# Intelligent Detection of Anomalies
*in*
# Telecommunications Customer Behaviour

*By*

**Isaac Olusegun Osunmakinde**
(segun@cs.uct.ac.za)

---

Supervisor : **Dr. Anet Potgieter**
*Agents Research Group (ARG) Laboratory*
Computer Science Department
University of Cape Town
Cape Town, South Africa.
E-mail: anet@cs.uct.ac.za

---

**Submitted to the Department of Computer Science,
University of Cape Town, South Africa.**

*In fulfilment of the requirements for the award of
Masters of Science (M.Sc.) degree
in Computer Science.*

September 1, 2006

# Abstract

Anomaly detection in telecommunications call data tries to discover deviant behaviour of individual subscribers, caused by customer churn or attrition, potential fraud, deliberate or unintended expensive mistakes in call data, bad debt risk, or even improper disconnection of communication lines. Malicious behaviour has been noted as one of the key causes of such anomalies, which have consequently led to unquantifiable losses of revenue to many telecommunication networks worldwide. Although the intentions of most subscribers to these networks are unknown when making phone calls, their behaviour pattern is reflected in their call data.

Recent studies have investigated the challenges of anomaly detection but have not given conclusive solutions to address this problem. In this work, we infer that if appropriate individual probabilistic models are used for anomaly detection, the true positive rates of the approaches can be maximized, while the false alarms can be minimized. In this research, we present a modelling technique that can efficiently facilitate anomaly detection that will help call analysts and managers with adaptive decision-making. We developed and implemented a Data Transformation System (DTS), a new Hybrid Genetic Algorithm (HGA) and an Anomaly Detection System (ADS) to address this challenge. The HGA mines Behavioural Bayesian Networks (BBNs) for individual subscribers from call data pre-processed by the DTS, and the ADS trains and acts on these BBNs. Our results are validated with publicly available data and with empirical evaluation results of the ADS, using real world land-line call data for subscribers of a local Telecommunication Services Provider (TSP). Hence, the results provide an illustrative approach of addressing anomaly detection in telecommunication networks.

**Keywords**: Genetic Algorithms, Bayesian Networks, Subscriber / Network Management, Anomaly Detection System.

# Acknowledgements

This is the most difficult part of this thesis because so many people assisted me, but I must mention a few. I first give Almighty God the glory, for His mercy endureth for ever, and thank Him for keeping me alive until this day.

I express my profound appreciation to my supervisor, **Dr Anet Potgieter** of the Computer Science Department, at the University of Cape Town. Thank you for your guidance, care, and constructive ideas throughout this research. I am grateful to the entire management and staff of Complex Adaptive Systems (Pty) Ltd, for their insightful encouragement. Many thanks to the staff members of the Computer Science Department, the authorities at the University of Cape Town, and the members of Agents and AIM research groups, at UCT.

More importantly, I thank the entire management of the following organisations for funding this research and enabling me to complete it: **Complex Adaptive Systems (CAS); the African Institute for Mathematical Sciences (AIMS), South Africa; UCT Faculty of Science** (waiver bursary); **Agents laboratory** (research grant); and **African Mathematics Millennium Science Initiative (AMMSI)**. I give kudos to *Nebula Systems* for providing me with the telecommunication call data with which I conducted experiments to refine my research methodology.

I thank my lovely wife, **Mrs. Cecilia Oluwaseyi Osunmakinde** for her support, the Adebos, every member of my family, **Prof. O. D. Makinde** of the University of Limpopo, and all my friends, for their indefatigable advice. Many thanks to the developers and maintenance staff of the open source software programs, such as *JavaBayes, Latex, Gnuplot, Weka* and the evaluation copy of *BayesiaLab.*

I say thank you and God bless you all.

**Isaac Olusegun Osunmakinde, 2006.**

# Contents

# List of Figures

# List of Tables

# List of Acronyms

AI.....Artificial Intelligence

ADS.....Anomaly Detection System

BBN.....Behavioural Bayesian Network

BN.....Bayesian Network

CDR.....Call Detail Record

CPD.....Conditional Probability Distribution

CPT.....Conditional Probability Table

CSV.....Comma Separated Variable

CCD.....Current Call Dataset

DAG.....Directed Acyclic Graph

DTS.....Data Transformation System

EA.....Evolutionary Algorithm

EDA.....Extended Dependency Analysis

EM.....Expected Maximisation

ESN.....Equipment Serial Number

GSM.....Global System Mobile network

HGA.....Hybrid Genetic Algorithm

MDL.....Minimum Description Length

MI.....Mutual Information

MIN.....Mobile Identification Number

MLE.....Maximum Likelihood Estimate

OSI.....Open System Interconnectivity

PIN.....Personal Identification Number

SIM.....Subscribers Identification Module

TSP.....Telecommunication Services Providers

HCD.....Historical Call Dataset

# Chapter 1

# Introduction

Telecommunications Service Providers (TSPs) that have the ability to detect changes in the behaviour of their subscribers, will be able to take timely actions in order to minimise their losses.

The term anomaly refers to an outlier or to a deviant pattern that can easily be spotted in small datasets but is hidden and requires intelligent detection in the massive amounts of call data generated in telecommunications environments. Common inconsistencies in call data are caused by customer churn or attrition, failure in networks [60], potential fraud [57], [43], [51], deliberate or unintended expensive mistakes made by telecommunications' workers, changes in subscribers' budget on phone calls, bad debt risk and improper disconnection of communication lines. Anomalies in telecommunication call data can also be caused by malicious behaviour, which results in the loss of billions of dollars and wastage of network resources worldwide each year.

The most common cause of anomalies is fraud, because it is committed intentionally and it is difficult to combat. Since fraud is spreading in telecommunication companies in various countries [57], [43], [25], some examples of South African networks that have the potential to be subjected to these anomalies include *Telkom, Cell-C, Vodacom and MTN* [21], [63]. To complicate the situation further, network carriers often do not want to admit that fraud exists as an anomaly in their systems, so that their subscribers do not suspect that fraud is a significant problem. If they do acknowledge it as a significant problem, it might cause churn or cause more subscribers to try to commit fraud. Instead, they prefer this problem to be solved confidentially. If the subscribers suspect fraud but nonetheless keep paying for debts that they did not incur and for services that they did not receive, and if the Telecommunication Services Provider (TSP) does not find a solution

to the problem, these customers may decide to seek alternative competing service providers.

Subscriber calls are characterised by an on-going transformation of their behaviour which appear in a non-linear order. That is, call behaviour changes at irregular intervals. An example is a subscriber who normally makes short period international calls at irregular intervals during off-peak periods. If this subscriber then makes long duration calls during peak periods, this behaviour contradicts the subscriber's call patterns. These deviations might have been caused by a mistake made by the TSP administration, potential fraud etc. In such a situation, the TSP must be able to recognise these deviations, learn from these deviations and take the appropriate actions in order to maintain security. This research includes an in-depth investigation into unlawful acts in fixed and mobile line networks, unsupervised modelling and experiments regarding the early detection of anomalies in fixed line call data. We could not perform tests on mobile call data due to inability of getting the data and the confidentiality issues on the networks.

Anomalies are often caused by fraud. Telecommunication fraud types can be classified into two categories: subscription and superimposed fraud [57], [38]. Subscription fraud results from an illegal subscription to a service using a false identity without the intention of paying for the service. Superimposed fraud is committed by impersonators who use a network without permission and incur additional charges to a caller's bill. Efficient detection of these anomalies can help call analysts to identify these anomalies when they occur, to understand how and why they have occurred, and to act appropriately to prevent recurrence. It will reduce the risk of call analysts making bad decisions and will minimise telecommunication revenue losses based on malicious behaviour.

In this research, we implemented three systems to recognise anomalies, learn from the anomalies and act upon the anomalies. The Data Transformation System (DTS) pre-processes the data and prepares the data for optimal interpretation. The Hybrid Genetic Algorithm (HGA) mines individual probabilistic models in the form of cause-effect relationships called Behavioural Bayesian Networks (BBNs). The Anomaly Detection System (ADS) uses Bayesian learning to train these models from the call data, and uses Bayesian inference in these models to detect anomalies.

Both Bayesian inference and Bayesian learning center around *Bayes' rule* for updating or revising beliefs in light of new evidence. In our research we want to calculate for example the probability that a call is anomalous given a particular subscriber's behaviour. That is; we want to deal with

expressions of the form:

$$Pr(a\ call\ =\ regular \mid Subscriber's\ behaviour).$$

The above expression represents the conditional probability that a call is regular, given a subscriber's behaviour,

or

$$Pr(a\ call\ =\ anomalous \mid Subscriber's\ behaviour).$$

the conditional probability that a call can be classified as an anomaly, given the subscriber's behaviour. Bayes' rule is covered in more detail in later sections.

## 1.1 Problem Definitions, Motivations and Research Questions

Service providers are greatly challenged when subscribers, who feel uncomfortable with their services, including for example: over-charging, not answering queries promptly, or generally bad customer service, change to competing carriers. It results in a loss of revenue [21], which can be attributed to anomalies. Service providers that can detect and adapt to anomalies will be able to minimise their losses.

Researchers have thus far focused more on anomaly detection in mobile call data, with little or no research being conducted on anomaly detection in land lines. Although the latter may not cause such a great a loss as in mobile phone lines, they are still part of carrier losses in general and thus need to be combated. For instance, the real data from the federal communications commission [20] states that carriers lose more than 150 million dollars per year and Brooks [7] estimate 900 million dollars per year on anomalies caused by fraud in United States. Telecommunication anomaly is a problem in many developed countries and the numbers are increasing in areas such as Africa [21], [63]. Frayne [21] currently reported that half a million dollars is lost to common fraud schemes each day, and the numbers are increasing in Africa.

Currently, one of the prevailing methods for combating anomalies is incorporated into *block crediting* [19] [28]. This means a bill is sent to the customer, who either approves or disapproves of the billed

amount. This can result in arguments, which could frustrate either party. Moreover, this approach is expensive and may contain errors.

Existing approaches to anomaly detection include rule-based systems [19], statistical systems [43], neural networks [6], Bayesian networks by segmenting users into groups [28] or distance-based systems [2]. These are powerful techniques but their anomaly detection accuracy are limited as discussed in the next paragraphs. A generalised limitation is that most of these techniques use summarised call data as models and are therefore regarded as being approximated. Our technique is different from these as we tried to minimise approximations.

Minimising approximations in experimental research was suggested as a contribution to the accuracy of scientific results [28], [58]. In light of this, the limitation of approximating a model can be improved by using probabilistic models for individual subscribers. Modelling subscribers individually is complex due to the high non-linearity of subscriber behaviours and due to the explosive technological advancement in phone services (see Chapter Two). In this research, we used individual modelling as one of the contributions to minimising approximation in the detection of anomalies.

The effectiveness of existing anomaly detection techniques are complicated by the fact that the call datasets are distributed over various sites, such as geographical service base stations, in different data format types, such as csv and databases. Furthermore, call datasets contain mixed attributes (continuous and nominal attributes). A nominal attribute may contain alphabetical names or numeric values (e.g. "Name" or "2006"). Continuous attributes contain only decimal / integer values (e.g. 2.4, 5.56) while a mixed attribute space is a collection of both continuous and nominal attributes in a dataset. The differences in the call datasets in various sites can be attributed to the problem of multiple vendors that prepare the datasets for TSPs. Most of the existing systems have access to few subscribers' call data but the real life data we used for our test was obtained from many service based stations.

Many of the existing methods disregard the mixed attribute space but detect anomalies in a continuous attribute space using absolute or distance-based approaches [2], which may result in the loss of information that will affect the detection qualities. In distance-based approach, Bay [2] defined a point value and a minimum deviation expected from that value. If a new value is to be detected, it must not deviate more than what is expected. Moreover, absolute approaches mostly use fixed values for comparisons, and it is difficult for these values to change as the behaviour of subscribers

change. For instance, a subscriber who has been noted to make five calls per day and suddenly changes his behaviour by making nine calls, will be classified as an anomaly by a distance-based approach. This may not necessarily be true.

Most of the existing systems use pre-defined detection rules [19], [6] pre-classified anomalies in training set or use segmentation of users into groups [28], [43] but their quality of detections and abilities in the detection of changing behaviour are questionable.

Most existing anomaly detection systems mentioned above operate in batch mode [65], [43]. This mode creates a lag time that can be exploited by deceitful subscribers. A reduced lag-time will enable service providers to detect and act upon anomalies faster, which will reduce the damage caused by these behaviours without the appropriate intervention.

In view of all these, we are motivated to use individual models that can adapt within a transactional processing system (daily processing). The following research questions are investigated in this study:

- How can call data be observed and transformed so that the underlying models of individual subscribers can be mined efficiently?

  Data transformation is worthy of investigation because Cantu [9] suggested that combining discretisers with evolutionary algorithm such as genetic algorithm, is an interesting future development. We therefore included a discretiser as a subsystem of our data transformation. One of the expectations of our data transformation was to contribute in maximising true positive rates and minimise false alarm rates of our anomaly detection results.

- Can a genetic algorithm be used to mine optimal Bayesian network models from telecommunications call records?

  Positive testimonies exist from early researchers about using a Genetic Algorithm (GA) to mine Bayesian network structures from data. Wai [67] demonstrated and showed that a GA is superior for discovering extremely good networks from large datasets.

- How can we develop a prediction model based on Bayesian networks so that anomalies can be detected in customers' behaviour?

  Prediction models based on Bayesian networks have been applied to several applications, such as medicine to diagnose diseases [40]. We are motivated to apply this technology to telecommunications data.

The objectives of our studies were to illustrate the capabilities of Bayesian networks technology and integrate them into our prototype 4.1 to overcome most of the problems stated in the previous paragraphs. Our findings on the research questions and how we achieved these objectives are discussed in the next section.

## 1.2 Research Approach

We started this research by investigating the deceitful and intentional telecommunications anomaly (fraud) types. The unlawful acts and how they are committed were investigated and are presented in later sections. We defined a structure in Figure 2.1 that shows how one fraud type is linked to another, for both landlines and mobile networks. The detailed investigation was carried out because deceitful anomalies are more difficult to detect than other anomaly types. We believe that our investigation results will provide insightful information to service providers in order to understand bad practices in their networks.

After carrying out an analysis on call data, we designed and implemented the DTS that transforms call data so that individual subscribers' models can be mined successfully. The DTS consists of three subsystems, including: the filtering subsystem, the discretization subsystem, and the user-collection subsystem, whose detailed functionalities are explained in Chapter Four. The DTS observes and transforms subscribers' call data, and consequently answers our first research question.

For the purpose of subscriber call behaviour modelling, we developed and implemented a new genetic algorithm (HGA) from the domain of evolutionary algorithms. The HGA uses information theoretic measures and mathematical components to mine BBNs for individual subscribers. Our individual BBN modelling technique is an improvement over the existing detection systems, which include rule-based systems [19], statistical systems [43], neural networks [6], Bayesian networks by segmenting users into groups [28] or distance-based systems [2]. These are discussed in detail in section, *Related Research in Telecommunications Anomaly Detection.* The HGA is a variant of classical genetic algorithms. It is unique in that we produced two generations and evolved higher generations avoiding redundant offspring. As a result of these, the HGA answers our second research question. More details are described in our methodology in Chapter Four.

For the purpose of intelligent telecommunications anomaly detection, we designed and implemented the Anomaly Detection System (ADS) from the domain of probabilistic and mathematical concepts.

The ADS acts on the BBNs and detects anomalies in mixed attribute datasets under a variety of constraints, including for example: minimising response time using transactional processing. We used daily transactions in this research for incremental training of the BBNs. [55] proposed the detection of outliers from data in real time. The ADS can do detection per single transaction (real time) but in our prototype, we process the detection in batches of daily transactions. This improves on most existing batch anomaly detection systems. The ADS answers our last research question. More details are described in Chapter Four.

Therefore, our fundamental objective is to maximise correct predictions and minimise incorrect predictions at an acceptable level. The next section presents our research contributions.

## 1.3 Research Contributions

This research has contributed the following knowledge: construction of a new algorithm called the Hybrid Genetic Algorithm (HGA), which falls in the research domain of Evolutionary Algorithms; the mining of *multiply-connected* Bayesian networks with our algorithms; building a Bayesian Network for each individual subscriber to improve on generalised anomaly detection models; and mining networks from mixed or nominal datasets(which is done by most existing systems [31], [30], [22]), as well as mining networks from numerical datasets.

The DTS, the BBN and the ADS implemented in this research contribute to the new research area of Bayesian networks applied to anomaly detection. We achieved adaptive intelligence by our HGA that mined individual subscriber models, trained by the ADS using daily transactions, and used by the ADS to detect anomalies in real time of daily batch transactions. The degrees of belief (or level of confidence) obtained from the ADS about anomalous calls will assist network carriers to make reliable decisions and will reduce false alarm rates.

Each of the DTS, HGA, mined BBNs and ADS contributes to a different aspect of solving the problems mentioned above with respect to observing and transforming call data, individual subscriber modelling, and improving the quality of anomaly detection.

The implementation results of this research are primarily intended to help TSPs to understand how to recover revenue losses through explanatory and exploratory approaches, pattern recognition, detection, and acting upon anomalies.

In view of these, the positive response from early users of telecommunication modelling was that this approach holds great promise that subscriber modelling may one day become a standard technique of business decision making, especially in emerging and rapidly changing telecommunication markets [12].

## 1.4 Principal Research Results

The implementation results of our genetic algorithm (HGA) have demonstrated a very good modelling capability, when validated with real life telecommunications call data; when evaluated with publicly available data [50], [49]; and compared with some open source software programs [69]. All these results were accomplished from the informative training sets that were generated from our DTS. The detail results are in Chapter Five.

The implementation of our ADS generated interesting detection results when we experimented with daily call detection and during three evaluation test cases. The overall average detection is 85.878 %. The ADS performs detection on calls that are made to both existing and new destination numbers. Experiments on daily call detection produced very good results. Test case one detected anomalies within every subscriber's behaviour and produced good degrees of beliefs (levels of confidence) on every detection. Test case two was to confirm the reliability of our ADS and to determine if it could differentiate between two subscribers' behaviours. The results of test case two clearly detected call patterns that belong to two different subscribers. Test case three was used to ascertain that known anomalies could be detected effectively by our ADS. The detail is in Chapter Five.

## 1.5 Organisation of thesis

Chapter One includes the introduction, the definition of the problems, the objectives, the motivations for using anomaly detection and the research questions. The principal results are also highlighted.

Chapter Two contains an in-depth literature review of telecommunications anomalies and the reviews of prior work done by other researchers in anomaly detection and Bayesian modelling.

Chapter Three includes the research background on probabilistic modelling and an in-depth de-

scription on mining Bayesian networks.

Chapter Four describes the main methodology, which contains the system model, the Data Transformation System (DTS), Bayesian modelling and the Hybrid Genetic Algorithm (HGA), and lastly, the Anomaly Detection System (ADS).

Chapter Five presents the implementations, simulations, the evaluation and the experimental results.

Chapter six contains the summary and interpretation of results, future research, contributions to knowledge, and concluding remarks. Also, we included the following at the appendix: implementation designs, implementation results and list of publications.

# Chapter 2

# Literature Review

## 2.1 Overview and the details of the problem

A number of causes of anomalies in subscriber behaviour have been identified in the introduction, namely: customer churn or attrition, potential fraud, deliberate or unintended expensive mistakes in call data, changes in subscribers' income, or even improper disconnection of communication lines. It is known that fraud in telecommunication is usually a *deceitful* type of anomaly [25], [51], which requires further elaboration. It can also be an indication that a customer is entering into bad debt.

In *bad debt* risk, a subscriber pays for his/her first two bills after opening an account, uses the network and never pays new bills again [57], [38]. This suspicious behaviour needs to be checked quickly. For instance, a subscriber is not expected to be making extremely high frequency costly calls within a very short period of time, when s/he has been used to making infrequent very cheap calls over a long period of time. This could be suspicious and regarded as anomalous.

According to Frayne [21], telecommunications industry worldwide currently loses half a million dollars to common fraud schemes each day, and the numbers are increasing in Africa. For example, operators in South Africa have already experienced major setbacks to innovative services as a result of fraud [21]. To understand the detection of anomalies, specifically fraud, the following sections clarify and classify the different types of fraud committed worldwide, including South Africa [63], [21] and describe how they are committed.

### 2.1.1 Anomalies in Telecommunications Call Data

Telecommunication services providers (TSP) could protect themselves and their subscribers against deceitful anomalies by using intelligent models or systems that can recognise, learn from and act on fraud. Figure 2.1 illustrates the interrelationships of the known fraud types. This diagram is a pictorial representation of fraud types discussed by various researchers [1], [7], [10]. This diagram includes only fraud and not other anomalies such as debt or possible churn. Churn in subscriber management means a subscriber quitting a service for another within a telecommunication network. Also, a review of anomaly detection techniques and their limitations are described in section; *Related Research in Telecommunications Anomaly Detection.*

When one receives a business offer over the telephone, one regards it as *telemarketing.* If the person online is anonymous, then it could be a fraudulent venture. It may be difficult for an intelligent model to trap telemarketing fraud because a telephone subscriber may voluntarily decide to become a victim of it [7]. The subscriber becomes a victim by accepting the business offer and by providing personal bank account information which can be used to make a false subscription to a network (see next section).

### 2.1.2 Fixed or Land Line Phones

Fraudsters operate on both fixed lines and mobile phones. Fraud is committed in these networks in both developed and underdeveloped countries. In 2001 half a million-fixed lines in South Africa were disconnected, many as a result of fraudulent activities [21]. Consequently, it made the country dropped from third to fifth on the list of African countries with the highest level of telephone marketing. Figure 2.1 shows that the major fraud types are *superimposed* and *subscription* fraud.

Fraud is said to be superimposed when an impersonator illegally steals resources from legitimate users by gaining access to their phone accounts. The impersonator is said to be an *insider* when the fraud is committed by an employee of a TSP. Superimposed fraud is *external* when committed by unscrupulous members of the general public.

In the case of subscription fraud, a fraudster illegitimately obtains a legal account and uses services without intending to pay the bills.

### 2.1.1 Anomalies in Telecommunications Call Data

Telecommunication services providers (TSP) could protect themselves and their subscribers against deceitful anomalies by using intelligent models or systems that can recognise, learn from and act on fraud. Figure 2.1 illustrates the interrelationships of the known fraud types. This diagram is a pictorial representation of fraud types discussed by various researchers [1], [7], [10]. This diagram includes only fraud and not other anomalies such as debt or possible churn. Churn in subscriber management means a subscriber quitting a service for another within a telecommunication network. Also, a review of anomaly detection techniques and their limitations are described in section; *Related Research in Telecommunications Anomaly Detection.*

When one receives a business offer over the telephone, one regards it as *telemarketing.* If the person online is anonymous, then it could be a fraudulent venture. It may be difficult for an intelligent model to trap telemarketing fraud because a telephone subscriber may voluntarily decide to become a victim of it [7]. The subscriber becomes a victim by accepting the business offer and by providing personal bank account information which can be used to make a false subscription to a network (see next section).

### 2.1.2 Fixed or Land Line Phones

Fraudsters operate on both fixed lines and mobile phones. Fraud is committed in these networks in both developed and underdeveloped countries. In 2001 half a million-fixed lines in South Africa were disconnected, many as a result of fraudulent activities [21]. Consequently, it made the country dropped from third to fifth on the list of African countries with the highest level of telephone marketing. Figure 2.1 shows that the major fraud types are *superimposed* and *subscription* fraud.

Fraud is said to be superimposed when an impersonator illegally steals resources from legitimate users by gaining access to their phone accounts. The impersonator is said to be an *insider* when the fraud is committed by an employee of a TSP. Superimposed fraud is *external* when committed by unscrupulous members of the general public.

In the case of subscription fraud, a fraudster illegitimately obtains a legal account and uses services without intending to pay the bills.

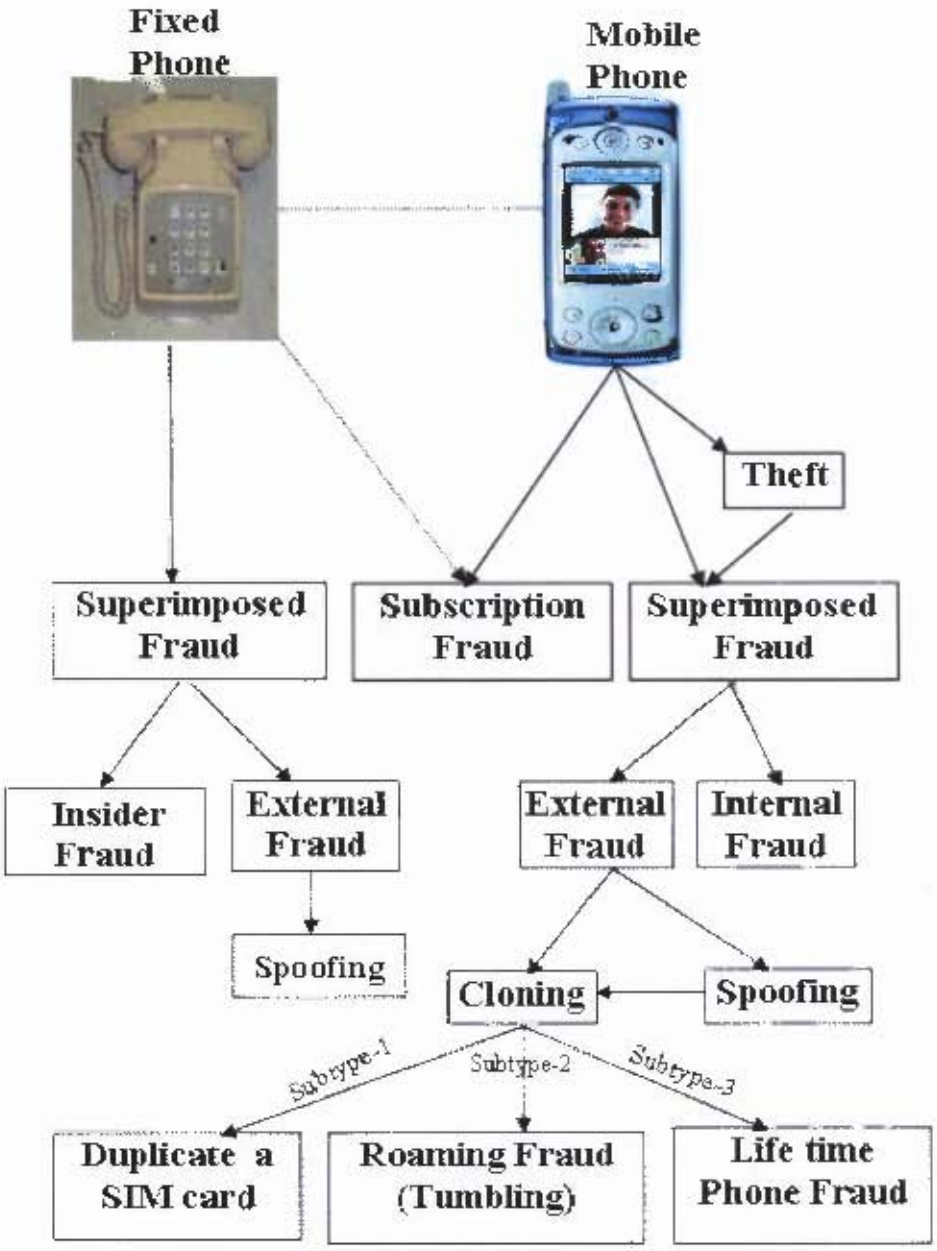


Figure 2.1: Schematic Classification of Deceitful Fraud

### Deficient Security in Land Phone Networks

To commit insider superimposed fraud, the TSP employee traces the phone line of wealthy customers (e.g. companies), taps the line, connects a receiver temporarily and makes calls for personal usage.

Figure 2.2: Public Landline Transmission Path

External superimposed fraud is committed similarly to insider superimposed fraud, in public places, possibly using the so-called local loop, and it can be done for commercial or for private use. A **local loop** is the transmission path from a legal customer's phone to the closest local office of the Telecommunication Services Provider (TSP). A hacker can also gain access to a subscriber network through several call trials to its PABX (Private Automatic Branch eXchange), which connects the trunk line to the PSTN (Public Service Telephone Network). *Spoofing* refers to a phone user who pretends to be someone else.

Figure 2.2 shows a public landline transmission path. Fraudsters can easily tap into other subscribers' accounts using specialised tools applied to the landline transmission paths. This is an example of external superimposed fraud.

In subscription fraud [5], a swindler steals the personal bank account data (e.g. account number) of someone who subscribes to the same or different network (TSP) and gains access to a land line service from a given telephone operator. The legal users may notice the subscription fraud only on their account, if they are very observant. Fixed line swindlers often stay only temporarily in a house or apartment, in order to conceal their whereabouts.

### 2.1.3 Mobile Phones

The subscription fraud explained above applies to mobile phone networks too. It is common and occurs frequently.

*Superimposed fraud* is committed when impersonators gain access to the accounts of legitimate users by stealing their unique Mobile Identification Number (MIN) and Equipment Serial Number (ESN) to perform *cloning* or *spoofing*. *Spoofing* refers to the action of a phone user who pretends to be someone else. A phone is said to be cloned when a Subscriber Identification Module (SIM) card is reprogrammed to transmit the MIN / ESN belonging to another legitimate cell phone [7]. The cloning variations [15] are given below:

- *Subtype-1* **duplicates** only one MIN / ESN on a new SIM card.
- *Subtype-2* is a type of **roaming fraud** where more than one MIN / ESN are stored on a SIM card. The continuous changing from one MIN / ESN to another is called **tumbling** because each call is billed to one MIN / ESN at a time. Thus, it is difficult for the owners to notice illegal calls.
- *Subtype-3* is **life time phone fraud**. In this case, the fraudsters could manually program any MIN / ESN stolen through handset keypads.

Subscribers are identified using a *smart card* called a SIM (Subscriber Identification Module), which contains the MIN / ESN. The MIN and ESN take the following forms:

MIN = Continent code + Country code + TSP type + Phone number.

ESN = Manufacturer's number + Serial number of the phone.

If a fraudster impersonates a legitimate user with a stolen MIN / ESN, the legitimate user is billed.

Research has shown that theft is another type of serious deceitful fraud in the mobile phone industry [25]. When a mobile phone is stolen, it is still superimposed on a legal user by spoofing the network. An example of a victim is a subscriber who is on contract phone and pays monthly bills to the TSP. Another example is the spoofing of a pre-paid phone with well loaded airtime. The stolen phone continues to be used by the fraudster until it is reported to the TSP for blockage, but only if the owner remembers his or her Personal Identification Number (PIN).

