance

Often in data analysis of real world applications, all the variables analyzed are not of equal significance. Consider a mobile network traffic measurement data which comprises of multi-dimensional data among which we choose two different variables x_{sum} which sums up the total duration of all calls that occurs in a cell and another variable x_{dc} which signifies the number of calls that have been dropped in the cell.

For a telecom network monitoring personnel, the variations in the variable x_{dc} are likely to be of more significance compared to that of the variable x_{sum} since the former is more indicative of failures in the network. The importance of a variable in anomaly detection can be adjusted by multiplying each scaled variable x_s by a weight factor corresponding to its relative importance. This is one way of incorporating knowledge of application domain and the significance of a variable in analysis into the anomaly detection process.

2.3.4 Data segregation through clustering

The process of separating data points into subsets that have a meaning in a particular problem's context is called cluster analysis [58]. Each cluster thus obtained represents a group of data points that have similar behavior. There are a variety of clustering mechanisms and algorithms which are used for different kinds of scenarios. All these methods try to achieve the same goal *i.e.* internal homogeneity and external separation through different mechanisms [59]. Two clustering techniques which are relevant to this thesis are listed in the following subsections.

2.3.4.1 K-means clustering

The *k-means* clustering technique is an example of a partitional clustering technique. Partitional clustering technique generates a single partition of the data in an attempt to recover natural groups present in the data. The partition is achieved using a certain objective function. The *k-means* clustering method partitions the dataset into *k* clusters which are mutually exclusive. Each of the obtained clusters are classified by the data points which belongs to the cluster and the centroid, or center of the cluster. A cluster centroid is defined as the point in the dataset to which the sum of the distances from each of the data points in the cluster is the minimum [60].

The *k-means* clustering technique uses an iterative algorithm which attempts to minimize the sum of distances from each of the data points to its cluster centroid. The data points are moved from one cluster to the other until the sum of distances cannot be decreased any further. As a result of this, each of the detected clusters end up being as compact and well separated as possible [60]. Detailed explanations about the algorithm can be referred in [58] and [59] and have been excluded for the sake of brevity.

The Davies-Bouldin Index (DBI) [61] is a commonly used measure which indicates the similarity of clusters. This is often used as a measure of the appropriateness of data partitions in those cases, where the density of data points in a cluster decreases as the distance from a vector characteristic of the cluster increases. The Davies-Bouldin criterion for cluster evaluation is based on a ratio of distances within a cluster and between clusters [61]. The DBI is defined by the equation 2-11, where D_{ij} the within-to-between cluster distance ratio for *ith* and *jth* clusters mathematically depicted as equation 2-12.

$$DBI = \left(\frac{1}{k}\right)\sum_{i=1}^{k} \max_{j\neq i} \{D_{ij}\} \qquad 2\text{-}11$$

$$D_{ij} = (\overline{d}_i + \overline{d_j})/\, d_{ij} \qquad 2\text{-}12$$

The average distance between each point in the i^{th} cluster and its centroid is denoted by $\overline{d_\iota}$ and $\overline{d_J}$ denotes the average distance between each point in the i^{th} cluster and the centroid of j^{th} cluster. Smaller values of DBI denote that the clustering has been done efficiently [60] and this property of the DBI is used to find the optimum number of clusters into which a dataset can be divided using the *k-means* clustering algorithm.

2.3.4.2 Hierarchical clustering

Hierarchical clustering techniques construct clusters by recursively partitioning the dataset in either a top-down or bottom-up fashion [62]. The two main kinds of hierarchical clustering are

- Agglomerative hierarchical clustering: In this kind of clustering each data point starts off being considered as a cluster on its own. These data points are further merged until the cluster structure is obtained.
- Divisive hierarchical clustering: In this technique, entire dataset is considered as one cluster. The cluster is then divided into sub-clusters, which are successively divided into sub-clusters till the required cluster structure is obtained.

A dendrogram is obtained as a result of either of the two techniques which represents the nested grouping of objects as well as the similarity levels [62]. From the dendrogram a desired similarity level is chosen and the clustering of the data objects is obtained by cutting the dendrogram at a desired similarity level.

Ward's minimum variance method [63] is one such criteria that is often used [64] in agglomerative hierarchical clustering. This criteria minimizes the total variance within the cluster. During each iteration, the pair of clusters which on merging leads to the minimum increase in total within-cluster variance are found out and these are merged.

2.4 Conclusions

This chapter provided the necessary background information required to comprehend this research work. Knowledge about the telecommunication network and calculation of KPIs, is critical in understanding the telecommunication network performance measurement data. Some preliminary experiments are performed on this data using the anomaly detection system based on SOMs, discussed in Section 2.2.5. The results of this experiment are further used to formulate the problem description in Chapter 3.

3 Problem description and evaluation criteria

This chapter describes the problem this thesis studies, and the evaluation criteria that will be used to evaluate the effectiveness of the solutions proposed.

Firstly, the dataset used in this thesis is described in Section 3.1. The subsequent Section 3.2 analyzes the specific problems in pre-processing this dataset. Finally, Section 3.3 formulates the evaluation criteria that are used to measure the effectiveness of the solutions proposed for the problems.

3.1 Traffic measurement data

The anomaly detection tests done as part of this thesis were performed on mobile network traffic measurement data obtained from *Nokia Serve atOnce Traffica* [8]. This component monitors real-time service quality, service usage and traffic across the entire mobile network owned by the operator. It stores detailed information about different real-time events such as call or SMS attempts, handovers etc.

The real-time data, stored in large distributed databases is exported to files in such a way that one such file contains traffic measurements aggregated at different levels of the hierarchical TMN system. Based on the goal of the anomaly detection experiment as well as the monitored network functionalities, KPIs are then derived out of these files and they form the data on which the anomaly detection experiment is carried out. Mobile traffic measurement data was monitored from 12134 telecom network cells in total. Fig 3-1 represents a view of the entire process.

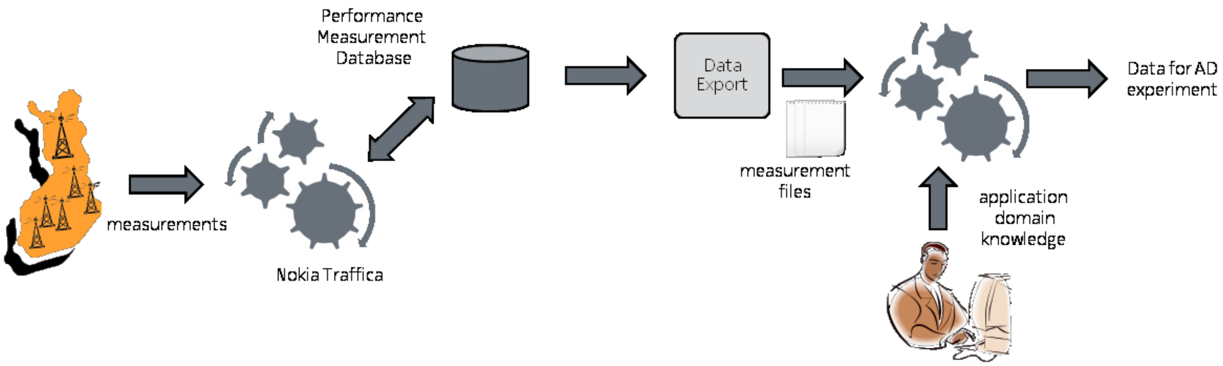


Fig 3-1 Process of deriving traffic measurement data for anomaly detection experiment

A set of 33 KPIs which are indicative of telecommunication activities like voice calls, handovers, SMS *etc.* were chosen for this thesis. Details about the names of the KPIs as well as their description are provided in Table 3-1.

17 of the 33 KPIs in the dataset represent measurement data related to calls. There are 5 KPIs which correspond to measurement data about handovers, 6 KPIs which correspond to SMS, and 5 KPIs which correspond to interruptions in operations. All KPIs except '*CALL_DURATION_SUM*' are counter KPIs, i.e., they have whole number values. 11 of the KPIs are positive indicators of performance and the remaining 22 indicate failures. The rows of the table are color coded to identify these positive and negative indicators.

Table 3-1: 33 Key Performance Indicators monitored for the anomaly detection experiment

Positive Indicator	**Negative Indicator**

KPI Name	**KPI Description**
OUTGOING_CALL_COUNT	Number of outgoing calls in the cell
INCOMING_CALL_COUNT	Number of incoming calls in the cell
ANSWERED_CALL_COUNT	Number of answered calls in the cell
CALL_DURATION_SUM	Total duration of all voice calls in the cell
VERY_SHORT_CALLS_COUNT	Number of calls with very short duration
SHORT_CALLS_COUNT	Number of calls with short duration
AVG_LENGTH_CALLS_COUNT	Number of calls with average duration
LONG_CALLS_COUNT	Number of calls with long duration
VERY_LONG_CALLS_COUNT	Number of calls with very long duration
CS_FAILURE_COUNT	Number of call set-up failures
CSC_FAILURE_COUNT	Number of call set-up failures due to errors in core network
CSR_FAILURE_COUNT	Number of call set-up failures due to errors in radio network
CSU_FAILURE_COUNT	Number of call set-up failures due to uncategorized errors
CD_FAILURE COUNT	Number of dropped calls
CDC_FAILURE COUNT	Number of dropped calls due to core network failures
CDR_FAILURE COUNT	Number of dropped calls due to radio network failures
CDU_FAILURE COUNT	Number of dropped calls due to uncategorized failures
HO_FAILURE_TO_NE	Number of failed handovers to the cell
HO_FAILURE_FROM_NE	Number of failed handovers from the cell
HOC_FAILURE_COUNT	Number of failed handovers due to core network problems
HOR_FAILURE_COUNT	Number of failed handovers due to radio network problems
HOU_FAILURE_COUNT	Number of failed handovers due to uncategorized reasons
SMS_SENT_COUNT	Number of text messages sent from the cell
SMS_RECEIVED_COUNT	Number of text messages received to the cell
SMS_FAILURE_COUNT	Number of failed text messages
SMSC_FAILURE_COUNT	Number of failed text messages due to errors in core network
SMSR_FAILURE_COUNT	Number of failed text messages due to errors in radio network
SMSU_FAILURE_COUNT	Number of failed text messages due to uncategorized errors in network
INT_TXN_CNT	Number of interrupted transactions
INT_TXN_FAILURE_CNT	Number of interrupted transactions due to failures
INT_TXNC_FAILURE_CNT	Number of interrupted transactions due to failures in core network
INT_TXNR_FAILURE_CNT	Number of interrupted transactions due to failures in core network
INT_TXNU_FAILURE_CNT	Number of interrupted transactions due to uncategorized failures in network

3.2 Problem description

The raw data exported from databases needs to pass through several levels of pre-processing before it can be used for a comprehensive anomaly detection experiment. Experiments done as part of this thesis (see Table 3-2) depicted that pre-processing techniques such as data cleaning, filtering of training dataset, scaling, weighting etc. have a huge impact on the number and the quality of anomalies detected.

As discussed in earlier sections, failures are a commonplace in telecom networks and hence telecom measurement data taken from live networks for training purposes are very likely to have high values for its failure counter KPIs and high failure ratios as well.

Studying the impact of different training set filtering mechanisms, on the quantity as well as quality of the anomalies detected is one of the important prerequisites for choosing a filtering technique for the data in hand.

Removing data points from NEs with high failure rates has a good impact on the quantity of anomalies detected as shown in Table 3-2. In this experiment, 1% of the data points which had highest failure ratio were removed from the training set. If the training set contains a significant number of NEs with poor performance (e.g. high failure rates), then the training phase assumes such poorly performing cells also as part of the norm and such cells are not detected in the analysis phase.

Table 3-2 Impact of filtering training set based on failure ratio

Scenario	Result
No training set filtering	**121** anomalies from **11** NEs
1% filtering using failure ratio	**573** anomalies from **121** NEs

With such a huge impact of training set filtering on detecting anomalies, it is important to choose the right way of eliminating the anomalous data points from the training set.

What are the appropriate filtering techniques that, when applied to telecom network measurements, lead to the detection of most meaningful anomalies?

This thesis analyzes different training set filtering techniques to find a solution to this problem.

The traffic in telecom networks vary highly depending on several factors such as time of the day, geographical location of the cell etc. [48]. Cells located in high traffic regions like city center can have huge differences in traffic patterns and KPI counters from cells located in suburban areas. In such cases, if there is not enough number of cells with the high traffic in the training set, the impact of analyzing the entire dataset for anomalies can have unpredictable values.

Moreover, there is a need for aggregating data counters at distinct intervals for cells with different traffic. This need can be justified with a sample fictitious case. Consider two cells *X* and *Y* located at distinct locations: *X* in the city center and *Y* far away from the city. Both of these cells have a traffic aggregation interval of 10 minutes. Two traffic records having the value of the KPI '*CALL_DURATION_SUM = 0*' for both the cells *X* and *Y* appears on the screen of a network monitoring personnel during peak traffic hours. These two observations definitely carry different meanings. The zero value of call duration record from the latter is very likely to be a reason of its geographical location, whereas that from the former could be due to malfunctioning network equipment.

Large aggregation intervals are not always optimal, as faults in cells with high traffic cannot be detected earlier.

How can early as well as meaningful detection of anomalies from cells of varying traffic types be done effectively?

Finding an answer to this question is the second main objective of this thesis.

Details about several data scaling techniques were given in Section 2.3.2. Choosing the appropriate scaling method for the KPI values in an anomaly detection experiment is tricky. This choice often depends on the distributions of each of the KPIs. The scaling method also defines the kind of anomalies that get detected as part of the analysis.

How can the anomaly detection capabilities of scaling techniques be compared? Which scaling techniques work well for telecom network measurement data and which do not?

Comparing different scaling techniques to find their effectiveness in anomaly detection is the third and final objective of this thesis.

3.3 Evaluation Criteria

Measuring the effectiveness of a pre-processing technique in detecting anomalies is a challenging task. What is considered as an anomaly depends on what is expected; the number of observations and the underlying distribution of the dataset [3]. Thus it becomes very significant to understand what the normal behavior of an entity is, before deriving how anomalous an observation of the entity in the dataset is. What might seem as an interesting anomaly to one person may not seem as an anomaly to another. The superiority of the different anomaly detection techniques is solely based on the subjective assessment by the end user [5].

However, to evaluate the effectiveness of a pre-processing technique in a consistent as well as unbiased manner, a three stage evaluation technique is adopted which is described in detail:

Quantitative evaluation: This sort of evaluation aims to find absolute as well as relative measures of performance in terms of numbers. Since reference data with labeled anomalies are not available for analysis, the thesis aims to find the **accuracy in detecting synthetic anomalies** that are inserted into actual data.

Firstly, a group of anomalous NEs are modeled based on certain parameters like traffic level, failure rates etc. Data points corresponding to these anomalous NEs (which in turn are anomalies) are programmatically generated and inserted into the analysis dataset at regular intervals.

The accuracy in detection of synthetic anomalous NEs as well as their observations from the analysis dataset would then give a quantitative measure of performance of a pre-processing technique.

Another metric of quantitative evaluation that will be used is the number of non-synthetic anomalous NEs and their observations that get detected using a pre-processing technique. The number of anomalies detected should not be too high or too low. The ideal number is expected to depend on the time duration of analysis, number of cells, number of elements in the dataset etc.

Qualitative evaluation: This type of evaluation aims at finding the overall **quality of anomalies** detected as a result of a pre-processing technique. The quality of anomalies will be perceived from a telecom network monitoring personnel's perspective. Two kinds of qualitative evaluation are adopted in this thesis.

The first, more scientific approach developed as part of this thesis is inspired from a technique of anomaly clustering used by *Kumpulainen et al.* in [64]. This technique uses **hierarchical clustering** to divide the anomalies detected into a fixed number of clusters such that the centroids of the clusters are sufficiently different from each other. The cluster centroids are then analyzed using a metric which defines the significance of the failures for KPIs like call setup failures, dropped calls, handover failures, SMS failures etc.

The failure significance metrics calculated as mentioned above, are further classified into multiple levels based on severity. The severity of an anomaly group is further derived from the severity levels of the calculated failure significance metric. A comparative study of the number of anomalies and their severity can provide a comprehensive qualitative evaluation of a pre-processing technique.

