QUANTITATIVE METHODS TO SEGMENT TARGET BUSINESSES IN A TELECOMMUNICATION ENVIRONMENT FOR SALES OPTIMISATION

Louis F. Dannhauser

Supervisor: Dr Joke Bührmann

Johannesburg, 2020

A dissertation submitted to the Faculty of Engineering and the Built Environment,
University of the Witwatersrand
in fulfilment of the requirements for the degree of Master of Science.

Dedication

I dedicate this research to:

God, the ultimate Father and inspiration, who taught me to earn respect through my work (1 Thes. 4:11-12), I praise Your name.

Jan Daniel Dannhauser, who was the most tolerant and accepting father on earth, I miss you.

Karen, my wife, who believed in me throughout with patience and encouragement, I love you.

Andrew, my friend who trusted me, even when I did not always trust him, I thank you sincerely.

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the Degree of Master of Science (in Industrial Engineering) at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.

Louis François Dannhauser

15 April 2021 at Johannesburg

Abstract

The dissertation focuses on studying various quantitative response-based, multivariate analysis (MVA) methods to segment a potential enterprise business to business (B2B) customer database for a telecommunications (telecom) company. Segmentation in the telecom industry is of high interest due to the competitive nature and the complexity of the sales environment. MVA methods fall into two categories, namely interdependence and dependence analysis. The interdependence analysis methods evaluated were *k*-means clustering (KMC) and particle swarm optimisation (PSO). With regard to dependence analysis, chi-square automatic interaction detection (CHAID) and artificial neural networks (ANN) were evaluated.

Applying a method in a business environment were assessed through testing some hypotheses and answering applicable research questions.

The quantitative evaluation of the interdependent and dependent methods were done in the form of test runs to measure quality of output and speed of processing. Based on the results, guidelines and recommendations were identified for business segmentation through application of the methods.

Acknowledgements

I wish to acknowledge the following persons or institutions for their contributions towards this thesis:

- Dr Joke Bührmann, my supervisor. Thank you for your endless patience, guidance, motivation and encouragement
- The enterprise business and billing teams in Tanzania, who provided data for the research
- This research was supported by a bursary from Vodacom (Pty) Ltd.

Contents

List of Figures

List of Tables

List of Acronyms

AD	Automatic Differentiation
AHP	Analytical Hierarchy Process
AI	Artificial Intelligence
AID	Automated Interaction Detection
ANN	Artificial neural network
ARPU	Average revenue per unit
B2B	Business to business
BP	Backpropagation
CA	Cellular Automata
CB	Covariance-based
CF	Collaborative filtering
CV	Critical value
CART	Classification and regression tree
CHAID	Chi-square automatic interaction detection
CNC	Computer numerically control
CNN	Convolutional neural network
CoDANN	Concepts of Design Assurance for Neural Networks
CPU	Central processing unit
DF	Degrees of freedom
DL	Deep Learning
DNN	Deep neural network
DOE	Design of experiments
EASA	European Union Aviation Safety Agency
ESS	Explained sum of squares
GDP	Gross domestic product
GPU	Graphics processing unit
GMDH	Group method of data handling
IID	Independent and identically distributed

ICT	Information and Communication Technologies
IOT	Internet of things
IT	Information technology (computers)
IT	Inferior temporal (neuroscience)
KMC	K-means clustering
LMS	Least mean square
LTP	Long-term potentiation
LSTM	Long short-term memory
M2M	Machine-to-machine
MAD	Michigan Algorithm Decoder
MAID	Multivariate extension of AID
MAE	Mean absolute error
MANOVA	Multivariate analysis of variance
MDS	Multidimensional scaling
MIP	Marketing investment planning
ML	Machine Learning
MLP	Multilayer perceptron
MRM	Multiple regression on distance matrices
MRT	Multivariate regression tree
MSE	Mean squared error
MSISDN	Mobile Station International Subscriber Directory Number
MVRM	Multivariate regression model
MVA	Multivariate analysis
NAP	Neural abstraction pyramid
NP-hard	Nondeterministic polynomial time hard
PCA	Principle component analysis
PCM	Pulse-code modulation
PLR	Polytomous logistic regression
PLS	Partial least squares

PG	Percent good
PSO	Particle swarm optimisation
REBUS	Response-based unit segment detection
RFM	Recency, Frequency, and Monetary
RMSE	Root mean squared error
RNN	Recurrent neural network
RSS	Residual sum of squares
SAM	Serviceable addressable market
SEM	Structural equation modelling
SEO	Search Engine Optimisation
SI	Swarm intelligence
SIC	Standard Industrial Classification
SOM	Self organising map
SOFM	Self organising feature map
TAM	Total addressable market
TDNN	Time delay neural network
THAID	Theta automatic interaction detection
TSP	Traveling salesperson problem
TSS	Total sum of squares
XAID	Exploratory automatic interaction detection