The superimposed fraud could be internal or external [43]. As in the case of land lines, internal fraud here is committed by employees who manufacture the SIM cards. Superimposed fraud is said to be external when unscrupulous people clone a stolen MIN / ESN onto a new smart card or keep on spoofing a stolen phone. When a stolen phone is reported, spoofing stops because the MIN / ESN is blocked, but the phone could still be used further for cloning.

**Deficient Security in Mobile Phone Networks**

For mobile subscription fraud, a cellular or GSM phone is collected on a contract agreement, where a periodic bill debits a legal bank account. Research shows that subscription fraud is mostly committed with regard to local calls, as these may not be noticed right away. Identification of this type of fraud will take time to be resolved by the TSP.

Internal fraud is caused by certain criminal employees of the SIM card company who sell MIN / ESN codes to fraudsters or use them themselves.

In the case of cloning, there is a computer program for sale on the Internet [68] that uses a search and replacement algorithm, removing the old MIN / ESN and replacing it with a new combination on the smart card. After executing the program, an electronic device called an Electrically Erasable Programmable Read Only Memory burner, (an**EEPROM burner**) uses ultraviolet light to clear and write onto the microchip of the SIM card.

### 2.1.4 The Unlawful Methods of Obtaining Security Numbers

#### Deficient MIN / ESN Security

The unlawful methods of trapping these security numbers are summarized below:

**Eavesdropping**: The fraudsters use an antenna, or a software-based electronic mobile communication scanner to trap valid MIN / ESN numbers by illegally monitoring the radio wave transmissions from the cell phones of legitimate subscribers.

**Cracking**: A specialised computer is used to read the *setup* menu of a mobile phone to trap the MIN / ESN. Handset technicians and motor vehicle mechanics are usually involved in this dubious business practice.

#### The Unlawful Techniques of Discovering Subscriber's PIN Numbers

The illegal methods [15] of acquiring PIN numbers are given below:

**Shoulder Surfing**: This is the technique of spying on users when they enter their PINs in public places through the use of cameras, telescopes or looking over their shoulders.

**Scanning / Cracking**: This involves the use of a computer and a sequential number dial-out program to obtain the PIN.

**Social Engineering**: This is a traditional criminal technique to trick gullible people into revealing their security number, believing that this is required for legitimate purposes.

### 2.1.5 The Constraints of Analog, Digital and ESN / MIN

Analog mobile phones (e.g. cellulars) are cheap and do not encrypt the MIN / ESN during transmission. This makes it easier for fraudsters to trap the security codes.

The MIN / ESN for digital phones (e.g. GSM) was initially well encrypted during transmission but was later cracked (made simpler) because of the decoding disparity from one country network to another [25]. Initially, when sophisticated encryption techniques were used to protect the security numbers for GSM handsets, there were restrictions on access and some countries found it difficult to decrypt these because of the expensive equipment required. As a result, internetwork linking

countries have far more lax security which is easier to breach. Little or no encryption is used with digital phones in these countries, which puts legitimate users at risk of being defrauded [26].

### 2.1.6 The Proposed Solution for South Africa

South African telephone operators intend to **blacklist** defrauded SIM cards which may disturb, but not stop lawbreakers. Blacklisting intends to make handsets and SIM cards useless. Currently, however, there is no uniform rule on blacklisting phones [59]. **Greylisting** involves blocking SIM cards but not handsets, an approach that is already in practice. These are imperfect solutions as the lawbreakers do not respect governmental policies. Consequently, making appropriate laws for the telephone industry is a challenge for policy makers. It becomes laborious for the government to keep making and enforcing laws to arrest people with fraudulent telephone activities. How many criminals who break these laws can be arrested? Therefore, more intelligent models are required to understand and detect anomalies in call data. When TSP detects too many anomalies in fixed line accounts, they cut down the phone lines as described by Frayne [21] for South Africa.

## 2.2 Prior Anomaly Detection and Modelling Research Applied to Bayesian Networks

This section presents the prior anomaly detection approaches in particular, to prevent telecommunications anomalies. These approaches are: rule-based, statistical, neural network, Bayesian networks and Distance-based. They are good methods but they require improvement. Many aspects, where they are limited in capabilities were stated in the problem definitions section in Chapter One. They are further discussed in detail below. Also, this section introduces the researchers that have worked on the two most popular techniques of modelling Bayesian networks.

### 2.2.1 Prior or Related Research in Telecommunications Anomaly Detection

As mentioned earlier, a number of methods have been used for telecommunication anomaly detection. These include: rule-based approaches, statistical techniques, neural networks, distance-based approaches, and Bayesian networks. It was presented in recent studies that unsupervised learning of Bayesian networks for anomaly detection, which was investigated in this research, is better than

countries have far more lax security which is easier to breach. Little or no encryption is used with digital phones in these countries, which puts legitimate users at risk of being defrauded [26].

### 2.1.6 The Proposed Solution for South Africa

South African telephone operators intend to **blacklist** defrauded SIM cards which may disturb, but not stop lawbreakers. Blacklisting intends to make handsets and SIM cards useless. Currently, however, there is no uniform rule on blacklisting phones [59]. **Greylisting** involves blocking SIM cards but not handsets, an approach that is already in practice. These are imperfect solutions as the lawbreakers do not respect governmental policies. Consequently, making appropriate laws for the telephone industry is a challenge for policy makers. It becomes laborious for the government to keep making and enforcing laws to arrest people with fraudulent telephone activities. How many criminals who break these laws can be arrested? Therefore, more intelligent models are required to understand and detect anomalies in call data. When TSP detects too many anomalies in fixed line accounts, they cut down the phone lines as described by Frayne [21] for South Africa.

## 2.2 Prior Anomaly Detection and Modelling Research Applied to Bayesian Networks

This section presents the prior anomaly detection approaches in particular, to prevent telecommunications anomalies. These approaches are: rule-based, statistical, neural network, Bayesian networks and Distance-based. They are good methods but they require improvement. Many aspects, where they are limited in capabilities were stated in the problem definitions section in Chapter One. They are further discussed in detail below. Also, this section introduces the researchers that have worked on the two most popular techniques of modelling Bayesian networks.

### 2.2.1 Prior or Related Research in Telecommunications Anomaly Detection

As mentioned earlier, a number of methods have been used for telecommunication anomaly detection. These include: rule-based approaches, statistical techniques, neural networks, distance-based approaches, and Bayesian networks. It was presented in recent studies that unsupervised learning of Bayesian networks for anomaly detection, which was investigated in this research, is better than

most detection methods [38]. This is because it does not require a prior knowledge of fraudulent data and can detect new anomalies from call data.

The *rule-based* techniques work best with user profiles containing explicit information, where anomaly criteria are encoded as rules [19]. This technique is not scalable and is difficult to manage because it requires explicit rules which is labour-intensive and time consuming, and involves programming for every imaginable possible anomaly. It is difficult to incorporate new anomaly types into such rules as they occur. The performance of the rule-based approach degrades drastically as the call data size grows. This rule was used by Fawcett [19] to classify calls in a subscriber's account, for example:

(Time-of-Day = Night) AND (Location = Bronx) $\Longrightarrow$ fraud.

The rule above means this customer does not make calls at night in Bronx. If he does, an alarm should be raised. This may not always be true since the behaviour of the customer can change at anytime. Our approach takes care of this effectively. This approach is described in anomaly indicators and detection results in Chapters Four and Five respectively.

The *statistical* methods of anomaly detection works well with univariate distributions, where the average call duration, the longest call duration and the average number of calls per day are usually compared with a pre-determined threshold [43]. As the threshold is taken at a particular time period, however, the statistical threshold model will not adapt to changes in the call behaviour of a given customer. If the number of calls for a day exceeds its normal behaviour (threshold), it is considered as anomalies which may not also be true.

The popular *neural network* technology [6] employs feature extraction, where summarised statistics are usually used to train models with Call Detail Records (CDR). The extracted features from CDR are pre-classified as comprising either of a regular or an anomalous call. This technique can detect anomalous calls that already exist in the training data. The detection quality of new types of anomalous calls can be questioned.

Hollmen [28] presented the use of a *Bayesian network* in anomaly detection by profiling users / behaviours into groups and used known different fraud scenarios. The Bayesian network was not mined from call data but in our research, we fully mined an individual model from data. Hollmen used a Bayesian network classifier drawn from expert knowledge with a prior knowledge of known fraudulent scenarios. The appropriateness of this method to detection of new anomalies

is questionable. Hollmen however reported that their experiment detected 85 % of anomalies. It was not clearly stated whether they used real life or generated call data but they proposed to test their method on mobile phone real-world size.

Bayesian networks have been successfully applied in medicine to diagnose diseases such as pneumonia [40], and speech recognition [66]. Mining Bayesian network models from data has also been successfully applied in several applications [16] but few of such Bayesian network technology have been applied to telecommunications call data. This may probably be due to its computational time or being a new research area, but our work illustrates a proof of concept on the capability of mining Bayesian network, and opens up future direction to improve mining speed. This is our new idea.

*Distance-based* anomaly detection is described in [2], [34]. This was already discussed in section *Problem Definitions, Motivations and Research Questions* of Chapter One. In this method, an absolute distance between a dataset and a defined point as a distance outlier is used. This approach focuses primarily on continuous attributes. Our approach differs from theirs as we can also process mixed datasets.

Simple time series analysis has been applied to understand and detect anomalies in various datasets other than telecommunications. [42] applied this model to identify disease outbreaks by studying the sale of Over-The-Counter (OTC) drugs in pharmacy shops. Their algorithm creates a map of a uniform rectangular N x N grid where each cell corresponds to a search region. A search is then conducted along all axis-aligned areas on the grid to find regions that have shown a recent anomalous increase in sales. The regions that show high deviation in sales from the estimated baselines are labelled as alerts of OTC sales that may indicate disease outbreaks. This is an absolute analysis method, whose efficiency earlier researchers [8] in telecommunication anomalies found questionable.

In view of all these methods above, we used and proposed individual modelling to facilitate anomaly detection systems. The next section introduces the prior work on modelling Bayesian networks.

### 2.2.2 Introduction to Prior Work on Modelling Bayesian Networks

We studied Hill-Climbing and consider Genetic algorithms in this research to model Bayesian networks that we used to monitor the behaviour of telecommunication subscribers. We introduced over six researchers' prior work on modelling Bayesian networks and the detailed background on

this Bayesian technology will be described in Chapter Three.

Hill climbing algorithms are greedy score-based algorithms that continuously move uphill and may be terminated when there is no further learning improvement in the specific network structures. A new Max-Min hill-climbing method was developed by Ioannis [30] as a variant of hill-climbing algorithms. It was confirmed that some of the operators used for modelling in this technique do not provide any theoretical guarantee but this is a future direction. Also, Josep [33] developed an incremental hill-climbing method as a variant of standard hill-climbing to learn a Bayesian structure from data. In his work, he confirmed that the initial pattern of variables presented for modelling determines its output. It therefore implies that there can be a problem of generating several models as opposed to obtaining one optimal network. This sensitivity to variable ordering problem requires an improvement. This prior work is discussed further in the next chapter.

A genetic algorithm (GA) is a stochastic search algorithm in which a large population of states is maintained where new states are generated by mutation and crossover processes [58]. Minimum Description Length Evolutionary Programming (MDLEP) was developed by Wai [67] to mine BN structures based on the MDL principle and on evolutionary programming. They developed three genetic operators which include: structure-guided mutation, knowledge-guided mutation, and freeze and defrost operators to mine Bayesian networks from data. Some of their operators can be combined to save modelling time. Larranaja [39] used a GA to generate variable orderings to evolve several networks during BN learning from complete data. His work is an alternative method of using GA to model networks from data. William [70] developed a genetic algorithm to solve the sensitivity to variable ordering problem encountered by Josep [33]. His work was like combining hill-climbing and GA. So, he hopes to use only GA to mine network in future research.

In this research, we are motivated to use GA to model individual subscribers behaviour based on the satisfactory testimonies of these related researchers. Further information on these concepts of these techniques and their applications are described in Chapter Three.

## 2.3 Conclusion

This chapter has expressed the existence of the problems in telecommunications anomalies, and its losses in both developed and underdeveloped countries. The strength and weakness of the prior research to combat the problems under investigation have been reviewed and related techniques of

modelling Bayesian networks were introduced.

The strength and limitations of different anomaly detection methods gave direction to the current research. As a result of this, the first step was to investigate fraud types and their relationships from the above literature references with the scientific methods of how they are committed. Moreover, the reasons (laxity) behind defrauding TSPs were reviewed in order to understand and assist call analysts in making decisions when they detect suspicious call records.

A proposed solution for South Africa and the review of prior work in this field were presented in this chapter. The strengths and weaknesses of the prior work and the gap that this research covers were discussed. It became clear that anomaly detection may not prevent fraud from being committed by criminals but it helps the TSPs to understand where and when anomalies set in to their networks' data. When they understand these, then they can tighten their networks' security. This can frustrate the criminals and thereby minimise their fraudulent activities.

The next chapter presents the relevant background required for this research.

# Chapter 3

# Research Background

## 3.1 Background on Probabilistic Modelling

This section explains the necessary concepts of knowing the past, understanding the present and predicting the future using probabilistic modelling. It cuts across the concepts of Artificial Intelligence, analysing Bayesian Network models, and related research in anomaly detection.

### 3.1.1 Types of Artificial Intelligence

Artificial Intelligence (AI) is a field of computer science that attempts to inculcate the intelligence of humans into a machine in order to solve problems. It is almost widely accepted that the two types of AI are *strong* and *weak* [50].

*Strong AI* is the creation of computer-based artificial intelligence that can reason and solve problems with self awareness. There has been no standard example of this yet, but strong AI assumes the human mind is software and the brain is purely hardware. Thus, the assumption is that a computer system can roughly approximate a human being, if it can include all the properties of human minds, such as consciousness.

*Weak AI* is the creation of computer-based artificial intelligence that can reason and solve problems only in a *limited* domain; e.g. a probabilistic model applied to detect anomalous call records in the telecommunication domain. Other examples include pattern recognition, image processing, neural networks, robotics etc.

A moderate level of progress has been made in weak AI [58], and this is viewed as *the set of Computer Science problems without good solutions.* We intend to make our own contribution to this field of study by using Bayesian Networks, as probabilistic models, to detect anomalous activities in telecommunication call records. This detection requires probabilistic reasoning with uncertain information. It primarily involves mining patterns from subscribers' behaviour together with the use of Bayesian inferences and differential analysis. The prediction of calls gives rise to degrees of beliefs that a call is regular or anomalous. This imposes a level of confidence (probability) on problem solving. The background on Bayesian networks will be reviewed in the next sections.

### 3.1.2 Fundamentals of Probability Theory and Bayes' Theorem

A Bayesian network represents a set of random variables and their causal interrelationships in a given problem domain. A Bayesian network requires discrete random values such that if there exists random variables $X_1, ..., X_n$ with corresponding values $x_1, ..., x_n$ then their joint probability is given by the term [53]:

$$Pr(X_1 = x_1, ..., X_n = x_n) \tag{3.1}$$

The important properties of discrete probability distributions are as follows [53]:

$$0 \leq Pr(X_i) \leq 1, \text{ where i} = 1, 2, ... \text{ n}$$

$$\sum_{i=1}^{n} Pr(X_i) = 1$$

These basic properties are used in Bayesian networks, which are multivariate models that uses Bayes' theorem to solve complex problems like detecting anomalies in callers' records. For instance, to understand the solution to this expression,

$$Pr(A\ call\ from\ 27720251590 = true \mid user's\ behaviour)\ = ?$$

we need to understand Bayes' theorem, which first requires the knowledge of conditional probability.

The concept of conditional probabilities forms the basic building block of Bayes' theorem. For any two random variables $X_i$ and $X_j$, the conditional probability of $X_i$ given $X_j$ is defined in equation 3.2.

$$Pr(X_i \mid X_j) = \frac{Pr(X_i,\ X_j)}{Pr(X_j)} \tag{3.2}$$

Using equation 3.2, together with the chain rule [50] of conditional probabilities, we have equation 3.3.

$$Pr(X_i,\ X_j) = Pr(X_i \mid X_j)Pr(X_j) \tag{3.3}$$

The order of choosing $X_i$ and $X_j$ does not matter in equation 3.3, as in equation 3.4.

$$Pr(X_j,\ X_i) = Pr(X_j \mid X_i)Pr(X_i) \tag{3.4}$$

Now, if we equate the right hand sides of the equations 3.3 and 3.4, we have equation 3.5.

$$Pr(X_i \mid X_j)Pr(X_j) = Pr(X_j \mid X_i)Pr(X_i) \tag{3.5}$$

Thus, Bayes' theorem is given as equation 3.6.

$$Pr(X_i \mid X_j) = \frac{Pr(X_j \mid X_i)Pr(X_i)}{Pr(X_j)} \tag{3.6}$$

From equation 3.6, $Pr(X_i \mid X_j)$ is called the *posterior probability.* The posterior probability of $X_i$ (the hypothesis) is obtained after making observations or studying the behaviour ($X_j$) of a user. It is the original degree of belief when the likelihood and prior are combined.

Also, $Pr(X_j \mid X_i)$ is the *likelihood* function of $X_i$ given $X_j$ in the posterior. It is taken as the conditional probability of what we know (the evidence $X_j$) based on what we do not know (the hypothesis $X_i$). In other words, it is the probability of seeing the observation $X_j$ given that the hypothesis $X_i$ is true.

In Bayes' theorem in equation 3.6, the *prior* probability of $X_i$, $Pr(X_i)$, is the probability of $X_i$ before making any observation or any inference.

The *marginal* probability of $X_j$ is mostly taken as a normalising constant and in expression 3.7,

$$\frac{Pr(X_j \mid X_i)}{Pr(X_j)} \tag{3.7}$$

it is called the *scaling factor* because it gives a measure of the impact that observations have on the belief of the hypothesis. Supposing variable $X_j$ has two values t and f in a dataset, the probability obtained by adding the joint probability of $X_j = t$ without considering f, is marginal. A good example is shown in equation 3.21.

### 3.1.3 The (In)dependencies in Bayesian Networks

Formally, a Bayesian belief network is a directed acyclic graph represented in expression 3.8.

$$G = \{V(G), A(G)\} \tag{3.8}$$

where V(G) implies expression 3.9,

$$V(G) = \{V1, ..., Vn\} \Rightarrow \ vertices\ of\ graph\ G \tag{3.9}$$

and a set of arcs is described in expression 3.10,

$$A(G) \subseteq V(G) \times V(G) \tag{3.10}$$

All these represent the causalities or dependencies among the probabilistic variables. A parent variable refers to the *cause* while a child variable means the *effect.*

Figure 3.1 shows the graphical representation of a Bayesian network model, which describes the qualitative knowledge about a particular domain. This probabilistic model gives an efficient description of multivariate probability densities. In our research, the nodes

$$n_i, \ \ where\ i = 1, 2, ..., 6.$$

could represent anomaly indicators with the dependency relationships as arcs. The following concepts will be continually used in the subsequent sections.

**Conditionally Dependent**: In Figure 3.1, $n_5$ is conditionally dependent on $n_1$; we denote and compute the probability as: $Pr(n_5 \mid n_1)$. Additionally,

- $n_4$ is conditionally dependent on $n_1$ and $n_2$; we denote and compute the probability as: $Pr(n_4 \mid n_1,\ n_2)$,
- $n_3$ is conditionally dependent on $n_2$; we denote and compute the probability as: $Pr(n_3 \mid n_2)$,
- $n_6$ is conditionally dependent on $n_4$ and $n_3$; We denote and compute the probability as: $Pr(n_6 \mid n_4,\ n_3)$,

**Conditionally Independent**: An *evidence* is a value that is seen or observed with certainty in real life. A node is said to be non-empty if it contains value. If $n_4$ is a non-empty (evidence) node then $n_6$ is conditionally independent of $n_1$ and $n_2$. In other words, this means $n_6$ does not depend on $n_1$ and $n_2$ but only depend directly on $n_4$. Also, the only path from $n_5$ to $n_4$ is blocked by node $n_1$ then, we say that $n_5$ is *d-separated* (conditionally independent) from $n_4$ [50]. Using Bayes' rule, we say that $Pr(n_5 \mid n_1)$ is conditionally independent of $Pr(n_4 \mid n_1,\ n_2)$. In fact, this is an important property that the belief network exploits to reduce the number of probabilities to be computed as a model which makes intractable problems feasible. That is, the theory of conditional independence makes a child node to consider only its direct parent nodes when computing probabilities.

**Unconditionally Independent**: A node is said to be empty if it does not contain a value or an evidence to propagate into other nodes. In Figure 3.1, suppose $n_2$ is empty then, we say that $n_3$ is unconditionally independent of $n_4$. If $n_2$ does not have an evidence to propagate into $n_4$, this probably implies that $n_3$ does not also propagate significant value into $n_2$. That is, by Bayes' theorem, we can also say $Pr(n_4 \mid n_1)$ is unconditionally independent of $Pr(n_3)$.

**Joint Probability**: A *query* node is a target node that we want to infer or whose probability we want to know. The joint probability of a model depends on what is expected in a query node given certain evidence. Using Bayes' theorem, the joint probability density distribution is defined in equation 3.11.

$$Pr(V_1, ..., V_n) = \prod_{i=1}^{n} Pr(V_i|\pi(V_i)) \tag{3.11}$$

where $\pi(V_i)$ represents a set of parent(s) of $V_i$. The joint probability distribution for the BN in Figure 3.1, is shown in equation 3.12.

**Conditional Probability Table (CPT)**: This is a form of table that contains the conditional probability distributions for discrete variables. Each row in a CPT contains the conditional probability of each node value for all the possible combinations of values for its parents. Each column must sum up to one [58]. Examples of CPTs are shown in figures 4.3 and 4.4, for some nodes on the Bayesian network in Figure 4.2.

$$Pr(n_1, n_2, n_3, n_4, n_5, n_6) = Pr(n_1)Pr(n_2)Pr(n_3 \mid n_2)Pr(n_4 \mid n_1, n_2)Pr(n_5 \mid n_1)Pr(n_6 \mid n_3, n_4) \tag{3.12}$$

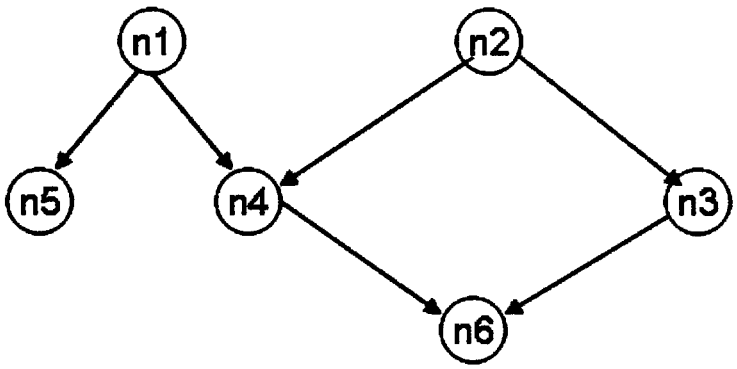


Figure 3.1: A Sample Bayesian Network Model

### 3.1.4 Further Characteristics of Bayesian Networks and Learning

Some important features of Bayesian networks apart from those mentioned in the previous two subsections are learning, probabilistic reasoning, level of confidence, and uncertainty.

**Learning**: The various scenarios in which Bayesian network learning can be applied are with known structures (models) with complete data (records), known structures with incomplete data, unknown structure with complete data, and unknown structures with incomplete data. The known structure implies supervised learning, while the unknown structure implies unsupervised learning.

Our research intends to understand the behaviour of individual subscribers by mining (unsupervised learning) structures from call data. Bayesian networks are mined from training data and the results are depicted graphically as qualitative knowledge and quantitative knowledge is captured in the conditional probability tables (CPTs).

**Probabilistic reasoning**: Causal (top down), Diagnostic (bottom up) and Explaining Away are the three patterns of reasoning observed in Bayesian networks [50].

The *causal inference* technique uses a cause to infer an effect. In other words, when a cause is observed, we can generate or predict the possible effects. For instance, from Figure 3.1, an example of causal inference is given by: $Pr(n_5 = true \mid n_1 = true)$.

The *diagnostic inference* technique uses effects (or symptoms) as evidence to infer causes. It is normally used in expert systems. From Figure 3.1, an example of diagnostic inference is: $Pr(n_1 = true \mid n_5 = true)$.

*Explaining away* is a reasoning technique where the change in belief is a possible explanation, if an alternative explanation is actually observed [50]. For instance, we may want to ask; what is the $Pr(n_4 \mid n_1)$ and $Pr(n_4 \mid n_6)$? These two results can be explained to make better decisions. This type of reasoning is called Berkson's paradox [50], and it uses a causal reasoning step embedded within a diagnostic reasoning.

In this research, almost all the observed nodes will be used as evidence nodes. These evidence nodes are used during reasoning as anomaly indicators in the subscribers' models, when detecting regular or anomalous calls.

Any of these reasoning techniques can be used to detect abnormal patterns from callers' records. This decision depends on the hypotheses to be tested on the belief models we generate in this research.

**Level of confidence**: The Bayesian network model uses a *degree of belief*, which is equivalent to probabilities, to express the confidence level of a result. For instance, if a Bayesian belief model

gives the output of a probability distribution about an entity such as: $Pr(calling\ destination = true) = 0.92$ and $Pr(calling\ destination = false) = 0.08$, we can then say that we have 92 % confidence that this is a regular call.

**Uncertainty**: Bayesian network models are used to reason about uncertain information. Uncertainty arises as a result of estimating from past (historical) and present behaviour to predict future behaviour. Uncertainty also arises when data is incomplete.

**Adaptive Systems**: A system is said to be *adaptive* if it can adjust to its environment. The behaviour of telecommunication subscribers changes on an ongoing basis. We called a Bayesian network that models an individual subscriber's behaviours, a Behavioural Bayesian Network (BBN). We used BBNs to adapt to changes through incremental learning.

Even though cycles can be included in some Bayesian networks [53], we do not consider cycles. This is beyond the scope of this research. In the next sections, we will discuss the differences between singly-connected and multiply-connected Bayesian networks but the next section states the theory of Maximum Likelihood Estimate (MLE) and Parameter Learning.

### 3.1.5 The Theory of Maximum Likelihood Estimate (MLE) and Parameter Learning

According to [58], the MLE is a reasonable approach to estimate probabilities for the parameters of a mined Bayesian network especially when a training dataset is large. It is a good approximation of Bayesian learning because it becomes difficult to prefer one parameter over another (biased) and, thus, uniform priors are assumed over all parameters. This is an unbiased estimate because it allows estimation to be driven by the dataset.

Supposing there are two variables x and y in a dataset with values that sum to N. If the parameter of x represents the proportion or the probability as $\gamma$ then, parameter of $y = 1 - \gamma$. Therefore, the likelihood of the dataset (d) with the required hypothesis (h) is given as follows [58]:

$$Pr(d|h_\gamma) = \gamma^x(1-\gamma)^y \tag{3.13}$$

The likelihood of h is given by the value of $\gamma$ that maximises the log likelihood:

$$L(d|h_\gamma) = logPr(d|h_\gamma) = \log(\gamma^x(1-\gamma)^y) \Rightarrow x\log\gamma + y\log(1-\gamma) \tag{3.14}$$

The next equations 3.15 to 3.17, differentiate L w.r.t $\gamma$, and equate to zero to obtain the maximum likelihood estimate of $\gamma$;

Since the differential of

$$\log\gamma = \frac{1}{\gamma} \tag{3.15}$$

then,

$$\Rightarrow \frac{dL(d|h_\gamma)}{d\gamma} = \frac{x}{\gamma} - \frac{y}{1-\gamma} = 0 \tag{3.16}$$

Therefore,

$$\gamma = \frac{x}{y+x} = \frac{x}{N} \tag{3.17}$$

This is the deduction of the instance counts which are sufficient statistics discussed by [46]. Hence, this formulates the basis of the quantitative knowledge of Bayesian learning that computes the MLE of Conditional Probability Tables (CPTs) associated with each node of a Bayesian network.

For the Bayesian learning, the CPTs are computed from the instances of training records and the conditional probability distribution of a BBN. For instance, Figure 4.2 shows the network of Dr Watson's grass. It has four nodes with two states each. The nodes are Cloudy(C), Sprinkler(S), Rainy(R) and Wetgrass(W) and the states are true(t) and false(f). The joint probability distribution is given in equation 3.18.

$$Pr(C,S,R,W) = Pr(C)\ Pr(S \mid C)\ Pr(R \mid C)\ Pr(W \mid S,R) \tag{3.18}$$

The probabilities in each CPT are calculated such that all the node states are computed. For example, the knowledge of node C is calculated as marginal probabilities. Since C has two states t and f, then the probabilities are calculated in equations 3.19 and 3.20.

$$Pr(C = t) = \frac{\#(Instances\ of\ C\ =\ t)}{Total\ number\ of\ instances\ of\ C} \tag{3.19}$$

$$Pr(C = f) = \frac{\#(Instances\ of\ C\ =\ f)}{Total\ number\ of\ instances\ of\ C} \tag{3.20}$$

These results are captured in the conditional probability table (CPT) associated with node C. Also, the conditional probabilities of node $R$ given node $C$ is calculated in equations 3.21 and 3.22.

$$Pr(R = t \mid C = t) = \frac{Pr(R = t,\ C = t)}{Pr(C = t)} \tag{3.21}$$

$$= \frac{\#(Instances\ of\ R = t,\ C = t)}{\#(Instances\ of\ C = t)} \tag{3.22}$$

for all values of states t and f.

Furthermore, the other probabilities in equation 3.18 are calculated in the same way. See Figures 4.3 and 4.4 for the CPTs of Rainy and Sprinkler nodes.

Having calculated the CPTs of the BBNs in our application, they are representations of the quantitative knowledge acquired from the call records about the behaviour of a particular phone user. Since learning is a continuous process, the BBNs are retrained on an ongoing basis. See incremental learning loop in Figure 4.1. The next section describes Bayesian inference as it is required for prediction in this research.

### 3.1.6 Bayesian Inference

After Bayesian parameter learning, the most important benefit of using Bayesian networks in real life applications is to carry out probabilistic inference. Bayesian inference is statistical inference

According to [8], absolute analysis in detection system is a fixed trigger system that raises an alert when anomalies are suspected in a dataset. Absolute analysis is more useful in the static world where patterns of operations are procedural, for instance, when analysing repairs of car faults or diagnosing diseases. It is therefore good at detecting *extremes* of fraudulent activities in telecommunications. In differential analysis, the trigger changes dynamically as the behaviour changes.

Differential analysis can be used to monitor the behavioural patterns associated with every telephone subscriber by studying his or her historical call activities. For example, a behavioural pattern detected as an anomalous call for subscriber X may be accepted as a regular call for subscriber Y. As subscribers' behaviours are representations of a dynamic world, absolute analysis would not be able to detect telecommunication anomalies and differential analysis is more appropriate for the implementation of the ADS presented herein.