In cases where the number of KPIs used in the analysis is much smaller, it is possible to find distinct anomaly groups without using any scientific clustering mechanisms. Human beings are capable of performing visual clustering in low dimensional spaces using tools like *Microsoft Excel* which comes handy in such scenarios. The usage of these kinds of tools can be useful in finding visually distinct anomaly groups. Severity levels were attached to a particular group of anomalies and this was used to classify the entire set of anomalies into different severity levels.

General evaluation: The motive behind this is to evaluate all other factors like robustness, scalability, computational complexity and several other practical implications of the pre-processing technique.

3.4 Conclusions

This chapter elaborated upon the nature of the performance measurement data used for the experiments in this research work. Further, it also described in detail the three specific problems this thesis tackles. This chapter also formulated suitable evaluation criteria for measuring the effectiveness of the solutions proposed in Chapter 4.

The following Chapter 4 substantiates the need for stable metrics that can be used for training set filtering, and derives a metric that overcomes the short-comings of many training set filtering techniques. It also proposes a method by which traffic measurement data from cells of varying traffic types can be analyzed meaningfully and compares the anomaly detection capabilities of various scaling techniques.

4 Novel approaches

This chapter introduces the pre-processing as well as other techniques that were developed as part of this thesis work. Firstly, the rationale for the usage of a metric beyond simple failure ratios is explained. The *failure significance metric (fsm)* is then derived mathematically. The usage of the failure significance metric in training dataset filtering and the expected outcome are explained further.

Secondly, an approach of classification of cells into multiple groups based on network traffic is introduced. The possible benefits that could be obtained from such sort of a classification are also explained. Further, this chapter proposes two ways of measuring the effectiveness of an anomaly detection technique: quantitative measure using synthetically modeled NEs and qualitative measure using *failure significance metric.*

4.1 Failure Significance Metric based training set filtering

Failures of different kinds and intensities are a commonplace in large telecom networks. Several KPIs such as Drop Call Ratio (DCR), Handover Failure Ratio, SMS failure ratio etc. are measured by network monitoring personnel to detect faulty NEs in the network. Experiments done as part of this thesis showed that, removing observations of faulty NEs from the training dataset leads to substantial changes in results of anomaly detection (see Table 3-2).

The impact of filtering *'high failure ratio'* data points from the training dataset was briefly described in Section 3.2. The number of anomalous NEs, as well the number of anomalies detected was significantly higher when training dataset filtering was used. Filtering of data points which have non-zero values of failure counters makes the training set too ideal and unrealistic. The usage of this training set filtering in anomaly detection experiments results in the detection of a large number of anomalies. A substantial proportion of these anomalies are not significant from the perspective of network management.

The failure ratio metric $fr(u,n)$ for u failures out of n attempts, defined in equation 4-1, does not take the magnitude of the number of attempts into account. Hence one failure out of one attempt, as well as, hundred failures out of hundred attempts, give the same resultant failure ratio.

$$fr(u,n) = (u/n) = \frac{failure\_count}{total\_attempts} \qquad \text{4-1}$$

If there has been a lot of activity and both numerator and denominator of the equation 4-1 are high, the failure ratio is a meaningful metric. However, if there has not been much activity, both numerator and denominator are low, and the resultant failure ratio metric can be randomly high.

Using $fr(u,n)$ as a metric for filtering the training set can have mainly two drawbacks: 1) it removes random points from the training dataset and overall quality of the training dataset cannot be guaranteed to be high 2) network monitoring personnel can be misguided by such wrong signals and is likely to spend their time analyzing an anomaly which might result due to a high failure ratio and low number of attempts.

It is possible to give thresholds for n and u, above which the failure ratio can be considered as a measure of failure. However, this approach too comes with its own set of problems. Regional, seasonal, daily, weekly and even hourly traffic variations can lead to such rules being unable to detect important anomalies. This leads to the need for defining a metric called the '*failure significance metric (fsm)*'.

4.1.1 The failure significance metric

The *fsm* is a metric that evaluates the significance of a failure based on the total number of attempts that have been made. The *fsm* metric balances the failure ratio so that the failure ratios based on small number of attempts are scaled to be of smaller value when compared to the failure ratio based on higher number of attempts. Equation 4-2 is a weighing function that helps in scaling a value based on the sample size 'n'.

$$f(n) = \frac{2}{1 + e^{\frac{w}{n}}} \qquad 4\text{-}2$$

The term 'w' is a term that can be used to adjust the sensitivity of the function. The value of this parameter could vary from 0 to 1. In cases when w is 0, there is no scaling based on number of attempts as the whole fraction becomes equal to 1. The behavior is opposite at the other end of the range (w =1). Fig 4-1 shows how the value of the scaling function $f(n)$ varies for different values of the sensitivity tuning parameter w.

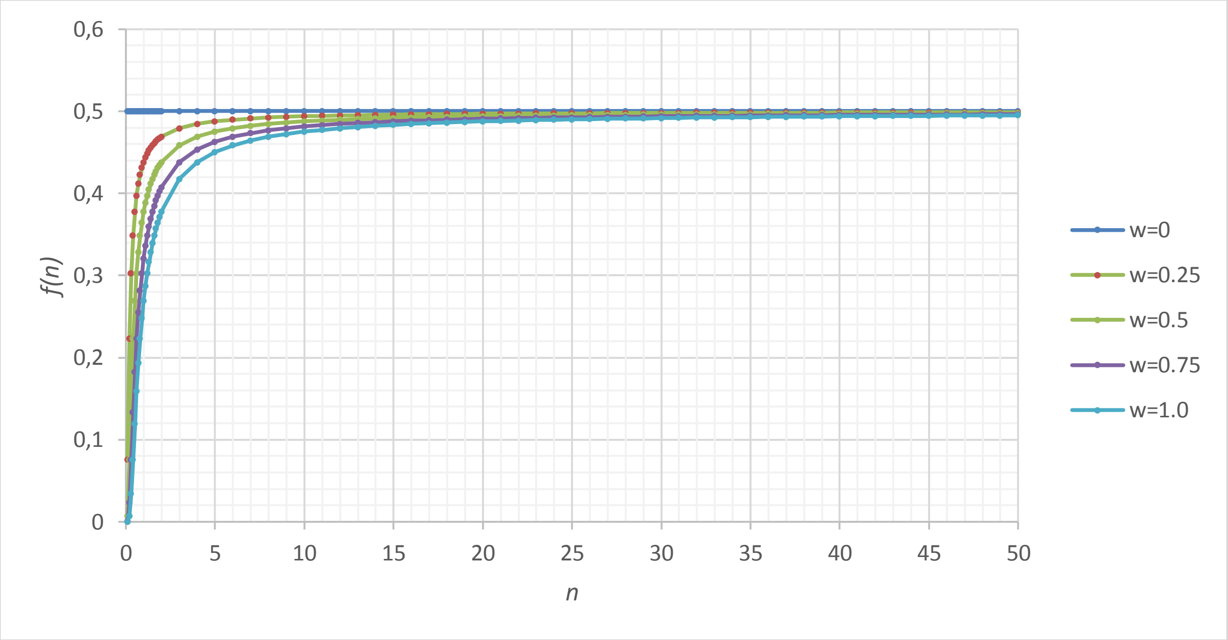


Fig 4-1 Graphical representation of variation of scaling function with sensitivity tuning parameter

When the number of attempts increases, the denominator increases, resulting in high value of $f(n)$. This property of the function can be used to scale the difference of a failure ratio from the average value of failure ratio in the training set. When the difference between the failure ratio of a data point and the average failure ratio of the training set is high, scaling is performed using an inverse metric of the number of attempts (for example 4-2). This results in a higher scaled value in cases where the number of attempts is high in contrast to cases where the number of attempts is low.

On applying this scaling function to the difference of the failure ratio with the average failure ratio gives a scaled value of failure ratio $fr_{scaled}(u,n)$ as shown in equation 4-3. Here $fr(u,n)_{avg}$ represents the average failure ratio in the entire dataset.

$$fr_{scaled}(u,n) = \frac{2}{1+e^{\frac{w}{n}}}\,(fr(u,n) - fr(u,n)_{avg}) \qquad 4\text{-}3$$

Let's consider u as the total number of failures that have happened in an NE during an aggregation interval, and u_{avg} as the average number of failures that occur for all the NEs during the same aggregation interval in training set. When the number of failures u is high, the impact it has on the network is also high. The value of logarithm of u to the base u_{avg} (which is also $((\log(u)\,/\,\log(u_{avg}))$ gives a measure of relative magnitude of failure count when compared to average failure counts. This factor defined here as an impact factor is given in equation 4-4.

$$i(u) = \frac{log(u+\delta)}{log(u_{avg}+\delta)} \qquad 4\text{-}4$$

The term δ of low magnitude is added to the numerator and denominator to eliminate the need to separately handle the zeroes in the data. Setting the value of $\delta = 1$ does not distort the result too much and provides a value which is close enough to the value of $\frac{log(u)}{log(u_{avg})}$.

The failure significance metric $fsm(u,n)$ is further obtained by multiplying the two terms as shown in equation 4-5.

$$fsm(u,n) = \frac{2}{1+e^{\frac{w}{n}}}\,\left(fr(u,n) - fr(u,n)_{avg}\right) * (\log(u+\delta)\,/\,\log(u_{avg}+\delta)) \qquad 4\text{-}5$$

Since $fr(u,n)$ is mathematically represented as u/n, $fsm(u,n)$ can also be represented by equation 4-6

$$fsm(u,n) = \frac{2}{1+e^{\frac{w}{n}}}\,(\left((u/n) - (u/n)_{avg}\right) * (\log(u+\delta)\,/\,\log(u_{avg}+\delta)) \qquad 4\text{-}6$$

Fig 4-2 shows two views of a three dimensional plot of the failure significance function obtained in Equation 4-5.

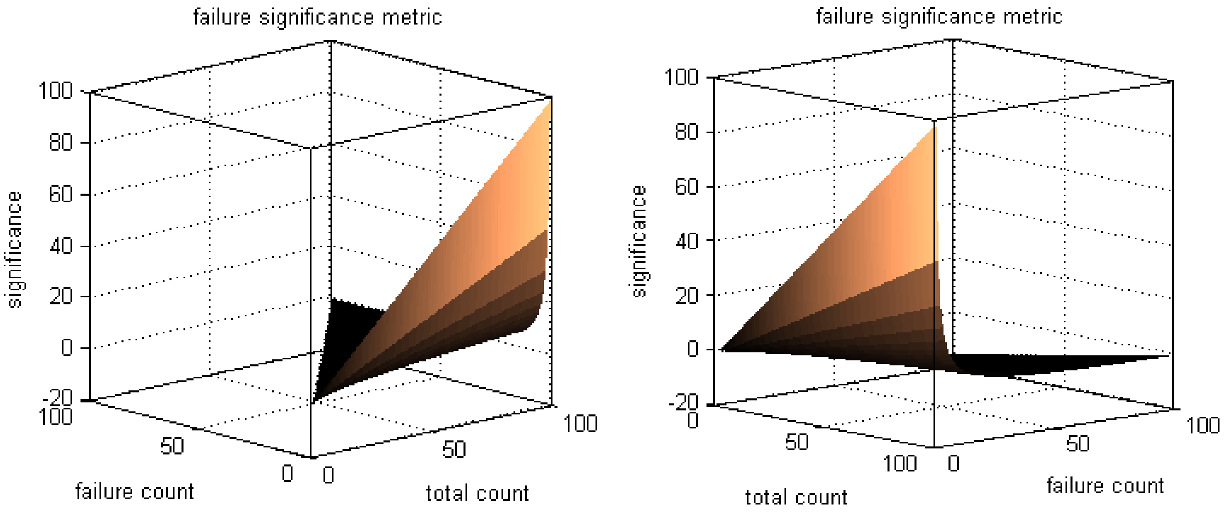


Fig 4-2 Graphical representation of the failure significance metric (w=0.5)

4.1.2 Significance metric in training dataset filtering

A network operator monitoring the network for anomalies will be interested in observing the network elements which have faults corresponding to high failure significance metric. The value of the *failure significance metric* can give a measure of the relative significance of a failure ratio of an observation among a group of observations. By removing the data points, which correspond to high failure significance metric from the training set, the observations in the analysis dataset which correspond to this sort of behavior will be detected as anomalies. This is a typical example of a case in which application domain knowledge is used in filtering the training data set.

Consider a set of n observations measured from m cells. The fsm based training set filtering approach removes the top k percentile of observations which have the highest value of the *fsm* metric. $k=1\%$ which corresponds to removing *1%* of observations with highest fsm provided good results in the tests done as part of this experiment.

4.2 Cell classification based on traffic

The need for aggregating data counters at distinct intervals for cells with different traffic patterns was described in Section 3.2. To achieve this, the first step is to identify the optimal number of groups into which the cells can be clustered and also the cluster centroids. Further, suitable aggregation intervals are chosen for each group based on the cluster centroids, in order to make the data in all cases to be comparable.

The number of calls occurring in the cell during a day is chosen as a metric for classification of cells. This number is calculated for 12134 cells in the dataset and *k-means* clustering with Davies-Bouldin Index (DBI) [61] is used to find the optimal number of clusters. The variation of the DBI with the number of clusters is shown in Fig 4-3. The optimal number of clusters is arrived to be at 2.

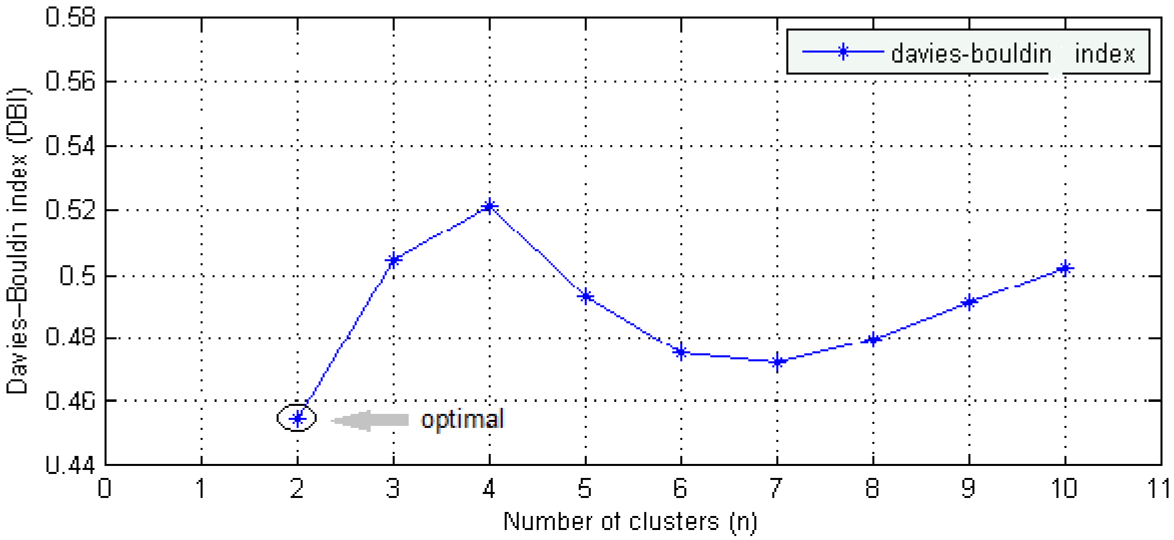


Fig 4-3 Variation of Davies-Bouldin index with number of clusters

Thus the cells are classified into two groups using the *k-means* clustering algorithms which gives the centers of the two clusters at c1 = 83.2 and c2 = 1967.3 respectively.

<u>*Group 1, Low traffic cells*</u>**:** Since the mean center of this group lies at 83.2 calls per day, this group will be further referred to as '*low traffic group (ltg)'.* This is the majority group which contains

approximately 97% of the cells (Fig 4-4). For this group of cells, the aggregation interval was chosen as 1 hour.