1. INTRODUCTION

Observing the growth in business to business (B2B) services over the past three decades, one important aspect comes to the fore. B2B companies from various industries like financial services, pharmaceuticals, energy provision and construction equipment leasing, have embraced the use of a marketing strategy (Baldock, 2005). These companies would describe marketing as a way to identify, attract and retain customers as profitably as possible. According to Baldock (2005) their marketing investment planning (MIP) considers all aspects to achieve this. As part of their MIP most companies gain enough market knowledge to predict or anticipate which customer segments are their most profitable. This is also confirmed in the Openview e-book where business leaders highlight that scaling a business is best not left to guesswork or instinct (Nguyen, 2012). This knowledge is not always successfully implemented due to a number of challenges, a lack of resources being the least. The major drawback, resulting in a suboptimal marketing strategy, is the lack of knowledge about the target market.

The target market for any B2B product or service is not one homogeneous mass. Rather, it can usually be divided into several distinct groups (Willan, 2014). Without a deep understanding of how a company's best current customers are segmented, a business often lacks the market focus needed to efficiently allocate and spend its precious human and capital resources (Nguyen, 2018). Rather than only evaluating segments, the focus has moved quickly to defining a segment, and finding the members of the segment so they can be targeted with direct marketing communications (Wyner, 1995). Response models such as discriminant analysis, classification and regression trees (CARTs), the Recency, Frequency, Monetary (RFM) method and Logistic Regression, have been used for this purpose. However, if there is considerable customer heterogeneity in the database, the models can be potentially misleading (Suh *et al.,* 1999). To reflect this heterogeneity, researchers have introduced ways to combine two or more methods. Years

later, Chernev (2007), from the Kellog school of management, showed how segmentation has become an integral part to prepare for a market positioning strategy. Baldock (2005) reported that pattern-recognition analytics can be used to get detail on the precise makeup of the complete customer experience, and seek to replicate that going forward. Currently, modelling the complete customer experience is rapidly becoming the norm. In case studies from industry, some business have already exploited the powerful pattern recognition capabilities of neural networks to shift funds towards more effective communication channels and to identify interactions for customer repeat purchase (Baldock, 2005). These are all built on effective segmentation strategies and proper analytical segmentation techniques.

This dissertation presents therefore a review of using these techniques as part of quantitative methods to segment target businesses for sales optimisation. The telecommunications industry was chosen as business where the research was conducted, due to the ever changing market of this industry and its impact on B2B services.

In order to convey the results of the research clearly, the subtle differences in the various terms used for segmentation, and its relation to target markets, are point out in the first section of this chapter. Note that one of the industry standard abbreviations for telecommunications is telecom. For brevity, this shortened reference to telecommunications are used, where applicable, in the rest of this dissertation.

1.1 Definitions

Throughout the research, mention will be made of a few types of entities common to a telecommunications business environment. To distinguish their meaning, definitions are given next in revised form (Collins, 2014):

- Company: An organisation of individuals conducting a commercial or industrial enterprise or business endeavour. Examples of companies are corporations, partnerships, associations, or small businesses.
- Customer: A legal entity that buys goods or services the same way a consumer would. In this research it will be a company trading with, or entering into an agreement with a telecom service provider to receive services that the company pays for.
- Employee: A person who is hired to work for another or for a company, business or firm in return for payment
- Subscriber: In this research, an employee of a company subscribing to telecommunication services as part of the agreement between the company and a telecom service provider, is called a subscriber to the telecom service provider.
- Line: In telecommunications, this is defined as a link between two points carrying communications. This communication line can be an invisible microwave, or a fibre-optic cable, or a copper cable, or a combination of these. A subscriber can get access to one or more communication lines as part of the services received from a telecom service provider. A communication access granted to the subscriber is referred to as a line.

Numerous definitions of segmentation may be found in publications and online. The Zyxo BlogSpot gives some very neat and concise definitions of variations of the term (Zyxo, 2010):

- Segmentation is the process of dividing real world entities (be it customers, images or characteristics) into groups, based on predefined boundaries.
- Customer segmentation is the practice of dividing a customer base into groups of entities that are similar in specific ways.

The customer base used for customer segmentation forms part of a market. A market is a place, an arrangement of organisations, procedures, social relations or infrastructures, where parties engage in exchange of goods and services.

Expanding on the above definitions, Kotler and Keller (2006) describe a market segment as a subset of a market. This subset is comprised of people or organisations sharing one or more characteristics that cause them to demand similar products and/or services based on qualities of those products, such as price or function.

A true market segment meets all the following criteria:

- distinct from other segments (different segments have different needs)
- homogeneous within the segment (exhibits common needs)
- responds similarly to a market stimulus, and can be reached by a market intervention (Kotler and Keller, 2006).