The next section describes the differences between singly and multiply-connected networks.

### 3.1.8 Differences Between Singly and Multiply-connected Networks

For singly-connected networks, there is only a single path between any two given nodes while there are more than one path between any two given nodes in multiply-connected networks. Figure 5.25 shows a representative example of a singly connected network mined from UCI Irish data [49], using the Weka program [69]. Also, we present in Figure 5.26, a good example of a multiply-connected network that we mined, with the same Irish data, using our genetic algorithm (HGA). In Figure 5.25, observe that there is only one path from *sepallength* to *class* while in Figure 5.26, there is more than one path from *petallength* to *class*.

Exact inference works well on singly-connected networks while approximate inference performs better on multiply-connected networks [31]. More information is provided on their differences by Russel [58].

Our HGA has the ability to mine multiply-connected networks. Multiply-connected Bayesian networks are important to us in this research because one variable can easily influence many others via more than one path in a network that models subscriber's behaviour.

## 3.2 Bayesian Networks and Machine Learning

Machine learning forms part of the field of AI. Russel [58] defines machine learning as the ability to adapt to new circumstances, to detect, and extrapolate patterns. A machine is said to learn whenever it gains knowledge, changes execution based on the information acquired in order to improve in its performance [50]. The field of machine learning that is applied in this research is computational intelligence.

Computational intelligence is the study of adaptive mechanisms that exhibit an ability to learn or adapt to new situations, to generalise, abstract, discover and associate [18]. This describes numerically based AI distinguished from symbolic AI. Some examples of machine learning studies used in computational intelligence mechanisms are discretization, inference, searching and optimisation algorithms such as using evolutionary algorithms for mining a Bayesian network. We researched and implemented some of these algorithms.

### 3.2.1 Mining Bayesian Structures from Telecommunications' Data to Encode Knowledge

We mine Bayesian network structures from data because it provides good knowledge models that can reveal the behaviour of a system [33]. Since historical data keeps track of every activity of a system, Bayesian networks can model this data, extract the hidden information and may even be able to explain the present to predict future activities. For instance, in a telecommunication system, the intention of phone users may not be known, but it is reflected in their call data. Therefore, the individual Bayesian network models that are mined from the data are good representations of the individual subscribers. Further reasons why the mining of Bayesian networks from data will be useful in anomaly detection include:

- Knowledge is expensive to acquire and most of the time, there are no experts to interpret environments or model a domain. Since data is cheap and has useful information about the environments, Bayesian networks can encode this hidden information as knowledge.
- For many real world applications, Bayesian networks effectively capture and reveal the hidden patterns in datasets and encodes the patterns in form of graphical structures and conditional (in)dependencies [53], [50].

- Bayesian networks (BNs) provide insight into knowledge discovered from data about a domain and it is capable of handling uncertainties [50]. Consider the subscribers' models in Figures 5.1 to 5.17. Decisions about anomaly detection are being made despite the uncertainty about subscriber behaviours, and relationships between destination number and call duration attributes, relationships between destination network and location attributes etc.

- During probabilistic learning, a BN can help to deal with hidden attributes and it can estimate missing data.

- When the behaviour of a system is known or static, then it does not require the network to learn or extrapolate future behaviour from data. However, if it is dynamic such as subscriber behaviour and if behaviour cannot be pre-determined, then machine learning techniques are necessary.

- Unlike with small datasets, when an organisational dataset is large, domain experts may find it difficult to manually build a Bayesian network model that can explain the activities in the data. A learnt model pattern from a given environmental state must be able to adapt to a changed environment in order to improve its intelligence.

- Subscribers' models evolve on an ongoing basis. It can only be mined from data because subscribers' call information grows rapidly and their behaviour changes continually.

### 3.2.2 Categories of Bayesian Learning

We can describe Bayesian learning as the task of finding the best Directed Acyclic Graph (DAG) that fits a training set of data, which results in qualitative and in quantitative knowledge stored in the Bayesian network. The qualitative knowledge is captured in the connection of nodes and the directed arcs. The directed arcs show causalities or dependencies among the nodes. The quantitative knowledge is captured in the Conditional Probability Tables (CPTs) in the data. Both the DAG and CPTs can be learnt from training data.

As stated earlier in the discussion of the characteristics of Bayesian networks, [50] and [46] classified Bayesian learning problems as follows:

**Known Structure and Complete Data**: In addition to our reasons for learning Bayesian structures, when human experts come up with an appropriate network that represents a problem

domain, then we say that the structure is known. It is left to learn the Conditional Probability Tables (CPTs) using the known structure and a complete dataset. A suitable machine learning algorithm for the CPTs using observed data is the Maximum Likelihood Estimate (MLE) [46].

**Known Structure and Incomplete Data**: By retaining the same meaning as in known structures above, incomplete data means that some data are missing in the training set. A solution to the partially observed data by machine learning algorithms is provided by the Expected Maximisation (EM) algorithm used to predict the missing values [50].

**Unknown Structure and Complete Data**: In addition to learning the CPTs for complete data, an unknown structure means that human experts cannot pre-determine the network that best fits a dataset. The solution is to search through the network space and use heuristic techniques to mine Bayesian structures from data. This is the case considered in this research.

**Unknown Structure and Incomplete Data**: With the usual meanings retained from the above discussion, a solution to this problem is simply to combine the EM algorithm and the heuristic techniques to be discussed.

Given this wide range of problem areas, we have decided to address only the problem of unknown structures and complete data in this research.

### 3.2.3 Techniques of Mining Bayesian Structures from Data

Mining Bayesian Networks from observed data is a machine learning application, which searches for a suitable structure that models a dataset. It is the first process of reasoning under conditions of uncertainty and it is often referred to as the discovery of causalities or dependencies in datasets [62] [54]. Many machine learning researchers have shown and accepted that, learning Bayesian Networks from data is an NP-hard problem [11]. NP-hard means finding a solution to a problem that runs in a Non-deterministic Polynomial time.

We present a simple combinatorial analysis in Figure 3.2, which illustrates the complexity of network learning problems. It is a combinatorial problem because the use of the movement operators such as arc reversals and arc deletions can continue indefinitely. Arc reversal is changing the direction of an arc between two nodes while arc deletion is the removal of an arc between two nodes.

Observe that there are infinitely many Directed Acyclic Graph (DAG) solutions in X, if $X =$

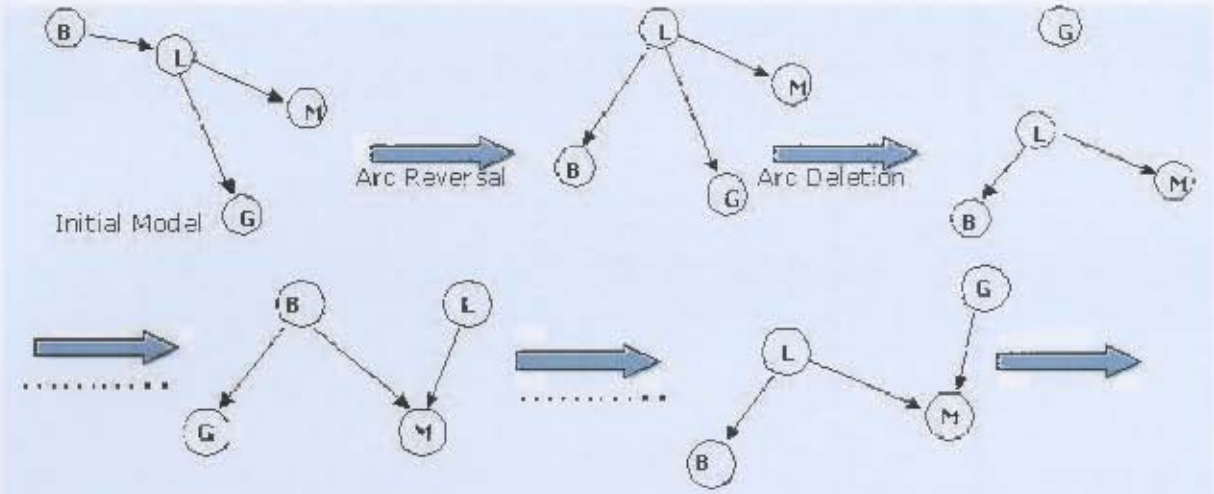


Figure 3.2: Combinatorial Analysis Problem in the Network Space

$\{x1, x2, ...xn\}(\equiv \{M, G, L, B\}$ in Figure 3.2) is a network space. Consequently, heuristic search techniques are being used in the literature to find optimal Bayesian network models from datasets.

According to Russel [58] an heuristic is an educated guess of learning, solving complex problems or making decisions. A heuristic-search technique is a search strategy that is guaranteed to find a good enough model in a reasonable time, but which may not always find an optimal or universal model from a dataset. The objective of every BN structure search algorithm is to keep finding edges and nodes that will give the best network in a reasonable finite time. This objective tends to minimise search time and space usage. In terms of storage, we are looking for a network that will minimise memory usage through the fitness of our objective function. The *objective function* of a structure learning algorithm is to find an optimal network with a global minimum score. More information is in the MDL section in Chapter Four. Popular examples of various search techniques to mine network structures are hill-climbing, simulated annealing and genetic algorithms. The details of these are described in the subsequent sections.

### 3.2.4 Hill-Climbing Methods applied to Mining Bayesian Network Structures

Hill climbing algorithms are greedy score-based algorithms. A greedy search is a best-first search in which the most promising node is chosen for expansion according to some specified rules. Furthermore, the hill climbing search algorithm continuously moves uphill and may be terminated when there is no further learning improvement in the specific network structures. It is useful to study the

details of hill-climbing because most advanced search and optimisation techniques use it as basis. Hill-climbing is popularly referred to as the K2 algorithm and it is characterised as follows in the next paragraphs according to [58] [22].

**Common Characteristics of Hill-Climbing Algorithms**

The common characteristics of hill-climbing algorithms are; it learns structures very fast; it does not maintain a search tree or space; it keeps only the *current state* (network) and the objective function value (network goal). The network goal depends on the design, its implementation and is commonly reached by convergence; and it does not look ahead beyond the immediate neighbor of the current state.

A schematic hill-climbing algorithm extracted from [45] and [58] is enumerated below. By definition, let *model* = *state* and *goal model score* = *objective function value* = *minimum score.*

**Schematic Hill-Climbing Algorithm**

1. Evaluate and score *initial model*
2. if *initial model score* $\equiv$ *goal model score* then exit,
   else *current model* = *initial model.*
3. Select a new operator to generate a *new model.*
4. Evaluate and score the *new model.*
5. if *new model score* $<$ *current model score* then, *current model* = *new model.*
6. if *current model score* $\approx$ *goal model score* then exit,
   else repeat step 3.

The commonly used operators are: addition, removal and reversal of arcs [50], [11], [30] as illustrated in Figure 3.2. They are specifically used to transform a current Bayesian Network to another during learning (see fig. 3.2). By definition, an **edge addition** operator is used to join two dependent nodes of a current network with an arc. An **edge removal** is used to delete an arc between two nodes, and it is normally used to break a cycle that was created in the current network. Also, an **edge reversal** is a transformation that requires a change of arc in the opposite direction between two nodes. Thus, the choice of an operator and the details of its mechanism differs from one

machine learning researcher to the other. With reference to Figure 3.2, we verified and simulated this method with a small block lifting dataset in [50] and its performance is shown in Figure 3.3.

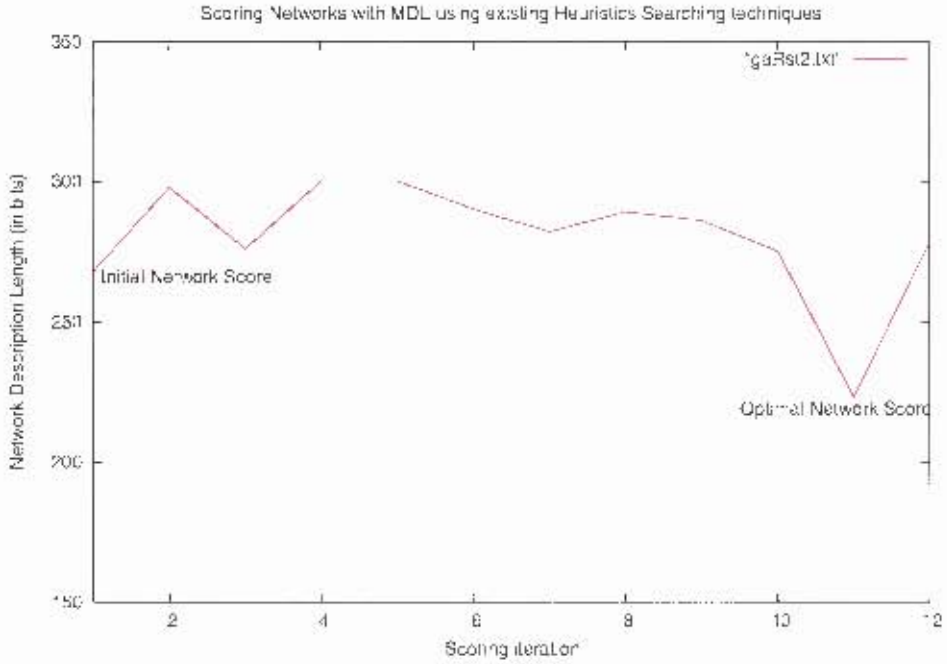


Figure 3.3: Heuristic methods: The behaviour of the hill climbing method when finding an optimal network for a block lifting data [50]

Despite the fact that hill-climbing is fast, it also gives rise to a further research question, viz; when does (or should) the search (choice of operators) stop to avoid the process becoming stuck or converging at a local minimum? A local minimum is an intermediate solution point for an iterative process. An example of a local minimum point in Figure 3.3, is (x, y) = (7, 280). To overcome this problem, many variants of the hill-climbing approach have been developed and are introduced as follows [58]:

**Stochastic hill-climbing** chooses variables at random with some probabilities of selection. It takes longer, but produces a better solution than the straightforward K2 approach.

**First-choice hill-climbing** implements stochastic hill-climbing, and randomly generates successors until one is generated that is better than the current model.

**Random-restart hill-climbing** carries out a series of hill-climbing searches with randomly generated initial models.

### 3.2.5 Further Related Research on the Hill-Climbing Algorithms applied to Bayesian Networks

The variants of hill-climbing research applied to many Bayesian network structures were investigated in order to understand their strengths and weaknesses.

Ioannis [30] presented a Max-Min hill-climbing method used to learn Bayesian Networks from data. They first reconstructed a skeleton of a Bayesian Network and then performed a Bayesian-scoring greedy hill-climbing search to orient the edges. The orientation of the edges used any of the hill-climbing operators described earlier, and mathematical mechanisms were defined to choose the operators. Consequently, Ioannis stated that the orientation phase did not provide any theoretical guarantee but they hoped to improve on it in the future.

Josep [33] proposed an incremental hill-climbing search to learn a Bayesian structure from data. He also combined the operators of K2 with an incremental search to improve on computational time. However, he explained that incremental hill-climbing algorithms are very sensitive to ordering effects. That is, when two sample orderings $O_1$ and $O_2$ of a training set are presented to the algorithm, it may output different domain models. This is a general research concern that must be avoided.

To this end, the following are further characteristics of hill-climbing as identified and discussed in [45] and [36].

#### Further Characteristics of Hill-Climbing

One characteristic of hill-climbing is its inability to backtrack. That is, to undo the action of an operator. This is as a result of its characteristic that it does not have memory. It is also trivial to program.

Another one is that its movement set design is critical. With reference to Figure 3.2, the movement set is the choice of operators such as reversal and deletion of arcs. This is a result of large network space and nodes dependency problems. There is no universal pattern of choosing operators at each mining step.

Also, if the number of moves with operators is enormous (e.g. in large datasets), the algorithm may be inefficient. Moreover, if the number of moves is small, the algorithm can get stuck easily.

In view of the above, we conducted further research into advanced approaches (presented in the following section) to mine structures from data.

### 3.2.6 Stochastic Techniques for Mining Bayesian Networks

The optimisation techniques of mining structures that are recommended for the purposes of this research are Simulated Annealing (SA) and Genetic Algorithms (GA). They are stochastic in nature because the probability that they use to select successors, such as offspring, is an improving function that minimises the objective function value. The algorithms are described as follows:

**Simulated Annealing** is an algorithm that combines hill-climbing with a *random walk*. The latter refers to a gradual movement to choose a new structure uniformly at random from a solution space of all possible networks. Although, the SA method has been proved in the literature to be better than hill-climbing, we are not motivated to use it to model telecommunication subscribers because:

1. it has more positive testimonies in solving manufacturing scheduling optimization problems than mining structures [58].
2. there has been little academic research work done on its performance in mining Bayesian network structures.

Hence, we decided to use the GA method.

#### Genetic Algorithms

A Genetic Algorithm (GA) is an evolutionary optimisation algorithm that solves problems by using the process of evolution [23]. Russel [58] describes a GA as a stochastic search algorithm in which a large population of states is maintained. New states are generated by mutation and crossover, which combine pairs of states from the population. A GA elicits a natural selection whereby offspring populate the next generation according to the fitness of the parents. The following in the next paragraphs are characteristics of a GA according to Russel [58] and Moore [45].

#### Common Characteristics of Genetic Algorithms

The characteristics of GA are as follows: it maintains a solution space or search tree (e.g. generated populations), it is computationally intensive because it takes a long time to run to completion during optimisation. Also, bitstring representation is critical in GA, if an application requires bits [45].

A bit is either a 1 or a 0. A bitstring is a sequence of 0's or 1's. An example of a bitstring is "10011". An application that requires bits are more prone to errors that can arise from bitstring manipulation.

Moreover, inner-loop optimisation is often possible [45]. This is to ensure decomposability of network components. In our research, we used inner-loop optimisation, which will be described in more detail later.

Also, GA can crossover contiguous chunks of the bitstrings instead of random bits, and states used may not necessarily be bitstrings [45]. It could use strings over other finite alphabets. An alphabet is defined as a basic symbol in an application, such as a bit 1, 0 or letters a, b, c etc.

A schematic GA as described and extracted from (Andrew Moore) [58] and [67] is as follows:

**Schematic Genetic Algorithm**

1. Identify initial population
2. Evaluate initial population
3. Generate a successive population
   (i) Parent selection
4. Generate offspring through the genetic operators
   (ii) Crossover
   (iii) Mutation
5. Evaluate the fitness and survival of individuals
6. Repeat steps 3 - 5 until the solution space is exhausted.

## MOTIVATIONS FOR RESEARCHING GENETIC ALGORITHMS

For the purposes of the current research, we are particularly interested in exploring GAs for the following reasons:

1. Its intelligent characteristics can save a lot of programming efforts. That is, the implementation of crossover, mutation and fitness function is clear but may probably require more programming work with other search algorithms.

2. It has wider academic applicability, and positive testimonies exist from early researchers in mining Bayesian network structures. For instance, Larranaja [39] showed that a GA is very good for mining suitable DAGs from complete datasets. Also, Wai [67] demonstrated and showed that a GA is superior for discovering extremely good networks from large datasets. Moreover, Moore [45] and Russel [58] described the GA as one of the best optimisation algorithms and as being highly versatile and useful for solving optimisation problems, specifically mining structures from data.

3. A GA will eventually reveal the true hidden relationships in a dataset but the process may take a long time.

Hence, we deduce from the above premises that a GA can reveal a true representation of underlying models in datasets more effectively than the greedy search methods. Prior to discussing the methodology used herein, it is necessary to include the following important fundamental definitions of GA terms, as acquired from the literature.

**Fundamental Definitions of Genetic Algorithm Terminologies**

1. A **Population** is a set of individuals. For instance in P = $\{\{x_1\}, \{x_2, x_5\}, \{x_6\}, ...\{x_n\}\}$, P denotes a population.

2. An **Individual** is represented as a string over a finite alphabet. For instance, $\{x_1\}, \{x_2, x_5\} \in$ P are individuals. In other words, $x_2$ and $x_5$ are alphabets or strings that are finite within the individual $\{x_2, x_5\}$. A form of string that is not an alphabet is a bitstring that was defined earlier.

3. The **Fitness Function** is used to evaluate the fitness of individuals in candidate selection. That is, the capability of the individuals to survive the generation. For instance, the Minimum Description Length (MDL) is an example of a fitness function. More description is provided in Chapter Four.

4. The **Objective Function** tries to find an optimal value for a fitness function. For instance, the objective function is to find the minimum network score, when mining Bayesian networks from data and when the MDL is used as the fitness function.

5. **Candidate Selection** is the choice of individuals in the population that are fit to achieve the objective of a problem definition. For instance, $x_1$ may be more probable to be the parent

of $x_2$ while $x_2$ may be less probable to be the parent of $x_1$. This is determined by the fitness function applied to BN structures.

6. A **Gene** is an entity or a subset in an individual. For instance, the orders or directions of $x_1,\vec{}x_2$ and $x_2,\vec{}x_1$ are separate genes of the individual $\{x_1, x_2\}$. For instance, in this research, this means that, the measurement of dependency of $x_2$ on $x_1$ is different from the measurement of dependency when the mutation gives $x_1$ depending on $x_2$.

7. **Reproduction** is the generation of offspring from parents. The common genetic operators such as *crossover* and *mutation* are used to carry out the reproduction process from one generation to another. The reproduction multiplies the population and therefore creates a solution space. An illustrative example will be shown in subsequent sections.

8. **Crossover** is generally considered as the most important genetic operation. This refers to the process of creating a new individual (offspring) through the combination of genetic materials of two parents. For instance, let $\{x_1\}$ *and* $\{x_2\}$ be parents in a network of population then, through crossover, they reproduce a new individual $\{x_1, x_2\}$. Also, see fig. 3.4 for a biological example.

9. **Mutation** introduces new genetic material into an existing individual. Introduction of new genetic material in structure mining refers to change of order or direction of the gene (variables) (see gene description above). For instance, we can mutate $x_1, x_2$ to become $x_2, x_1$. Also, see fig. 3.4 for biological example. We can see from the left hand side of the mutation picture that, a chemical process acts on the genetic material and generates new genetic material on the right side. However, observe that they maintain the same structure but different chemical constituents.

In the next sections, we investigate how existing GA researchers applied these GA concepts to mine Bayesian structures.

### 3.2.7 Further Related Research using Evolutionary Algorithms (EA) to Mine Bayesian Network Structures from Data

Genetic algorithms are popular specialisations of evolutionary algorithms. Wai [67] developed the Minimum Description Length Evolutionary Programming (MDLEP) to mine BN structures based

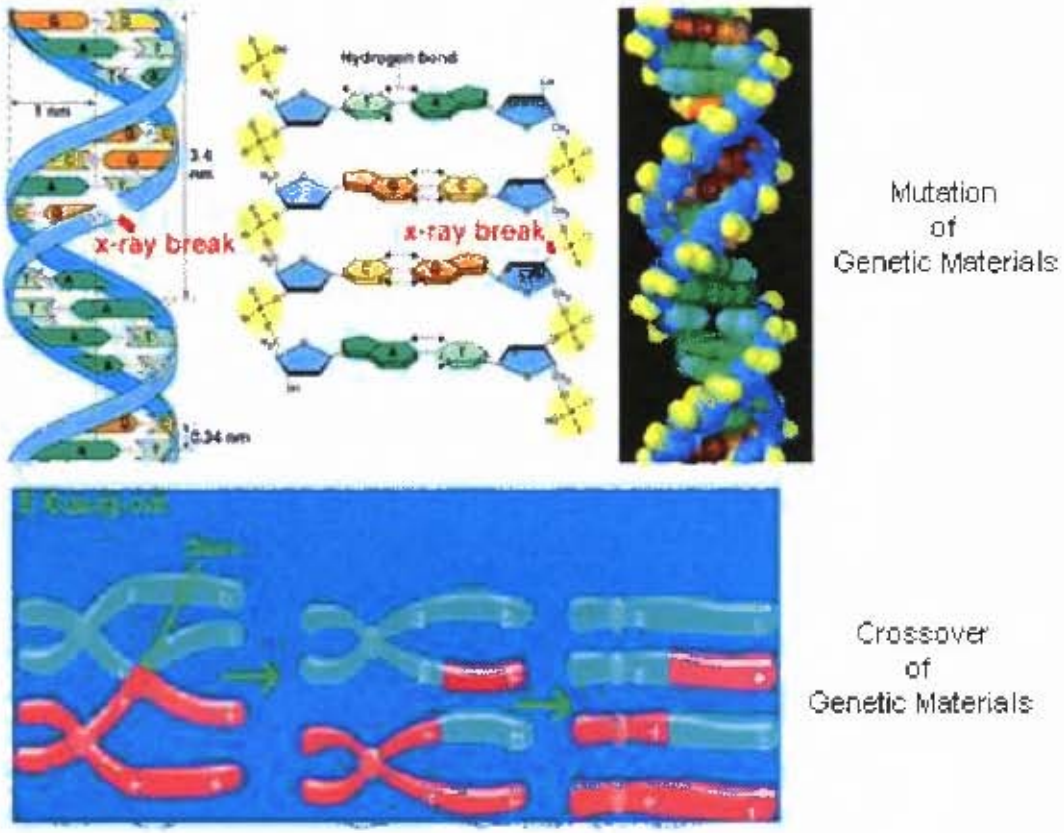


Figure 3.4: Mutation [47] and CrossOver [14] Processes

on the MDL principle and on evolutionary programming. In their approach, they integrated the MDL metric from information theory and developed variants of classical genetic operators which include: structure-guided mutation, knowledge-guided mutation, and freeze and defrost operators to mine Bayesian networks from data.

These operators reproduced new offspring from a parent network. Specifically, they defined structure-guided mutation as randomly adding edges, deleting edges, reversing edges and moving one node to another. That is, an edge is randomly added between two nodes if not already exist. If the edge already exists, it deletes it. Also, it randomly changes the direction of an edge and increases / decreases the number of parents of a child node to get a new structure.

Wai [67] defined knowledge-guided mutation as using MDL to determine which edges should be removed or inserted in a network. For every edge it adds or removes, it computes and keeps its MDL. That is, if it decides to add a new edge to the current structure, it first considers and compares with the already stored MDL score and chooses the lower MDL score. Also, the freeze

and defrost operators were used to identify and repair redundant nodes. That is, when there are several modifications to the network and there is no improvement, it prevents further changes on the network. This variant type of a GA is claimed to discover extremely good networks. Some of their operators can be combined to save time.

Larranaja [39] used a GA to generate variable orderings that are representations of chromosomes. The variables evolved several networks during BN learning from complete data. They used genetic operators, such as rank selection to select parents for reproduction, crossover and mutation to generate new individuals. They showed in their experiments that a GA can discover good networks from data. There was no clarity, though, on how their results proved better than what they addressed in the literature survey of their paper.

William [70] identified sensitivity to variable ordering as a shortcoming of most greedy score-based algorithms such as K2. Variable ordering is the combination of attributes to form nodes in Bayesian structure learning. When an algorithm is sensitive to ordering of network variables (A, B, C, D), it implies that the emerged optimal network of input pattern B, C, D, A will probably be different from input pattern D, A, C, B. They developed a genetic algorithm to mitigate this problem by searching the permutation space of variables using a fitness function. Specifically, William et al. used an order crossover (OX) to generate a population from six variable nodes. For instance, OX reproduces offsprings

$$O_1 = \{6\ 2\ \underline{5\ 3}\ 1\ 4\}$$

and

$$O_2 = \{5\ 3\ \underline{6\ 2}\ 4\ 1\}$$

from parents

$$P_1 = \{3\ 4\ \underline{6\ 2}\ 1\ 5\}$$

and

$$P_2 = \{4\ 1\ \underline{5\ 3}\ 2\ 6\}$$

Also, they implemented mutation as swapping of number positions as shown. Moreover, they refer to $P_1, P_2, O_1$ *and* $O_2$ as individuals. However, they claimed that they have minimised part of the K2 limitations and therefore, proposed that their results have the potential to only use the GA to mine networks in future research.

It is interesting to note that a similar approach is used by Myers [48] who also made representations of GA elements and operators such as gene, crossover etc. They developed a variant of a genetic algorithm as described below. They illustrated structure mining with five variable nodes, where each node is represented as a gene and a network structure is represented as chromosome. Myers used uniform crossover and mutation as add, delete and reverse operators that are commonly used in K2 algorithms. They illustrated with two parents simultaneously as follows:
Parents ($P_1$ *and* $P_2$) $\rightarrow$

$$P_1 = [\{A\}\{B\ A\}\ \underline{\{C\ A\}}\ \{D\ B\ C\}\underline{\{E\ C\}}],\quad P_2 = [\{A\ B\}\ \{B\}\underline{\{C\ B\}}\ \{D\ B\}\underline{\{E\ C\ D\}}]$$

offspring ($O_1$ *and* $O_2$) $\rightarrow$

$$O_1 = [\{A\}\ \{B\ A\}\ \{C\ B\}\ \{D\ B\ C\}\ \{E\ C\ D\}],\ O_2 = [\{A\ B\}\ \{B\}\ \{C\ A\}\ \{D\ B\}\ \{E\ C\}]$$

These operators are used to modify the chromosome structure, but they are prone to cyclic structures. The improvement on what they used as genetic operators paves the way for future research.

In the next section, we will describe the differences between the GA and the K2 algorithms.

### 3.2.8 Common Differences Between Hill-climbing and Genetic Algorithms

Similarly to hill-climbing, a genetic algorithm starts with an initial model, usually the worst model, that can iteratively be optimised to a goal model. Apart from this similarity, the following are the differences between the two approaches:

- The number of iterations for a GA can be pre-estimated from the population size while it is determined at convergence point in the K2 algorithm.
- The K2 does not generate a solution space because it does not have memory but grabs its neighbour. Unlike the K2, a GA has memory and generates populations using the genetic operators [58], [45].
- A GA is stochastic in candidate selection while K2 operators are greedy in their search approach [58].

- The K2 is terminated when there is notice of convergence but this convergent point may be stuck at a local minimum [45], [58]. Unlike the K2, a GA runs into completion by exhausting the whole network space.
- The K2 is fast with good models but a GA is computationally intensive with better models [70].
- When two sample orderings $O_1$ *and* $O_2$ of a training set are presented to the K2, it may output different models but a GA can take any ordering and will present the same model [70].

## 3.3 Conclusion

In this chapter, we observed that anomaly detection in various datasets is a problem of reasoning with uncertainty.

Probabilistic models (Bayesian Networks) provide the ideal technology to reason in the presence of uncertainty [53]. In this research, we use the Bayesian networks technology to detect the anomalies. This chapter has introduced the basic building blocks of probabilistic modelling, including: probabilities, Bayes' theorem and reasoning techniques. In this research, we used Bayesian networks to model individual subscriber behaviour as cause-effect relationships, which we used to detect anomalies. We discussed the different categories of Bayesian learning.