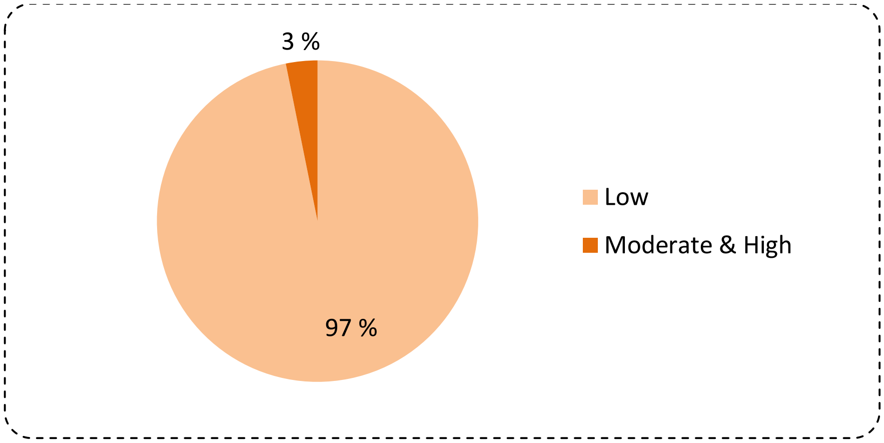


Fig 4-4 Traffic Distribution of cells

<u>Group 2, Moderate and high traffic cells</u>: Since the mean center of this group lies at 1967.3 calls per day, this group will be further referred to as '*moderate and high traffic group (mhtg)'.* This is the minority group which contains only 3% of the total cells. For this group of cells, the aggregation interval was chosen as 15 minutes. The choices of aggregation interval for both groups were done through studying the magnitude of the data generated.

Once the data from the two groups of cells are segregated, they are to be trained and analyzed separately.

4.3 Quantitative measure of anomaly detection technique

Choosing the scaling technique for anomaly detection in a dataset is a difficult task. A variety of scaling techniques were discussed in Section 2.3.2, each trying to solve a different kind of problem. Range based scaling was suggested to be very good for revealing clusters [52] and [53], and *z-score scaling* is claimed to be good in detecting anomalous observations. Robust scaling techniques using Median as well as Robust Logarithmic scaling claim to reduce the impact of outliers in anomaly detection. This section of the thesis proposes a method of quantitative evaluation of anomaly detection capabilities of a particular technique.

The proposed method of measuring quantitatively the anomaly detection capabilities of a pre-processing technique is by adding synthetic anomalies into the analysis dataset.

- Firstly, the training set data passes through a system which calculates the statistics of each of the KPIs including mean, standard deviation etc.
- A set of synthetic NEs (cells in the current scenario) are modeled with their fault conditions. For example, two synthetic NEs with different traffic levels and failure percentages are modeled in Table 4-1.

Table 4-1 Synthetic NEs and their fault condition

Synthetic NE Id	Traffic	Failure percentage
30000	5 to 10 times average traffic	(60 to 100)%
30001	2 to 5 times average traffic	(30 to 80)%

- The analysis data is then passed through a system which takes in the models of the synthetic anomalous NEs as input, and adds data points which pertain to the synthetic NEs' behavior.
- Normal anomaly detection steps are then carried out and two metrics are calculated: number of synthetic anomalous NEs detected and number of synthetic anomalies detected.

Fig 4-5 shows a view of the entire process.

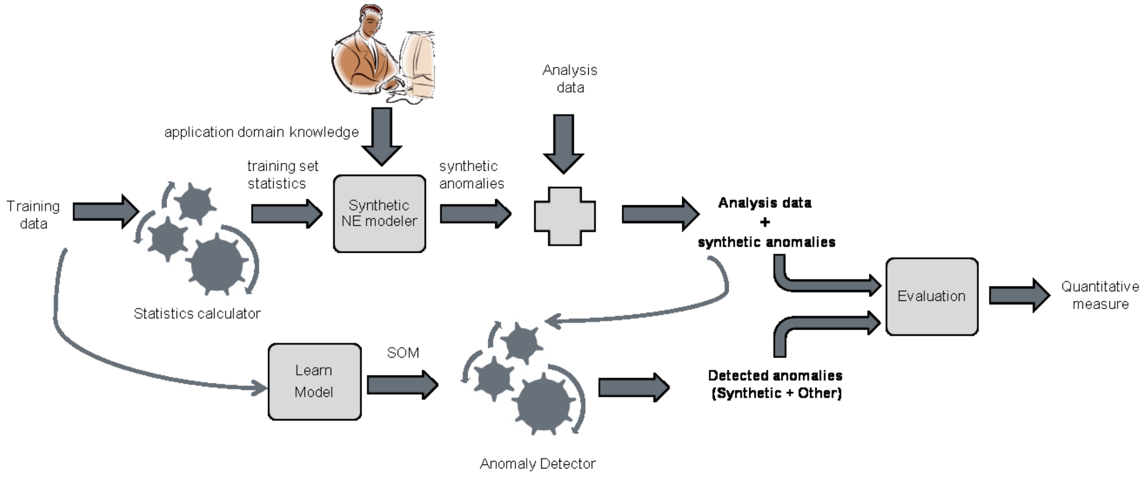


Fig 4-5 Process adopted for quantitative measure of anomaly detection technique

4.4 Qualitative measure of anomaly detection technique

This section proposes a method of qualitative evaluation of anomaly detection capabilities of a technique using the *failure significance metric* derived in 4.1.1. This approach is inspired from a technique of anomaly clustering using '*hierarchical clustering*' used by *Kumpulainen et al.* [64].

The proposed technique uses hierarchical clustering to divide the anomalies detected into a fixed number of clusters such that, the centroids of the clusters are sufficiently different from each other. The goal of this step is measure the number of distinct kinds of anomalies detected using an anomaly detection technique.

The cluster centroids are then analyzed to find the '*failure significance metric'* for KPIs like call setup failures, dropped calls, handover failures, SMS failures etc. This step attempts to classify an anomaly group based on the KPIs of the cluster center. The failure significance metrics calculated as mentioned above, are further classified into multiple levels based on severity (refer Fig 4-6).

The overall severity of an anomaly group is further derived from the severities of its contributing features. By comparing the number of distinct anomaly groups and their severities, a comparative measure of quality of anomaly detection can be obtained. Further, comparing the total number of anomalies and their severities can also be used as a measure of quality. Fig 4-7 shows a view of the entire process.

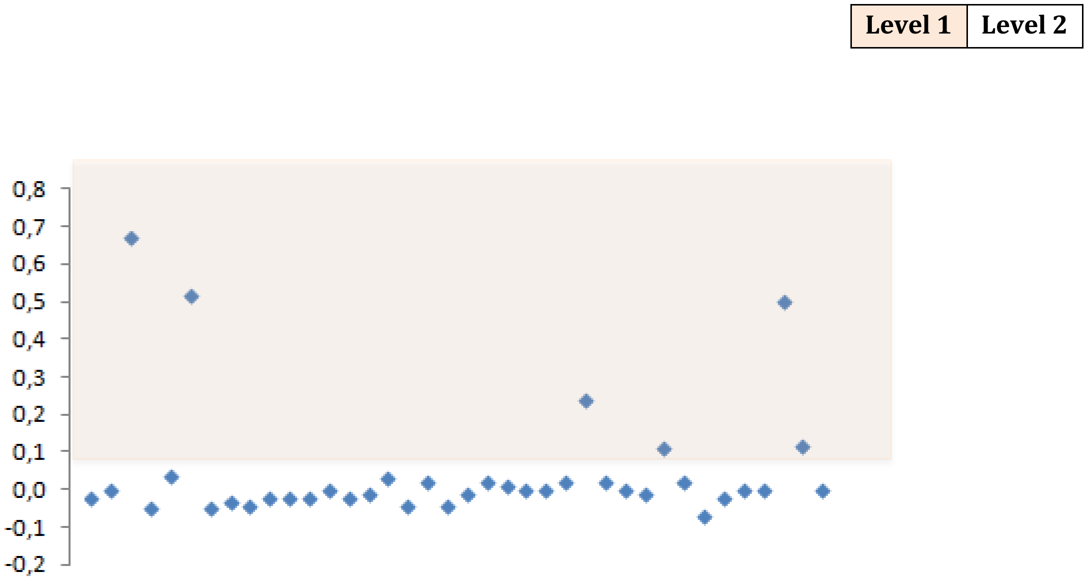


Fig 4-6 Classification of fsm into severity levels

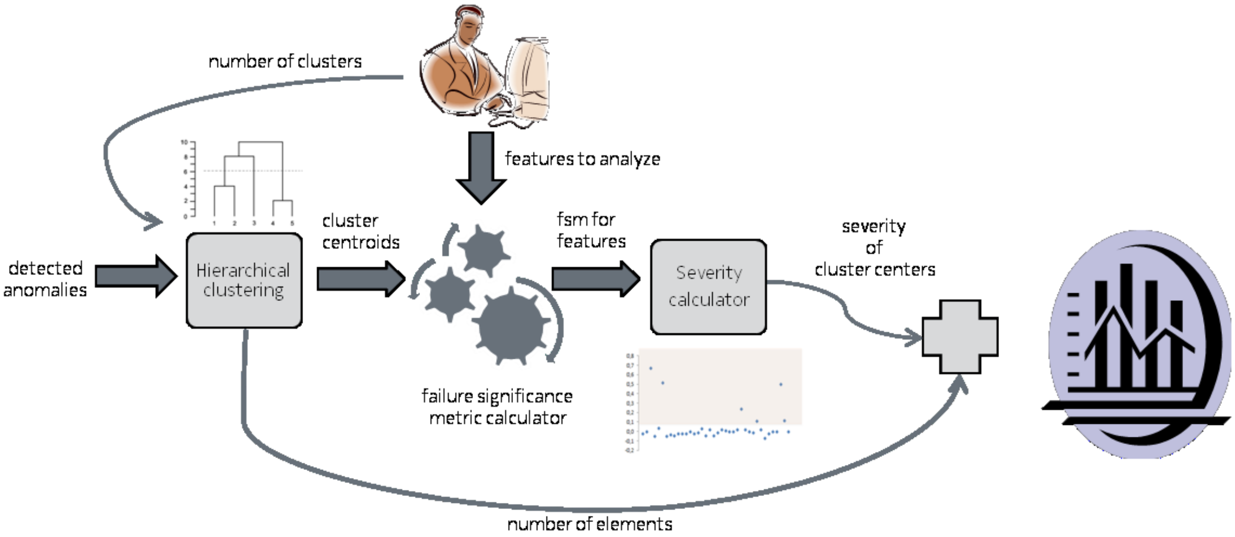


Fig 4-7 Process adopted for qualitative measure of anomaly detection technique

4.5 Conclusions

This chapter introduced the novel approaches that were developed as part of this research. The rationale behind the formulation of the *failure significance metric* and its use in dataset filtering was discussed. A technique of cell classification using *k-means* clustering was introduced. Further, two techniques of measuring the effectiveness of an anomaly detection technique were also introduced. Chapter 5 proceeds to further evaluate the effectiveness of the novel approaches using the criteria defined in Chapter 3.

5 Evaluation of novel approaches

This chapter provides an evaluation of the novel approaches discussed in Chapter 4. Firstly in Section 5.1, the *failure significance metric* based training set filtering technique is evaluated by comparing it with other training-data cleaning techniques. In section 5.2, the approach of classification of cells based on network traffic is put to test, to find out whether this produces any significant improvement in the anomaly detection. Finally, section 5.3*,* evaluates different scaling techniques in terms of their ability to detect anomalies using the approaches of quantitative and qualitative evaluation proposed in Section 4.3 and 4.4 respectively.

For all experiments in chapter, Self-Organizing Maps were used and the parameters used for the training phase are given in APPENDIX A: SOM parameters used in training phase.

5.1 Failure significance based training set filtering

5.1.1 The experiment

The impact of the *failure significance metric* based training set filtering technique is evaluated by measuring the SMS counters of a group of 11,749 cells over a period of two days starting from *1st April 2014*. The measurement data from the first day (*April 1st*) is chosen as the training data set and the data from the subsequent day (*April 2nd*) is chosen as the analysis data set. A set of five KPIs were chosen for this experiment. The chosen KPIs are *SMS_SENT_COUNT, SMS_RECEIVED_COUNT, SMS_FAILURE_COUNT, SMSC_FAILURE_COUNT* and *SMSU_FAILURE_COUNT.* The descriptions of these KPIs were listed in Table 3-1.

In order to measure the percentage anomaly detection, a set of five anomalous cells (with error conditions described in Table 5-1) are modeled programmatically and synthetic observations corresponding to them are added into the analysis dataset. The performance of a technique is evaluated in 5 realms: 1) number of synthetic anomalies detected, 2) number of synthetic anomalous NEs detected, 3) total number of anomalies detected, 4) total number of anomalous NEs detected and 5) quality of anomalies detected.

Table 5-1 Synthetic NEs and their error conditions

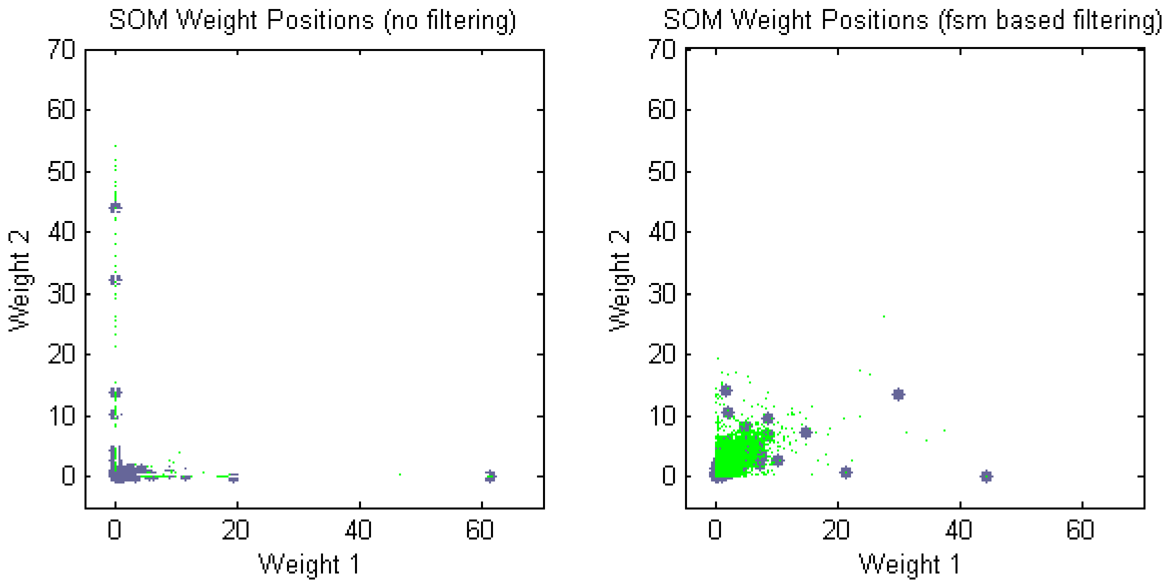
Synthetic NE Id	Total SMS count	SMS failure percentage
30000	(10 to 18) times average traffic	(60 to 100)%
30001	(5 to 9) times average traffic	(60 to 100)%
30002	(2.5 to 4.5) times average traffic	(30 to 60)%
30003	(0.5 to 0.9) times average traffic	(60 to 100)%
30004	(0.5 to 0.9) times average traffic	(30 to 60)%

In order to measure comparatively the anomaly detection capabilities of the *fsm* based filtering technique with other techniques, three other methods were chosen: 1) *SOM smoothening* method (refer section 2.3.1.1), 2) Percentile based training set filtering and 3) Failure ratio based training set filtering (refer section 2.3.1.2). In each of the cases, 1% of the observations from the training set were filtered out. A quantitative and qualitative analysis of the anomalies detected is provided in the subsequent sections.

5.1.2 Illustration

Fig 5-1(a) and Fig 5-1(b) represents the structure of the SOM (in linear scale) obtained in two cases: without training set filtering and with 1% *fsm* based filtering respectively. The darker and bigger dots represent the positions of the neurons and smaller green dots represent the data points. As can be seen, the structure of the SOM looks very different in both cases.