The market that is segmented, according to the definitions above, is also referred to as the total addressable market (TAM) or total available market, with reference to the revenue opportunity for a product or service. Blank and Dorf (2012) described a serviceable addressable market (SAM) or 'served available market' as the part of the TAM that can actually be reached to use a product or service. In the telecom industry, the SAM is a market with access to the telecommunication service provider's existing infrastructure. A target market is a group of customers within a SAM, at which a business

aims its marketing efforts and resources, and is a subset of the total market for a product or service (Blank and Dorf, 2012).

Target marketing is a marketing technique that targets a group of potential customers with specific characteristics (that is, the portion of the TAM that can be reached and at which marketing efforts are aimed by a business). Target market segmentation (a more specific term for customer segmentation) is the division of potential customers in a given market into discrete groups (Nguyen, 2012).

Frank Wyman (2005) encapsulated all the above in the definition of segmentation for targeting customers to provide the most client-actionable strategy:

"The dividing of a market's customers into subgroups in a way that optimises the firm's ability to profit from the fact that customers have different needs, priorities, and economic levers."

1.2 Research Context

The division of potential customers into mutually exclusive groups relies on information about the customer, derived from firmographic data. Firmographics (also known as firm demographics) is a set of characteristics available to a company for identifying prospective customer organisations. Firmographic variables relate to organisations in the same way as demographics relates to people (Smith, 2013).

Any segmentation method should have the following characteristics (Stuntebeck, 2012):

- Solid analysis of firmographic data;
- Clarity on the method used;
- Proper management of the segmentation process; and
- Buy-in from the sales management teams.

The best segmentation method involves performing an analysis of this data and then summarising it. Traditionally, sales departments have not operated with any formally defined sales strategies or processes. In many

cases, each salesperson develops their own non-documented sales approach with few metrics in place to measure performance (Yan *et al.*, 2015). At best, a qualitative or non-quantitative approach is usually followed, leveraging various customer attributes conceptualised through conversations with business stakeholders and customer focus groups to gather pointed data. This information represents consumer experiential behaviour. Analysts assign subjective segments for targeted campaign treatments through documented business rules for segmentation.

1.3 Research Motivation

In the traditional segmentation approach, a focused and effective sales strategy to address the target market is lacking. Market segmentation would play a crucial role in establishing such a target market strategy, which would lead to the development of an effective business strategy (Simkin, 2008). Although segmentation through business rules would lead to better results towards a target market strategy than a non-documented sales approach, the results would be suboptimal. With the current trend towards data-driven marketing, there are much better ways to achieve a market strategy goal, by using analytical or quantitative segmentation analysis (Grover, 2016).

Since marketing became a discipline on its own, some statistical methods have been used to segment the consumer-driven market based on demographic, geographic, economic and psychological parameters. However, B2B target market segmentation is based on more recently developed metrics. Stuntebeck (2012) and Nguyen (2012) gave a relevant and practical basis for business segmentation. Their proposed guidelines, in conjunction with relevant practical considerations, may still be implemented in business and telecom industries today.

However, both authors highlighted the fact that further review and testing are needed to enhance the process. Specific and continuous analysis of the segmentation methods used in a given industry will shed more light on the best approach to a segmentation scheme.

Due to its competitive nature, the telecom industry is an industry that performs best in applying the most efficient method for B2B market segmentation. It is a fact that telecom market segmentation is crucial for winning the intense competition for telecom enterprise business (Sheng and Xu, 2006). However, due to the complexity of enterprise business in the telecom industry, enterprise segmentation has been neglected, in favour of product development. Although literature does exist, there are very few practical guidelines on customer-focused B2B market segments in this industry (Simkin, 2008).

1.4 Problem Statement

The purpose of this research is to compare different quantitative methods for segmenting a telecom B2B target market. The most practical methods were analysed and guidelines recommended for business segmentation in a specific telecom company in Africa.

1.4.1 Research objectives

The objectives of the study are:

1. Review the analytical methods from literature that will be best for grouping data to identify segments.
2. Evaluate which of these methods may be applied to B2B target market segmentation. For effective sales targeting, a robust analytical method is proposed.
3. Analyse the appropriate method/s, using real-world enterprise market data and finding practical segments for use in a telecom company.
4. Propose factors to consider for the practical implementation of the chosen method/s to segment enterprise customers in a specific telecom environment.

1.4.2 Research hypotheses

If the quantifiable parameters that describe a business are used, a method can be found to classify the business as part of a group of businesses with similar properties. The hypotheses tested on this method are:

1. This method is quantitative, repeatable and can be used to reduce or eliminate manual intervention to classify businesses.
2. The chosen method is practically useful in the telecom industry for targeting enterprise customers.
3. As a result, each business with similar properties (or in the same segment) can be targeted through the appropriate sales channels.