Our approach is to study the behaviour (call patterns) of every subscriber and to build an individual model in order to achieve effective anomaly detection. We have seen that mining Bayesian network structures from data is essential, especially for call data. We use a genetic algorithm (HGA) to mine BBNs for individual subscribers. The details of genetic algorithms and the motivations for choosing it for this research have been presented.

To this end, we have reviewed the theories and background needed in the chapters that follow.

# Chapter 4

# Methodology

Our first research question is, *How can call data be observed and transformed so that the underlying models of individual subscribers can be mined efficiently?* In order to address this research question, we propose the Data Transformation System (DTS) that transforms the data so that individual subscriber models can be mined effectively from the call data. One should recall that, anomalies in telecommunications call data are difficult to understand because it is necessary to discover the intentions of subscribers and to identify inconsistencies in calling behaviour. Such inconsistencies cannot be observed directly from subscribers' call data and we need machine learning algorithms that can discover and model the true calling activities of every subscriber. We therefore propose a new hybrid genetic algorithm (HGA) in order to reply to our second research question; *How can optimal Bayesian network models be mined from telecommunications call records*?

We also propose the Anomaly Detection System (ADS), to detect anomalies from observed call data. This replies our last research question; *How can Bayesian network models be used to detect anomalies in customers' behaviour?* This thesis does not only apply machine intelligence algorithms to telecommunications industries but also contributes to machine learning algorithms, specifically for mining Bayesian network structures from data.

The next section describes the methodology that we used in this research.

## 4.1 System Model

The system model illustrated in Figure 4.1 gives an overview of our research methodology.

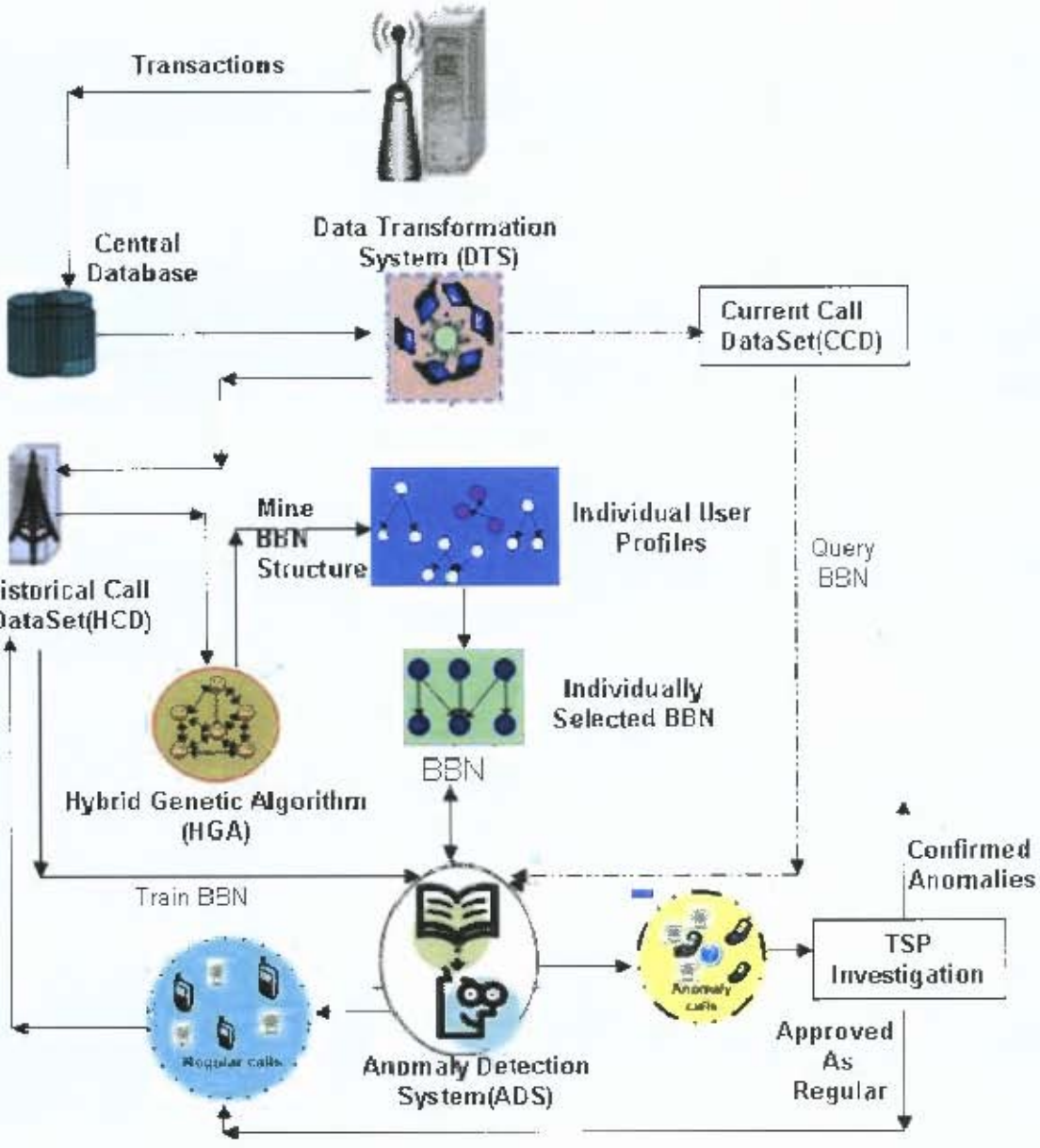


Figure 4.1: Research System Model

As shown in Figure 4.1, the three major system components are: the DTS, the HGA that mines the individual subscriber Bayesian networks (BBNs), and the ADS. The call records for all subscribers reside in a central database. The central database is a collection of call transactions from all base

stations of a network service provider. A base station is a service station for a TSP. Furthermore, the central database contains both the historical and the current call transactions for all subscribers. In our prototype, we process our call data grouped into daily transactions. Daily calls are stored in the Current Call Dataset (CCD). Historical call data refers to calls made since first subscribing to the network, and is stored in the Historical Call Dataset (HCD). In future research, we aim to handle transactions per call in real time and not group them into days.

The Data Transformation System (DTS) prepares the data for use by the rest of the system. It forms the pre-processing phase of this system. The HGA uses the data prepared by the DTS to mine an optimal Bayesian network (BBN) for each subscriber from the individual subscriber calls, contained in the HCD.

The ADS uses Bayesian learning, Bayesian inference and differential analysis, which will be discussed in detail in the subsequent sections. During Bayesian learning, the ADS uses the HCD to train an evolved BBN from the HGA to populate CPTs. The ADS acts on a trained BBN to classify call records in the CCD for a given subscriber. Call data identified as regular is added to the HCD to incrementally train the BBN model, while the anomalous call alerts the call analysts to investigate. The anomalous calls can be finally confirmed by the TSP as regular or anomalous. If it is confirmed as regular, it will be included into the HCD for training. If it is confirmed as anomalous, then it is totally excluded from the training set.

The addition of regular calls to the HCD is used to improve the accuracy of the BBN before the ADS acts upon it. The use of this incremental training is strongly encouraged to minimise error rates, commonly called false alarms. This incremental training causes the BBN models to evolve with the dynamic world of individual changing behaviours, which would not have been possible if the models were static.

### 4.1.1 Data Acquisition and Data Understanding

There are confidentiality issues that make it difficult to obtain telecommunication call data. For the purpose of studying telecommunication anomalies, we were able to acquire land line call data from a servicing firm to Telkom South Africa. This real life dataset is a good generalisation of networks that have mobile call data. It contains almost all the attributes expected in any telephone network. For instance, most of the attributes in Table 5.4 that we used in our land line data are also presented

in the mobile phone data used by Burge [8], Hollmen [28], Fawcett [19]. The identities of the subscribers were filtered out in order to protect the confidentiality of individual subscribers. One of the interesting advantages of this research is that we applied our proposed modelling technique (HGA) successfully to other applications [56] and it can therefore be applied to mobile networks with confidence.

In order to conduct our call data analysis, we explored the use of a data analysis tool called SQLGate for the MySql database. This tool is used to generate structured datasets such as CSV (Comma Separated Variable), XML (Extensible Markup Language), database or text files, as required. Thus, any of these formats can serve as input specifications to a mining structure algorithm depending on requirements.

We tried to ensure the universality of our architecture and also that our modelling design would be reproducible by using publicly available datasets to test our implementations. The common publicly available datasets are in the UCI repository [49], which most machine learning researchers use to learn and evaluate their networks. Among other datasets used for testing this research is the dataset containing the operations of a block-lifting machine [50].

During the data analysis phase, we identified the attributes of a Call Detail Record (CDR) in the call data. The call attributes are: {*originating-no, call-date, duration, destination-no, destination-net, location, call-cost, peak/off-peak, record-code* }. How datasets and their attributes are acquired and observed for modelling are described in the next section.

### 4.1.2 The Data Transformation Systems (DTS)

It is known from knowledge discovery and data mining literature that data transformations can help to better understand the data that enters into machine learning algorithms [4]. In this research, we packaged the required data transformation activities into a number of processing subsystems. The DTS is a decomposable system that observes and preprocesses call data for modelling and prediction. This system is general, not only limited to telecommunication data, and can therefore be used in other applications. One of the objectives of our DTS is to produce a suitable informative training dataset for the modelling architecture. This ensures a meaningful interpretation of patterns mined with the Bayesian networks.

The DTS is a suite of three simple, interactive and cascaded subsystems, which include the filtering,

discretization and user-collection subsystems. For more detail on the designs, consult the class diagram in Figure A.1 in Appendix A.

The **filtering subsystem** receives telecommunication call data, processes the data and sends the results to the discretization subsystem, more specifically:

- Fetches call data from the central database;
- Reports the number of call records and attributes;
- Reports the number of subscribers;
- Reports subscribers' call data if required;
- Ensures attribute names are self explanatory;
- Avoids initial possible overparameterisation by excluding unnecessary attributes;
- Creates new attributes by automatically generating node names as one-to-one mapping to the original attribute names. This assists in taking care of long names, manage memory, avoid clumsiness and ensures confidentiality to call attribute names.

The **discretization subsystem** uses the output of the filtering subsystem. Discretization applied to numerical attributes is well known to statisticians and other machine learning researchers. The *equal width* algorithm is an unsupervised discretization algorithm that is very suitable to anomaly detection [4], [9]. It divides data sets into intervals of equal width and the widths of the intervals depend on the patterns of the available data values, producing an informative telecommunication dataset in our case. The functionalities of the discretization subsystem are as follows:

- For simplicity, it discretises the number of attributes depending on user specifications. Our implementation accepts numerical attributes as nominals, except if the user specifies it to be discretized.
- It generates a default number of intervals but allows flexibility. That is, a user can specify the desired number of intervals as an alternative to the unsupervised generated default intervals.
- It automatically identifies date attributes and generates the corresponding days of the week.

- During discretization, the generated intervals maintain the data types, e.g. integer inputs will be discretized as integer intervals, etc.

We identified and implemented the equal-width algorithm [24] as follows:

**Pseudo-code for Equal-Width Discretization Algorithm**

**Inputs:** (i) X[i], N, C, (n)

(ii) Default n: max(M/3C, min(C+1, M))

*Comments:* n = desired number of intervals, M = Number of distinct elements in X[i], C = number of classes in the input set, N = Number of points, and X[i] = column elements to be discretized.

---

1. Find the minimum value Xmin and the maximum value Xmax of X[i], i = 1, . . ., N
2. IntervalWidth = (Xmax - Xmin)/n
3. For r = 0 to n

   e[r] = Xmin + IntervalWidth * r

**Outputs e[0:n]:** Vectors of discretized boundaries. That is, discretized intervals such as: $[e_0, e_1], [e_1, e_2], ..., [e_{n-1}, e_n]$

---

The **user-collection subsystem** receives the output of the discretization subsystem and collects data for individual users using the originating call numbers. Every subscriber will have a subset of the HCD and a subset of the CCD data. The HCD subset serves as the training data for the HGA to mine an individual Bayesian model (BBN), while the CCD subset is used as the query data by the ADS. The user-collection subsystem performs the following activities:

- Among all the call attributes in the central database, HCD and CCD subsets are collected using subscribers' originating call numbers as key.

- Collection is performed in batch or on only one originating call number. That is, we may want to generate CCD and HCD subsets of only one subscriber at a time or generate CCD and HCD subsets for all subscribers.
- Originating call numbers are automatically deleted from the training set because it does not contribute information to relationship discovery during modelling, as it is the same for all call records for the same subscriber.
- In general, for other datasets, apart from call data, if a user does not specify the inclusion of originating call numbers, then all attributes are included in the training set.
- It introduces incremental learning by loading summarised statistics of distinct values found on all attributes of a user collection. This helps to speed up modelling.

On completion of all activities of the DTS, the final informative CCD and HCD subsets are generated for use by the rest of the system.

## 4.2 Bayesian Modelling and the Hybrid Genetic Algorithm (HGA)

To facilitate the modelling of *multiply-connected* Bayesian network structures, we propose new technique called a Hybrid Genetic Algorithm (HGA). This does not only adopt the classical GA operators but also integrates information theoretic measures as learning components and mathematical concepts as population constructions. Specifically, the information theoretic measures that we used include: Mutual Information (MI), Shannon's information content, and Minimum Description Length (MDL) (see next sections). Mutual information is a model that finds the probability that two variables share information. The mathematical component part of the HGA is the PowerSet lattice, which implements the crossover operator. Moreover, the PowerSet lattice generates a population space from which the best networks can emerge.

The classical GA is a framework or generalisation of HGA. The initiative used to research and reconstruct some of these models (e.g. EDA, PowerSet, etc) to achieve the fundamental functionalities or elements of GA is the power of HGA. The classical GA such as the pseudocode shown in the next section is a framework [18] whose operators such as crossover, mutation, and optimization processes are required to be redefined. As Russel [58] supports the variants of algorithms, any

algorithm that is a framework cannot be implemented directly but it must be reconstructed to be operational. The development of HGA is a new idea as part of our methodology to modify and experiment models (e.g. EDA, MDL, PowerSet, etc) from various fields of information theory and mathematical set theory to redefine the operators of classical GA.

Moreover, related algorithms were developed from the frameworks of classical GA and even hill-climbing algorithms. As described earlier, Wai [67] developed the Minimum Description Length Evolutionary Programming (MDLEP) algorithm using the MDL principle and evolutionary programming as a variant of classical GA. Similarly, Ioannis [30] developed a Max-Min hill-climbing (MMHC) algorithm that uses some mathematical operators as a variant of classical hill-climbing algorithm.

One of the interesting features of this methodology is that, the components of the HGA are implemented as decomposable systems. This decomposability means that the functionalities of every component such as the MI, perceives their inputs and produce their outputs differently from other HGA components. Furthermore, decomposability makes this algorithm ideally suited to be implemented using distributed agents. In the next section, we describe the HGA in more detail.

### 4.2.1 Pseudocode for Genetic Algorithm

In order to give better direction to our implementation, and in order to incorporate the schematic genetic algorithm in the literature review, we extracted the following pseudocode of a GA from [18] and [58]:

---

Pseudo-code for Genetic Algorithms

---

1. Let $g = 0$ be the generation counter
2. Initialise a population $C_g$ of N individuals; $C_g = \{\vec{C}_{g,n} \mid n = 1, ..., N\}$
3. While no convergence

- Evaluate the fitness of each individual $\vec{C}_{g,n} \in C_g$
- Perform cross-over

1. Select two individuals $\vec{C}_{g,1}$ and $\vec{C}_{g,2}$

2. Produce offsprings from $\vec{C}_{g,1}$ and $\vec{C}_{g,2}$

- Perform mutation

1. Select one individual $\vec{C}_{g,n}$

2. mutate $\vec{C}_{g,n}$

- Select the new generation $C_{g+1}$

*Crossover* is generally considered as the most important genetic operation, which is the process of creating a new individual by combining the genetic materials of two parents. Similarly to crossover, *mutation* introduces new genetic material into an existing individual.
The development of these genetic operators depend on their applications.

The definitions of GA terms in our literature review give better understanding to the GA pseudo-code above. An iterative scheme, when applied to mining Bayesian network structure from data, *converges* when the candidate network with the best fitness emerges.

Looking at steps one and two in the pseudo-code above, the initial population of N individuals corresponds to the attributes of a training set. That is, the generation $C_g$ where g = 0, represents the parents in the form of vectors. From step three, we evaluate the fitness of each individual, and score each variable $\vec{C}_{g,n}$ in the generation $C_g$. In the crossover step, two individual variables $\vec{C}_{g,1}$ and $\vec{C}_{g,2}$ are selected from the parent set and combined to produce a new subset of variables. This new subset is referred to as the *offspring*. Also, selecting an individual $\vec{C}_{g,n}$ for mutation means changing the genetic material of an offspring. This implies changing the order or direction of elements in subsets. However, this is not noticeable or relevant to us in so-called singleton sets. In the set theory of mathematics, a singleton set is an individual with only one variable.

In view of this, selection of individuals for reproduction and fitness of the network model is stochastic (probabilistic). In the following sections, we describe how we applied genetic algorithms in our research.

### 4.2.2 Mining Bayesian Network Structure using the HGA

By maintaining the usual meanings of the GA terms and considering the pseudocode above, the HGA methodology that we used to mine Bayesian network structures is described as follows:

The HGA mining begins with an initial population P, say $\{\{x_1\},\{x_2\},\{x_3\},\{x_4\}\}$ called parents, and of which the independent network: $(x1),(x2),(x3),(x4)$ is formed. Every parent in the initial network is evaluated with the MDL model scoring measure (see subsequent sections; *Approach: Minimum Description Length (MDL)*). Every parent in P such as $\{x_1\}$ produces offspring during several crossover processes with other parents such as $\{x_2\},\{x_3\},\{x_4\}$. The mathematical Poset lattice (see subsequent section; *Mathematical PowerSet Lattice*) implements this crossover process and is advantageous to our network space because it guards against cyclicity, which we do not allow in this research.

Furthermore, we implemented a network checker that confirms the absence of the cycles. There is an equiprobable selection of every other parent for producing a child subset. This means that the probability of producing offspring from a pair of parents is 0.25 each for the above initial population. To avoid backtracking, to avoid worse scoring structure or to avoid time wastage in evaluating an entire candidate network structure using MDL, we introduced an inner loop by using MI (see subsequent sections; *Approach: Mutual Information (MI)*) to check if an offspring produced is probabilistically fit as a candidate sub-structure in the entire Directed Acyclic Graph (DAG).

For every new offspring produced that survived, such as $\{x_1,x_2\}$, the prior probability that $\{x_1\}$ or $\{x_2\}$ is a parent of each other, is 0.5 respectively. Since they are equiprobable, it becomes difficult to decide which should be the parent. So, a new offspring $\{x_2,x_1\}$ is produced using the EDA (see subsequent sections; *Approach: Extended Dependency Analysis (EDA)*) by mutating $\{x_1,x_2\}$. There is also equal prior probability 0.5 of $\{x_2\}$ or $\{x_1\}$ being a parent of each other. It is also difficult to decide between the two. Therefore, these two different individuals will respectively formulate candidate sub-networks.

In order to decide which attribute is the parent, the two candidate sub-networks compete with each other and the winning candidate sub-network will emerge. This means that a more fitting candidate sub-network will serve as a building block to the entire DAG. The selection of a more fitting candidate sub-network is guided by Shannon's information content (see subsequent sections; *Approach: Extended Dependency Analysis (EDA)*) in which the EDA decides and chooses a candi-

date sub-network that will minimise the score of the DAG. This optimization process is repeated until the whole solution space is exploited and the best Bayesian network model emerges.

The new generations of individuals can be produced from the current generation of offspring such as $\{\{x_1, x_2\}, \{x_1, x_3\}, \{x_1, x_4\}, ..., \{x_3, x_4\}\}$. In contrast to this, one of the distinguishing features of our methodology is that we control the new generation of individuals to avoid repetition of already existing individuals. The larger subsets of individuals will be broken down into smaller ones at the mining stage. Since the possible smaller subsets of individuals are produced in the older generations, we observed that less time may be wasted if the reproduction of newer generations is controlled.

Given the above, our structure mining methodology is a variant of the classical genetic algorithm covered in existing research on genetic algorithms.

### 4.2.3 The Proposed Architecture for the Hybrid Genetic Algorithm

The HGA uses the output of the DTS, and consists of five subsystems namely, the PowerSetLattice, the HGA, the fitness function (scoring), the EDA, and the MI. The training set can be presented in XML, as a database file, a CSV or text file formats as required. In our implementation, a text file serves as input to the PowerSetLattice and the HGA subsystems. Furthermore, the HGA uses the scoring subsystem, which can be implemented by using a Bayesian method, MDL or Maximum A Posteriori (MAP). MAP [58] is a statistical learning method that can be used to estimate an unknown variable from a scientific data. MAP is similar to maximum likelihood (ML) but it includes a prior knowledge on the unknown variable to be estimated. We used the MDL in our implementation.

As shown in the implementation design in Figure A.2 in Appendix A, the **PowerSetLattice subsystem** implements the crossover process using data attributes. The functionalities are stated below:

- It extracts attribute names as input to the initial population.
- It prepares the parents as a set and generates ordered sub-sets of offspring on equal probabilities.
- It releases the sub-sets of offspring on request to the HGA and keeps maintaining the population as it grows.

The **Hybrid-Genetic-Algorithm subsystem** is the coordination subsystem that controls the other subsystems including, the PowerSetLattice, scoring subsystem, the EDA subsystem, and evolves the best Bayesian network models for individual subscribers. Its functionalities are highlighted as follows:

- The HGA starts by requesting the parents set from the PowerSetlattice subsystem and prepares the attributes as independent nodes of an initial network.
- It gets the corresponding data for the attributes received from the PowerSetLattice subsystem.
- It sends both the initial network and its data to the scoring sub-system that uses the MDL for evaluation and remembers the score.
- For the purpose of optimization, the HGA randomly requests a offspring sub-set from the PowerSetlattice and sends it to EDA to calculate its fitness.
- If the offspring is fit, the HGA sends the initial network and the offspring sub-set to the MDL for evaluating the new network. The new network score result optimizes the initial network score.
- For the purpose of monitoring possible overparameterization, which is generally a problem of large networks, the HGA carries out a pruning process. The pruning keeps the number of parents of a network node at k-value [11].
- The HGA ensures that the solution space, generated by crossover and mutation processes, is exploited by searching the entire network space in order to emerge the best structure.

The **EDA subsystem** is used to check the fitness of offspring sub-sets received from the HGA subsystem. That is, it performs a dependency analysis of sub-set attributes. The functionalities are as follows:

- It receives an offspring sub-set from the HGA and uses it to find the probability associated with its fitness as a candidate sub-network.
- It uses the MI as an inner loop to first find the probability that the two attributes of the sub-set share significant information.

- If the probability of information sharing is significant, the EDA forms a candidate sub-network from the two attributes of the sub-set, mutates the sub-set and uses Shannon's information content to evaluate the two candidate sub-networks.
- From the fitness competition, the wining candidate sub-network emerges and it is sent back to the HGA.

The **MI subsystem** is the innermost loop that is used by our HGA to find the probability that two attributes of a offspring sub-set share information. We observed from our implementation that the probabilities of two attributes sharing information are commutative. That is, they are equal. The MI performs the following functions:

- It receives the offspring sub-set from the EDA.
- It computes the probability of information sharing between the attributes of the sub-set and gives feedback to the EDA subsystem.

### 4.2.4 The Mathematical PowerSet Lattice used by HGA

In mathematical set theory, if S is a finite set, then the PowerSet of S represented as Pw(S), is the set of all subsets of S. It is a formalised concept of ordering or arranging elements of a set. The magnitude of a set S, is the number of elements in the set. Suppose the cardinality or magnitude of $\|S\|$ = n elements then, the PowerSet of S implies

$$Pw(S) = 2^n \ subsets$$

For instance, if set $S = \{x1, x2, x3\}$, then the PowerSet of S implies

$$Pw(S) = \{\{\}, \{x1\}, \{x2\}, \{x3\}, \{x1, x2\}, \{x1, x3\}, \{x2, x3\}, \{x1, x2, x3\}\}$$

This is an algebraic concept that we use to generate populations of the HGA. The individuals of the initial population $S = \{x1, x2, x3\}$, are referred to as parents which reproduce offspring (the subsets of Pw(S)). This population of parents are modified via crossover and mutation processes, which elicit sexual reproduction and natural adaptation.

### 4.2.5 HGA Approach: Mutual Information (MI)

In probability and information theory, Mutual Information (MI) [23] [32] is an intelligent model that measures information of a random variable X found in a random variable Y. That is, a measure of the amount of information sharing between two nodes of a network. Its unit of measurement is a bit and the model is given below:

$$I(X,Y) = \sum_{x,y} Pr(x,y) \times \log_2 \frac{Pr(x,y)}{Pr(x)Pr(y)} \tag{4.1}$$

where, $Pr(x,y)$ = joint probability distribution of x and y while $Pr(x)$ and $Pr(y)$ are the marginal probabilities of x and y respectively. We observe that MI has been applied in many areas of machine learning. Some of its important properties are stated below.

**Properties of Mutual Information**

- MI performs better on probabilistic related variables.

- $I(X,Y) = I(Y,X)$. This implies that, MI is commutative or symmetric.

- $I(X,Y) \geq 0$

Suppose X and Y are independent, then X contains no information about Y and neither does Y contain information about X. Therefore, their mutual information will be zero. This is obvious in the relation below:

$$\log \frac{Pr(x,y)}{Pr(x)Pr(y)}$$

$$= \log 1 = 0$$

Thus, MI is a measure of independence such that

$$I(X,Y) = 0, \;\; iff \; X \; and \; Y$$

are independent random variables. This implies that,

$$I(X,Y) > 0, \;\; iff \; X \; and \; Y$$

are dependent (i. e. shares information).

### 4.2.6 HGA Approach: Extended Dependency Analysis (EDA)

The original Extended Dependency Analysis (EDA) is a reconstructability analysis technique that evaluates the level of significance that exists between variables of a training set [61]. The level of significance between two given attributes is determined by their information sharing and the link threshold. More information on link threshold is described in the performance evaluation of HGA in Chapter Five. In addition to the original EDA, our methodology includes the mutation process of the genetic algorithm to reproduce a new candidate sub-network; it furthermore finds the dependency between the two variables using the theorem of Shannon Information Content [41], and produces the wining candidate sub-network.

It receives a offspring sub-set from the HGA and produces a candidate sub-network. It passes candidates of not more than two variables at a time to the MI.

Shannon defines information content as a natural measure of an outcome $x$ as:

$$h(x) = \log_2 \frac{1}{Pr(x)} \tag{4.2}$$

So, for all possible outcomes of $x$, the information content is defined as:

$$h(x) = \sum_x \log_2 \frac{1}{Pr(x)} \tag{4.3}$$

It is measured in bits. The smallest number of bits possible to store the information of $\mathbf{x}$ effectively is used. For example, the information content represented using 67.51 bits is preferred to using

98.31 bits to represent the same information.

### 4.2.7 HGA Approach: Minimum Description Length (MDL)

One of the important components that ensures that a GA is robust is its fitness function, which in our research measures the optimality of a candidate network (Bayesian network). The Minimum Description Length (MDL) is suitable for measuring the fitness of Bayesian networks as models of datasets [50]. We use an *objective function* based on the MDL to minimise the number of bits required to store a dataset and its corresponding Bayesian Network. Our **HGA** iteratively invokes the **MDL** until the minimum score is obtained. In principle, the MDL is made up of two two components as follows:

$$L(D,\ B)\ =\ \sum_{i=1}^{m} log_2 \frac{1}{Pr(v_i)} + \frac{|B|\ log_2\ m}{2} \tag{4.4}$$

where D = training dataset,

L = length of bits required to store both the dataset and its corresponding network.

$|B|$ = number of parameters in B, defined as:

$$B = \sum_{i=1}^{N} (j_i - 1)k_i \tag{4.5}$$

where $j_i$ and $k_i$ are the cardinalities of the child node and its parents set respectively. Also N = number of all nodes in a network, $m$ = number of samples and $\frac{log_2\ m}{2}$ = number of bits that are appropriate to represent each numeric parameter. Recall that, a *candidate network* is the network presented to MDL for scoring. Also, $Pr(v_i)$ = probability of a sample of D. This is computed for all values of $\{i, ..., m\}$. That is;

$$Pr(D) = Pr(v_i, .....v_m) \tag{4.6}$$

Note that, $P(v_i)$ is a probability distribution computed using a candidate network.

Therefore, the first component (information content) generative

$$\sum_{i=1}^{m} log_2 \frac{1}{Pr(v_i)} \tag{4.7}$$

measures the required length, in bits, to store the *dataset* only and the second component

$$\frac{|B|\ log_2\ m}{2} \tag{4.8}$$

measures the required length, in bits, to store the *network model*. Thus, equations 4.7 and 4.8 are added to give the MDL scoring measure for the optimal network.

### 4.2.8 Potential Benefits of the Modelling Architecture

The following in the next paragraphs are the potential benefits of our modelling architecture.

Most of the previous researches use one operator for crossover and another to guide against cycles respectively [67]. In order to save space and time, a mathematical Poset lattice is used here to implement the crossover process and partly guides against cycles.

Another benefit is that our system is not restricted by datasets types. Unlike most existing systems that can only mine networks from mixed or nominal datasets, our system can mine networks from datasets that contain numerical values that are not discretized into intervals.

One of the distinguishing features of this methodology is that, the new generation of individuals is controlled to avoid repetition of already existing individuals. This saves time during the mining phase.

Another important benefit is that our HGA implementation is not terminated by convergence, but terminates when the solution space is exhausted. During convergence, other methods may become stuck at a local minimum [58].

The other benefits are: the use of MI as an inner loop helps to ensure that the probability of an optimized new candidate network score does not get worse. This saves time. Also, our system is a general application, and has been successfully applied in other applications [56]. In addition, our HGA can mine multiply-connected networks.

In the next sections, we will discuss how we use our system to detect anomalies.

## 4.3 Behavioural Bayesian Networks (BBNs) and the Anomaly Detection System (ADS)

Up to this point, we have shown and described the methodology that we use to model individual subscribers' BBNs. The ADS uses the individual BBNs to understand current call data from historic call data, modelled by the BBNs, and to identify regular and anomalous calls in the current call data. This section presents the investigation results with respect to the last research question of this thesis.