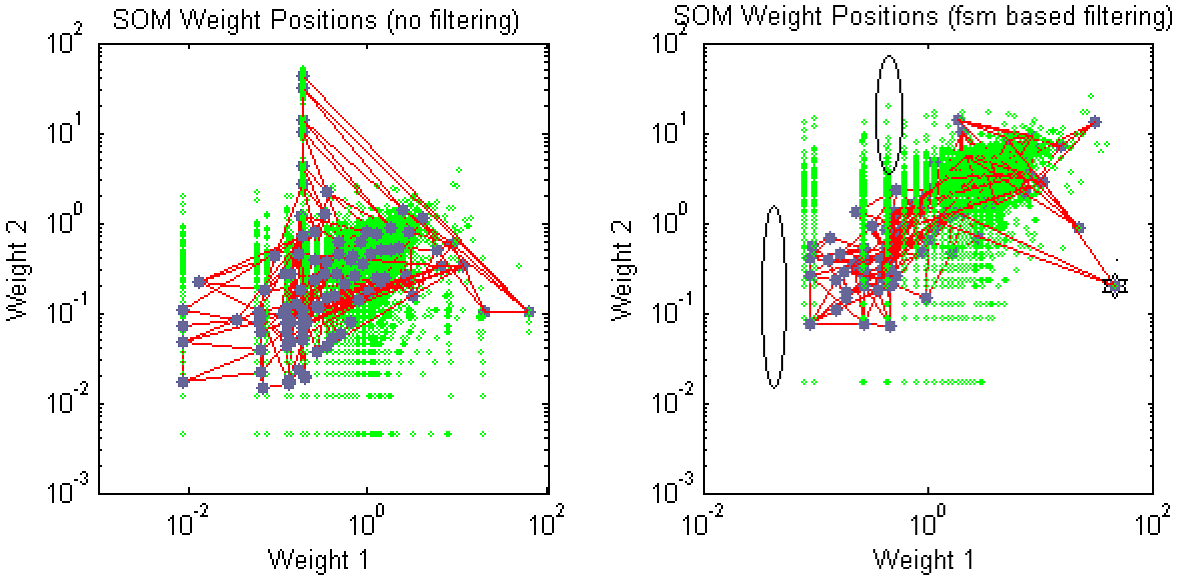
Logarithmic scales are suitable to understand in more detail the structure of SOMs in both cases as shown in Fig 5-2(a) and Fig 5-2(b). The main regions where the structures of the SOMs are different are marked with ellipses on the figure to the right. These regions correspond to data points which have high *fsm* values. Since, the data points from this region are removed due to the filtering, the neurons which correspond to this behavior are removed as well.



(a)

(b)

Fig 5-1 SOM weight positions (linear scale): (a) no training set filtering, (b) fsm based training set filtering



(a)

(b)

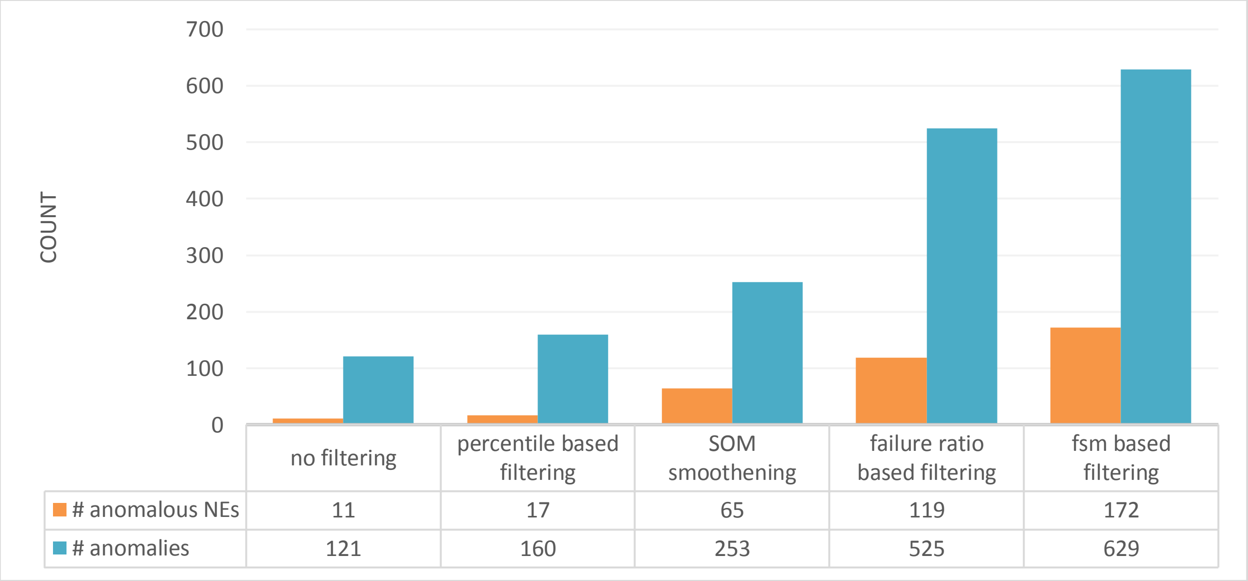
Fig 5-2 SOM weight positions (log scaled): (a) no training set filtering, (b) fsm based training set filtering

5.1.3 *Quantitative evaluation of approach*

The two subsequent sections evaluate quantitatively the training set filtering techniques.

5.1.3.1 Non-synthetic anomalies

Fig 5-3 summarizes the results in terms of the total number of anomalies and anomalous NEs detected in each of the scenarios. The number of anomalies detected using the *fsm* based filtering technique is higher when compared to all other techniques. The results clearly show that filtering the training set has a huge impact on the number of anomalies and anomalous NEs detected.

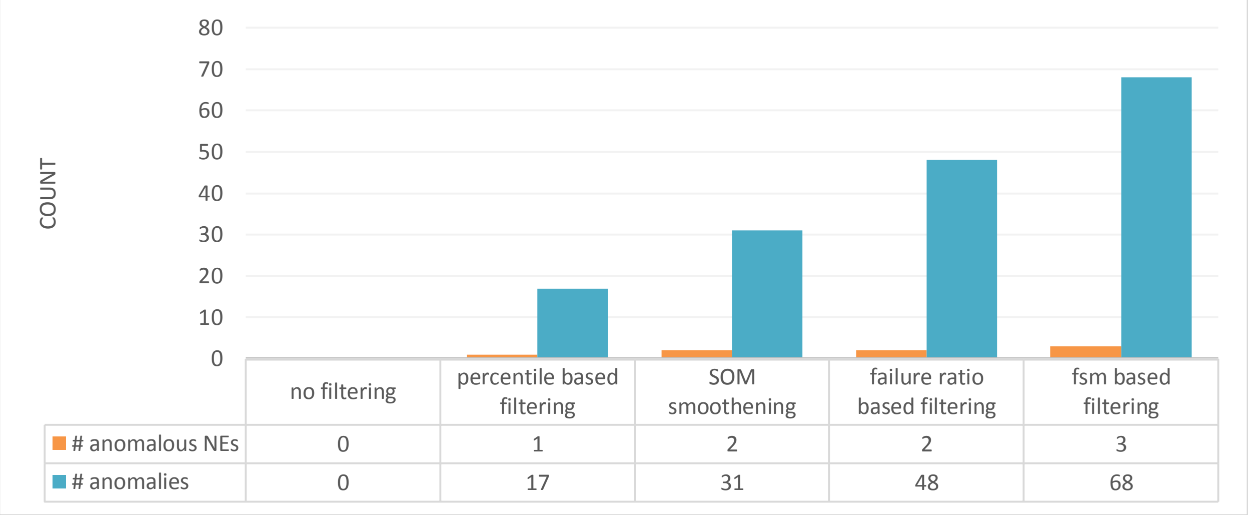


	no filtering	percentile based filtering	SOM smoothening	failure ratio based filtering	fsm based filtering
🟧 # anomalous NEs	11	17	65	119	172
🟦 # anomalies	121	160	253	525	629

Fig 5-3 Statistics of the detected non-synthetic anomalies and anomalous NEs

5.1.3.2 Synthetic anomalies

Fig 5-4 summarizes the results in terms of the number of synthetic anomalous NEs and synthetic anomalies detected in each of the scenarios. The number of anomalies detected using the *fsm* based filtering technique is higher when compared to all other techniques. It is interesting to note that approximately 43% of the synthetic anomalies still remain undetected even using the *fsm* based filtering technique. In the absence of training set filtering techniques, none of the anomalous NEs nor their anomalous observations could be detected.



	no filtering	percentile based filtering	SOM smoothening	failure ratio based filtering	fsm based filtering
🟧 # anomalous NEs	0	1	2	2	3
🟦 # anomalies	0	17	31	48	68

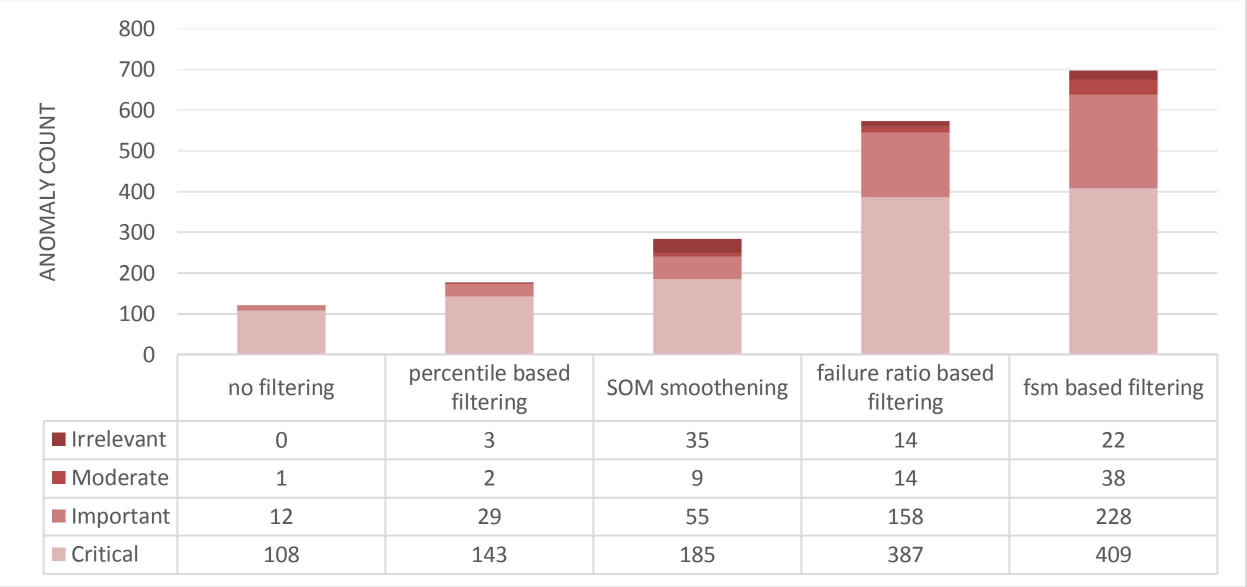
Fig 5-4 Statistics of detected synthetic anomalies (out of 120) and anomalous NEs (out of 5)

5.1.4 Qualitative evaluation of anomalies

Since, this experiment monitored relatively fewer KPIs (only 5), a manual analysis of quality of anomalies was not a tedious task and hence was chosen in this case. Detailed analysis of the entire set of anomalies exhibited ten different kinds of anomalies. The detected typed of anomalies along with their severity are provided in APPENDIX B: Anomaly group Id, description and its severity.

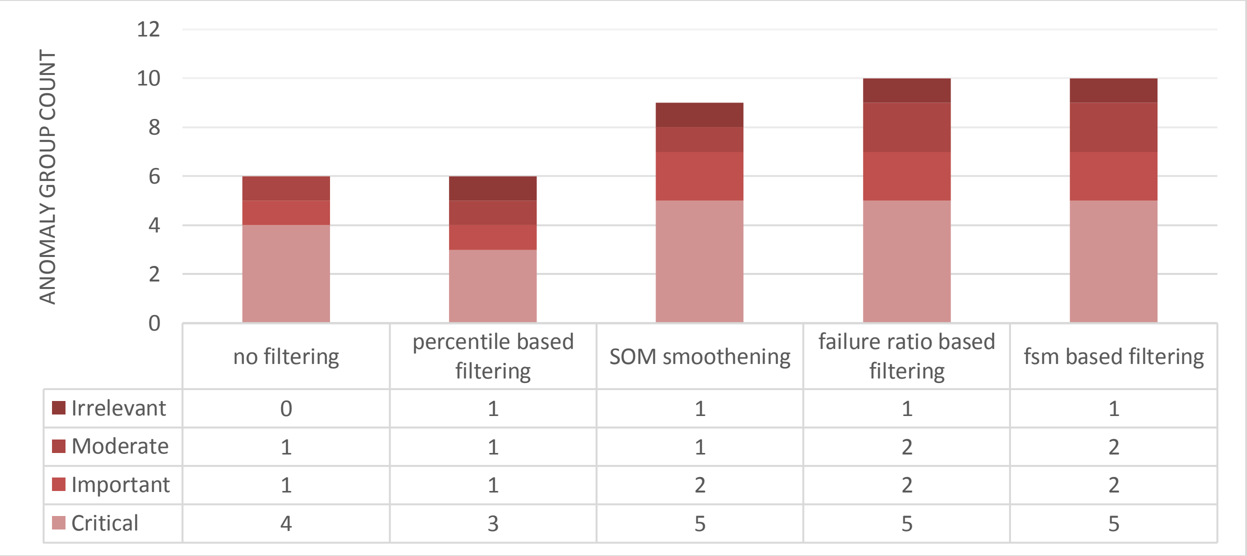
Fig 5-5 gives a relative representation of the number of anomalies with their criticality levels. Failure significance metric based training set filtering outperforms all other filtering techniques on the basis of the number of critical and important anomalies detected. The interesting point to note here is that, *fsm* based training set filtering finds approximately four times the number of critical anomalies that are found without any sort of filtering. This is a very significant difference. However, it is also important to note that the *fsm* based filtering does not outperform other methods in terms of the proportion of critical anomalies detected out of the total number of anomalies detected.

Fig 5-6 gives a relative representation of the number of anomaly groups with their criticality levels. As can be seen here, the number of distinct anomaly groups found by using the application domain knowledge incorporated filtering techniques (failure ratio and fsm) is higher than the cases which do not employ it.



	no filtering	percentile based filtering	SOM smoothening	failure ratio based filtering	fsm based filtering
Irrelevant	0	3	35	14	22
Moderate	1	2	9	14	38
Important	12	29	55	158	228
Critical	108	143	185	387	409

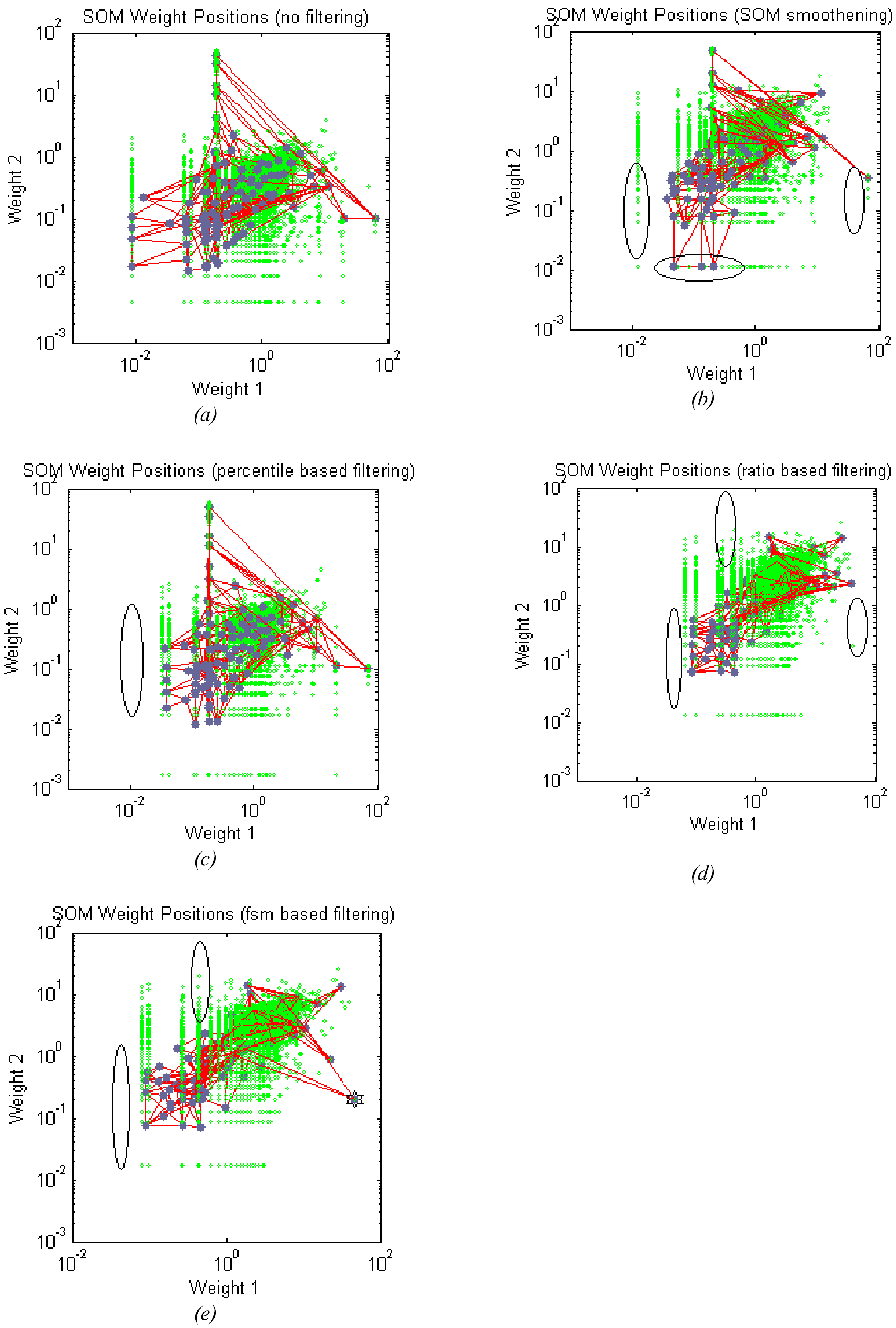
Fig 5-5 Qualitative analysis: number of anomalies and their criticality (synthetic + non-synthetic)



	no filtering	percentile based filtering	SOM smoothening	failure ratio based filtering	fsm based filtering
Irrelevant	0	1	1	1	1
Moderate	1	1	1	2	2
Important	1	1	2	2	2
Critical	4	3	5	5	5

Fig 5-6 Qualitative analysis: number of anomaly groups and their criticality (synthetic + non-synthetic)

The impact of each of the training set filtering technique on the structure of the SOM is depicted in Fig 5-7(a-e).