1.4.3 Research question

The principal question to answer in the research can be formulated as:
How do the various techniques mentioned in this research compare, for market segmentation on an enterprise customer base of a telecommunications company in Africa?

1.4.4 Scope

The literature study focuses on the following areas for clustering, pattern recognition and grouping:

a. Interdependence analysis methods
b. Dependence analysis methods.

1.4.5 Limitations

Focus of the study is on methods previously considered or researched for customer segmentation, in order to show how these can be implemented effectively in a telecom environment with business customers. Hence, the analysis of the appropriate method for segmentation is limited to the following quantitative methods:

a. K-means clustering or KMC
b. Particle Swarm Optimisation or PSO
c. Chi-square Automatic Interaction Detection or CHAID
d. Artificial Neural Networks or ANN.

A number of entities, in the form of firmographic data for companies of various sizes, comprising up to 22 measurable parameters, were considered. For the assessment, 3 362 companies with up to 900 000 subscribers out of 2 000 000 employees were analysed. This aligned with statistical sample sizes, while still being sufficient to demonstrate the effectiveness of the quantitative methods used for analysis.

Note that only available market and customer data of a telecom company in one specific African country outside South Africa was used.

1.4.6 Structure of research

Segmentation methods in this research are based on multivariate analysis (MVA) techniques in two categories, namely interdependence and dependence analysis. In Chapter 2, which encompasses a literature review on interdependence analysis methods, some of the methods widely used today without reliance on external factors are described. Methods relying on external factors, i.e. dependence analysis methods, are reviewed for their relevance as segmentation methods in Chapter 3. The methodology used for data preparation and the analysis of the abovementioned methods is discussed in Chapter 4. The analysis of the data and methods with results is explained in Chapter 5. The conclusion and final comments on choosing a quantitative method for segmentation are given in Chapter 6.

Extracts from the datasets used for analysis are listed in Appendix A, together with input variables per test for each method. Detailed results in the form of tables or graphic illustrations are given in Appendix B. Source code or description of components used for the algorithms are presented in Appendix C.

2. LITERATURE REVIEW: INTERDEPENDENCE ANALYSIS METHODS

Interdependence analysis refers to a subset of multivariate segmentation techniques that group customers based on similar characteristics (Chulis, 2012). The most popular interdependent or unsupervised methods are the group of clustering methods.

One group of clustering methods is hierarchical clustering, which repeatedly links pairs of clusters until every data object is included in the hierarchy. Hierarchical clustering methods may be either agglomerative or divisive, involving a bottom-up or top-down approach, respectively. Agglomerative hierarchical methods start with each customer in their own cluster, which is then merged with other clusters based on inter-cluster distances (Jain and Dubes, 1998). Divisive methods start with all customer points in one cluster, and incrementally divide the points into an increasing number of clusters (Bührmann, 2016).

Latent class methods, on the other hand, represent a structural equation modelling (SEM) approach that uses probability modelling to maximise overall fit-to-find groups in datasets of multivariate categorical data. Hierarchical and latent class clustering methods would require that research be wholly dedicated to them, due to the many variations and computational complexity, and latent variables not observed.

A two-step clustering method to segment telecom customers, using *k*-means and survival character-based analysis, were found in literature (Chen *et al.,* 2007). The nature of survival analysis is projection of time, analysing the time before an event, e.g. churn, occurs, as opposed to segmenting a population or market at a given time. This research assesses two non-hierarchical methods, where the relationships between clusters are undetermined, namely *k*-means clustering and PSO.

2.1 K-means Clustering (KMC)

The KMC method is one of a group of algorithms called partitioning methods. The problem with partitional clustering may be formally described as: Given n objects in a d-dimensional metric space, determine a partition of the objects into k groups or clusters, such that the objects in a cluster are more similar to each other than to objects in different clusters (Jain and Dubes, 1988).

The solution to this problem is straightforward:

Select a clustering criterion; then, for each data object, select the cluster that optimises the criterion. The KMC algorithm initialises k clusters by arbitrarily selecting one object to represent each cluster. Each of the remaining objects is assigned to a cluster and the clustering criterion is used to calculate the cluster mean. These means are used as the new cluster points and each object is reassigned to the cluster to which it is most similar. This continues until there is no longer a change when the clusters are recalculated.

The basic algorithm is shown next, as presented by Hamilton (2012) for knowledge discovery in databases and by Gorban and Zinovyev (2008) as k-means principal points.

The basic k-means clustering algorithm

Input: The number of cluster k and a dataset X containing n objects.