Consider the following scenarios that we studied as a basis for the ADS:

- It becomes clear in BBN models that the phone calls of a subscriber S1 are clearly different from subscriber S2. For instance, S1 can make international calls regularly, but S2 can make them once in a while.
- If S1 makes several international calls from January to March, stops doing so in April-June and later makes this type of call once after June, does this mean that it could be a fraudulent activity?
- If S2 makes local calls to location A every first week of the month and makes a similar local call to location A or a new location B at the end of a particular month after many years, is this a fraudulent activity?
- Consider the synthetic historical call data in Table 4.1. Will a high or small number of call instances, such as 54 in row 1 and 1 instance in row 2 respectively, influence the prediction of a similar call instance to be either anomalous or regular?
- Also, will a new call instance that has not been observed in the history of calls, as in Table 4.1, be predicted as anomalous or regular?

It is obvious from the above scenarios that a constant threshold for all subscribers will not be able to detect anomalies. Therefore, these diverse scenarios and other unforeseen dynamic situations necessitate the use of differential analysis. We implement the ADS model using the following

Table 4.1: An illustrative synthetic historical call data for originating number: 27216861776

| Destination number | Location | Duration | Peak / Off-peak | Number of instances |
|---|---|---|---|---|
| 27216850231 | $> 50km$ | 120 | p | 54 |
| 27216850231 | $> 50km$ | 120 | x | 1 |
| 27216850231 | $< 50km$ | 120 | p | 7 |
| 27216850231 | $< 50km$ | 120 | x | 27 |
| 27767343614 | $> 50km$ | 120 | p | 3 |
| 27767343614 | $< 50km$ | 120 | x | 2 |
| 27767343614 | $< 50km$ | 60 | p | 4 |
| 27767343614 | $< 50km$ | 60 | x | 2 |

components: [i] *Bayesian learning and Sampling from Database*, [ii] *Bayesian Inference*, [iii] *Call Detection* and [iv] *Degree of Detection.* These components are described in the subsequent sections. First, though, Chapter Three describes the computation of BBN model parameters as a derivation from the Maximum Likelihood Estimate (MLE). The probability values of Bayesian learning parameters must be estimated prior to inferences. The genetic algorithm mines Bayesian network structures from data. These BBNs are then used by the ADS during Bayesian learning.

### 4.3.1 Example of Bayesian Inference Describing Prediction as Related to Our Methodology

[31] and [58] have identified exact Bayesian inference to be tractable to small networks. That is, exact inference is NP-hard due to its exponential complexities of space and time on multiply connected networks. Fortunately, approximate Bayesian inference can be identified as a replacement to query multiply connected networks, which is used in this thesis. The BucketTree algorithm is an example of an approximate inference algorithm implemented in JavaBayes [13]. We embedded the BucketTree inference engine of JavaBayes in our implementations. JavaBayes [13] is open source software for probabilistic reasoning. Examples of probabilistic reasoning using this inference engine will be illustrated by using the popular Dr Watson's grass example of a Bayesian network in Figure 4.2.

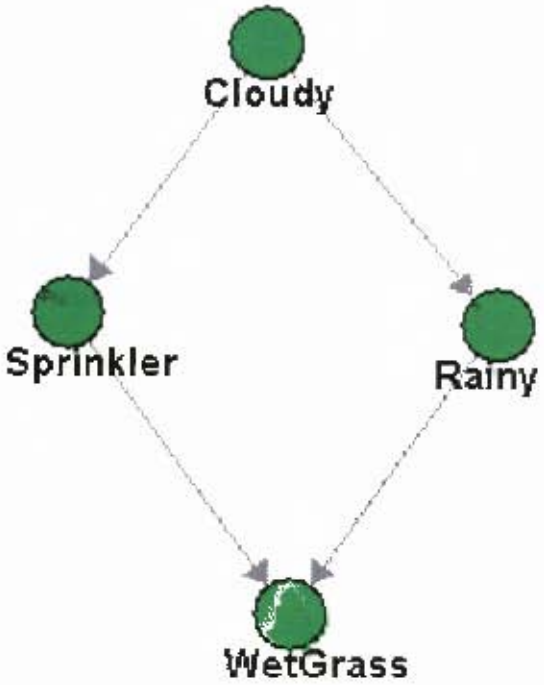


Figure 4.2: Dr Watson's grass Bayesian Network Model

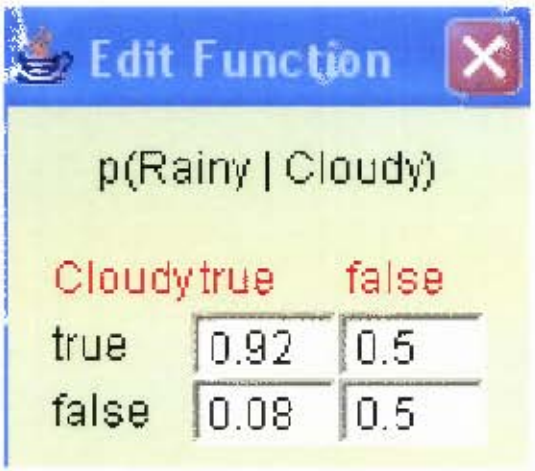


Figure 4.3: Sampled Conditional Probability Table (CPT) for Rainy node

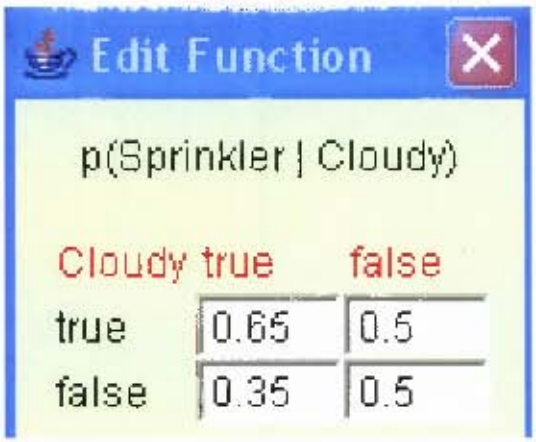


Figure 4.4: Sampled Conditional Probability Table (CPT) for Sprinkler node

Suppose the CPTs of Watson's nodes are as in Figures 4.3 to 4.6, and have been estimated using Bayesian learning. Figure 4.3 shows the CPT of node Rainy, given Cloudy node. The values true and false on the rows are from Rainy node while the values true and false on the columns are from Cloudy node. Similar explanation applies to Figures 4.4 to 4.6. With the advantage of information propagation in Bayesian networks, we made the following queries in equations 4.9 and 4.10, and


Figure 4.5: Sampled Conditional Probability Table (CPT) for WetGrass node

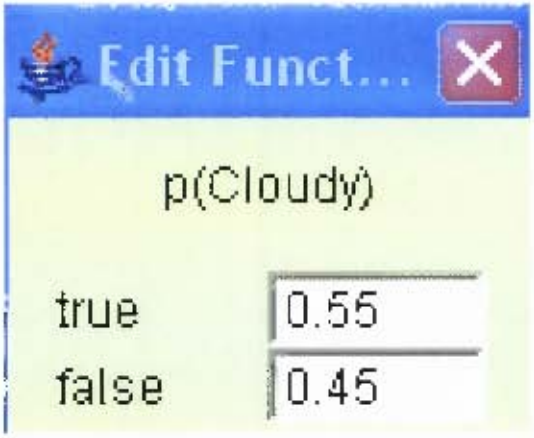


Figure 4.6: Sampled Conditional Probability Table (CPT) for Cloudy node

obtained the Bayesian inference results using the inference engine of JavaBayes.

$$Pr(Rainy = true \mid Cloudy = false,\ Sprinkler = true,\ WetGrass = true) = 0.5246913580246914 \tag{4.9}$$

The Bayesian inference result in equation 4.9 shows we are over 52 % sure that, it rains given that the sprinkler is on, the grass is wet, and it is not cloudy.

$$Pr(Cloudy = true \mid Rainy = true,\ Sprinkler = true,\ WetGrass = true) = 0.7451291345718 17 \tag{4.10}$$

Also, the inference result in equation 4.10 shows we are over 74 % sure that, it is cloudy given that it rains, the sprinkler is on, and the grass is wet. We used inference as one of the components that describes the mechanisms of Anomaly Detection System in the next section.

### 4.3.2 Mechanisms of Anomaly Detection System

This subsection describes the derivation of our mechanisms for Anomaly Detection System (ADS).

Qualitative detection of anomalies in call data requires a historical knowledge of subscribers' call behaviours. Recall from the literature review of this thesis that many of the related research in anomaly detection have used methods that express historical knowledge as rules and segmented related users into groups. These methods use the rules for classifying call records based on pre-classified anomalies but did not take unexpected call instances into account. Also, most of these methods segmented users into groups based on approximate related behaviours, but did not take into account that a user's regular behaviour may be another user's anomaly. With this, *true* calls may be misclassified as *false* calls and vice versa. Consider these scenarios to emphasize the points:

- Suppose call attributes $V_1$ implies $V_2$ and $V_2$ implies $V_3$ then, it can be inferred that $V_1$ implies $V_3$ but this can be an anomaly using just background rules to decide classifications.
- As a background rule for a subscriber's call, if a call instance $C_1$ shares information with another call instance $C_2$, then $C_2$ will not be regarded as sharing information with $C_1$. Whereas with inference, it makes more sense to say that $C_1$ shares information with $C_2$ implies $C_2$ shares information with $C_1$. [31] has more examples.

It would have been more appropriate if previous methods had used inference to predict anomalies by understanding historical knowledge first. Hence, we have integrated Bayesian inference as a component into our ADS. However, Bayesian inference may not be sufficient in itself to predict calls as being either regular or anomalous. This is because a call instance whose historical record is very low may be inferred as anomalous, even if this is obviously not right. For instance, using Table 4.1 as historical knowledge, we can infer the probability that the subscriber makes a phone call to a given destination number given certain evidences as:

Also, in the implementation of our ADS, the BBN is trained by computing its CPT using the training set (see previous sections for the computations of CPTs in Bayesian learning).

**Call Prediction**: Call prediction represented as $P_E^{BN}(\vec{e})$ is the probability of $E = \vec{e}$ inferred from the Bayesian network. This is computed using the Bayesian inference engine of JavaBayes embedded in our ADS implementation. Recall that for every subscriber, the prediction of calls in CCD is a daily transaction.

**Call Detection**: Call detection is the decision making aspect of our ADS using the computed sampled and inferred probabilities. As a scientific test / decision, this step suspects that a phone call record is either *regular* or *anomalous*. From our empirical test and the results of our investigation, we therefore define call detection as follows:

$$Pr_E^{BN}(\vec{e}) >= Pr_E^{\Xi}(\vec{e}) \implies \textit{Regular Call Suspected}$$

That is, if the inferred probability is greater or equal to the sampled probability, then regular calls may be suspected. If the inferred probability is less than the sampled probability, then, we can suspect the phone call to be anomalous. That is:

$$Pr_E^{BN}(\vec{e}) < Pr_E^{\Xi}(\vec{e}) \implies \textit{Anomaly Call Suspected}$$

Automatic detection of anomalous calls can reduce workload of telecommunications service providers. However, this ADS is very helpful to them when a subscriber does complain of having received and paid for call bills that were not incurred. The service providers can easily confirm by using the ADS if such calls in the bill were truly anomalous or regular to ensure customer satisfaction. *Regular* implies that it follows the historical call patterns behaviour of the subscriber, while *anomalous* means it does not. To this end, how do we compute the level of confidence when making decisions about such calls?

**Degrees of Detection**: Degrees of detection refers to the level of beliefs with regard to the call detection decisions. This result is expressed as a percentage and it will help call analysts and managers to make final decisions. A deviation *below* the sample probability is computed as follows:

$$(\frac{|Pr_E^{\Xi}(\vec{e}) - Pr_E^{BN}(\vec{e})|}{Pr_E^{\Xi}(\vec{e})} \times 100)\% = \mathbf{X.xx}\% \tag{4.14}$$

A deviation *above* the sample probability is computed as:

$$(\frac{|Pr_E^{BN}(\vec{e}) - Pr_E^{\Xi}(\vec{e})|}{Pr_E^{BN}(\vec{e})} \times 100)\% = \mathbf{X.xx}\% \tag{4.15}$$

This is analogous to the computation of percentage errors in mathematics. Equations 4.14 and 4.15 are interesting because they support the concept of adaptive intelligence with their changing inputs. That is, the outputs use differential analysis due to their regular update in input values. We want to say that, we are **X.xx** % sure that a call is *anomalous* or *regular*. Thus, the implementation of the ADS results is shown in Chapter Five of this thesis. For more detail on the ADS design, consult the class diagram in Figure A.3 in Appendix A. The next section describes how indicators can be identified for use in ADS.

## 4.4 Identification of Anomaly Indicators

This section describes our methodology on how we identified anomaly indicators (the most interesting attributes) in our mined networks. Burge [8] defined the meanings and classified anomaly indicators as primary and secondary, but he could not give specific examples. As one of our contributions, we improved on these definitions and applied it to Bayesian networks.

As discussed in Chapter One and Two, recent studies have investigated the challenges of anomaly detection but have not given conclusive solutions to address this problem. In this research, we infer that if appropriate anomaly indicators for individual subscribers are used for detection, the true positive rates of anomaly detection approaches can be maximized, while the false alarms can be minimized. One of our primary objectives was to identify significant or interesting attributes to make anomaly detection techniques effective.

Having analyzed the Bayesian modelling of subscriber behaviour and having used the theories of causalities [53], we derived a qualitative measure for computing the most significant nodes (indica-

tors) in the mined networks. Similarly to Jaroszewick [31] work, we defined the most significant of all nodes in a mined network as in equation 4.16.

$$Max(V_k) = \sum_{j=1}^{n} c_j + [\varepsilon], \forall V_k \in G \tag{4.16}$$

where c is the number of causalities for every node in the network G. Also, the $[\varepsilon]$ is optional, when two node measures are equal, and it depicts the number of independencies that may be caused by the removal of the node. Some of the related efforts to identify interesting nodes can be found in [31], but they mostly rely on prior knowledge of domain experts. However, the knowledge of the telecommunications domain expert may fail due to their inability to update knowledge about the current trends in the telecommunication calls and also due to large datasets.

With reference to equation 4.16, the most significant node is referred to as the most interesting attribute, and it is **x3** in Figures 5.1 to 5.17. The most interesting attribute can also be referred to as the primary anomaly indicator because it has the highest number of causalities. For each subscriber, this node carries a lot of information and are very helpful for detecting anomalies. Observe that x3 has the highest number of causalities such as, 4 in Figure 5.1, 5 in Figure 5.3, 4 in Figure 5.5, etc, on the networks. This is a particularly interesting attribute that is common to the majority of the subscribers. This shows that, x3 shares information with most of the other nodes on the networks. It will thus be important to consider *Destination-no* (x3) as a primary anomaly indicator in conjunction with the longest call duration, which is normally used in existing anomaly detection research.

Furthermore, in a situation where there is need for more than one anomaly indicator, a *secondary indicator* can also be identified from the network. The secondary indicator is the node with the next highest number of causalities to the primary indicator. For instance, in Figure 5.1, the primary indicator is **x3** with 4 causalities as stated above and the secondary indicator is **x6** with 3 causalities. Therefore, **x6** which is *Call-cost* may also be needed during anomaly detection.

Thus, the results of our modelling technique efficiently identified the indicators that can enhance anomaly detection methods. It also provides qualitative knowledge that will be useful to call analysts, managers and anomaly researchers. We used these indicator results in the implementations

of our system as presented in the following section.

## 4.5 Techniques of Evaluating BBN models

A number of techniques are required to evaluate our methodologies, such as the HGA and the ADS. The new HGA designed herein, which mines BBNs from data, can be evaluated using *structural differences* as used by most of the researchers in our literature [67]. The only way to justify the universality of our architecture and to assure that our modelling design is reproducible is to use publicly available datasets in our implementations. We evaluated the performance of the ADS using a *cross validation* technique, which is visualised by a *confusion matrix* as described by [37].

### 4.5.1 Structural Evaluation Techniques

We have further incorporated structural evaluation technique, which is defined in equation 4.17.

$$\sum_{i=1}^{n} \phi_i \tag{4.17}$$

$\sum \phi_i$ is the sum of the symmetric differences between the mined network and the known network. In contrast, it is the sum of the similarities between the two networks. The evaluation takes it beyond the sum of the symmetric differences in the learned network and the known network. The additional possible measures are: [1] MDL score of the optimal network, [2] Duration of mining the network structure.

Also, according to [35], two networks are equivalent if they have the same skeleton and the same set of collapsing edges called *V-structures*. Hence, these evaluation measures are applied in Chapter Five. Evaluations of mined networks are performed based on the information provided with the known networks or with the publicly available datasets.

### 4.5.2 Cross Validation

Cross Validation is a technique used to ensure the performance of a learned network. This is to ascertain the accuracy of probabilistic prediction of a query node on the network. According to [58], it is done by setting aside $\frac{1}{K}$ fraction of the known data by using it to test the prediction of

a hypothesis induced from the remaining data. The common values of K are 5 and 10 and this method is therefore also known as **leave-one-out** cross validation.

For instance, if the size of a known dataset is 100 and $K = 5$ *or* $K = 10$, then, the test data size will be 20 or 10, while the training data size will be 80 or 90 respectively. In anomaly detection, test data is CCD and training data is HCD. The detailed result can be visualised as a confusion matrix.

### 4.5.3 Confusion Matrix

A confusion matrix is a tabular structure that contains the number of actual and predicted classifications from a Bayesian network model. The performance of a model as computed with cross validation is shown as the occurrence values in the matrix. With reference to Figure 4.2, if we suppose that our query node is a sprinkler, then we have the confusion matrix as in Table 4.2.

The predicted values with Bayesian inference are on the columns while actual values from dataset are on the rows. From Table 4.2, w = # of correct predictions that an instance is false, x = # of incorrect predictions that an instance is true, y = # of incorrect predictions that an instance is false, and also z = # of correct predictions that an instance is true.

Table 4.2: Confusion Matrix for Dr Watson's Network

| | false | true |
|---|---|---|
| false | w | x |
| true | y | z |

The following can be computed from the confusion matrix:

[i] The **percentage accuracy** of the model

$$= (\frac{w+z}{w+x+y+z}) \times 100 \tag{4.18}$$

that is, total percentage of predictions that were correct.

[ii] The **false positive rate**

$$= (\frac{x}{w+x}) \times 100 \tag{4.19}$$

that is, the percentage of false instances that were incorrectly predicted as true.

[iii] The **true positive rate**

$$= (\frac{z}{y+z}) \times 100 \tag{4.20}$$

that is, the percentage of true instances that were correctly predicted as true.

[iv] The **false negative rate**

$$= (\frac{y}{y+z}) \times 100 \tag{4.21}$$

that is, the percentage of true instances that were incorrectly predicted as false.

[v] The **true negative rate**

$$= (\frac{w}{w+x}) \times 100 \tag{4.22}$$

that is, the percentage of false instances that were predicted correctly as false.

## 4.6 Conclusion

In Figure 4.1, we presented our research system model.

We designed the DTS that observes and preprocesses datasets, specifically call data. The DTS subsystems include the filtering, the discretization and the user-collection subsystems. They were designed to easily prepare the CCD and the HCD subsets for individual subscribers. This enhances the performance of our entire system development.

Our new Hybrid Genetic Algorithm (HGA) is a variant of the classical genetic algorithm. The HGA subsystems were designed in order to be decomposable and is therefore ideally suited to be implemented using intelligent agents, which we will explore in future research.

The use of inner-loops, the control of redundant offspring and the capability to exhaust the solution space makes our HGA modelling architecture reliable. The interactions of the HGA with other systems such as the DTS and the ADS allow the individual user profiles, namely individually selected BBNs.

Also, our ADS allows the changing behaviour of individual subscribers for example when a subscriber makes a phone call to new destination number. This minimises false alarms. More detail on this will be described in the description of the ADS in Chapter Five. Moreover, the use of differential analysis is an improvement in our detection system because it allows on-going changing behaviour of subscribers. Any inconsistent call record in the CCD can easily be trapped by our detection system.

For adaptive intelligence of our ADS, we included the incremental training loop from the ADS to the HCD in our system model. A potential problem for our ADS is if insufficient data is used as the training set, it may weaken our detection.

# Chapter 5

# Implementation Results, Simulations and Evaluations

This chapter presents the implementation results of our algorithms and methodologies as described in the previous chapters. Specifically, we provide the implementation results of our DTS, HGA that mines the individual subscriber Bayesian networks (BBNs), and the ADS. The implementation of our DTS pre-processes real world landline call data for TSP subscribers and produces suitable training datasets. By masking the subscribers' phone numbers, we have ensured the confidentiality of network carriers, whose policy it is to protect telecommunication customers.

The implementation of our Hybrid Genetic Algorithm (HGA) used the training dataset prepared by the DTS. The true calling behaviour of telecommunication subscribers is reflected in their call data, which we visualised as individual Bayesian network models (BBNs). We also visualised the simulation results of the HGA optimizations during the mining of the BBNs. The variabilities of the different subscriber network models provide evidence of differences in individual calling activities that can facilitate the understanding and detection of anomalies in individual subscriber behaviours.

Also, the implementation of our Anomaly Detection System (ADS) that used the BBNs that were mined from the HGA demonstrates the detection of anomalies using daily call data. We obtained good results which we will discuss in the next sections.

Furthermore, this chapter presents three different evaluations and validations of our technologies. In the first place, to validate the universality of our architecture, we used publicly available datasets.

We have conducted a number of experiments to benchmark the performance of our HGA methodology. We used evaluation data in [49] and [50], and the performance results were promising. In the second evaluation, we conducted a test case on our ADS. It measured the performance of ADS accuracies in the detection of regular and anomalous calls for individual subscribers. Lastly, we tested the sensitivity of the ADS in discriminating between different subscribers. The evaluations and the ADS accuracies were good.

The next section describes the programming language and the hardware that we used for our implementation.

## 5.1 Development Environment of Our Algorithms

The system that we implemented consists of the algorithms and the various models presented in the previous chapters. It consists of the three subsystems, namely the DTS, the HGA, and the ADS. The system was implemented using the Java programming language.

The hardware configuration is:
- Pentium(R) 4, CPU 3.00 GHz, 512 MB of RAM.

The operating system is:
- Microsoft Windows XP platform.

Specifically, our software implementations are in Java 5.0 using Eclipse as the IDE (Integrated Development Environment) [17]. Certain representative results are shown in the sections that follow.

## 5.2 Data Transformation System (DTS) Results

The DTS preprocesses and produces a suitable informative training dataset for the modelling architecture. The subsystems of DTS are: the filtering, the discretization and the user-collection subsystems.

Using the phone calls dataset, the filtering subsystem preprocessed data for over 160 subscribers, and more than 73,000 calls with 9 attributes. The major input requirements of DTS are the data types (CSV, text, database etc) it reads which must include originating call numbers on the first

Table 5.1: Some call records for subscriber 145521100, which were extracted before being processed by the DTS

| Orig-No | Call-date | Durat | dest-no | dest-net | Location | Call-cost | p/o | rec-code |
|---|---|---|---|---|---|---|---|---|
| 145521100 | 2004/04/30 | 21 | 145571002 | National | 0-50Km | 0.49 | P | 2250 |
| 145521100 | 2004/05/06 | 26 | 145521613 | National | 0-50Km | 0.49 | P | 2250 |
| 145521100 | 2004/05/13 | 90 | 117807666 | National | >50Km | 1.3 | P | 2250 |
| 145521100 | 2004/05/18 | 41 | 12124268783 | International | USA | 1.81 | X | 2250 |

**Legend**: Orig-No = originating-No, Durat = Duration, dest-no = destination-No, dest-net = destination-net, rec-code = record-code, and p/o = peak/off-peak.

column of the call dataset. The output of the filtering subsystem serves as the input for the discretization subsystem, and its functionalities maintain discrete values in discrete intervals, and continuous values in continuous intervals (see columns Duration and Call-cost in Table 5.2).

The user-collection subsystem receives output from the discretization subsystem and extracts an informative call dataset for each individual subscriber with the attributes and node names. For instance, the DTS observed and preprocessed the original sampled call data in Table 5.1. It may be difficult to understand the table through inspection. The preprocessed informative result is shown in Table 5.2. This table is a mixed dataset. Most other anomaly detection researchers use only continuous attributes [2]. This mixed dataset is more meaningful because of the inclusion of the days of the week and the discretization results. We shortened some column names in the tables and we provided their meanings as a legend below Table 5.1.

According to the TSP that provided our test data (Nebula Systems), the peak period calls are made during the daytime, while the off-peak period is over-night. We excluded originating-No from modelling a subscriber's behaviour because it does not exhibit any cause-effect relationships to other nodes. In other words, the originating-No in the dataset is the same for all records and does not contribute any information to mine Bayesian networks (BBNs). The training set is used by the HGA algorithms to mine the BBNs and helps in better interpretation by the ADS. For more detail on the DTS implementation design, consult the class diagram in Figure A.1 in Appendix A.

Table 5.2: The training set observed for subscriber 145521100 after being processed by the DTS

| Call-date | Duration | destination-No | dest-net | Location | Call-cost | peak/off-peak |
|---|---|---|---|---|---|---|
| Sunday | 0-44 | 145571002 | National | 0-50Km | 0.0-1.25874 | P |
| Sunday | 0-44 | 145521613 | National | 0-50Km | 0.0-1.25874 | P |
| Sunday | 87-131 | 117807666 | National | >50Km | 1.25874-2.51748 | P |
| Friday | 0-44 | 12124268783 | International | USA | 1.25874-2.51748 | X |

## 5.3 Population Construction with Crossover Process Results

With reference to the genetic algorithm terms defined in our literature review and discussion of the methodology, the mathematical Poset lattice generates the network space, from which optimal Bayesian network models are mined. The attributes of the training set are used to construct the population from one generation to another. Table 5.3 shows the results of a solution space for candidate network models generated by our HGA using the call data.

Table 5.3: The crossover process results of Hybrid Genetic Algorithm using the attributes in table 5.4 of our call data

| | |
|---|---|
| Second Generation | {x6,x7}<br>{x5,x6} {x5,x7}<br>{x4,x5} {x4,x6} {x4,x7}<br>{x3,x4} {x3,x5} {x3,x6} {x3,x7}<br>{x2,x3} {x2,x4} {x2,x5} {x2,x6} {x2,x7}<br>{x1,x2} {x1,x3} {x1,x4} {x1,x5} {x1,x6} {x1,x7} |
| First Generation | {x1} {x2} {x3} {x4} {x5} {x6} {x7} |

We define the first generation as the generation that consists of singleton sets called *parents*, while the second generation consists of sets that are the offspring of crossover processes (see Table 5.3). Each offspring is a set containing only two attributes. Higher generations of offspring are monitored with the use of the inner-loops in the implementation of our HGA. Higher generations of offspring without control will introduce redundancies and our inner-loops consist of the EDA and the MI described in Chapter Four.

For instance, a new offspring {x1, x2, x3} can be constructed, which belong to the third generation and may not be fit as a sub-candidate network. The following individuals: {x1, x2}, {x1, x3} and {x2, x3}, which are already in the second generation, formed the new parents of the offspring {x1,

x2, x3}. During the mining process, any of the individuals {x1, x2}, {x1, x3} or {x2, x3} is selected during fitness competition as a sub-candidate network. The competition is performed using the EDA and higher generations of offspring form new sub-candidate networks. Therefore, instead of producing several offspring of more than two attributes, the second generation offspring generate only the ones with the best fitness for the third or higher generations.

We use this method to save reproduction and construction time, to manage memory and to reduce redundant offspring, which may not be fit as sub-candidate networks.

## 5.4 Visualisations of Subscribers' Behavioural Bayesian Networks (BBN) Mined By the HGA

In this section, we describe the implementation results of our hybrid genetic algorithm using the algorithms and models described in Chapter Four. We visualised nine subscribers BBNs as the implementation results of the HGA.

The HGA subsystems that were implemented include the fitness function (MDL), the EDA, the MI etc. For more detail on the implementation design, consult the class design in Figure A.2 in Appendix A. The HGA generated individual subscriber models using Java which can be presented in XML-BIF, database, CSV (comma separated variable) or text file formats. We have used JavaBayes to visualise the BBN models.

The following Figures in 5.1 to 5.17 are some of the subscribers' BBN models that were generated in this research using preprocessed call data such as Table 5.2. Each of these BBN models was mined with different data sizes depending on the call behaviour of individual subscriber. On average, every model was mined by the HGA using 650 call records. Table 5.4 lists the call attributes and their corresponding node names in the Bayesian networks. Originating number (x0) is not used as part of the modelling because it stays the same for individual subscribers.

The variabilities of these networks illustrate the difference in individual subscribers behaviour. This confirmed that a single general model cannot be used effectively to model all subscribers.

The patterns of calls are reflected and identified in the relationships mined from the call attributes. These relationships are referred to as qualitative knowledge. The quantitative knowledge is captured in the CPTs. During Bayesian inference, the ADS uses the qualitative and quantitative knowledge

to detect anomalies.