SOM Weight Positions (no filtering)

(a)

SOM Weight Positions (SOM smoothening)

(b)

SOM Weight Positions (percentile based filtering)

(c)

SOM Weight Positions (ratio based filtering)

(d)

SOM Weight Positions (fsm based filtering)

(e)

Fig 5-7 SOM weight positions in logarithmic scale for different filtering techniques

5.1.5 General evaluation of the approach

This technique of filtering the training set, based on the failure significance metric values was found to give good results in anomaly detection from real telecom network KPI counters. Calculating the failure significance metric of an observation from a group of observations is a simplistic process and can be applicable in other domains which deal with relative KPIs as well.

A cut-off value of 1% has been used throughout this thesis for evaluation of this approach. This was found to be a suitable value for the dataset in hand. Higher values of cut-off percentage can lead to more ideal datasets and can lead in detection of not so significant anomalies as well. The cut-off percentage provides a mechanism to indirectly control the number of anomalies detected.

5.2 Cell classification based on traffic

This section evaluates the benefits as well as drawbacks of the *cell classification based on traffic* approach introduced in Section 4.2. An experiment is performed on real telecom network measurement data to find the number of distinct synthetic anomalous NEs which get detected in two cases: with and without grouping of cells. Another metric that is used for evaluation of this approach is the percentage of synthetic anomalies that get detected by using each of the two approaches. A qualitative analysis of the non-synthetic anomalies using '*failure significance metric*' is also attempted and the observations are recorded.

5.2.1 The experiment

A set of 33 KPIs were chosen and monitored over a period of two days. The data from the first day is chosen as the training data set and the data from the subsequent day is chosen as the analysis dataset. The SOM parameters used in the training phase are given in APPENDIX A: SOM parameters used in training phase.

As mentioned in Section 4.2, the cells were classified into two groups based on the number of voice calls in them during a day using the *Davies Bouldin Index* [61]: the low traffic group '*ltg*' and the moderate and high traffic group '*mhtg'*. An aggregation interval of 1 hour was used for the cells in *ltg* and an aggregation interval of *15* minutes was used for the cells in *mhtg.*

19 faulty cells are modeled synthetically, and anomalous observations corresponding to them are added into the analysis dataset. Details about the cell identifiers and their fault condition description can be found in APPENDIX C: 19 Synthetic Cells and their fault description. The percentage of anomalous NEs as well as their observations that get detected in both cases (with and without grouping) is expected to give a quantitative measure of the effectiveness of the approach. For the ungrouped dataset, two aggregation intervals of *15* minutes as well as *60* minutes are chosen so that effective comparison can be made with the grouping approach.

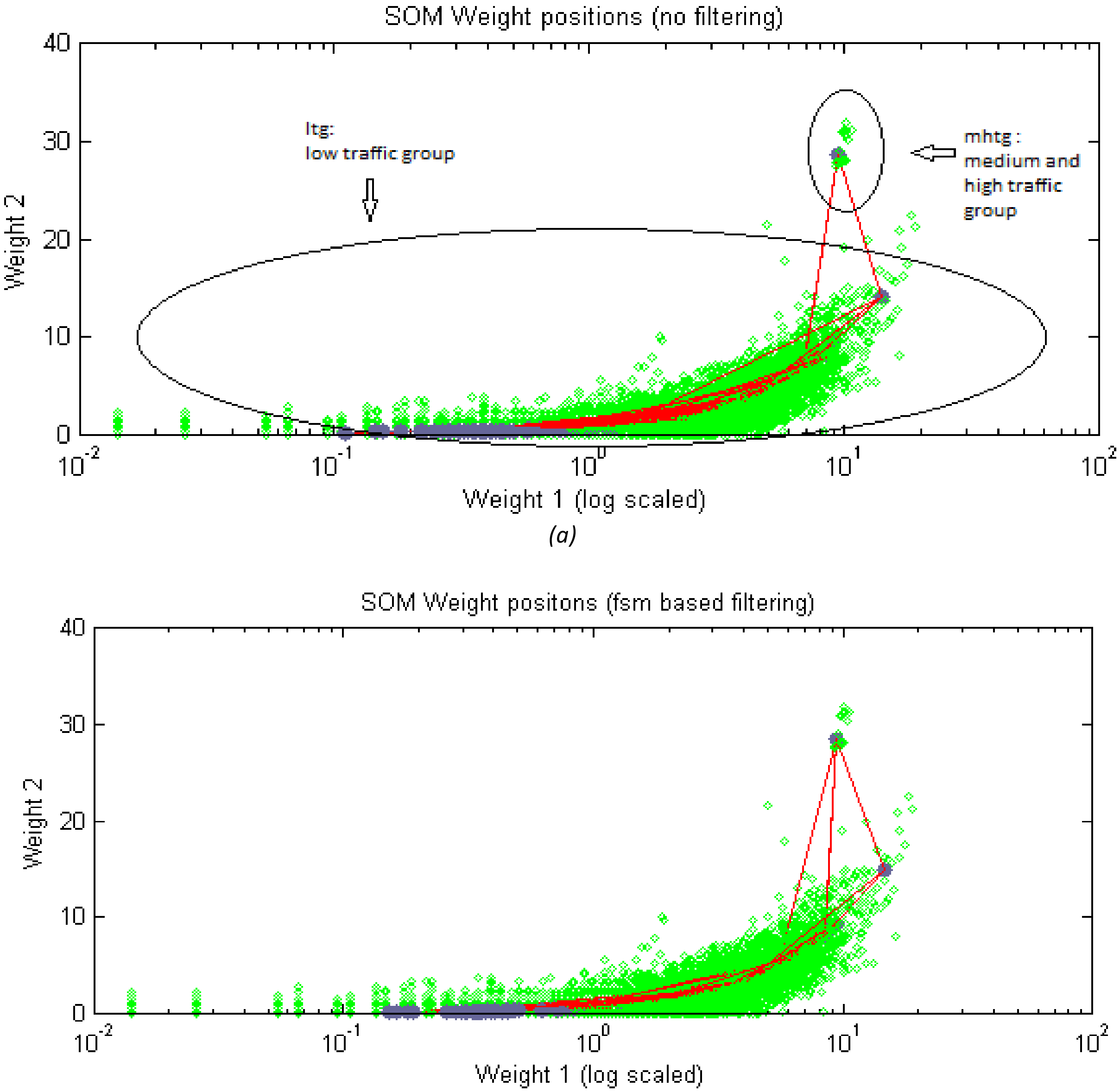
The *Neural Net Clustering App* of *MATLAB R2014a* has been used to visually comprehend the effect of clustering of cells on the structure of the SOM. The impact of the *failure significance metric* based training set filtering on the structure of the SOM is also studied.

5.2.2 Illustration

Fig 5-8 (a) shows the structure of the SOM obtained by training the entire dataset of *12134* cells for a period of 1 day. Logarithmic scales are used in the x-axis as this was the most suitable to show the structure of the SOM in terms of its weight positions. There are two clear groupings of observations in the entire dataset that are marked and designated in Fig 5-8 (a).

Note that it is not possible to clearly demarcate the two groups in the figure as there could be instances of low traffic in *mhtg* which could lead to the data point falling in the *ltg.*

Fig 5-8 (b) represents the structure of the SOM obtained after removing observations from the training set corresponding to high *failure significance metric*. It is not possible to visualize the structural differences between the two SOMs. One possible reason for this could be that the number of data points in the training set is too high, and hence the effect is not visible in terms of SOM structure.



(a)

(b)

Fig 5-8 Ungrouped SOM weight positions: (a) no training set filtering (b) fsm based training set filtering

Fig 5-9 represents the structure of the ltg SOM obtained in two scenarios. The figure on the left represents the SOM weight positions without filtering from the training set. The figure on the right represents the SOM weight positions obtained post the *fsm* based training set filtering. The main visual changes in the SOM structure are highlighted on the figure on the left. The removal of the data points corresponding to high *failure significance metric* has produced considerable change in the position of the neurons. However it is not possible to see the data points that were removed from the training set. This again could be a result of the huge size of the dataset.

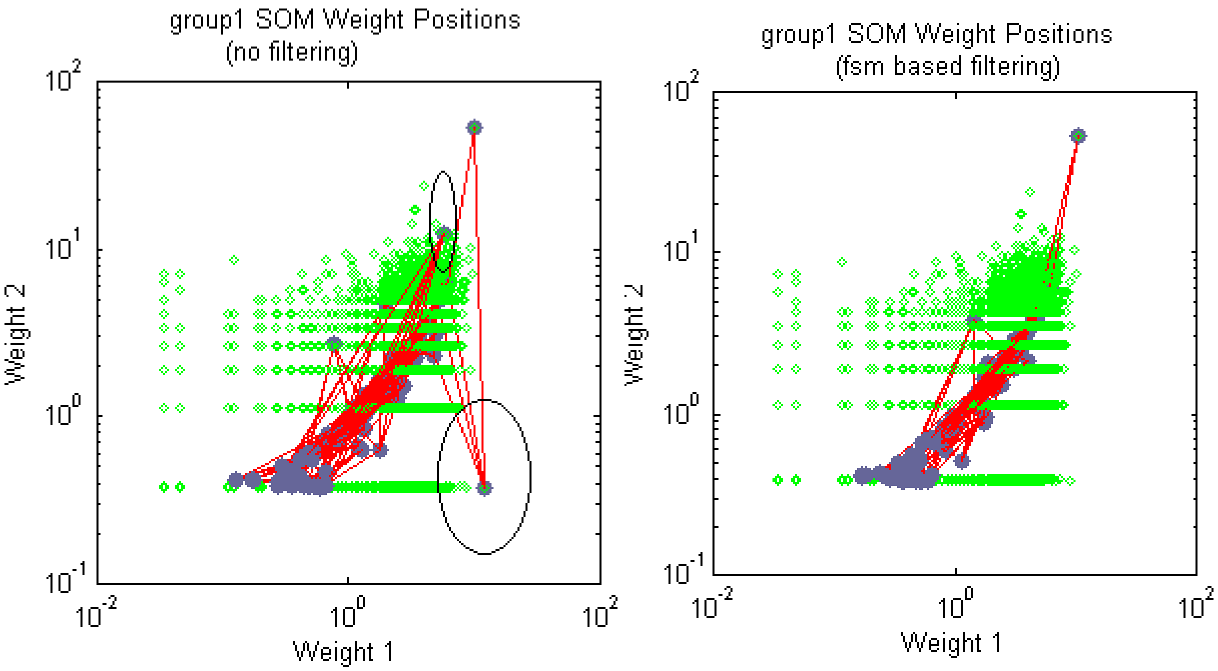


Fig 5-9 group1 (ltg) SOM weight positions: without training set filtering (left) and failure significance based training set filtering (right)

Similarly, Fig 5-10 represents the structure of the *mhtg* SOM. Since this is a smaller dataset, the effect of the training set filtering can be seen visually. The bunch of data points that were present near the lower right corner of the figure on the left is not present in the figure on the right. The absence of these data points has brought about a considerable change in the structure of the SOM.

Another way to interpret this is based on the *failure significance metric*. Since the data points that get removed by the *fsm* based training set filtering correspond to large attempts and high failure ratio, it can be assumed that the missing data points on the second figure correspond to this behavior. Without the presence of the training set filtering technique, the SOM learns this behavior and assumes that it is part of normal behavior of the network (justified by the presence of a neuron in this region). As a result of this, if in the analysis phase, there are data points, which correspond to this behavior, they will not be detected.

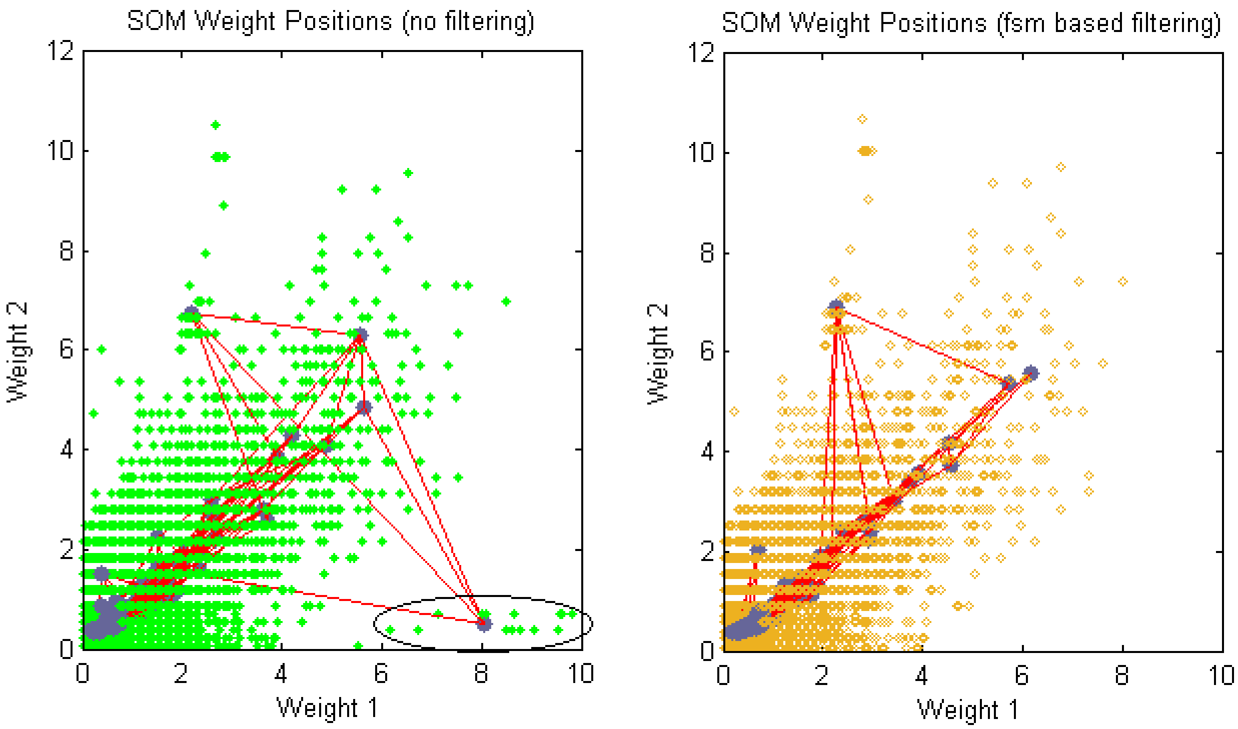


Fig 5-10 group2 (mhtg) SOM weight positions: without training set filtering (left) and fsm based training set filtering (right)

5.2.3 Quantitative evaluation of approach

Table 5-2 gives a summary of the results obtained in this experiment. Since the number of synthetic anomalous data points added into the analysis dataset was different in each case, a numerical comparison of the number of synthetic anomalies detected is not an accurate metric for comparison. Hence percentage of synthetic anomalies detected is used a metric for the comparison.