1. Select k clusters arbitrarily from X.
2. Initialise cluster centres with those k clusters, by choosing an initial set of random k points $Y = \{y_1,.., y_k\}$, $y_1 \in \mathbb{R}^n$ from $x_i \in X$
3. Loop:
   a) Partition by assigning, or reassigning all data objects to their closest cluster centre, or

      Partition X into subsets K_i, $i = 1,\dots,k$ of data points, chosen by their proximity to y_k :

      $$K_i = \left\{\mathbf{x}: \mathbf{y}_i = arg \min_{\mathbf{y}_j \in Y} dist(\mathbf{x}, \mathbf{y}_j)\right\}$$

      or choose $\mathbf{x}$ where $\|\mathbf{x} - \mathbf{y}_i\|^2$ is a minimum.

   b) Compute new cluster centres as the mean value of the objects in each cluster,

      $$\mathbf{y}_i = \frac{1}{|K_i|} \sum_{x \in K_i} \mathbf{x}, i = 1, \dots, k$$

   c) Repeat loop until no change in cluster centre calculation can be found (or complete convergence).

Output: A set of k clusters that minimizes the squared-error criterion.

Figure 2.1: A k-means cluster algorithm (Gorban and Zinovyev, 2008)

2.1.1 Background

James MacQueen first used the term, *k*-means (MacQueen, 1967), based on the method of Edward Forgy (1965). In 1957, Stuart Lloyd of Bell Labs proposed the *k*-means algorithm as a technique for pulse-code modulation (PCM), but this proposal was only published as a journal article much later (Lloyd, 1982). Therefore, *k*-means is sometimes referred to as the Lloyd-Forgy method, as Forgy essentially published the same method.

The original idea around this clustering method dates back to 1956, when Hugo Steinhaus published an article called, 'Sur la division des corps matériels en parties', translated to 'on the division of material bodies into parts' (Steinhaus, 1956). This article by Hugo Steinhaus is the first to explicitly formulate, in terms of a predictable solution, the problem of subdividing a whole into parts by *k*-means, also called 'nuées dynamiques' (dynamic clouds).

The KMC method is still widely used for segmentation, although it is one of the oldest algorithms for cluster analysis. To realise how cluster analysis relates to segmentation, it is important to note the subtle distinction between the terms, clustering vs. cluster analysis and segmentation vs. customer segmentation:

- Clustering is the process whereby objects are organised into groups where the members have some similarity.
- Cluster analysis is a tool for setting up boundaries for groups of objects or observations.
- Segmentation is the process whereby real-world entities are divided into groups based on pre-defined boundaries.
- Customer segmentation is the practice of applying segmentation specifically to a customer base.

2.1.2 Related research

A few applications of KMC for target market segmentation are found in literature, specifically for the consumer market. With regard to the B2B market, there is room for further evaluation of clustering techniques. In the researched literature, some alternative methods are also considered based on KMC as a starting point.

The use of the KMC method for classification, specifically in the telecom industry, has been researched rather extensively, mostly in China. The most applicable research is briefly described below.

- **Mobile customer cluster analysis with call detail records.**
  Lin (2007) used customer cluster analysis as an important data mining technique to improve mobile operators' competitiveness and customer value. This proved to be manageably effective for the mobile telecom industry. Most telecom carriers cluster their mobile customers by billing system data. Here, however, mobile customers are clustered based on call detail records to also assess consumer behaviour (Lin, 2007).
- **Improved *k*-means clustering of telecom enterprise customers.**
  Zhao, Zhang and Liu (2010) introduced an improved *k*-means algorithm to initialise cluster centres in the cluster analysis phase, following the requirements for customer segmentation in telecom enterprises. The results revealed greater improvement in efficiency and accuracy than the original method. Using the average revenue per unit and minutes of usage on subscriber units for enterprise customers, the segmentation obtained proved significant for differentiated services to customers, product design and the recommendation of phone packages (Zhao *et al*., 2010).
- **Telecom customer segmentation based on cluster analysis**
  Cai Qiuru *et al*. (2012) expounded that the telecom industry is typically a data-intensive industry where data mining applications prove to be good guidance in marketing strategies. Using detailed and comprehensive analysis of the classic *k*-means clustering method, they proposed small

business customer segmentation for the Changzhou telecom industry in the Jiangsu province (Qiuru *et al.*, 2012). Practical results show the *k*-means method to be an effective and successful resolution for customer segmentation in telecom companies, bringing services closer to the customers.

2.2 Particle Swarm Optimisation (PSO)

PSO is described in a summary by Mijwel (2018) as a computational method that optimises a problem by having a population of candidate solutions, here called particles, and moving these particles around in the search space, according to simple mathematical formulas for the particle's position and velocity. Each particle's movement is influenced by its local best-known position, but is also guided towards the best-known positions in the search space, which are updated as better positions are found by other particles. This is expected to move a swarm of particles to the best solutions (Mijwel, 2018).