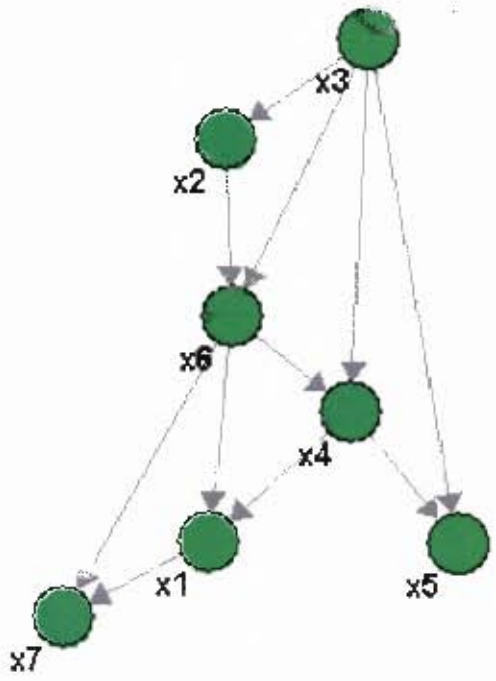


Figure 5.1: The Bayesian network (BBN) mined from the historical call data of subscriber 145521137

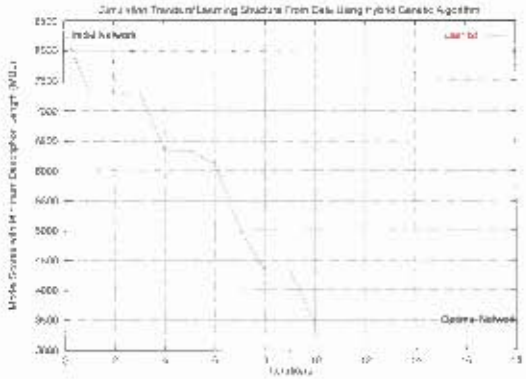

Figure 5.2: Optimization of Network Model: The behaviour of the Bayesian network when searching for the best model that fits the call data of subscriber 145521137

Table 5.4: Attributes of Phone Call Records

| Attributes | Node names |
|---|---|
| originating-No | x0 |
| Call-date | x1 |
| Duration | x2 |
| destination-no | x3 |
| destination-net | x4 |
| Location | x5 |
| Call-cost | x6 |
| peak/off-peak | x7 |

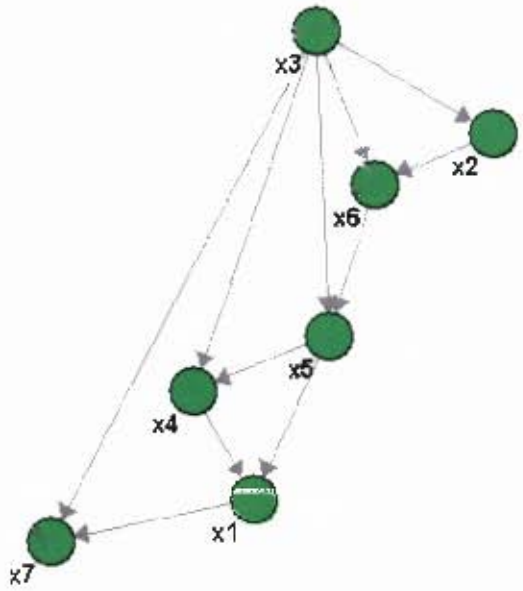


Figure 5.3: The Bayesian network mined from the historical call data of subscriber 145521198

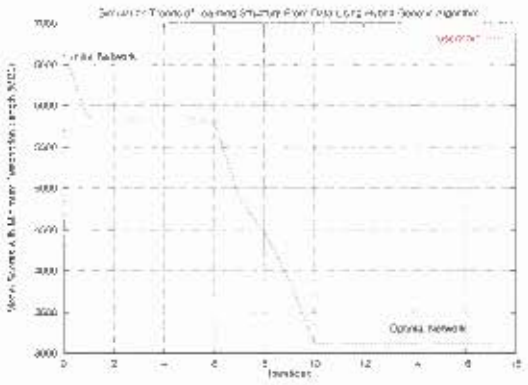

Figure 5.4: Optimization of Network Model: The behaviour of the Bayesian network when searching for the best model that fits the call data of subscriber 145521198

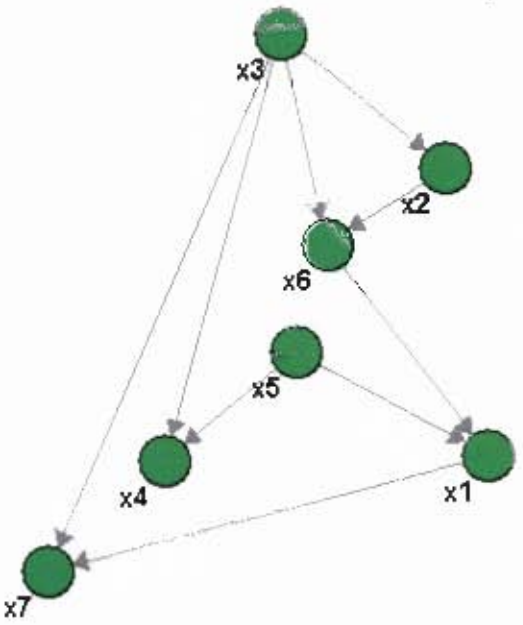


Figure 5.5: The Bayesian network mined from the historical call data of subscriber 145571025

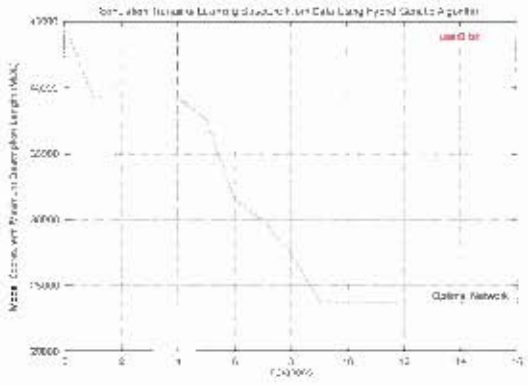

Figure 5.6: Optimization of Network Model: The behaviour of the Bayesian network when searching for the best model that fits the call data of subscriber 145571025

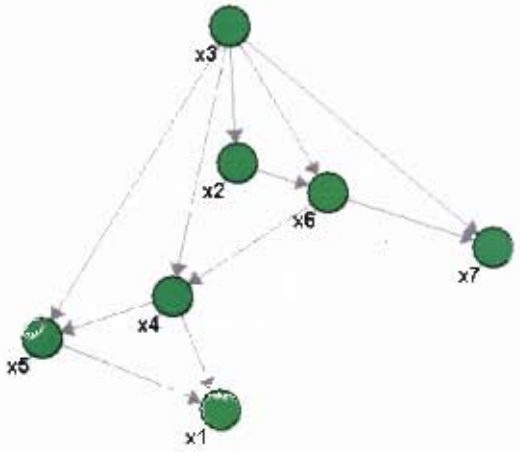


Figure 5.7: The Bayesian network mined from the historical call data of subscriber 145571042

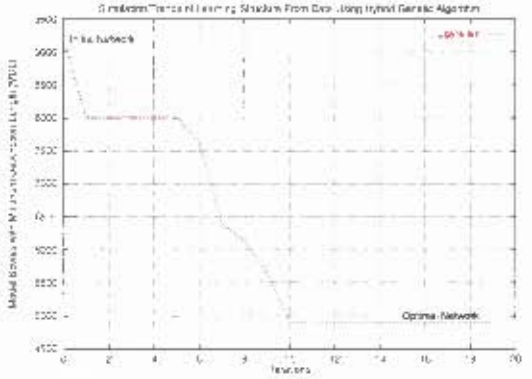

Figure 5.8: Optimization of Network Model: The behaviour of the Bayesian network when searching for the best model that fits the call data of subscriber 145571042

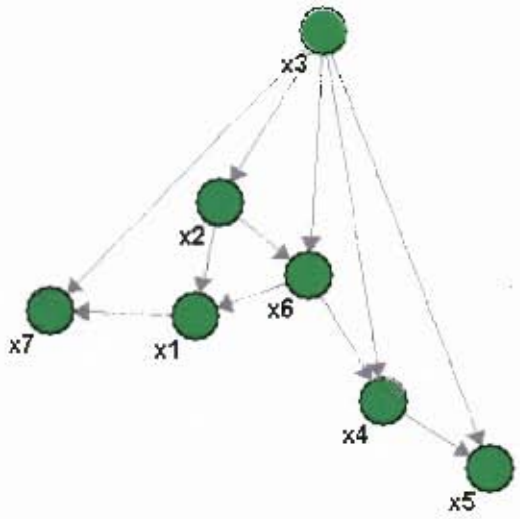


Figure 5.9: The Bayesian network mined from the historical call data of subscriber 145571050

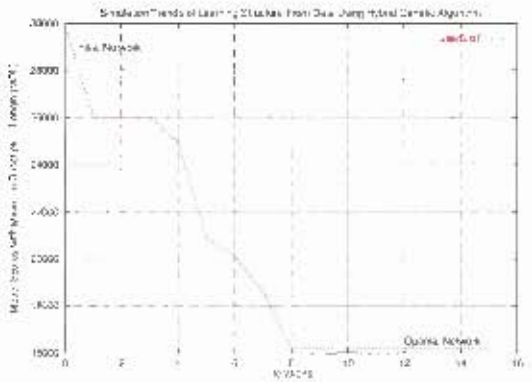

Figure 5.10: Optimization of Network Model: The behaviour of the Bayesian network when searching for the best model that fits the call data of subscriber 145571050

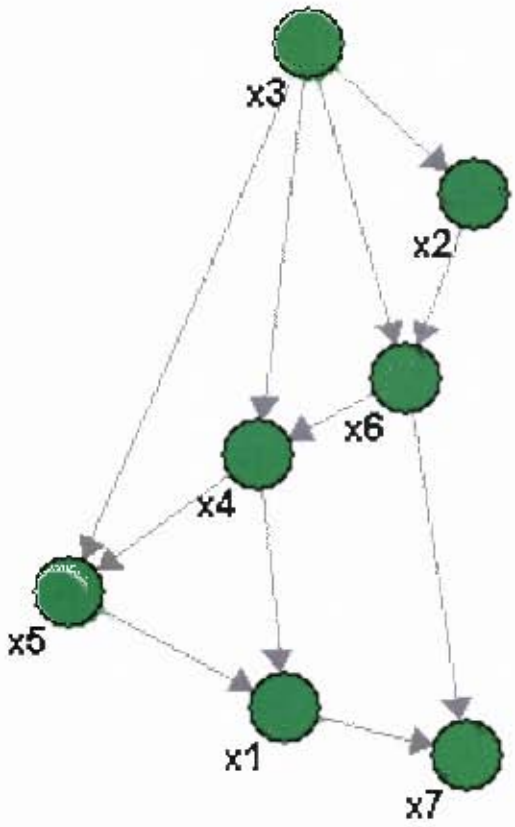


Figure 5.11: The Bayesian network mined from the historical call data of subscriber 145571179

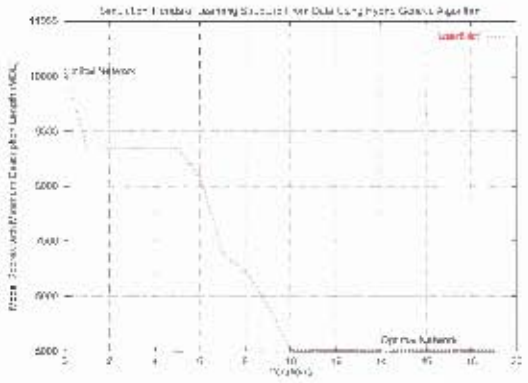

Figure 5.12: Optimization of Network Model: The behaviour of the Bayesian network when searching for the best model that fits the call data of subscriber 145571179

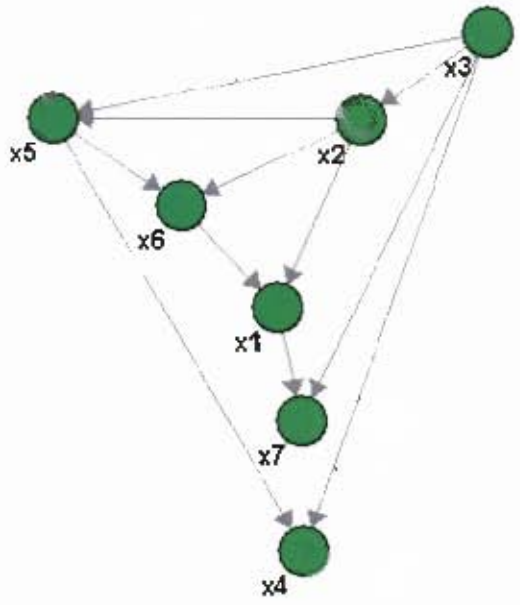


Figure 5.13: The Bayesian network mined from the historical call data of subscriber 145571055

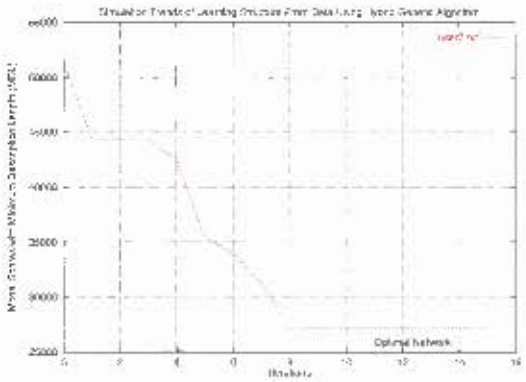


Figure 5.14: Optimization of Network Model: The behaviour of the Bayesian network when searching for the best model that fits the call data of subscriber 145571055

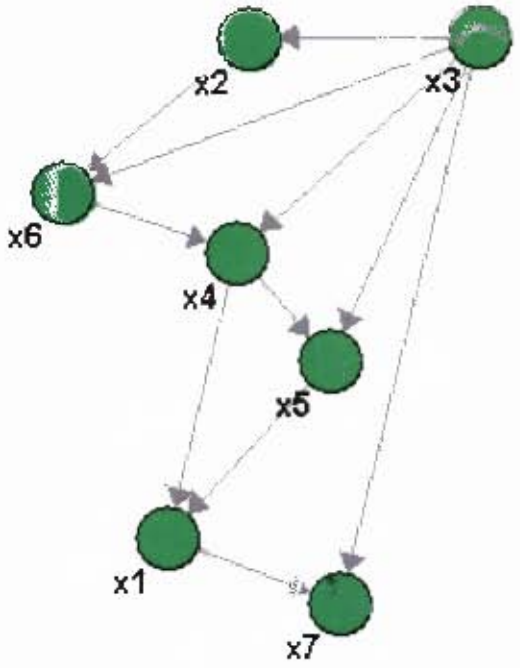


Figure 5.15: The Bayesian network mined from the historical call data of subscriber 145571046

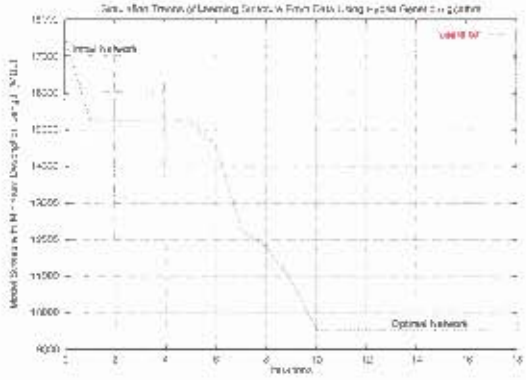


Figure 5.16: Optimization of Network Model: The behaviour of the Bayesian network when searching for the best model that fits the call data of subscriber 145571046

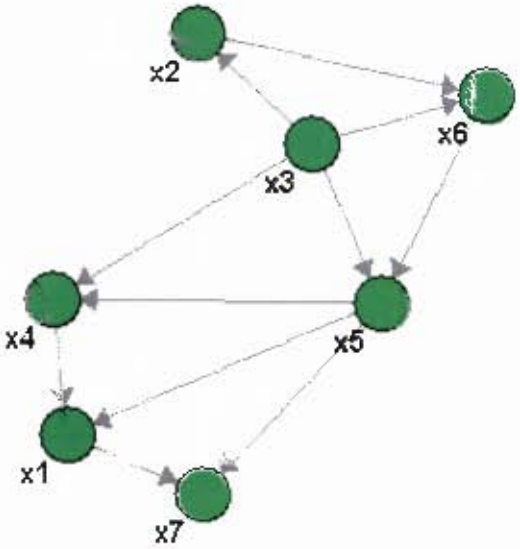


Figure 5.17: The Bayesian network mined from the historical call data of subscriber 145571030

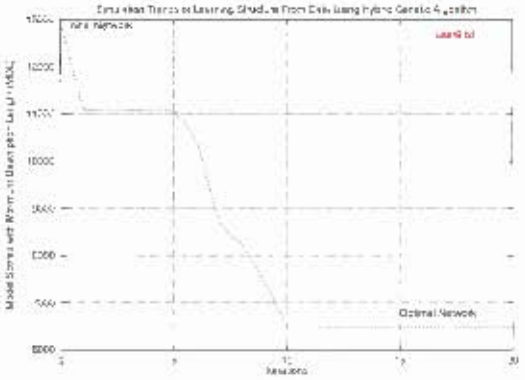

Figure 5.18: Optimization of Network Model: The behaviour of the Bayesian network when searching for the best model that fits the call data of subscriber 145571030

## 5.5 Simulation Results of the HGA

In this section, we describe and present the simulation results of HGA by visualising the optimisation results of nine BBNs.

By looking at the subscribers models in the previous section, we simulated the structure learning process by using the network optimisation results. The objective of the fitness function (MDL) used in the HGA is to find the optimal *global minimum scores* for the BBN. If the implementation of the HGA is terminated by convergence, when searching for optimal network, it may become stuck at the *local minimum score*. Our genetic algorithm takes time to exhaust the entire solution space but eventually ends up in a global minimum score. During scoring, we use the MDL; the lower the score, the better the network.

It is already confirmed by Russel [58] that any search algorithm (e.g. genetic algorithm) that completes the entire solution space will always find an optimal global minimum score. In the development of our HGA, it is certain with PowerSet that solution space must always complete as shown in table with population construction 5.3. Without using genetic algorithm terms, observe that every variable completely pairs with all other variables. It makes it easy for scoring the entire

solution space at implementations. Furthermore, to be certain about the global minimum scores, we repeated mining the BBNs in Figures 5.1 to 5.17 at times without number, same results were obtained. If there were local minimum scores, there would have been at least one different result.

We wrote scripts using open source software called *Gnuplot* [64] to simulate and visualise the optimisation results generated from our system. The simulation results that correspond to the subscribers' models are shown in Figures 5.2 to 5.18.

By studying Figures 5.2 to 5.18, it is evident that there are points on the graphs that could have been local minima. These are horizontal points at the middle of the graphs that appear as local convergence. These periods could be wastage of time but advanced fitness functions may minimise this.

## 5.6 Performance Evaluations of the HGA

This section presents the performance of our Hybrid Genetic Algorithm (HGA) in mining Bayesian network models from datasets.

We used publicly available datasets with the evaluation information supplied with them, and open source software to evaluate our modelling architecture. As discussed in Chapter Four, we used public datasets in the UCI repository [49] and the dataset provided by Nilsson [50]. We used Weka [69] and an evaluation copy of a machine learning software package called BayesiaLab [3] to evaluate some of our results as shown in the next sections.

### 5.6.1 Structural Evaluations by using the Nilsson Network as Benchmark

This subsection provides structural evaluations by measuring the similarities between the network mined by our HGA and the Nilsson network using the same dataset. We confirmed this measurement by comparing the resulting network with the networks mined using two different software packages.

Nilsson [50] provides, together with the benchmarking network, a small dataset that contains the operation of a block-lifting machine. Nilsson models the behaviour of the machine with its battery, monitors if the arm of the machine moves when holding the block, and monitors if a block is liftable.

Also, a gauge was used to indicate the status of the battery, if it was fully charged. However, no information is provided on whether the network was defined by a domain expert or whether it was mined from the data. Nilsson used the data to explain the scoring measure during the mining of Bayesian networks.

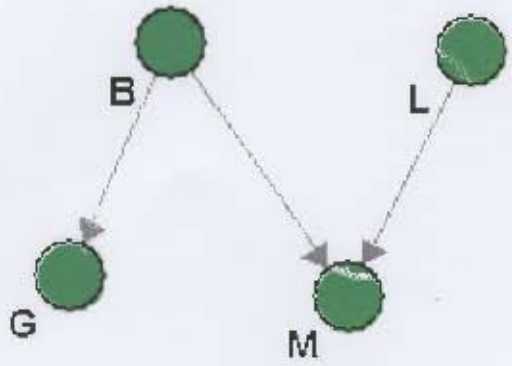


Figure 5.19: Block-Lifting Model From Nilsson

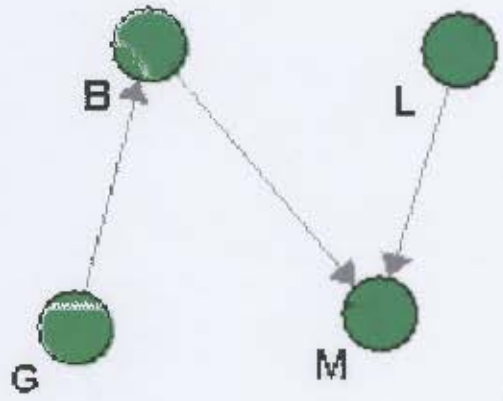


Figure 5.20: Block-Lifting Model From Implemented Hybrid Genetic Algorithm (HGA)


Figure 5.21: Block-Lifting Model From Genetic Algorithm of Weka

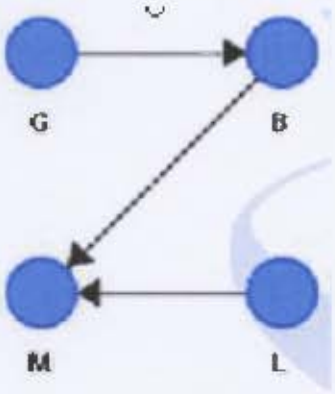


Figure 5.22: Block-Lifting Model From BayesiaLab Genetic Algorithm

In order to draw comparisons with Nilsson's network, we mined the dataset with a similar genetic algorithm to ours in *Weka*. We further mined the same dataset with another similar genetic algorithm of BayesiaLab. The Bayesian networks from the three sources, as well as the one generated from the implementation of our own new HGA, are shown in Figures 5.19 to 5.22.

From the results in Figures 5.19 to 5.22, the similarity of our model as compared with the Nilsson, Weka, and BayesiaLab models were computed and presented in Table 5.5. We have computed the measures of similarity in percentages using equation 4.17. By comparing Nilsson's network in Figure 5.19 with Figure 5.20, it shows that there is an opposite orientation between nodes G and B. Since there is one symmetric difference on the arc with the two networks, the similarity is computed as:

$$(2/3 * 100)\ \%$$

$$\Longrightarrow 66.67\%$$

As compared with Figures 5.21 and 5.22, the symmetric difference is zero. This implies that the similarity is:

$$(3/3 * 100)\ \%$$

$$\Longrightarrow 100\%$$

The percentage similarities can change depending on the datasets and the parameters used in the learning algorithms.

Table 5.5: Hybrid Genetic Algorithm (HGA) Modelling Similarities

| | Nilsson | Weka GA | BayesiaLab GA |
|---|---|---|---|
| HGA | 66.67% | 100% | 100% |

The significance of the similarity measures is to ensure that a newly developed search algorithm can discover at least some related links between network variables as existing search algorithms. Even though many Bayesian network researchers [27] said since arcs are not unique, only links should be considered but not the arcs when comparing two networks, we still include both arcs and links in some of our measurements.

### 5.6.2 Structural Evaluations by using UCI Machine Learning Repository Datasets as Benchmarks

In this subsection, we present an evaluation of our HGA, using the repository datasets of the University of California, Irvine (UCI) [49]. We observed that most of these datasets were not used for finding relationships in Bayesian networks because no network was made available [49]. They were used for statistical classification purposes based on the information supplied with the datasets [49]. Also, most of the complete datasets without missing values have relationships that involve the attributes of the class node such as Figure 5.23.

We tested our HGA implementation using the nursery dataset from UCI, which contains 12,960 instances and 9 attributes. We compared our network with the number of arcs in the network as presented by Josep [33]. Our mined nursery network is shown in Figure 5.23.

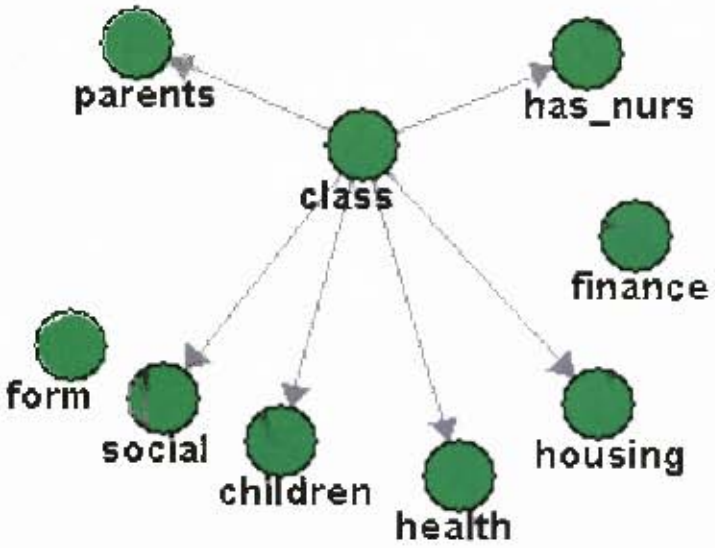


Figure 5.23: Nursery Bayesian Network Model

Josep [33] does not present the network structure mined from the dataset but discovered 8 links in his nursery network. In comparison, we obtained 6 links as shown in Figure 5.23. This difference does not mean that either Josep's result or our model result is poor. It has already been identified by William [70] that, in the real-world applications of structure learning, there is no *gold standard model* for a dataset.

We validated the statement made by William using the Iris dataset [49] mined by Weka, BayesiaLab,

and our HGA. In comparison, the networks were similar with some differences in the relationships between nodes as shown in Figures 5.24 to 5.26.

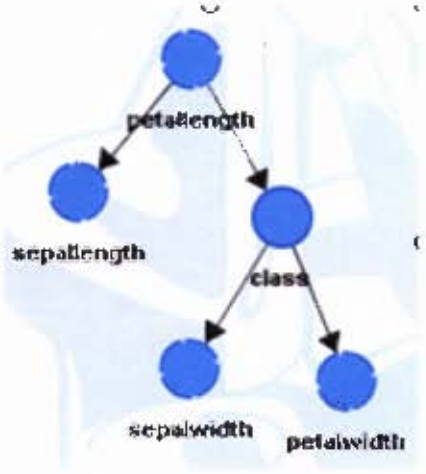


Figure 5.24: Iris Model Mined From UCI Data Using Genetic Algorithm of BayesiaLab Evaluation Copy

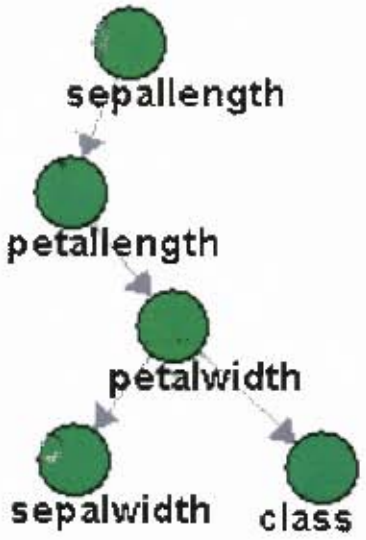


Figure 5.25: Iris Model Mined From UCI Data Using Genetic Algorithm of Weka

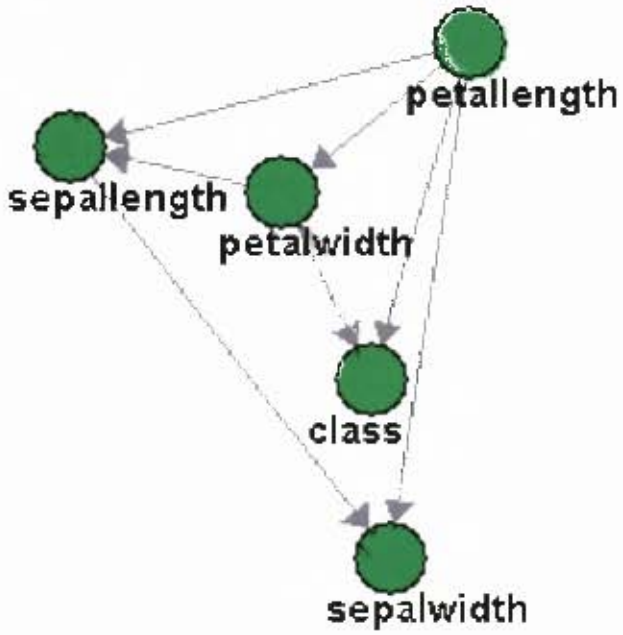


Figure 5.26: Iris Model Mined From UCI Data Using Hybrid Genetic Algorithm

The differences could be as a result of $\epsilon$, where $\epsilon$ is the link threshold, such as 0.01, 0.05, etc., as used in different implementations [31]. As discussed in Chapter Four, one of the functionalities of

the EDA is to determine if the link (or information sharing) between two attributes is significant or not. A link is significant if the information sharing between the attributes is greater than or equal to $\epsilon$.

Like most existing systems, we can mine networks from mixed (numerical and nominal) or nominal (textual) datasets. In addition, our system can also mine networks from numerical datasets that are not discretized. Most existing systems do not have this capability but instead need to discretize the dataset. Figure 5.27 shows the Bayesian network that we mined from computer hardware dataset obtained from UCI using the IIGA. The dataset contains 209 Instances and 7 attributes, and was not discretized.

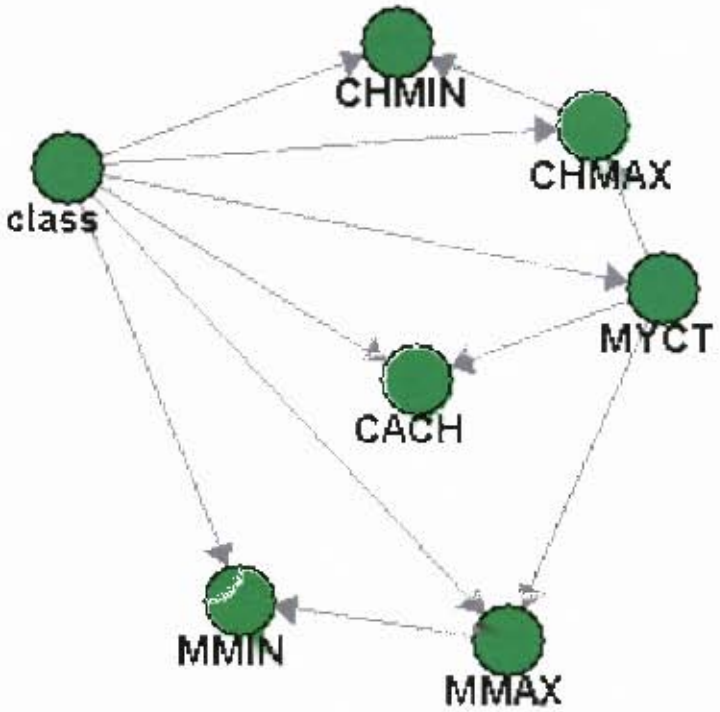


Figure 5.27: Computer Hardware Bayesian Network Model

It is evident that our IIGA found more meaningful relationships in comparison with Weka and BayesiaLab. For instance, more relationships are better in a network because of the reasons on more information propagation stated about multiply-connected networks in Chapter Three. Also, the evidence of the meaningful relationships is observed on the reliable detection results, 85.878 % we obtained in Chapter Five.

The network in Figure 5.27 contains 11 arcs, the mining took 5 seconds and the minimum description

length score of the network is 2372.535 bits.

The similarities of our HGA model results as compared with the existing models used in this section made our modelling architecture applicable to other applications [56].

The next section describes the identification of most interesting attributes in mined networks. This is useful and was used to improve the effectiveness of our ADS.

## 5.7 BBN Applications to Telecommunications Anomaly Detection

In this section, we present our modelling architecture as applied to anomaly detection in telecommunications call data. We experimented with our ADS by using five subscribers and we performed two evaluation test cases on the ADS. The experiments of the ADS detected possible anomalies in daily calls. The ADS was first evaluated with respect to its effectiveness in detecting anomalies within a subscriber's behaviour while the second test confirmed if it could identify one subscriber's calls using another subscriber's model. The ADS gave good and interpretable results.