The number of synthetic anomalous NEs detected by using the cell classification method is higher than the number of synthetic anomalous NEs detected without it. However the percentage of synthetic anomalies detected by using this approach is slightly lower than the cases without cell classification.

The effectiveness of this approach can depend on multiple factors. This approach is more suitable in cases where at least one traffic group is under-represented in the entire dataset. For example, in the current analysis, the *mhtg* group was under-represented because it formed only 3% of the entire number of the cells (see Fig 4-4). During an aggregation interval, the number of observations which correspond to *mhtg* will only be a fraction of the total number of observations.

Table 5-2 Quantitative evaluation of cell classification approach

No of groups	**Aggregation interval in minutes**	**Synthetic NEs detected**	**% of synthetic anomalies detected**
1	15	16 / 19	84 %
1	60	16 / 19	84.2 %
2	ltg – 60 mhtg – 15	18 / 19	81.42 %

In such cases, by using a fixed number of neurons for the entire ungrouped dataset, the minority *mhtg* group might not be appropriately represented. Hence, minute variations in the counters of such group, which can cause a greater impact on the quality of service of the network, might not be detected as anomalies. In this approach *50%* of the neurons are dedicated to learning each group. A suitable factor to decide the number of neurons to be dedicated to each group could be the fraction of the whole traffic that is carried by the group of cells. This has not been evaluated as part of this thesis due to time constraints.

5.2.4 Qualitative evaluation of anomalies

Since the number of KPIs used in this test is large, manual classification of anomalies and identifying the severity/quality of individual anomalies is a task that is extremely time-consuming. Hence, the '*Qualitative measure of anomaly detection technique*' proposed in section 4.4 is used in this scenario.

1. Hierarchical clustering was able to identify a total of 38 different kinds of anomalies.
2. The cluster centers (centroids) were calculated to find the failure significance metric corresponding to known fault areas such as call setup failure, dropped calls, handover failures, SMS failures and overall call failures.

3. Scatter plots of the failure significance metric for each of the features revealed that using two different levels of severity for each of the functionality is optimal (see APPENDIX D: FSM based severity classification – (Cell Classification) for details).
4. Using the *fsm* of each fault area, the centroids are classified into different severity levels. In the current analysis, if any of the fault areas of the cluster centroids are of '*level 1*', the centroid itself is considered to be of '*level 1*', else the centroid is of '*level 2*' (see Table 5-3). '*Level 1*' is considered as more severe compared to '*Level 2*'.

Table 5-3 Anomaly groups and their classification

Level 1	Level 2

Anomaly Id	Call Setup Failure Significance	Call Dropped Failure Significance	Handover Failure Significance	Overall Call Failure Significance	SMS Failure Significance Metric
A1	L2	L2	L2	L2	L2
A2	L2	L2	L2	L2	L1
A3	L1	L1	L2	L2	L2
A4	L2	L1	L2	L2	L2
A5	L2	L1	L2	L2	L2
A6	L1	L2	L2	L2	L2
A7	L2	L2	L2	L2	L2
A8	L2	L2	L1	L1	L2
A9	L2	L2	L2	L2	L2
A10	L2	L2	L2	L2	L2
A11	L2	L2	L2	L2	L2
A12	L2	L2	L2	L2	L1
A13	L2	L1	L2	L2	L1
A14	L2	L2	L2	L2	L2
A15	L2	L2	L1	L1	L2
A16	L2	L1	L2	L2	L2
A17	L2	L2	L2	L2	L2
A18	L2	L1	L2	L1	L1
A19	L2	L2	L2	L2	L1
A20	L2	L2	L2	L2	L2
A21	L2	L1	L2	L2	L2
A22	L2	L2	L2	L2	L2
A23	L2	L2	L2	L2	L2
A24	L2	L1	L2	L2	L1
A25	L2	L1	L2	L2	L1
A26	L1	L2	L2	L2	L1

A27	L2	L2	L2	L2	L1
A28	L2	L2	L2	L2	L2
A29	L2	L2	L2	L2	L1
A30	L1	L2	L2	L2	L2
A31	L2	L1	L2	L2	L2
A32	L2	L2	L2	L2	L2
A33	L2	L2	L2	L2	L2
A34	L2	L2	L1	L1	L2
A35	L2	L1	L1	L1	L1
A36	L1	L2	L2	L2	L1
A37	L1	L1	L2	L2	L2
A38	L2	L2	L1	L2	L1

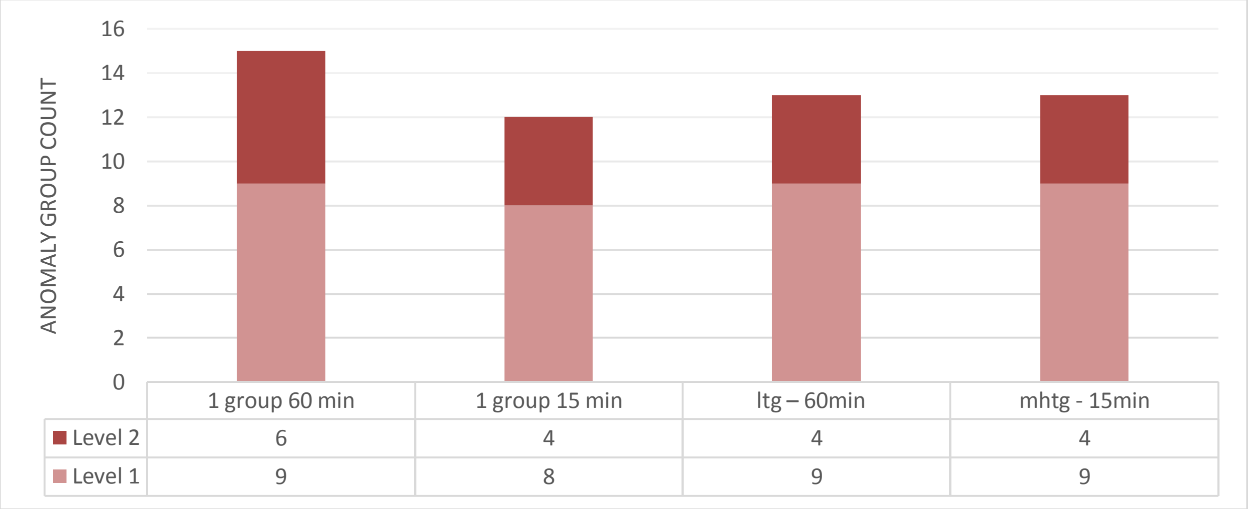


Fig 5-11 Qualitative evaluation (number and types of anomaly groups) detected using the cell classification approach

As can be seen from results in Fig 5-11, nine different types of '*level 1'* anomalies were able to be detected from a minute fraction 3% of the entire number of cells (*mhtg* group). The total number of distinct anomaly groups detected with cell classification approach is higher than in the case without classification. The choice of a low aggregation interval such as 15 minutes is not optimal for the entire traffic group of cells as the number of distinct anomaly groups found (12) in this case is the lowest among all scenarios tested.

Largest number of distinct anomalies were detected in the case when an aggregation interval of 60 minutes was used for the entire dataset. This implies that larger aggregation intervals are needed to find meaningful anomalies from a group of cells having very different traffic patterns and volumes. The clear drawback in using large aggregation intervals is that, the errors in cells with high volume of traffic can be found at the end of the aggregation interval, which is high. Late detection of anomalies in large traffic volume groups is not optimal, as the impact that this can have on the revenue and QoS of the network are high too. This further emphasizes the need to segregate the data set into logical groups and analyzing them separately for anomalies.

5.2.5 General evaluation of the approach

This is not a robust method because of multiple reasons. Classification of cells into groups based on their traffic level can lead to cases when there is no logical grouping and classification may not have clear boundaries. If long enough duration is not used, then the classification can lead to bad results. Even the classification of the cells and their groups' needs to be reviewed periodically as a region which had low mobile traffic could change to a region with high traffic due to seasonal activities like sports events, music concerts etc.

5.3 Choosing the scaling technique

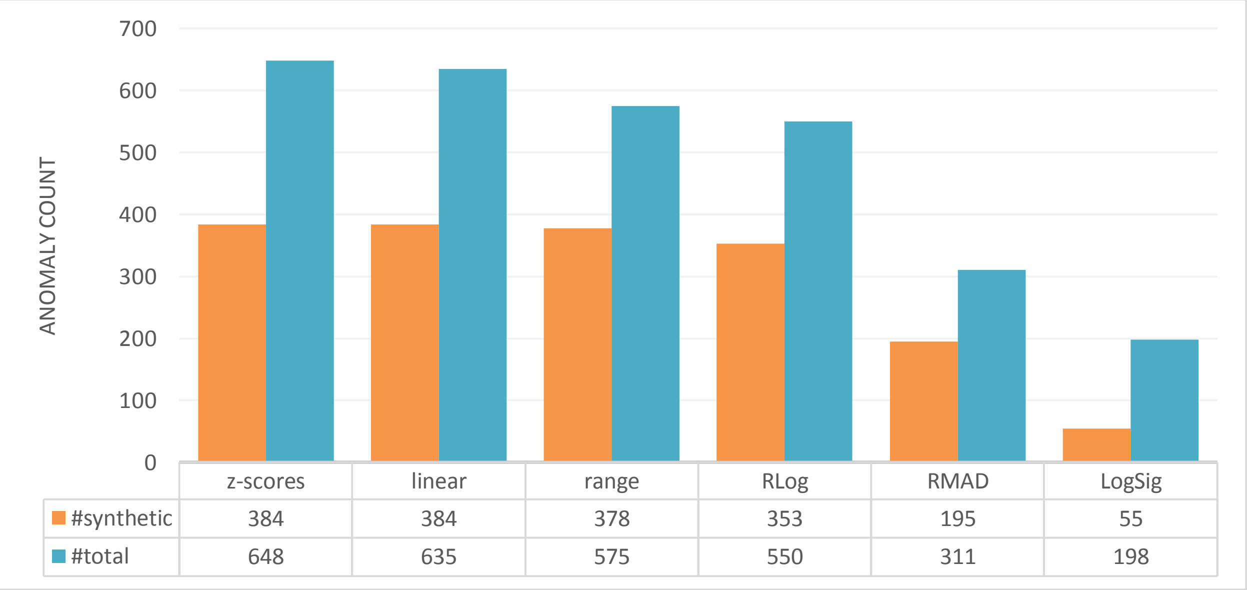
This section evaluates some of the scaling techniques proposed in section 2.3.2 to measure their ability to detect synthetic anomalies from real telecom network data. A qualitative analysis is also done to measure the quality of anomalies detected by using the scaling techniques chosen for evaluation.

5.3.1 The experiment

The training data, analysis data, synthetic anomalies etc. in this experiment are the same as in the previous experiment. KPI counters of 12134 cells were used in this experiment and the same synthetic anomalies as in the previous experiment were added to the analysis data set. The 6 scaling methods that were evaluated are *z-scores* (section 0), *linear scaling* (section 2.3.2.2), *range scaling* (section 2.3.2.4), *Robust Logarithmic* or *RLog scaling* (section 2.3.2.8), *RMAD scaling* (section 2.3.2.7) and *LogSig* scaling (section 0).

5.3.2 Quantitative evaluation

The performances of ‘z-score scaling’ and ‘linear scaling’ are very identical in terms of the number of anomalies detected. In both cases, the number of synthetic anomalies detected is the same. Thus it is safe to assume that, the ‘*mean-centering*’ technique adopted by the former does not bring substantial difference quantitatively. The performance of both *range scaling* and *robust scaling* are good too (see Fig 5-12).

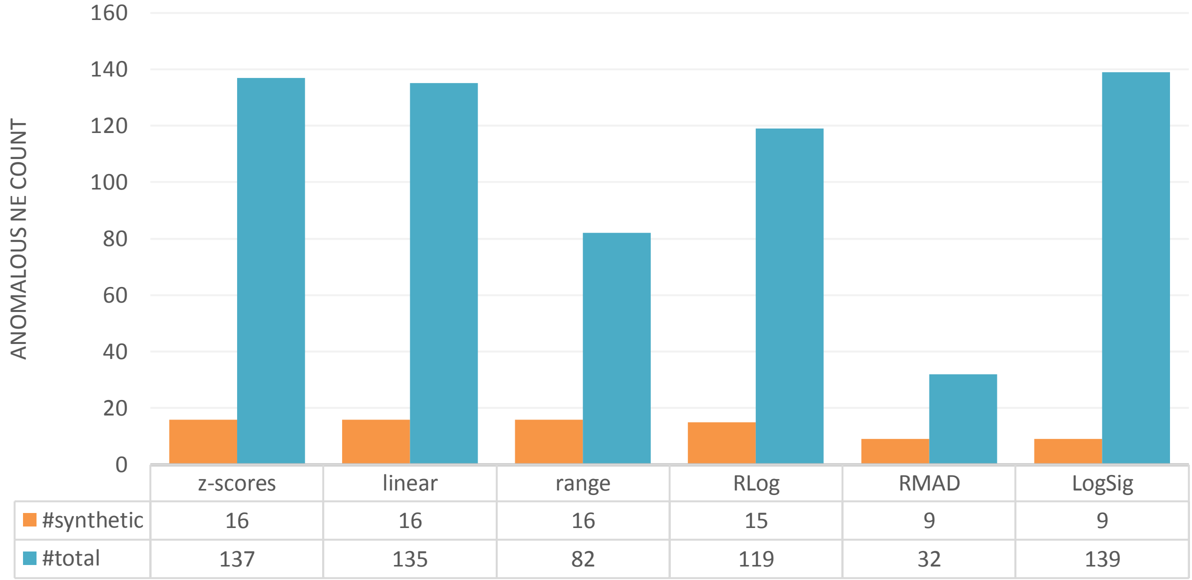


	z-scores	linear	range	RLog	RMAD	LogSig
🟧 #synthetic	384	384	378	353	195	55
🟦 #total	648	635	575	550	311	198

Fig 5-12 Quantitative evaluation of different scaling techniques on the detection of anomalies (synthetic and non-synthetic)

However RMAD scaling performs very badly for the current dataset. The reason for this is that in telecom network data, medians are not a very good measure of centers. The scaled values of several KPIs were zero and this leads to the failure of the technique in detecting anomalies. It is not fair to compare *LogSig scaling* with the other techniques as the goal of *LogSig* scaling is to learn knowledge about the main body of the data which corresponds to the normal mode of operation [50].

The interesting thing to note from Fig 5-13 is that *LogSig scaling* detects the largest number of distinct NEs among all scaling techniques. This should be because of the unique property of this technique to detect slight variations from the normal behavior of body.



	z-scores	linear	range	RLog	RMAD	LogSig
#synthetic	16	16	16	15	9	9
#total	137	135	82	119	32	139

Fig 5-13 Quantitative evaluation of different scaling techniques on the detection of anomalous NEs (synthetic and non-synthetic)

5.3.3 Qualitative evaluation of anomalies

The method of qualitative evaluation of anomalies with their criticality adopted in this experiment is the same as the one used in section 5.2.4*.*

1. Hierarchical clustering was able to identify a total of 26 different clusters of anomalies from all the scaling techniques combined.
2. The cluster centers (centroids) were calculated to find the failure significance metric corresponding to known fault areas such as call setup failure, dropped calls, handover failures, SMS failures and overall call failures.
3. Scatter plots of the failure significance metric for each of the features revealed that using two different levels of severity for each of the functionality is optimal (see APPENDIX E: FSM based severity classification – (Scaling techniques) for details).