Formally, according to Bratton and Kennedy (2007), the problem of PSO is:

Let $\mathbf{f}$: $\mathbb{R}^n \rightarrow \mathbb{R}$ be the cost function that has to be minimised. A candidate solution in the form of a vector of real numbers is an argument to the function. As output, a real number is determined, which indicates the objective function value of the given candidate solution. The gradient for $\mathbf{f}$ is not known. The goal is to find a solution a for which $\mathbf{f}(a) \leq \mathbf{f}(b)$ for all b in the search space, which would mean a is the global minimum.

Let S be the number of particles in the swarm, each having a position $x_i \in \mathbb{R}^n$ in the search space and a velocity $v_i \in \mathbb{R}^n$. Let p_i be the best-known position of particle i and let the vector $\mathbf{g}$ be the best-known position of the entire swarm.

The algorithm termination criterion is number of iterations, or a solution with adequate objective function value (Bratton and Kennedy, 2007).

Figure 2.2 shows a simplified form of the standard PSO algorithm as given by Maurice Clerc (2012).

The particle swarm optimisation (PSO) algorithm

For each particle $i = 1,...,S$ do:

7. Initialise the particle's position with a uniformly distributed random vector: $x_i \sim U(b_{lo}, b_{up})$,

where b_{lo} and b_{up} are the lower and upper boundaries of the search space

8. Initialise the particle's best-known position to its initial position: $p_i \leftarrow x_i$

9. if $\mathbf{f}(p_i) < \mathbf{f}(\mathbf{g})$ then update the swarm's best-known position: $\mathbf{g} \leftarrow p_i$, and initialise the particle's velocity: $v_i \sim U(-|b_{up}-b_{lo}|, |b_{up}-b_{lo}|)$

10. While a termination criterion is not met do:

For each particle i = 1,..., S do:

For each dimension d = 1,..., n do:

Pick random numbers: $r_p, r_g \sim U(0,1)$

Update the particle's velocity:

$v_{i,d} \leftarrow \omega\, v_{i,d} + \varphi_p\, r_p\, (p_{i,d} - x_{i,d}) + \varphi_g\, r_g\, (\mathbf{g}_i - x_{i,d})$,

Update the particle's position: $x_i \leftarrow x_i + v_i$

where ω, φ_p, and φ_g are selected to control the behaviour and efficiency of the PSO method.

11. If $\mathbf{f}(x_i) < \mathbf{f}(p_i)$ then:

update the particle's best-known position: $p_i \leftarrow x_i$

12. If $\mathbf{f}(p_i) < \mathbf{f}(\mathbf{g})$ then:

update the swarm's best-known position: $\mathbf{g} \leftarrow p_i$

Figure 2.2: The particle swarm optimisation algorithm (Clerc, 2012)

2.2.1 Background

With reference to fish schooling, the sociobiologist Edward Osborne Wilson (1975) wrote, "In theory at least, individual members of the school can profit from the discoveries and previous experience of all other members of the school during the search for food. This advantage can become decisive, outweighing the disadvantages of competition for food items, whenever the resource is unpredictably distributed in patches" (quoted by Kennedy and Eberhart, 1995). This statement suggests that social sharing of information offers an evolutionary advantage, a hypothesis fundamental to the development of PSO (Kennedy and Eberhart, 1995). The notion of members profiting from the previous experience of all other members also applies to other social behaviour in, for example, herds, schools, flocks and humans.

A number of scientists have created computer simulations of various interpretations of the movement of organisms in a bird flock or fish school.

PSO combines two main component approaches:

- The first, artificial life (A-Life), an extension of biomimicry[1], is a field of study wherein researchers examine systems related to natural life, its processes, and its evolution, through the use of simulations with computer models, robotics, and biochemistry (Langton, 1997). PSO links to A-Life through bird flocking, fish schooling, and swarming theory
- Evolutionary computation is a family of algorithms for global optimisation inspired by biological evolution, and the subfield of artificial intelligence (AI) and soft computing for processing these algorithms (Koza, 1992). PSO relates to evolutionary computation, by using both genetic algorithms and evolution strategies.

By simulating social behaviour, tools and ideas may be derived from computer graphics and social psychology research. With regard to computer