We have designed, implemented and evaluated the HGA to mine Bayesian network structures. As presented in the preceding sections of this chapter, the evolved BBNs represent the behaviour of individual subscribers. We tested the effectiveness of our models using the Anomaly Detection System (ADS), to detect regular and anomalous phone calls. In our ADS implementation, we used primary and secondary anomaly indicators as *query nodes.*

Recall from the preceding section that the primary and secondary anomaly indicators for Figure 5.1 are Destination-no and Call-cost respectively. Using the BBN of Figure 5.1, examples of anomaly detection queries used during Bayesian inference by the ADS are given in equations 5.1 and 5.2. We used the secondary anomaly indicator as a query node as in equation 5.2 because subscribers regularly make calls to new destination numbers. This makes the primary anomaly indicator in equation 5.1 incapable of giving a correct inference result. Therefore, we allowed calls made to new destination numbers by changing the query nodes.

$$Pr(DestinationNo = 828050678 \mid Location = \text{" } > 50km\text{"}, Duration = 120, CallDate = Wednesday, DestinationNet = Cellular, CallCost = 7.55244, Peak/OffPeak = P) \quad (5.1)$$

$$Pr(CallCost = 7.55244 \mid Location = \text{" } > 50km\text{"}, \ Duration = 120, CallDate = Wednesday,$$
$$DestinationNet = Cellular, Peak/OffPeak = P, DestinationNo = 828050678) \quad (5.2)$$

As shown in our system model (see Figure 4.1), the HGA mined Bayesian network models (BBNs) for individual subscribers using historical call datasets (HCD). The ADS used the HCD subsets for individual subscribers to train the evolved BBNs and to update the CPTs. The ADS used the trained BBNs to detect regular and anomalous calls from the CCD subsets for individual subscribers. The regular calls are used to incrementally train the BBNs but the anomalous calls are excluded for TSP investigation. Only those anomalous calls identified by the TSP as regular calls, are added to the HCD.

We used daily transactions in this research. Our ADS is a able to detect anomalies in real time but in the prototype, we process daily transactions. The ADS minimises response time using daily transactional processing. This improves on most existing batch anomaly detection systems.

## EXPERIMENTAL RESULTS OF ANOMALY DETECTION ON DAILY CALLS IN CURRENT CALL DATASETS (CCD)

The following are examples of the daily detection results that were generated from our implementations using five subscribers.

### ADS Results for Subscriber 1

From our implementations, we present a HCD subset for subscriber *145521137*, in Table 5.6. The meanings of the column names are set out on the legend at the bottom of the table.

The complete dataset for the HCD subset for Table 5.6 was used by our HGA to mine the Behavioural Bayesian Network in Figure 5.1.

Our ADS trained and used the model in Figure 5.1, to act upon the daily call records, and anomalous and regular calls were detected as shown in Table 5.7. The table shows the call detection results for the phone calls that were made on the last day, *Saturday*. All regular call events are used to incrementally train the model to actualise adaptive intelligence. Observe that the degree of belief for each call event is very clear and can therefore assist call managers to make reliable decisions.

For example, the first call event in Table 5.7 was detected as *regular* with degree of belief of 83.489

%. This gives the call manager the confidence to claim the cost from the subscriber. However, the last call event result that was detected as *regular*, only has a degree of belief of 4.441 %. This low percentage is due to the use of the secondary anomaly indicator rather than the primary anomaly indicator. This is an example of a call made to a possible new phone number because this particular call record has a characteristic behaviour reflected in the historical behaviour of this subscriber.

**ADS Results for Subscriber 2**

The subset of the HCD for subscriber *145521198*, is shown in Table 5.8, which was used by our HGA to mine the BBN in Figure 5.3.

The model in Figure 5.3 was used by the ADS, and anomalous and regular calls were detected as shown in Table 5.9. This table shows the call detection results for the phone calls that were made on *Sunday*. Observe that the first call event was detected as *anomalous* with the degree of belief of 54.615 %. This gives the call manager the confidence to see that there is indeed an inconsistency in the call patterns of this subscriber. This might have been attributed to one of the causes stated in our introduction, namely unintended expensive mistakes, potential fraud etc, and the appropriate action can be taken by the TSP.

The similar ADS results for the other three subscribers are included in Appendix B.

Table 5.6: Observed training set for subscriber 145521137

| x1 | x2 | x3 | x4 | x5 | x6 | x7 |
|---|---|---|---|---|---|---|
| Wednesday | 218-261 | 828050678 | Cellular | (Vodacom) | 6.293699999999999-7.55244 | P |
| Wednesday | 87-131 | 849001100 | Cellular | (Cell-C) | 2.51748-3.77622 | P |
| Friday | 44-87 | 1023 | SpecialServices | 0-50Km | 0.0-1.25874 | P |
| Saturday | 0-44 | 824442726 | Cellular | (Vodacom) | 1.25874-2.51748 | P |
| Saturday | 131-174 | 835575170 | Cellular | (MTN) | 3.77622-5.03496 | P |
| Sunday | 44-87 | 824442726 | Cellular | (Vodacom) | 1.25874-2.51748 | P |
| Sunday | 0-44 | 824442726 | Cellular | (Vodacom) | 1.25874-2.51748 | P |
| Tuesday | 0-44 | 145924180 | National | 0-50Km | 0.0-1.25874 | X |
| Thursday | 0-44 | 836838283 | Cellular | (MTN) | 1.25874-2.51748 | P |
| Monday | 87-131 | 829778104 | Cellular | (Vodacom) | 1.25874-2.51748 | P |
| ,, | ,, | ,, | ,, | ,, | ,, | ,, |

**Legend**: x1 = Call-date, x2 = Duration, x3 = Destination-no, x4 = Destination-net, x5 = Location, x6 = Call-cost, x7 = Peak/off-peak

Table 5.7: ADS Detects Current Calls for subscriber 145521137

| Daily Call Events |
|---|
| [Saturday, 44-87, 828581169, Cellular, (Vodacom), 1.25874-2.51748, P]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 83.489% |
| [Saturday, 0-44, 836431694, Cellular, (MTN), 1.25874-2.51748, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 30.372 % |
| [Saturday, 0-44, 834143703, Cellular, (MTN), 1.25874-2.51748, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 65.186 % |
| [Saturday, 44-87, 822302000, Cellular, (Vodacom), 1.25874-2.51748, P]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 45.603 % |
| [Saturday, 87-131, 834105953, Cellular, (MTN), 1.25874-2.51748, P]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 4.441 % |
| ,, |

Table 5.8: Observed training set for subscriber 145521198

| x1 | x2 | x3 | x4 | x5 | x6 | x7 |
|---|---|---|---|---|---|---|
| Friday | 44-87 | 113291200 | National | >50Km | 0.0-1.25874 | P |
| Friday | 44-87 | 145663811 | National | 0-50Km | 0.0-1.25874 | P |
| Saturday | 44-87 | 183971500 | National | >50Km | 0.0-1.25874 | P |
| Sunday | 0-44 | 145521710 | National | 0-50Km | 0.0-1.25874 | P |
| Thursday | 87-131 | 117807444 | National | >50Km | 1.25874-2.51748 | P |
| Thursday | 0-44 | 828057570 | Cellular | (Vodacom) | 1.25874-2.51748 | P |
| Monday | 0-44 | 828032536 | Cellular | (Vodacom) | 1.25874-2.51748 | P |
| Wednesday | 131-174 | 824673337 | Cellular | (Vodacom) | 3.77622-5.03496 | P |
| ,, | ,, | ,, | ,, | ,, | ,, | ,, |

**Legend**: x1 = Call-date, x2 = Duration, x3 = Destination-no, x4 = Destination-net, x5 = Location, x6 = Call-cost, x7 = Peak/off-peak

Table 5.9: ADS Detects Current Calls for subscriber 145521198

| Daily Call Events |
|---|
| [Sunday, 44-87, 832280426, Cellular, (MTN), 1.25874-2.51748, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 54.615 % |
| [Sunday, 44-87, 117807444, National, >50Km, 0.0-1.25874, P]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 55.557 % |
| [Sunday, 0-44, 834581134, Cellular, (MTN), 1.25874-2.51748, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 54.615 % |
| [Sunday, 44-87, 828058678, Cellular, (Vodacom), 1.25874-2.51748, P]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 92.862 % |
| [Sunday, 0-44, 832510192, Cellular, (MTN), 1.25874-2.51748, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 51.779 % |
| ,, |

### 5.7.1 Performance Evaluation of the Anomaly Detection System (ADS)

Since telecommunication services providers (TSP) cannot easily make call data available to researchers, the scientific alternative is the use of cross validation techniques as discussed in Chapter Four. We used the 90% and 10% cross validation technique to evaluate the performance of our implemented Anomaly Detection System (ADS). We present three *test cases* as shown in the subsequent sub-sections.

We used cross validation for the subscribers, whose daily calls were classified in the preceding subsection. We evaluated the accuracy of ADS by comparing the *expected call data* and the actual *calls detected* for every subscriber.

**TEST CASE ONE: ANOMALY DETECTION WITHIN A SUBSCRIBER'S PROFILE**

In test case one, we identified regular and anomalous calls from every subscriber's call data. The 90% of a subscriber's call data was used as a training set while the 10% was chosen at random as a test set. This was repeated for the five subscribers and the ADS accuracies were computed as shown in the next sections.

**ADS Accuracies for Subscriber 1**

Using the cross validation technique, the expected call data from the test set is shown in Table 5.10. This is because domain experts did not expect any anomalies. That is, none of the training records was known as anomalous before modelling.

Table 5.10: Expected Call Detection Results for subscriber 145521137.

| | Total |
|---|---|
| Regular Calls | 40 |
| Anomaly Calls | 0 |

The confusion matrix generated from our implementations, in respect of subscriber 145521137, is shown in Table 5.11.

The ADS accuracy for subscriber 145521137, is **87.5 %** . This is the true positive rate while the

Table 5.11: Confusion Matrix Results for subscriber 145521137

| Expected; Predicted | Regular | Anomaly |
|---|---|---|
| Regular: | 35.0 | 5.0 |
| Anomaly: | 0.0 | 0.0 |

false alarm rate is **12.5 %**

That is, the anomalous calls are regarded as false call detections.

**ADS Accuracies for Subscriber 2**

The expected call data from the test set is shown in Table 5.12. The confusion matrix generated in respect of subscriber 145521198 is shown in Table 5.13.

The ADS accuracy for subscriber 145521198 is **58.065 %**. This is the true positive rate while the false alarm rate is **41.935 %**

Table 5.12: Expected Call Detection Results for subscriber 145521198

| | Total |
|---|---|
| Regular Calls | 31 |
| Anomaly Calls | 0 |

Table 5.13: Confusion Matrix Results for subscriber 145521198

| Expected; Predicted | Regular | Anomaly |
|---|---|---|
| Regular: | 18.0 | 13.0 |
| Anomaly: | 0.0 | 0.0 |

Similar ADS accuracies for the other three subscribers are shown in Appendix C.

**Overall Accuracies of the ADS**

The overall accuracy (true positive rate) of our anomaly detection system can be defined as the average of the ADS accuracies for the five subscribers. The overall accuracy from our implementations is:

82.880 %

In addition to the interpretations for the degrees of beliefs of the ADS results, we now present visualised interpretations as shown in Figure 5.28. From the detection results on test case one, the simulation trends of the true positive rates and the false alarms (anomalies) are shown in Figure 5.28. We specifically referred to anomalies here as false alarms because they were not expected. More information is provided in test case one in Chapter Five. It is clear from the graph that we maximised correct predictions (true positive rates) and minimised false alarms.

When anomalies are detected in call records, caused by e.g. unintended expensive mistakes, potential fraud etc, the ADS output will assist the call analysts to quickly locate the call records where possible anomalies may have occurred. They no longer have to search through all the call records looking for anomalies. With respect to the ADS quality for test case one, the overall **true positive rate** (accuracy) of our ADS was 82.880%.


Figure 5.28: Prediction Rates of Anomaly Detection System for Test Case One

## TEST CASE TWO: ANOMALY DETECTION BETWEEN TWO SUBSCRIBERS' PROFILES

This test case is to show that our ADS can detect calls that were made by a subscriber, using the model of another subscriber.

90% of the HCD for a particular subscriber was used as a training set while 10% was chosen at random as a test set from another subscriber's HCD. For instance, 90% of training set in Table 5.6 was used to model call patterns of subscriber 145521137. 10% in Table 5.8 for subscriber 145521198, was used as a test set. The expected call detection results is shown in Table 5.14.

Table 5.14: Test case two: Expected Call Detection Results using subscriber 145521137's model

| | Total |
|---|---|
| Regular Calls | 0 |
| Anomaly Calls | 30 |

We expected all the detected calls to be anomalous because it is obvious that the call records do not belong to subscriber 145521137. The sampled detection results when ADS acted on the test set is shown in Table 5.15 and the confusion matrix is shown in Table 5.16.

Table 5.15: Test case two: ADS Detects Calls in 145521198's data using 145521137's model

| Daily Call Events |
|---|
| [Saturday, 0-44, 117807248, National, >50Km, 0.0-1.25874, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 100.0 % |
| [Friday, 44-87, 113291200, National, >50Km, 0.0-1.25874, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 100.0 % |
| [Thursday, 44-87, 828256810, Cellular, (Vodacom), 1.25874-2.51748, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 100.0 % |
| [Sunday, 0-44, 117807020, National, >50Km, 0.0-1.25874, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 100.0 % |
| [Monday, 44-87, 145555291, National, 0-50Km, 0.0-1.25874, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 100.0 % |
| ,, |

For test case two, the ADS accuracy for subscriber 145521137 is **100.0 %**. This shows that our

Table 5.16: Test case two: Confusion Matrix Results using subscriber 145521137's model

| Expected; Predicted | Regular | Anomaly |
|---|---|---|
| Regular: | 0.0 | 0.0 |
| Anomaly: | 0.0 | 30.0 |

ADS is reliable.

We also repeated this experiment by using subscriber 145521198's data as training and subscriber 145521137's data as test set. The expected call detection results is shown in Table 5.17.

Table 5.17: Test case two: Expected Call Detection Results using subscriber 145571198's model

| | Total |
|---|---|
| Regular Calls | 0 |
| Anomaly Calls | 39 |

We also expected all the calls in the test set to be anomalous. The detection results of the ADS acting on the test set is shown in Table 5.18 and the confusion matrix is shown in Table 5.19.

For test case two, the ADS detection accuracy for the subscriber 145521198 is **100.0 %**. This again confirms our ADS to be reliable.

Thus, it is clear from test case two's results that, the ADS can efficiently classify calls as belonging to another subscriber if the BBN of a different subscriber was used as the model. This will typically occur during subscription fraud when a handset is stolen.

Table 5.18: Test case two: ADS Detects Calls in 145521137's data using 145521198's model

| Daily Call Events |
|---|
| [Friday, 0-44, 219494280, National, >50Km, 0.0-1.25874, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 100.0 % |
| [Monday, 44-87, 824616747, Cellular, (Vodacom), 1.25874-2.51748, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 100.0 % |
| [Saturday, 0-44, 834105953, Cellular, (MTN), 1.25874-2.51748, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 100.0 % |
| [Thursday, 0-44, 842002025, Cellular, (Cell-C), 1.25874-2.51748, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 100.0 % |
| [Sunday, 0-44, 845993456, Cellular, (Cell-C), 1.25874-2.51748, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 100.0 % |
| ,, |

Table 5.19: Test case two: Confusion Matrix Results using subscriber 145521198's model

| Expected; Predicted | Regular | Anomaly |
|---|---|---|
| Regular: | 0.0 | 0.0 |
| Anomaly: | 0.0 | 39.0 |

## TEST CASE THREE: ANOMALY DETECTION USING TEST SET THAT INCLUDES INTRODUCED ANOMALIES

Test case three is another scientific way to ascertain that our ADS can convincingly detect anomalies from subscribers' daily calls.

Like previous test cases, we used 90% of the HCD as a training set to model individual subscriber behaviour and we randomly generated 10% of anomalous test data called noise. The noise records were generated as known anomalies using the properties or data types of the real dataset. We repeated these for eight subscribers. The ADS uses most of the BBN models in Figures 5.1 to 5.17

to act upon these anomalies. Since all the test records are known anomalies, we expected to have 100% detected anomalies. For example, Table 5.20 shows some detected call results for subscriber 145521137.

Table 5.20: Test case three: some calls detected by the ADS using model for subscriber 145521137

| Daily Call Events |
|---|
| [[Wednesday, 314-319, 218521142, National, >50Km, 2.088-2.588, P]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 98.975 % |
| [Wednesday, 299-304, 145920661, MobileInternational, >50Km, 5.430-5.930, X]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 98.975 % |
| [Tuesday, 168-173, 726765711, National, >50Km, 9.754-10.254, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 97.950 % |
| [Monday, 273-278, 145928463, National, 0-50Km, 1.468-1.968, P]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 94.060 % |
| [Friday, 481-486, 118987000, National, >50Km, 5.699-6.199, X]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 97.030 % |
| ,, |

In this case, the calls detected as anomalies are regarded as true detection, while the calls detected as regular calls are false alarms. For the efficiency of the ADS, the accuracies of the detection are summarised in Table 5.21. For every subscriber, we computed the rate of calls detected as percentages. The correctly detected calls are computed as true positive rates, while the incorrectly detected calls or false alarms are the error rates. In other words, in this test case, the average accuracy of correctly detected anomalies is 74.754 %, while incorrect detection is 25.246 %.

One of the objectives of our methodology is to maximise true positive rates and minimise false alarms. One can see in Table 5.20 that, for every subscriber, the true positive rate is greater than the error rate. We cannot have a group model for subscribers who behave in a similar way. If we do not have individual models to separate subscribers' behaviours, the false detection rates will be

Table 5.21: Test Case Three: The anomaly detection accuracies for eight subscribers

| Subscribers' Originating Number | True Positive Rates (%) | False Alarms (%) |
|---|---|---|
| 145521137 | 70.0 | 30.0 |
| 145571179 | 88.889 | 11.111 |
| 145571030 | 75.0 | 25.0 |
| 145521198 | 66.667 | 33.333 |
| 145571046 | 72.727 | 27.273 |
| 145571050 | 77.778 | 22.222 |
| 145571055 | 63.634 | 36.366 |
| 145571042 | 83.333 | 16.667 |
| Average accuracies | 74.754 | 25.246 |

greater than the true positive rates. Figure 5.29 visualises these detection results.

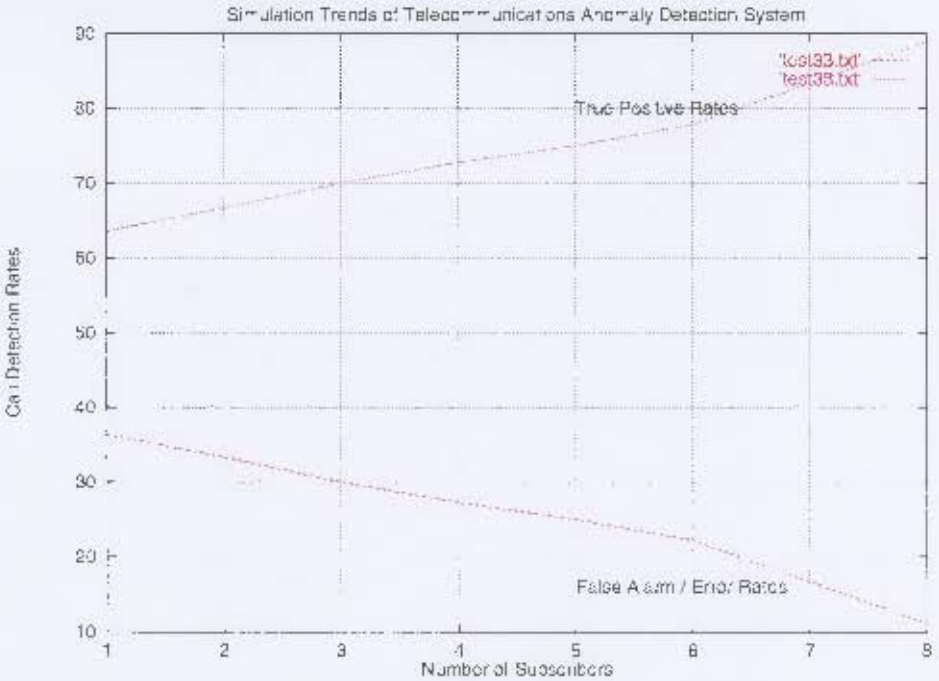


Figure 5.29: Prediction Rates of Anomaly Detection System for Test Case Three

For the performance of our ADS, we summarised in table 5.22, the three test cases for the overall average of anomalies detected.

Our methodology is more interesting than most of the prior work in our literature review since our overall average detection accuracy is 85.878 %. For instance, as discussed in our literature review, Hollmen detection result was 85 %. Unlike our three test cases, he did not report more

Table 5.22: Summary of results' accuracies for the three test cases

| Test Case | Accuracy (%) |
|---|---|
| Test Case One | 82.880 |
| Test Case Two | 100.0 |
| Test Case Three | 74.754 |
| Overall Average Accuracy | 85.878 |

than one test case and probably used a generated data because he proposed to collaborate with mobile networks.

## 5.8 Conclusion

For modelling subscribers' behaviour and detecting anomalies from call data, the DTS, the HGA and the ADS have been implemented as solutions. We applied our system to landline networks but it can easily be applied to mobile data as well. Our implementation addressed all of the research questions presented in Chapter One.

The implementation results of our DTS, which were used by the rest of our system, have validated the statement made by machine learning researchers [4]. Berka [4] said preprocessing of data can help to better understand the data that enters into machine learning algorithms. The DTS generated suitable training sets for the HGA to mine subscribers' profiles (BBNs). Moreover, the DTS is general in that it can preprocess datasets for other applications other than telecommunications.

The implementation results of the HGA, that mined individual BBNs, have shown variabilities in subscribers' behaviour and as a result, it has shown that the use of one model to group related users together in anomaly detection is questionable. Also, the simulation results of our HGA showed that it is good for every modelling technique to exhaust the solution space before termination rather than using convergence which can get stuck at local minima. Since we obtained good results in our evaluations, it encouraged our HGA to be applied to other applications [56].

The identification and the use of anomaly indicators as query nodes in our ADS implementations gave our system the capability to allow calls made to new destination numbers or existing destination numbers. We obtained very good overall detection accuracy, 85.878 %. Our first test case on the ADS gave reliable results, 82.880 %. The second test case was to differentiate two subscribers

which proved 100 % accurate. The third test case detected 74.754 % of known anomalies introduced into the call data. Also, the use of daily call detection is an added advantage which will reduce the lag time in the existing batch detection systems.

# Chapter 6

# Conclusions, Summary of Results and Future Research

In this dissertation, we have designed and implemented a Data Transformation System (DTS) for preprocessing datasets, a new genetic algorithm (HGA) for individual subscriber modelling and an Anomaly Detection System (ADS) for detecting anomalies in telecommunications call data. This concluding chapter will interpret our results, highlight our contributions, and state the future directions of this research.

## 6.1 Summary and Interpretation of Results

In Chapter Two, we described how anomalies are caused in telecommunications call data. Specifically, we reviewed the details of fraud in order for TSPs to understand bad practices in their networks, and provide insightful information to assist call analysts in making decisions when they detect suspicious call records. Related research in telecommunications anomaly detection has been described. These investigations enabled us to understand the strength and weaknesses of existing detection system so as to mitigate their problems.

In Chapter Three, we reviewed the background on probabilistic modelling and Bayesian networks. We used this knowledge to differentiate singly and multiply-connected networks. We described the relationships between Bayesian networks and machine learning. Various techniques of mining Bayesian network structures from data were discussed. We discussed our motivations for using

genetic algorithms.

In Chapter Four, we presented our overall research system model where we described the detail designs and functionalities of the DTS, the HGA and the ADS. We described the Hybrid Genetic Algorithm (HGA) that mines Bayesian networks from data using information theoretic measures and mathematical components. We discussed how the ADS uses BBN models to detect anomalies and we introduced the evaluation techniques used to measure the performance of our system.

In Chapter Five, we presented the implementation results and the empirical evaluations of our algorithms and models. Our results demonstrated that individual subscriber modelling using the HGA is better than existing methods that group related users together and also better than existing methods that pre-classify anomalies. This was presented as compared with Hollmen's [28] result above. With anomaly indicators, we also showed that our ADS effectively detects anomalies from call data with subscribers making calls to existing or new phone numbers using landline call data. Some of the daily detection results presented with low degrees of beliefs are examples of calls made to new destinations.

Our outcomes have shown that our experimental results on daily call detection, test cases one, two and three will provide solutions to the causes of anomalies stated in Chapter One. Specifically, bad debt risk and fraud can be trapped during anomaly detection using daily calls. Moreover, the graphs in Figures 5.28 and 5.29 shows the trends of minimising false alarms and maximising true detections. The results will assist call managers to make adaptive decisions and they can possibly tightened their networks.

## 6.2 Future Research

Even though we have established a proof of concept about Bayesian networks applied to telecommunications anomaly detection, our system is constrained with mining time, limited to large data but takes a long time for very massive datasets, and the capability to handle missing data has not been incorporated into this system. Our future research work is shown in the next paragraph.

As a future direction, we aim to incorporate incremental mining into our hybrid genetic algorithm so as to incrementally mine Bayesian networks from massive datasets. Also, there is need to investigate and fully explore more advanced fitness functions than the MDL to score Bayesian networks. This

will help to improve on the possible local convergence encountered on the optimization trends in Figures 5.2 to 5.18. We want to make the HGA more robust by exploiting the ADTree [44] speed power and incorporate other machine learning capabilities such as the handling of missing data using the Expected Maximisation (EM) algorithms.

Also, we believe that the improvement on real time detection of ADS from daily grouped records to real time detection per call will offer many advantages to the ADS; we aim to incorporate intelligent agents into our systems since we have designed them as decomposable systems; and we hope to collaborate with a mobile telecommunication service provider for further validation and evaluation of these new technologies.

## 6.3 Contributions to Knowledge

Even though, our HGA may take long to mine subscriber profiles from massive datasets, it eventually produces an optimal network. Our research has provided a general solution to mine individual subscriber models and to detect anomalies in telecommunications call data. It opens up many future research directions. Our research has made the following contributions:

- We constructed a new algorithm called the Hybrid Genetic Algorithm (HGA) to mine networks from datasets. The use of inner-loops that prevent the HGA network optimization from getting worse, the control of redundant offspring and the capability to exhaust the network solution space makes our HGA reliable.

- The HGA is not difficult to understand because it is a simple variant of classical genetic algorithms unlike other GA methods with complex designs, which are difficult to implement. The interactions of the HGA with other systems such as the DTS and the ADS show its capability as a decomposable system.

- We mined multiply-connected Bayesian networks with our HGA. Multiply-connected networks have more relationships in between the nodes which improves probabilistic reasoning.

- The variabilities of individual subscriber model as mined by HGA provide enough evidence to show that grouping related subscribers together and the use of pre-classified anomalies for detection system cannot be effective.

- The HGA mines networks from mixed or nominal datasets (which is done by most existing systems), as well as mining networks from numerical datasets that is not discretized.
- The use of primary and secondary anomaly indicators as query nodes give the ADS the capability to tolerate subscribers making calls to new destination numbers. This understanding avoids false alarms.
- The achievement of adaptive intelligence using Bayesian networks and the differential analysis in the ADS help to avoid false detections.
- The use of the degree of belief (or level of confidence) obtained from our ADS about suspicious calls will assist network carriers to make reliable decisions, and will reduce false alarm rates.

## 6.4 Concluding Remarks

The most important contribution of this research was the intelligence we achieved with the ADS. The intelligence of the ADS is a result of the following: modelling individual BBNs, incremental training of the BBNs, use of anomaly indicators, minimising approximations which include the use of mixed datasets (numerical and textual data).

The HGA successfully mined individual subscribers' models which make it suitable to applications in various academic areas such as [52], [56]. The HGA provided reliable evaluations for modelling similar networks from datasets in various domains.

From the empirical evaluation test cases, the good performance of the ADS provided an illustrative approach to addressing anomaly detection in telecommunications networks.

It is regrettable that we could not have access to mobile call data but we are fortunate that our landline data contains relevant attributes that are equivalent to the mobile data.

# Bibliography


[1] Adams, D. (1996). Thieves Answer Call for Mobile Phones. The Age, Page A3. May 25.

[2] Bay, D. & Schwabacher, M. (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule. *In proc. of 9th annual ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Pages 29-38.

[3] BayesiaLab (2005). http://www.bayesia.com/

[4] Berka, P. & Bruha, I. (1998). Discretization and Grouping Preprocessing Steps for Data Mining. *Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, Pages: 239 - 245.

[5] Bhargava, B., Zhong, Y. and Lu, Y. (2003). Fraud Formalization and Detection. *In Proceeding of Data Warehouse and Knowledge Management Conference (DaWak)*, Pages 330-339.

[6] Bounsaythip, C. & Rinta-Runsala, E. (2001). Overview of Data Mining for Customer Behaviour Modelling. *Technical Report*, TTE1-2001-18, VTT Information Technology, Finland.

[7] Brooks, T. & Davis, M. (1994). Are Your Phone Bills Fraud Free? Security Management, vol. 38, no. 4, Pages 67-68.

[8] Burge, P., Shawe-Taylor, J., Moreau, Y., Preneel, B. & Stoermann, C. (1997). Fraud Detection and Management in Mobile Telecommunications Networks. *Proceedings of the 2nd European Conference on Security and Detection*, IEE Conference publication, Pages 91-96, London.

[9] Cantu-Paz, E. (2001). Supervised and unsupervised discretization methods for evolutionary algorithms. *Workshop Proceedings of the Genetic and Evolutionary Computation Conference* (GECCO-2001), Pages 213-216.

[10] Cellular, O. (1994). Cellular Fraud Facts. *Proceedings of International Crime Stoppers Conference*, Hawaii, September, See also: Internet http://www.wireless101.com/new/fpf/fraud.htm.

[11] Chickering, D., Heckerman, D. & Meek, C. (2004). Large-Sample Learning of Bayesian Networks is NP-Hard. *The Journal of Machine Learning Research*, Volume 5, Pages: 1287 - 1330, MIT Press.

[12] Collings, D. (1999). Agent-based Customer Modeling. *Computing in Economics and Finance, Society for Computational Economics*, No 1352.

[13] Cozman, F. (2001). JavaBayes. *Bayesian Networks in Java*, University of Sao Paulo. http://www.cs.cmu.edu/ javabayes/Home/.