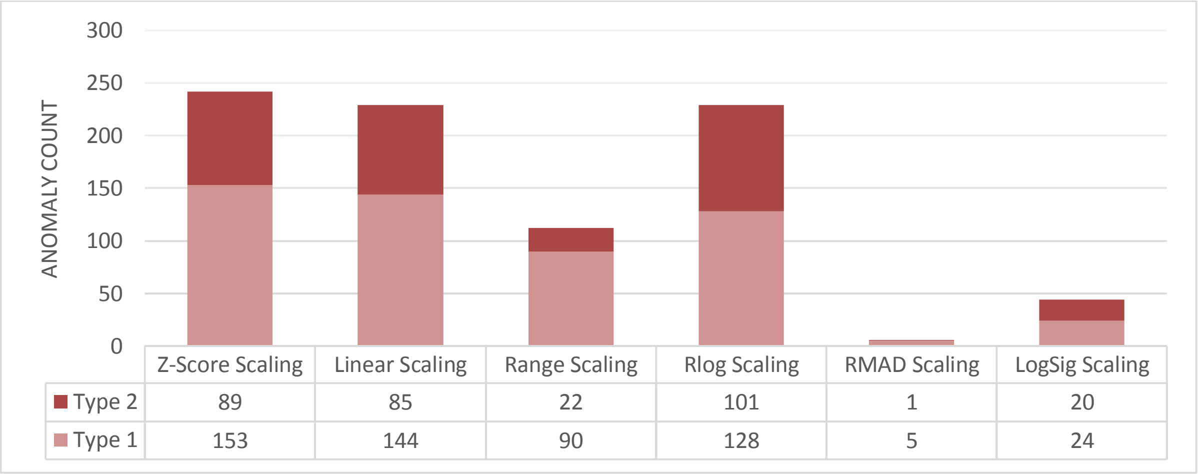
4. Using the *fsm* of each fault area, the centroids are classified into different severity levels. In the current analysis, if any of the fault areas of the cluster centroids are of ‘*level 1*’, the centroid itself is considered to be of ‘*level 1*’, else the centroid is of ‘*level 2*’ (see Table 5-4).

Table 5-4 Anomaly groups and their classification

Level 1	Level 2

Anomaly Id	Call Setup Failure Significance	Call Dropped Failure Significance	Handover Failure Significance	Overall Call Failure Significance	SMS Failure Significance Metric
A1	L2	L2	L2	L2	L1
A2	L2	L2	L2	L2	L2
A3	L2	L2	L1	L1	L1
A4	L2	L2	L1	L1	L2
A5	L2	L2	L2	L2	L2
A6	L2	L1	L2	L2	L2
A7	L2	L2	L2	L2	L2
A8	L2	L2	L2	L2	L2
A9	L2	L2	L2	L2	L1
A10	L2	L2	L2	L2	L2
A11	L2	L2	L1	L1	L2
A12	L2	L2	L1	L1	L2
A13	L1	L2	L2	L2	L2
A14	L2	L2	L2	L2	L2
A15	L2	L2	L1	L1	L2
A16	L2	L2	L2	L2	L2
A17	L2	L2	L1	L1	L2
A18	L2	L2	L2	L2	L2
A19	L1	L1	L2	L2	L1
A20	L2	L2	L1	L1	L1
A21	L1	L1	L2	L2	L2
A22	L2	L2	L1	L1	L2
A23	L2	L1	L2	L2	L2
A24	L2	L1	L1	L1	L2
A25	L2	L1	L2	L2	L2
A26	L2	L2	L2	L2	L2

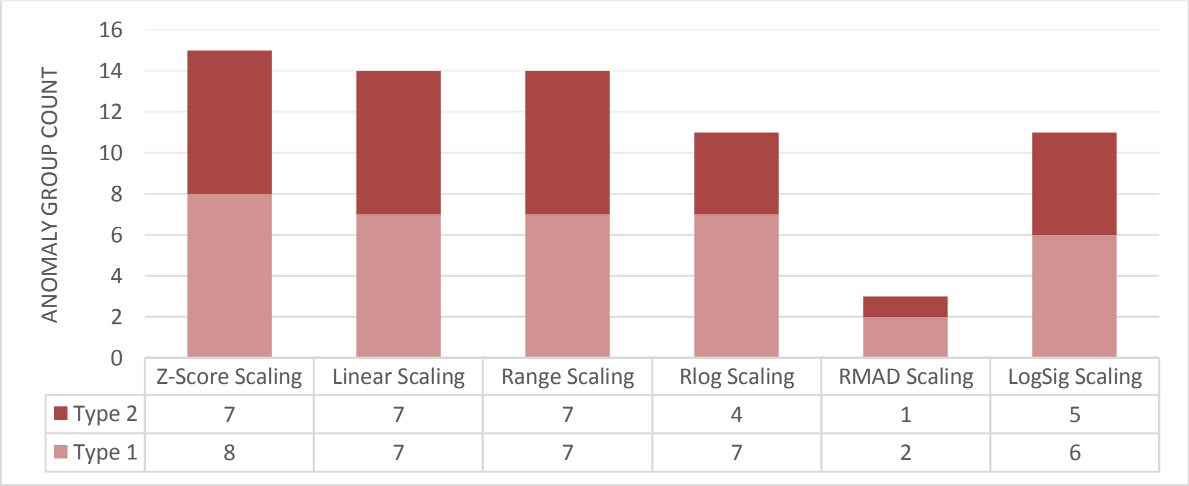
As can be seen from results in Fig 5-14, the results of the three technique z-score, linear scaling and RLog scaling techniques are comparable. The qualities of anomalies detected by the other techniques are much lesser.



	Z-Score Scaling	Linear Scaling	Range Scaling	Rlog Scaling	RMAD Scaling	LogSig Scaling
🟥 Type 2	89	85	22	101	1	20
🟥 Type 1	153	144	90	128	5	24

Fig 5-14 Number and types of anomalies detected using different scaling techniques

Fig 5-15 depicts the number of distinct anomaly groups detected by using different scaling techniques. The key thing to note here is that LogSig scaling detects some interesting high severity anomalies which could not be detected by other techniques. Joint usage of this scaling technique along with another scaling technique such as linear scaling or RLog scaling could provide interesting results [50].



	Z-Score Scaling	Linear Scaling	Range Scaling	Rlog Scaling	RMAD Scaling	LogSig Scaling
🟥 Type 2	7	7	7	4	1	5
🟥 Type 1	8	7	7	7	2	6

Fig 5-15 Number and distinct anomaly groups detected using different scaling techniques

5.4 Conclusions

This chapter evaluated the effectiveness of the novel approaches suggested in Chapter 4. Training set filtering using *fsm* was found to detect the highest percentage of synthetic anomalies (53%) and anomalous NEs (3 out of 5). In the absence of training set filtering, the results were poor.

The cell classification approach detected the largest number of synthetic anomalous NEs. Moreover, the percentage of synthetic anomalies detected with and without the cell classification approach were found to be similar. The qualitative evaluation approach introduced in Section 4.4 was found to be very effective in identifying and classifying anomalies for the scaling technique selection experiment as well as the cell classification experiment.

The next chapter concludes this thesis with the key outcomes of the various experiments performed and providing recommendations for future work.

6 Summary and conclusions

This thesis studied the effect of various pre-processing techniques on the quality and quantity of anomalies detected from telecom network measurement data. The study was concentrated on three areas of pre-processing: training set filtering, training set clustering/segregation based on logical groups and data transformation or scaling.

A novel approach of filtering training dataset using a newly introduced metric called 'failure significance' was introduced and its impact on anomaly detection was evaluated. This technique was able to detect the largest number of synthetic anomalies as well as anomalous NEs from a mixture of real network data and synthetic anomalies. The overall quality of anomalies (measured from a network monitoring personnel's point of view), was also found to be the highest. The experiments done as part of this thesis clearly show the need for filtering/cleaning the training dataset before using it in any kind of anomaly detection mechanism.

Further, an approach based for classification of cells into different groups based on their traffic levels and their impact on the anomaly detection experiments was demonstrated using real telecom network data. This approach was found to be good in scenarios where the number of cells belong to different traffic groups vary highly. On real telecom network data, the usage of this technique yielded good results in detecting the highest number of synthetic anomalous NEs.

Two methods of evaluation of effectiveness of an anomaly detection technique were introduced. A quantitative approach, which measures the ability to detect synthetic anomalies from a set of anomalous and non-anomalous data, from a live telecom network, was used to compare the effectiveness of pre-processing techniques. An approach for measuring the quality of anomalies was introduced, which uses hierarchical clustering and '*failure significance metric'* to determine the overall quality of anomalies detected.

The quantitative and qualitative methods were further used to compare the effectiveness of 6 scaling techniques in detecting anomalies.

Future Work: Scaling techniques play a very important role in anomaly detection experiments. Several KPIs used in this research work are different in nature. Some of them represent counters while some represent measures of time. This thesis did not explore the mixed usage of different scaling techniques in the same analysis. This is one possible area where considerable work could be done.

Tests done as part of this thesis revealed that when KPIs are analyzed in logical groups, they can lead to more meaningful results. For example, analyzing KPIs related to voice calls separately from those related to SMS. This is also one potential area of future work.

The next logical step after anomaly detection is to show these anomalies to the end user. Post-processing of anomalies to classify them into logical groups, finding severities, visualizing them etc. are huge challenges in this area that needs to be tackled in the near future.

Bibliography

[1] P. Kumpulainen and K. Hätönen, "Anomaly detection algorithm test bench for mobile network management," *MathWorks/MATLAB User Conference Nordic. The MathWorks Conference Proceedings,* p. 8, 2008.

[2] F. Chernogorov, J. Turkka, T. Ristaniemi and A. Averbuch, "Detection of Sleeping Cells in LTE Networks Using Diffusion Maps," *Vehicular Technology Conference (VTC Spring),* no. 73, pp. 1-5, 2011.

[3] P. Kumpulainen, Anomaly Detection for Communication Network Monitoring Applications,Doctoral Thesis in Science & Technology, Tampere: Tampere University of Technology, 2014.

[4] P. Kumpulainen, M. Särkioja, M. Kylväjä and K. Hätönen, "Finding 3G Mobile Network Cells with Similar Radio Interface Quality Problems," *IFIP Advances in Information and Communication Technology, Engineering Applications of Neural Networks,* vol. 363, pp. 392-401, 2011.

[5] P. Kumpulainen, M. Kylväjä and K. Hätönen, "Importance of scaling in unsupervised distance-based anomaly detection," *Proceedings of IMEKO XIX World Congress. Fundamental and Applied Metrology,* no. September 6-11, pp. 2411-2416, 2009.

[6] T. Kohonen, Self-Organizing Maps, Berlin: Springer, 1997.

[7] H. Yin, "The Self-Organizing Maps: Background, Theories, Extensions and Applications," *Computational Intelligence: A Compendium: Studies in Computational Intelligence ,* vol. 115, pp. 715-762, 2008.

[8] Anonymous, “Serve atOnce Traffica,” Nokia Solutions and Networks Oy, [Online]. Available: http://networks.nokia.com/portfolio/products/customer-experience-management/serve-atonce-traffica. [Accessed 16 December 2014].

[9] Anonymous, “Traffica - Nokia Networks,” 2012. [Online]. Available: http://networks.nokia.com/system/files/document/traffica_-_brochure.pdf. [Accessed 16 December 2014].

[10] M. Subramanian, Network Management: An introduction to principals and practice, Addison-Wesley, 2000.

[11] K. Hätönen, Data mining for telecommunications network log analysis. Doctoral Thesis, Helsinki: Helsinki University, 2009.

[12] M. Mouly and M.-B. Pautet, The GSM system for mobile communications, Palaiseau, France, 1992.

[13] S. Palat and P. Godin, The UMTS Long Term Evolution: From Theory to Practice, Wiley, 2009.

[14] R. L. Freeman, Telecommunication system engineering, 4th Edition, Wiley, 2004, p. 991.

[15] ITU, "Maintenance: Telecommunications Management Network: TMN Management Services: Overview," *Recommendation M.3200,* no. 10/92, p. 32, 1992.

[16] R. J. Burke, Network Management: Concepts and Practice, A Hands-On Approach, 2003: Prentice Hall.

[17] J. Suutarinen, "Performance Measurements of GSM Base Station System. Thesis (Lic.Tech.)," Tampere University of Technology, Tampere, 1994.

[18] S. Hamalainen, H. Sanneck and C. Sartori, LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency, John Wiley & Sons, 2012.

[19] P.-N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining, Addison-Wesley, 2005.

[20] A. Lazarevic, A. Ozgur, L. Ertoz, J. Srivastava and V. Kumar, "A comparative study of anomaly detection schemes in network intrusion detection.," *In Proceedings of the Third SIAM International Conference on Data Mining,* 2003.

[21] M. Agyemang, K. Barker and R. Alhajj, "A comprehensive survey of numeric and symbolic outlier mining techniques.," *Intelligent Data Analysis, IOS Press,* vol. 10, no. 6, p. 521–538, 2006.

[22] J. Jiang and S. Papavassiliou, "A network fault diagnostic approach based on a statistical traffic normality prediction algorithm. Global Telecommunications Conference, 2003. GLOBECOM '03," *IEEE,* vol. 5, pp. 2918-2922, 2003.

[23] R. Fujimaki, "Anomaly Detection Support Vector Machine and Its Application to Fault Diagnosis," *Data Mining, Eighth IEEE International Conference,* pp. 797-802, 2008.

[24] A. J. Höglund, K. Hätönen and A. S. Sorvari, "A computer host-based user anomaly detection system using the self-organizing map," *IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN),* vol. 5, pp. 411-416, 2000.

[25] M. Kylväjä, P. Kumpulainen and K. Hätönen, "Information Summarization for Network Performance Management, In: M. Laszlo, J.V. Zsolt, (eds.)," *Proceedings of the 10th IMEKO TC10 International Conference on Technical Diagnostics, Budapest, Hungar,* pp. 167-172, 2005.

[26] P. Kumpulainen and K. Hätönen, "Local Anomaly Detection for Network System Log Monitoring. In: Konstantionis, M. & Lazaros, I. (eds.). EANN 2007," *Proceedings of the 10th International Conference on Engineering Applications of Neural Networks,* no. August, pp. 29-31, 2007.

[27] M. Anisetti, C. A. Ardagna, V. Bellandi, E. Bernardoni, E. Damiani and S. Reale, "Anomalies Detection in Mobile Network Management Data," *Advances in Databases: Concepts, Systems and Applications,* vol. 4443, pp. 943-948, 2008.

[28] F. Chernogorov, "Detection of Sleeping Cells in Long Term Evolution Mobile Networks", Master's Thesis in Mobile Technology," University of Jyväskylä, Jyväskylä, 2010.

[29] T. Fawcett and F. Provost, "Adaptive Fraud Detection," *Data Mining and Knowledge Discovery,* vol. 1, no. 3, pp. 291-316, 1997.

[30] J. Hollmén and V. Tresp, "Call-based Fraud Detection in Mobile Communication Networks using a Hierarchical Regime-Switching Model," *Proceedings of the 1998 Conference (NIPS'11) Advances in Neural Information Processing Systems, MIT Press,* p. 889–895, 1998.

[31] R. J. Bolton and D. J. Hand, "Statistical Fraud Detection: A Review," *Statistical Science,* vol. 17, no. 3, p. 235–255, 2002.

[32] Y. Kou, C.-T. Lu, S. Sinvongwattana and Y.-P. Huang, "Survey of fraud detection techniques," *IEEE International Conference on Networking, Sensing and Control, 2004,* vol. 2, pp. 749- 754, 2004.

[33] V. Chandola, A. Banerjee and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR),* vol. 41, no. 3, p. 58, 2009.

[34] V. Hodge and A. Jim, "A Survey of Outlier Detection Methodologies,," *Artificial Intelligence Review, Springer Netherlands,* vol. 22, no. 2 / October, 2004, pp. 85-126, 2004.

[35] The Oxford Dictionary of English, Revised Edition ©, Oxford University Press, 2005.

[36] M. Kylväjä, K. Hätönen, P. Kumpulainen, J. Laiho, P. Lehtimäki, K. Raivio and P. Vehviläinen, "Trial Report on Self-Organizing Map Based Analysis Tool for Radio Networks," *Vehicular Technology Conference,* vol. 4, pp. 2365 - 2369, 2004.

[37] S. O. Haykin, Neural Networks and Learning Machines (3rd Edition), Pearson Prentice Hall, 2008.

[38] S. O. Haykin, in *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR, pp. Sections 9.1, 9.2, 9.3, 9.4.

[39] R. R. Beale and T. Jackson, in *Neural Computing - An Introduction*, Institute of Physics Publishing, pp. Sections 5.1, 5.2, 5.3, 5.4, 5.5.