[1] Biomimetics, is a name coined by Otto Schmitt in the 1950s for the transfer of ideas and analogues from biology to technology (Vincent *et al* 2006)

graphics, the first precursor to PSO was the work of Reeves (1983). The first mention of particles was as part of a method for the modelling of fuzzy objects. The method, called particle systems, is used to display objects that are dynamic, or not easily represented by polygons, or bird-like flying objects (called boids[2]). As cited by Gonzalo and Martínez (2012), it was based on simple laws: each boid would only be aware of its immediate proximity or neighbourhood; it would avoid collision with other boids; it would try to move according to the average velocity of its neighbours; and it would not leave the flock. In a similar kind of simulation than the particle system, Frank Heppner, a zoologist, and Ulf Grenander, a statistician, created a stochastic nonlinear model for coordinated bird flocks (Heppner and Grenander, 1990). In 1995, James Kennedy, a social psychologist, and Russell C. Eberhart, an electrical engineer at Purdue School of Engineering and Technology, discovered a method for optimisation of continuous non-linear functions through simulation of a simplified social model of bird flocking, fish schooling, and swarming theory, in particular. They modified the rules for the particle system of (Reynolds, 1987) to propose the PSO algorithm (Kennedy and Eberhart, 1995). For example, they suppressed the non-collision rule, in order to apply the new algorithm to optimisation problems, using a mathematical function as fitness function for each individual of the flock (particle). With these modifications, they estimated that the behaviour of the group resembled more that of a swarm than that of a flock; therefore, according to Gonzalo and Martínez (2012), the new algorithm was called particle swarm optimisation (PSO).

The algorithm began as a simulation of a basic collective environment, with the agents (points representing living organisms) being birds that were

[2] The name 'boid' corresponds with a shortened version of 'bird-oid object' and is incidentally also a New York metropolitan pronunciation of 'birds' (García-Gonzalo and Fernández-Martínez, 2012).

collision-proof. The original intent was to graphically simulate the elegant, but unpredictable manoeuvring of a bird flock (Kennedy and Eberhart, 1995). Unfortunately, the flock quickly settled into a common, unchanging direction. Therefore, besides the nearest-neighbour velocity-matching proposition, a stochastic variable called 'craziness' was introduced, randomising the velocity vectors.

According to Kennedy and Eberhart (1995), Heppner's bird simulations were a way to introduce a dynamic force into the simulation (Heppner and Grenander, 1990). The birds flocked around a roost, a position on the pixel screen that attracted them until they finally landed there. This eliminated the need for a variable such as craziness, as the simulation took on a life of its own.

PSO, therefore, differs from evolutionary computation methods such as an artificial neural network (ANN), in that the population members, called particles, are flown unsupervised through the problem hyperspace. The PSO algorithm is rather a method for simulating swarm intelligence (SI). Parsopoulos and Vrahatis (2010) described SI as a branch of artificial intelligence (AI) that studies the collective behaviour and emergent properties of complex, self-organised, decentralised systems with social structure (as cited by Srivastava and Kumbharvadiya, 2014). This was not the first published use of the term, swarm intelligence. Bonabeau *et al.* (1999) defined swarm intelligence in a biological context as the emergent collective intelligence of groups of simple agents.

The preferable term, particle, is used, instead of the biological term, agent and particle swarm optimisation is an algorithm to achieve the best solution for a particular problem in SI. PSO is a metaheuristic global optimisation method, as it makes few or no assumptions about the problem being optimised and can search very large spaces of candidate solutions.

Herewith a short explanation of the algorithm as it was represented in the code based on the tutorial given by (Ballardini, 2018b), with notation from Van der Merwe and Engelbrecht (2003).

Since each particle represents a position in the dimension space, the aim was to adjust this position, according to the best position of the particle found thus far, and the best position in the neighbourhood of that particle.

To accommodate this, each particle stores these values:

- The particle's current position (x_i)
- The particle's current velocity (v_i)
- The particle's best position found thus far (y_i).

At each iteration, the particle's position is updated to

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \tag{2.1}$$

At this point, it is worthwhile to remember that PSO is a useful method to simulate social behaviour. Therefore, the behaviour of each particle, in terms of its velocity, was based on the inertia weight, w, applied to the previous velocity, together with acceleration constants, c_1 and c_2, and random numbers r_1 and r_2 from a uniform distribution U(0; 1) per dimension. These form the basis of three societal components that influence the particle's velocity and ultimate position, Ballardini (2018b):

- The previous velocity: a fraction of the previous velocity has an effect on a particle's current moving state. The particle conducts inertial movement according to its own velocity, so the fraction used as parameter w is called the inertia weight.
- The cognitive component: a function of the distance of the particle related to its best-achieved distance, i.e. the particle's move resulting from its own experience. Therefore, parameter c_1 is called the cognitive learning factor or cognitive acceleration factor.
- The social component: a function of the distance of the particle from the best particle found thus far (i.e. the best of the personal bests). This result takes into account the best-achieved distance over all the

particles in the swarm, or the global (or local) optimal position in the swarm (y_g). It represents information shared and co-operation among the particles, where the particle's movement is influenced by other particles' experience in the swarm. The parameter c_2 is, therefore, called the social learning factor or social acceleration factor.