[14] BioTech Adventure. (2005). Crossover process. Oklahoma State University. http://biotech-adventure.okstate.edu/

[15] Delaney, P. (1993). Investigating Telecommunications Fraud. *Criminal and Civil Investigation Handbook*, 2nd ed., McGraw-Hill Inc. New York.

[16] Dimitris, M. (2003). Learning Bayesian Network Model Structure from Data. *PhD dissertation*, CMU-CS-03-153, School of Computer Science. Carnegie Mellon University.

[17] Eclipse (2003). IDE for Java. Sun Microsystems. http://www.eclipse.org/

[18] Engelbrecht, P. (2002). Computational Intelligence. John Wiley & Sons Inc. The Atrium, Southern Gate, Chichester, West Sussex PO198SQ, England.

[19] Fawcett, T. & Provost, F. (1997). Adaptive Fraud Detection. *Data Mining and Knowledge Discovery*, Kluwer Academic Publishers, Boston, CA. Vol 1, Pages 291-316.

[20] Federal Communications Commission (Fcc). (2005). Consumer and Governmental Affairs Bureau, 445 12th St. S.W., Washington. http://www.fcc.gov/cgb/consumerfacts/cellphonefraud.html.

[21] Frayne, M. (2006). Interconnect billing- Making Telecommunications Work for Africa, *Annual Telecommunications Report*, Intec Telecom Systems. http://www.connect-world.com/Articles/old-articles/MikeFrayne.htm

[22] Friedman, N. & Goldszmidt, M. (1998). Learning Bayesian Networks from Data. *Tutorial, American National Conference on Artificial Intelligence Madison*, AAAI Press, San Mateo, CA.

[23] Gell-Mann, M. (2001). The Quark and the Jaguar. *Adventures in the Simple and the Complex, Abacus inc. Lancaster, London WC2E7EN.*

[24] Georgiopoulos, M. & Anagnostopoulos, G. (2005). Experiments with Decision Tree Classifiers — Discretization of Numerical Attributes. *Combined Research and Curriculum Development (CRCD) in Machine Learning at University of Central Florida.*

[25] Gillian, D. (1999). The changing face of fraud : phone fraud. *3rd National Outlook Symposium on Crime in Australia : Mapping the Boundaries of Australia's Criminal Justice System Rydges Hotel, Canberra.*

[26] GSM Association. (2006). GSM World. http://www.gsmworld.com/history/page15.htm.

[27] Hofmann, R., & Tresp, V. (1996). Discovering Structure in Continuous Variables Using Bayesian Networks. *Advances in Neural Information Processing Systems Vol. 8 (NIPS 96), Pages 500-506, MIT Press*, Cambridge MA.

[28] Hollmen, J., Tresp, V., Taniguchi, M., & Haft, M. (1998). Fraud Detection In Communications Networks Using Neural And Probabilistic Methods. *Proceedings of the 1998 IEEE Int. Conf. in Acoustics, Speech and Signal Processing (ICASSP'98)*, Vol. 2, Pages 1241-1244.

[29] Hood, L, (1998). Agent based modelling, available at www.brs.gov.au/social-sciences/kyoto/hood2.html.

[30] Ioannis, T., Laura, B. & Constantin, A. (2004). A Novel Algorithm for Scalable and Accurate Bayesian Network Learning. *In 11th World Congress on Medical Informatics (MEDINFO)*, Pages 711-5, San Francisco, Califonia.

[31] Jaroszewick, S. & Simovici, D. (2004). Interestingness of Frequent Itemsets using Bayesian Networks as Background Knowledge. *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Pages: 178 - 186, ACM press.

[32] Jie, C., David, B., Weiru, L. (1997). Learning Belief Networks from Data, an Information

Theory Based Approach. *Proceedings of the Sixth International Conference on Information and Knowledge Management*, Pages: 325 - 331, ACM press.

[33] Josep, A. (2005). Incremental Methods for Bayesian Network Structure Learning. *AI Communications*, Volume 18, Pages 61-62, IOS Press, Nieuwe Hemweg 6B, 1013 BG Amsterdam.

[34] Knorr, E. & Raymond T. Ng, (1999). Finding Intensional Knowledge of Distance-Based Outliers. *Proceedings of the 25th International Conference on Very Large Data Bases*, Pages 211 - 222, Morgan Kaufmann Publishers Inc.

[35] Koivisto, M. & Sood, K. (2004). Exact Bayesian Structure Discovery in Bayesian Networks. *The Journal of Machine Learning Research*, Volume 5, Pages 549 - 573, MIT press.

[36] Kohavi, R., George, J. (1997). Wrappers for Feature Subset Selection. *Artificial Intelligence*, Vol. 97, Issue 1-2, Pages 273 - 324, Elsevier Science Publishers Ltd, Essex, UK.

[37] Kohavi, R. & Provost, F. (1998). Special Issue on Applications of Machine Learning and the Knowledge Discovery Process. *Journal of Machine Learning*, Pages 271-274, Kluwer Academic Publishers, Boston, Manufactured in the Netherlands.

[38] Kou, Y., Lu, C. Sirwongwattana, S. & Yo-Ping, H. (2004). Survey of Fraud Detection Techniques. *Networking, Sensing and Control, IEEE International Conference*, Page(s) 749-754 Vol.2, IEEE.

[39] Larranaga, P., Kuijpers, C., Murga, R. & Yurramendi, Y. (1996). Learning Bayesian Network Structures by Searching for the Best Ordering with Genetic Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, Pages 487-493.

[40] Lucas, P. (2001). Bayesian Models in Medicine. *The European Conference on Artificial Intelligence in Medicine (AIME'01) Cascais, Portugal.*

[41] Mackay, D. (2003). Information Theory, Inference and Learning Algorithms. Cambridge Press.

[42] Maheshkumar, S., Neill D. & Moore, A. (2005). Detecting anomalous patterns in pharmacy retail data. *Proceedings of the KDD workshop on data mining methods for anomaly detection.* http://rods.health.pitt.edu/LIBRARY/2005

[43] Michael, H., Lambert, D., Pinheiro, C., & Sun, D. (2002). Detecting Fraud in the Real World. *Handbook of massive data sets*, Pages 911 - 929, Kluwer Academic Publishers.

[44] Moore, A., Lee M. (1998). Cached Sufficient Statistics for Efficient Machine Learning with Large Datasets. *Journal of Artificial Intelligent Research* Vol. 8, Pages 67-91.

[45] Moore, A. Iterative Improvement Search: Hill-Climbing, Simulated Annealing, WALK-SAT, and Genetic Algorithms. School of Computer Science, Carnegie Mellon University. http://www.cs.cmu.edu/~awm/.

[46] Murphy, K. (2002). Dynamic Bayesian Networks. *Representation, Inference and Learning.* PhD thesis, UC Berkeley, Computer Science Division.

[47] Biology Department. (2005). Mutation Process. Brooklyn College. http://academic.brooklyn.cuny.edu/biology/bio4fv/page/molecular

[48] Myers, J., Kathryn, L. & DeJong, K. (1999). Learning Bayesian Networks from Incomplete Data using Evolutionary Algorithms. *Proceedings of the Genetic and Evolutionary Computation Conference*, Vol. 1, Pages 458-465, Morgan Kaufmann Publishers.

[49] Newman, D.J., Hettich, S., Blake, C.L. & Merz, C.J. (1998). UCI Repository of Machine Learning Databases, Irvine, CA: University of California, Department of Information and Computer Science. http://www.ics.uci.edu/~mlearn/MLRepository.html

[50] Nilsson, N. (1998). Artificial Intelligence. *A new synthesis*, first edition, San Fransisco USA. Morgan Kaufmann Publishers.

[51] Osunmakinde, I. O. & Potgieter, A. (2005). Telecommunications Fraud Detection using Bayesian networks. *Project of the African Institute for Mathematical Sciences (AIMS)*, South Africa.

[52] Osunmakinde, I. O. & Potgieter, A. (2006). Agent-Based Behavioural Modelling for Anomaly Detection in Call Data from Telecommunication Networks. *Proceedings of Southern African Telecommunications Networks and Applications Conference* (SATNAC), Pages 8-9.

[53] Pearl, J. (1988). Probabilistic reasoning in Intelligent systems. *Networks of Plausible Inference, Morgan Kaufmann Publishers.*

[54] Pearl, J. (2000). Causality, Models, Reasoning, and Inference. *Cambridge University Press*, Page 384.

[55] Pequeno, K. (1997). Real time fraud Detection: Telecoms next big step. Telecommunications (America Edition), Vol. 31, No. 5, Pages 59-60.

[56] Potgieter, A., April, K.A., Cooke, R.J.E. & Osunmakinde, I. O. (2006). Understanding Social Complexity, Journal of Emergence: Complexity and Organization (E:CO), Accepted for Publication.

[57] Rosset, S., Murad, U., Newmann, E., Idan, Y. & Pinkas, G. (1999). Discovery of fraud rules for telecommunications challenges and solutions. *In Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Pages 409-413. ACM Press.

[58] Russel, S. & Norvig, P. (2003). Artificial Intelligence. *A Modern Approach*, 2nd edition, Prentice Hall Series Inc. New Jersey 07458.

[59] Sapa, (2005). Crime Initiative Future, Cape Times. 14 April, Page 3.

[60] Sequeira, K. & Zaki, M. (2002). Anomaly-based Data Mining for Intrusions. *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Pages 386-395.

[61] Shannon, T. & Zwick, M. (2002). Directed Extended Dependency Analysis for Data Mining. *The Journal of Kybernetes*, Vol. 33, Pages 973-983, Emerald Group Publishing Limited.

[62] Spirtes, P., Glymour, C. & Scheines R. (2000). Causation, Prediction, and Search. MIT press, second edition, Cambridge, Massachussets.

[63] Telecoms Fraud - Africa. (2006). Summary: Understanding and Implementing Practical Strategies To Improve Your Capabilities to Effectively Detect, Manage and Reduce Fraud In Your Network. http://www.iir-events.com/IIR-conf/Telecoms/EventView.aspx?EventID=814

[64] Thomas W., Colin K. (2004). Gnuplot. http://www.gnuplot.info/

[65] Voelker, M. (2006). Staying a Step Ahead of Fraud. *Intelligent Enterprise Magazine,http* : $//www.intelligententerprise.com/showArticle.jhtml?articleID = 191901987$

[66] Wachsmuth, S. & Gerhard, S. (2002). Bayesian networks for speech and image integration, *Eighteenth national conference on Artificial intelligence*, Pages: 300 - 306, American Association for Artificial Intelligence Publishers.

[67] Wai, L., Man, L., Kwong, S. & PoShun, N. (1999). Using Evolutionary Programming and MDL Principle for data mining of Bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, Pages 174-178, IEEE Computer Society.

[68] Walters, D. & Wilkinson, W. (1994). Wireless Fraud, Now and in the Future: A view of the problems, some solutions. Mobile Phone News, October 24, Pages 4-7.

[69] Weka. (1993). Machine Learning Software. University of Waikato. http://www.cs.waikato.ac.nz/~ml/weka/

[70] William, H., Haipeng, G., Benjamin, P. & Julie, S. (2002). A Permutation Genetic Algorithm For Variable Ordering In Learning Bayesian Networks From Data. *Proceedings of Genetic and Evolutionary Computation Conference*, Pages 383-390, Morgan Kaufmann Publishers Inc.

# Appendix A: Implementation Designs

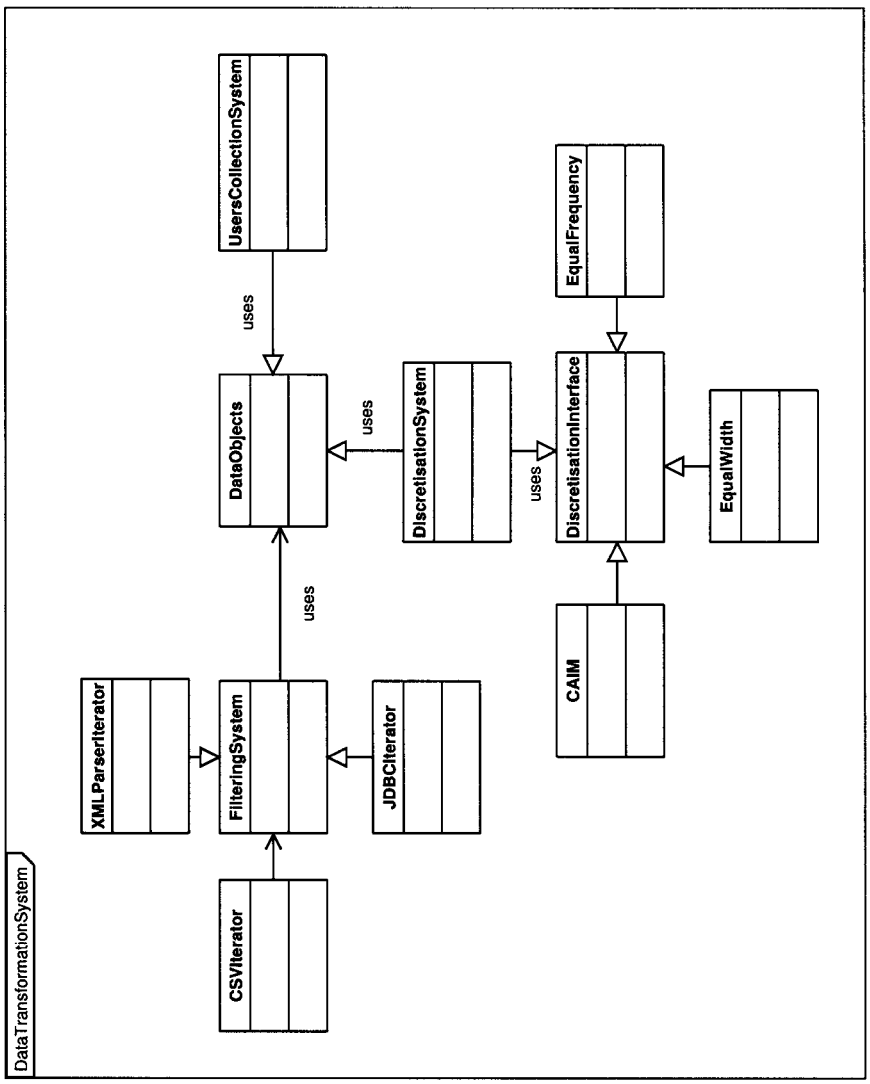


**Figure A.1: Class Diagram of Data Transformation System (DTS)**

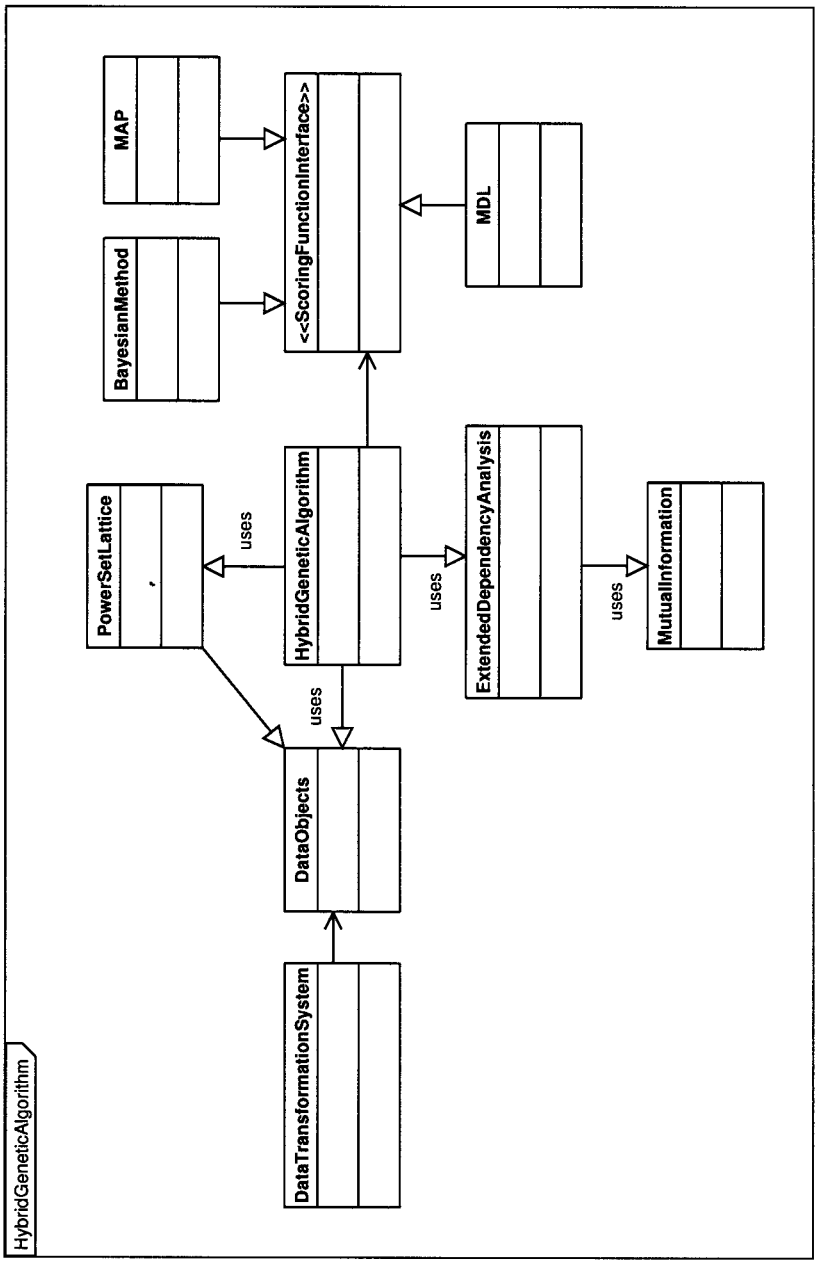


**Figure A.2: Class Diagram of Hybrid Genetic Algorithm (HGA)**

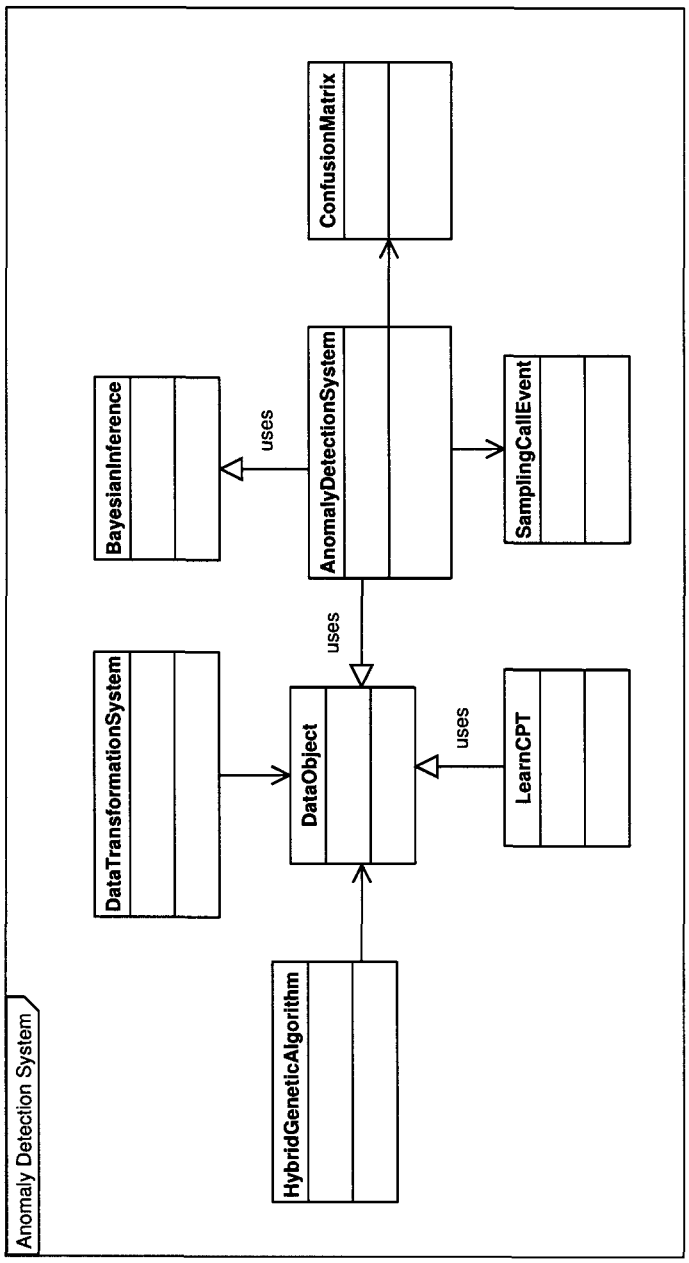


**Figure A.3: Class Diagram of Anomaly Detection System (ADS)**

# Appendix B: Implementation Results

## EXPERIMENTAL RESULTS OF ANOMALY DETECTION ON DAILY CALLS IN CURRENT CALL DATASETS (CCD)

### ADS Results for Subscriber 3

Table B.1 shows the subset of the HCD for subscriber *145571179*, which was used by our HGA to mine the Behavioural Bayesian Network in Figure 5.11.

The ADS used the model in Figure 5.11 and detected the daily call records as shown in Table B.2. The table shows the call detection results for the phone calls that were made on *Thursday*. The first call event in the table was detected as *anomalous* with the degree of belief of 95.890 %. As was the case with the previous two subscribers, this should trigger the TSP to take the appropriate actions.

### ADS Results for Subscriber 4

The subset of the HCD for subscriber *145571042*, is shown in Table B.3. The table was used by the HGA to mine the BBN in Figure 5.7.

Also, the ADS used the model in Figure 5.7, and consequently detected daily calls as shown in Table B.4. The table shows the call detection results for the phone calls that were made on *Friday*. All the detection results in the table also take their usual interpretations as stated above.

### ADS Results for Subscriber 5

The subset of the HCD for subscriber *145571030*, is also shown in Table B.5. The HGA used the call records and produced the network in Figure 5.17. Furthermore, the ADS used the model in Figure 5.17, acted upon the corresponding daily call records, and detected daily calls as shown in Table B.6. Observe that, the detected calls were made on *Wednesday*.

Table B.1: Observed training set for subscriber 145571179

| x1 | x2 | x3 | x4 | x5 | x6 | x7 |
|---|---|---|---|---|---|---|
| Thursday | 261-305 | 5.41E+11 | International | Argentina | 25.174799999999998-26.43354 | X |
| Friday | 0-44 | 145928463 | National | 0-50Km | 0.0-1.25874 | P |
| Saturday | 305-349 | 1164622700 | National | >50Km | 1.25874-2.51748 | X |
| Sunday | 44-87 | 123001000 | National | >50Km | 0.0-1.25874 | X |
| Tuesday | 44-87 | 14123695071 | International | USA | 1.25874-2.51748 | X |
| Wednesday | 0-44 | 117124000 | National | >50Km | 0.0-1.25874 | X |
| Monday | 0-44 | 145557056 | National | 0-50Km | 0.0-1.25874 | P |
| Monday | 87-131 | 145555283 | National | 0-50Km | 0.0-1.25874 | P |
| ,, | ,, | ,, | ,, | ,, | ,, | ,, |

**Legend**: x1 = Call-date, x2 = Duration, x3 = Destination-no, x4 = Destination-net, x5 = Location, x6 = Call-cost, x7 = Peak/off-peak

Table B.2: ADS Detects Current Calls for subscriber 145571179

| Daily Call Events |
|---|
| [Thursday, 0-44, 183810180, National, >50Km, 0.0-1.25874, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 95.890 % |
| [Thursday, 0-44, 145928463, National, 0-50Km, 0.0-1.25874, P]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 95.419 % |
| [Thursday, 392-436, 145521261, National, 0-50Km, 1.25874-2.51748, P]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 97.209 % |
| [Thursday, 0-44, 145556040, National, 0-50Km, 0.0-1.25874, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 91.780 % |
| ,, |

Table B.3: Observed training set for subscriber 145571042

| x1 | x2 | x3 | x4 | x5 | x6 | x7 |
|---|---|---|---|---|---|---|
| Wednesday | 0-44 | 119388443 | National | >50Km | 0.0-1.25874 | P |
| Wednesday | 0-44 | 118735119 | National | >50Km | 0.0-1.25874 | P |
| Thursday | 0-44 | 18593127172 | International | USA | 0.0-1.25874 | X |
| Friday | 0-44 | 117124000 | National | >50Km | 0.0-1.25874 | X |
| Saturday | 0-44 | 145928463 | National | 0-50Km | 0.0-1.25874 | X |
| Sunday | 44-87 | 128040300 | National | >50Km | 1.25874-2.51748 | P |
| Monday | 0-44 | 145521517 | National | 0-50Km | 0.0-1.25874 | X |
| Tuesday | 131-174 | 126644157 | National | >50Km | 0.0-1.25874 | X |
| ,, | ,, | ,, | ,, | ,, | ,, | ,, |

**Legend**: x1 = Call-date, x2 = Duration, x3 = Destination-no, x4 = Destination-net, x5 = Location, x6 = Call-cost, x7 = Peak/off-peak

Table B.4: ADS Detects Current Calls for subscriber 145571042

| Daily Call Events |
|---|
| [Friday, 784-828, 112777000, ISPNational, >50Km, 11.32866-12.587399999999999, P]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 44.019 % |
| [Friday, 0-44, 145928463, National, 0-50Km, 0.0-1.25874, P]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 68.847 % |
| [Friday, 174-218, 116368113, National, >50Km, 2.51748-3.77622, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 28.287 % |
| [Friday, 0-44, 437353335, National, >50Km, 0.0-1.25874, P]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 95.207 % |
| [Friday, 44-87, 145966727, National, 0-50Km, 0.0-1.25874, P]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 96.136 % |
| [Friday, 349-392, 2603321122, International, Zambia, 16.36362-17.62236, P]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 97.604 % |
| ,, |

Table B.5: Observed training set for subscriber 145571030

| x1 | x2 | x3 | x4 | x5 | x6 | x7 |
|---|---|---|---|---|---|---|
| Wednesday | 0-44 | 123001000 | National | >50Km | 0.0-1.25874 | X |
| Thursday | 0-44 | 123001000 | National | >50Km | 0.0-1.25874 | X |
| Friday | 44-87 | 145970789 | National | 0-50Km | 0.0-1.25874 | X |
| Saturday | 87-131 | 145970789 | National | 0-50Km | 0.0-1.25874 | X |
| Sunday | 87-131 | 415077777 | National | >50Km | 0.0-1.25874 | X |
| Tuesday | 0-44 | 117124000 | National | >50Km | 0.0-1.25874 | X |
| Monday | 44-87 | 123001000 | National | >50Km | 0.0-1.25874 | P |
| ,, | ,, | ,, | ,, | ,, | ,, | ,, |

**Legend**: x1 = Call-date, x2 = Duration, x3 = Destination-no, x4 = Destination-net, x5 = Location, x6 = Call-cost, x7 = Peak/off-peak

Table B.6: ADS Detects Current Calls for subscriber 145571030

| Daily Call Events |
|---|
| [Wednesday, 0-44, 118875553, National, >50Km, 0.0-1.25874, X]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 57.598 % |
| [Wednesday, 436-479, 152905429, National, >50Km, 2.51748-3.77622, X]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 57.598 % |
| [Wednesday, 0-44, 123001000, National, >50Km, 0.0-1.25874, X]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 95.511 % |
| [Wednesday, 44-87, 145526000, National, 0-50Km, 0.0-1.25874, P]<br><br>**Call Detection**: Anomaly Call Suspected<br>**Degree of Belief**: 23.026 % |
| [Wednesday, 44-87, 19546005957, International, USA, 2.51748-3.77622, X]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 89.634 % |
| [Wednesday, 44-87, 117124000, National, >50Km, 0.0-1.25874, X]<br><br>**Call Detection**: Regular Call Suspected<br>**Degree of Belief**: 97.755 % |
| ,, |

## TEST CASE ONE: ANOMALY DETECTION WITHIN A SUBSCRIBER'S PROFILE

### ADS Accuracies for Subscriber 3

Table B.7 shows the expected call data from the test set using the cross validation technique. The confusion matrix in respect of subscriber 145571179 is also presented in Table B.8.

Furthermore, ADS accuracy in respect of subscriber 145571179, is **77.358 %**. This is the true positive rate while the false alarm rate is **22.642 %**

Table B.7: Expected Call Detection Results for subscriber 145571179

| | Total |
|---|---|
| Regular Calls | 53 |
| Anomaly Calls | 0 |

Table B.8: Confusion Matrix Results for subscriber 145571179

| Expected; Predicted | Regular | Anomaly |
|---|---|---|
| Regular: | 41.0 | 12.0 |
| Anomaly: | 0.0 | 0.0 |

### ADS Accuracies for Subscriber 4

Table B.9 also shows the expected call data from the test set. Table B.10 shows the confusion matrix in respect of subscriber 145571024. Also, the ADS accuracy for subscriber 145571024 is **97.633 %**. This is the true positive rate while the false alarm rate is **2.367 %** .

Table B.9: Expected Call Detection Results for subscriber 145571024

| | Total |
|---|---|
| Regular Calls | 169 |
| Anomaly Calls | 0 |

Table B.10: Confusion Matrix Results for subscriber 145571024

| Expected; Predicted | Regular | Anomaly |
|---|---|---|
| Regular: | 165.0 | 4.0 |
| Anomaly: | 0.0 | 0.0 |

**ADS Accuracies for Subscriber 5**

The expected call data from the test set is also shown in Table B.11 using the cross validation technique. The confusion matrix generated from our implementations, in respect of subscriber 145571030 is set in Table B.12.

The ADS accuracy for subscriber 145571030 is **93.846 %**. This is the true positive rate while the false alarm rate is **6.154 %**

Table B.11: Expected Call Detection Results for subscriber 145571030

| | Total |
|---|---|
| Regular Calls | 65 |
| Anomaly Calls | 0 |

Table B.12: Confusion Matrix Results for subscriber 145571030

| Expected; Predicted | Regular | Anomaly |
|---|---|---|
| Regular: | 61.0 | 4.0 |
| Anomaly: | 0.0 | 0.0 |

# Appendix C: List of Publications

The following academic papers were published from this dissertation:

1. Osunmakinde, I. O. & Potgieter, A. (2006), "Agent-Based Behavioural Modelling for Anomaly Detection in Call Data from Telecommunication Networks", Southern African Telecommunications Networks and Applications Conference (SATNAC), 2006.

   **REVIEWERS' COMMENTS**

   - The paper as a whole is interesting.
   - The paper is based on an interesting idea and seems technically sound.
   - This paper has the potential to be a great paper.

2. Potgieter, A., April, K.A., Cooke, R.J.E. & Osunmakinde, I. O. (2006). Understanding Social Complexity, Journal of Emergence: Complexity Organization (E:CO), Accepted for Publication, 2006.