[40] K. Gurney, in *An introduction to neural networks*, CRC Press, 1997, pp. Sections 8.1, 8.2, 8.3.

[41] J. A. Hertz, A. S. Krogh and R. G. Palmer, in *Introduction to the theory of neural computation*, Addison-Wesley Publishing Company, 1991, pp. Sections 9.4, 9.5.

[42] P. Kumpulainen and K. Hätönen, "Local anomaly detection for mobile network monitoring," *Information Sciences,* vol. 178, no. 20, pp. 3840-3859, 2008.

[43] T. K. Kohonen, E. Oja, O. Simula, A. Visa and J. A. Kangas, "Engineering applications of the self-organizing map," *Proceedings of the IEEE,* vol. 84, no. 10, p. 1358 –1384, 1996.

[44] K. Raivio, O. Simula, J. Laiho and P. Lehtimäki, "Analysis of mobile radio access network using the selforganizing map," *Proceedings of Eighth International Symposium on Integrated Network Management,IFIP/IEEE,* p. 439 –451, 2003.

[45] P. Ozdzynski, A. Lin, M. Liljeholm and J. Beatty, "A parallel general implementation of Kohonen's self-organizing map algorithm: performance and scalability," *Neurocomputing: Computational Neuroscience Trends in Research 2002,* Vols. 44-46, no. June, p. 567–571, 2002.

[46] M. Izadi and R. Safabakhsh, "An improved time-adaptive self-organizing map for high-speed shape modeling," *Pattern Recognition,* vol. 42, no. 7, p. 1361–1370, 2009.

[47] X. Chen and X. Yan, "Using improved self-organizing map for fault diagnosis in chemical industry process," *Chemical Engineering Research and Design,* vol. 90, no. 12, p. 2262–2277, 2012.

[48] P. Kumpulainen and K. Hätönen, "Characterizing Mobile Network Daily Traffic Patterns by 1-Dimensional SOM and Clustering. In: Jayne, C. Yue, S. & Lazaros, I. (eds.).," *Proceedings of the 13th International Conference on Engineering Applications of Neural Networks 20-23 September 2012, London, UK. CCIS, Springer,* vol. 311, pp. 325-333, 2012.

[49] J. Laiho, K. Raivio, P. Lehtimäki, K. Hätönen and O. Simula, "Advanced Analysis Methods for 3G Cellular Network," *IEEE Transactions on Wireless Communications,* vol. 4, no. 3, pp. 930 - 942 , 2005.

[50] K. Hätönen, S. J. Laine and T. Similä, "Using the logsig-function to integrate expert knowledge to self-organizing map based analysis," *IEEE International Workshop on Soft Computing in Industrial Applications (SMCia),* no. June 23-25, pp. 145-150, 2003.

[51] J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques (3rd edition), Morgan Kaufmann, 2011.

[52] R. Gnanadesikan, J. R. Kettenring and S. L. Tsao, "Weighting and selection of variables for cluster analysis," *Journal of Classification,* vol. 12, pp. 113-136, 1995.

[53] G. W. Milligan and M. C. Cooper, "A Study of Standardization of Variables in Cluster Analysis," *Journal of Classification,* vol. 5, no. 2, pp. 181-204, 1988.

[54] E. M. Knorr, R. T. Ng and R. H. Zamar, "Robust space transformations for distance-based operations," *Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining,* pp. 126-135, 2001.

[55] Anonymous, “BIRT Analytics 4.2 Technical Summary of New Features,” Actuate Corporation, 2013.

[56] R. A. Maronna, D. R. Martin and V. J. Yohai, Robust Statistics: Theory and Methods, John Wiley & Sons, Chichester, 2006.

[57] P. Filzmoser, R. Maronna and M. Werner, "Outlier identification in high dimensions," *Computational Statistics & Data Analysis,* vol. 52, no. 3, pp. 1694-1711, 2008.

[58] A. K. Jain and C. R. Dubes, Algorithms for Clustering Data, New Jersey: Prentice Hall PTR, 1988.

[59] R. Xu and D. C. Wunsch, “Survey of Clustering Algorithms,” *IEEE transactions on Neural Networks,* vol. 16, no. 3, pp. 645 - 678 , 2005.

[60] Anonymous, “k-Means Clustering,” Mathworks, [Online]. Available: http://se.mathworks.com/help/stats/k-means-clustering.html. [Accessed 20 December 2014].

[61] B. W. Donald and D. L. David, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 1, no. 2, p. 224–227, 1979.

[62] O. Z. Maimon and L. Rokach, Data Mining and Knowledge Discovery Handbook, Springer, 2010.

[63] J. J. H. Ward, “Hierarchical Grouping to Optimize an Objective Function,” *Journal of the American Statistical Association,* vol. 58, no. 301, pp. 236-244, 1963.

[64] P. Kumpulainen, K. Hätönen, O. Knuuti and T. Alapaholuoma, "Internet traffic clustering using traffic header information," *Joint International IMEKO TC1 + TC7 + TC13 Symposium,* 2011.

APPENDIX A: *SOM parameters used in training phase*

Parameter	Value
Neuron count	50
Training iteration count	10
Initial learning rate	8.0
Final learning rate	0.1
Default ‘p’ value threshold	1.0
‘p’ value storage threshold	5.0
Length of data point	60
Detection interval	60
Length of data point	60
Neighborhood function	Gaussian
Neighborhood shape	rectangular
Neighborhood decreasing	non-linear

APPENDIX B: *Anomaly group Id, description and its severity*

Group Identifier	Anomaly Type Description	Severity
A1	1. Sent SMS count = 0 2. High received SMS count 3. Failure percentage (~100%) 4. Reason for failure - Uncategorized	Critical
A2	1. Received SMS count = 0 2. High sent SMS count 3. Failure percentage (~100%) 4. Reason for failure – Core network	Critical
A3	1. Received SMS count = 0 2. High sent SMS count 3. Failure percentage (~100%) 4. Reason for failure – Uncategorized	Critical
A4	1. Moderate/high sent SMS count 2. Moderate/high received SMS count 3. Failure percentage (90 - 100%) 4. Reason for failure – Core network	Critical
A5	1. Moderate/high sent SMS count 2. Moderate/high received SMS count 3. Failure percentage (90 - 100%) 4. Reason for failure – Uncategorized	Critical
A6	1. Moderate/high sent SMS count 2. Moderate/high received SMS count 3. Failure percentage (30 - 90%) 4. Reason for failure – Core network	Important
A7	1. Moderate/high sent SMS count 2. Moderate/high received SMS count 3. Failure percentage (30 - 90%) 4. Reason for failure – Uncategorized	Important

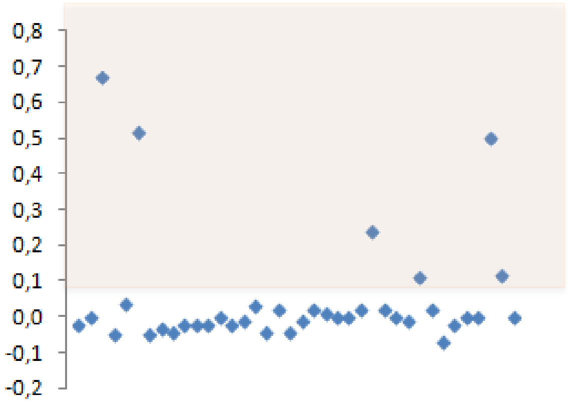
A8	1. Moderate/high sent SMS count 2. Moderate/high received SMS count 3. Failure percentage (10 - 30%) 4. Reason for failure – Core network	Moderate
A9	1. Moderate/high sent SMS count 2. Moderate/high received SMS count 3. Failure percentage (10 - 30%) 4. Reason for failure – Uncategorized	Moderate
A10	1. Moderate/high sent SMS count 2. Moderate/high received SMS count 3. Failure percentage (0 - 10%)	Irrelevant

APPENDIX C: 19 *Synthetic Cells and their fault description*

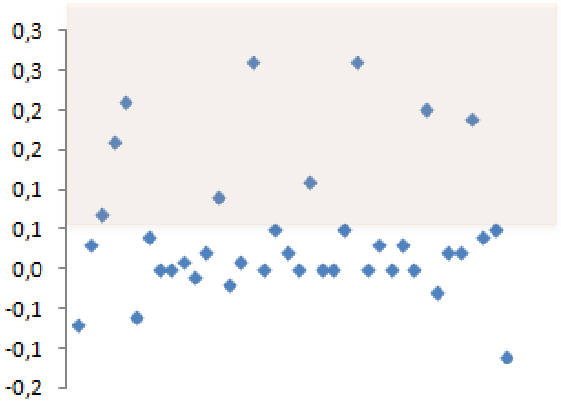
Cell ID	Fault Description
20001	All KPIs have zero value
30001	(80-100%) SMS failure + high number of SMS
30002	(30-80%) SMS failure + high number of SMS
30003	(80-100%) SMS failure + moderate number of SMS
30004	(30-80%) SMS failure + moderate number of SMS
60001	(80-100%) dropped calls + high number of calls
60002	(30-80%) dropped calls + high number of calls
60003	(80-100%) dropped calls + moderate number of calls
60004	(30-80%) dropped calls + moderate number of calls
50001	(80-100%) call setup failure + high number of attempts
50002	(30-80%) call setup failure + high number of attempts
50003	(80-100%) call setup failure + moderate number of attempts
50004	(30-80%) call setup failure + moderate number of attempts
70001	(80-100%) handover failures + high number of calls
70002	(30-80%) handover failures + high number of calls
70003	(80-100%) handover failures + moderate number of calls
70004	(30-80%) handover failures + moderate number of calls
40001	(70-90%) of the calls are of very short length + high number of calls
40002	(70-90%) of the calls are of very short length + low number of calls

APPENDIX D: *FSM based severity classification – (Cell Classification)*

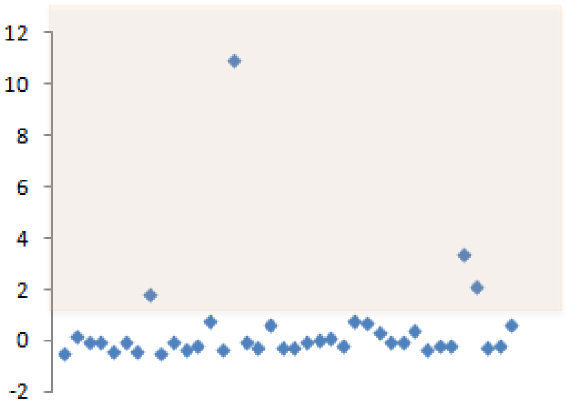
Level 1	**Level 2**



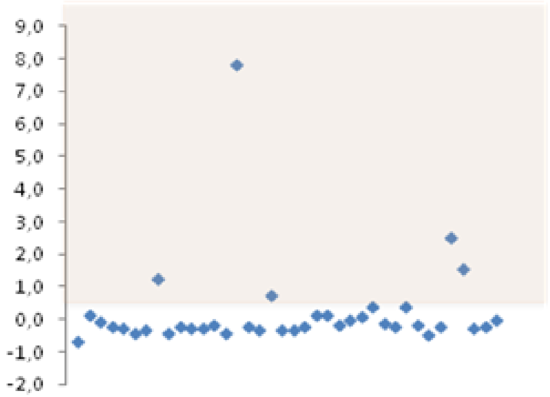
(a) call set up failure fsm categorization



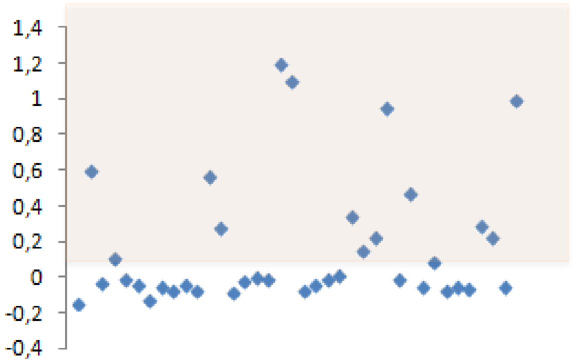
(b) dropped calls fsm categorization



(c) handover failure fsm categorization



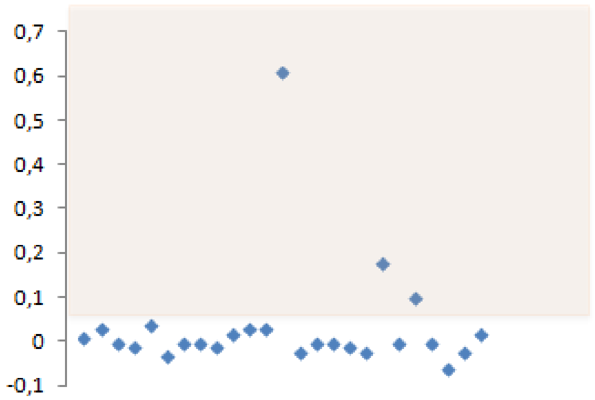
(d) overall call failure fsm categorization



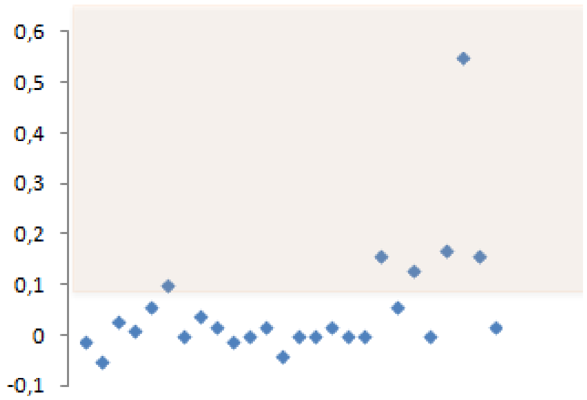
(e) SMS fsm categorization

APPENDIX E: *FSM based severity classification – (Scaling techniques)*

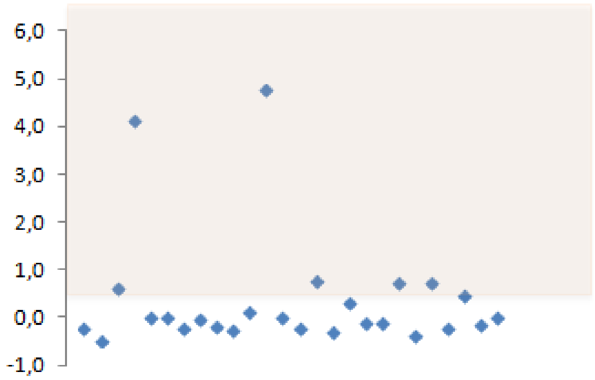
Level 1	Level 2



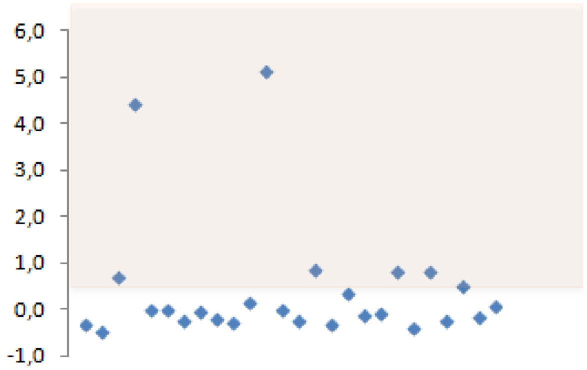
(a) call set up failure fsm categorization



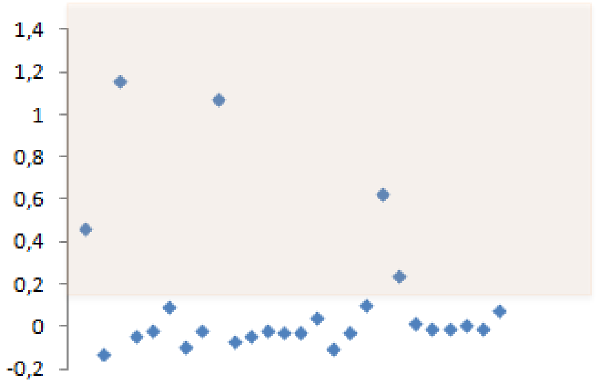
(b) dropped calls fsm categorization



(c) handover failure fsm categorization



(d) overall call failure fsm categorization



(e) SMS fsm categorization