The particle's velocity, which forms part of the position in equation (2.1), may now be expressed in terms of the societal components as *velocity* = *inertia* + *cognitive* + *social*, or

$$v_{i,k}(t\ +\ 1) = wv_{i,k}(t) + c_1\, r_{1,k}(t)\left(y_{i,k}(t) - x_{i,k}(t)\right) + c_2 r_{2,k}(t)\left(y(t) - x_{i,k}(t)\right), \quad (2.2)$$

with $k = 1, \dots, N_d$ (N_d = number of dimensions)

Here, the three societal components are indicated by

$$\begin{aligned} inertia &= wv_{i,k}(t)\,, \\ cognitive &= c_1\, r_{1,k}(t)\,, \\ social &= c_2 r_{2,k}(t)\left(y(t) - x_{i,k}(t)\right). \end{aligned} \quad (2.3)$$

The above sociological influences, further described by Wang, *et al.* (2018), are illustrated in Figure 2.3 as an iteration scheme for a particle.

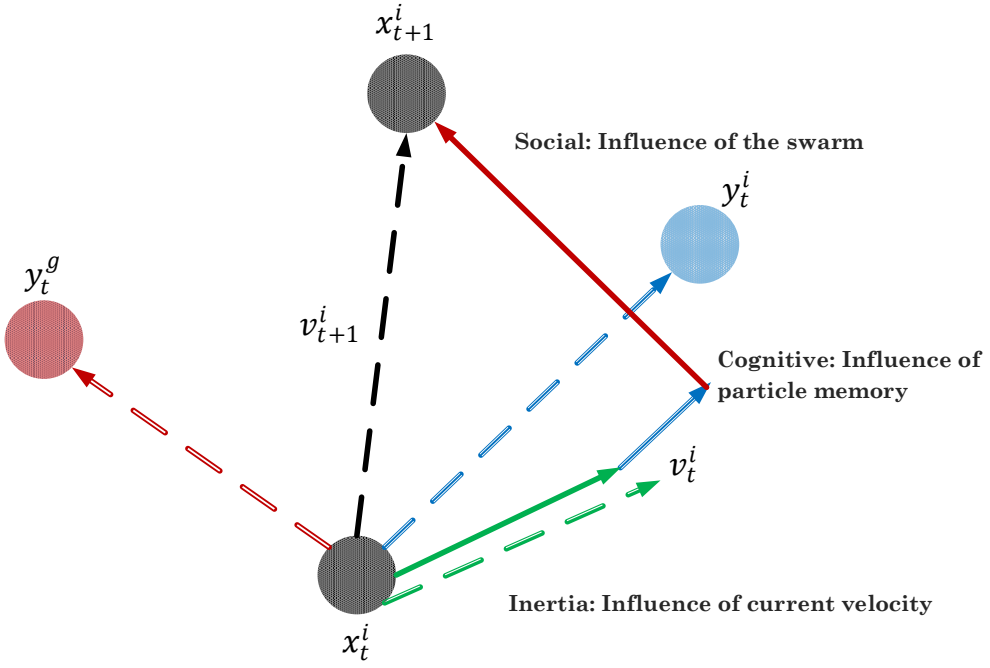


Figure 2.3: Influences on a particle (Wang *et al*., 2018)

The personal best position of a particle i is updated, if the fitness value in the current step was lower than the previous fitness value of the particle. This is calculated using equation (2.4) without any loss of generality.

$$y_i(t+1) = \begin{cases} y_i(t) & if\ f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & if\ f(x_i(t+1)) < f(y_i(t)) \end{cases} \quad (2.4)$$

The PSO algorithm usually executes continuous iterations of the equation (2.2) and equation (2.4), until a specified number of iterations has been reached. Alternatively, the algorithm may halt when the velocities are close to zero, where a minimum was reached in the optimisation process.

As described by Van der Merwe and Engelbrecht (2003), two basic approaches to PSO (**gbest** and **lbest**) exists based on the interpretation of the neighbourhood of particles. Equation (2.4) reflects the **gbest** version of PSO where the neighbourhood of the particle is basically the entire swarm. In the **lbest** PSO model, the swarm is divided into overlapping neighbourhoods, and the best particle of each neighbourhood is found. For the **lbest** PSO approach, the social component of equations (2.2) and (2.3) changes to

$$c_2 r_{2,k}(t)\left(\hat{y}_{j,k}(t) - x_{i,k}(t)\right) \quad (2.5)$$

where $\hat{y}_j$ is the best particle in the neighbourhood of the i^{th} particle.

In the two approaches, the social components are either bound to the entire swarm (**gbest**) or the current neighbourhood of the particle (**lbest**).

2.2.2 Related research

A study by Ricardo Poli (2008) revealed an interesting analysis of publications on the applications of PSO. Figure 2.4 was created here based on the results from the study of Poli (2008). It depicts the proportion of the application areas found in publications up to 2008. Note that the application areas directly or indirectly related to segmentation or telecommunications are shown in blocks, with darker borders closest to the scope of this study.

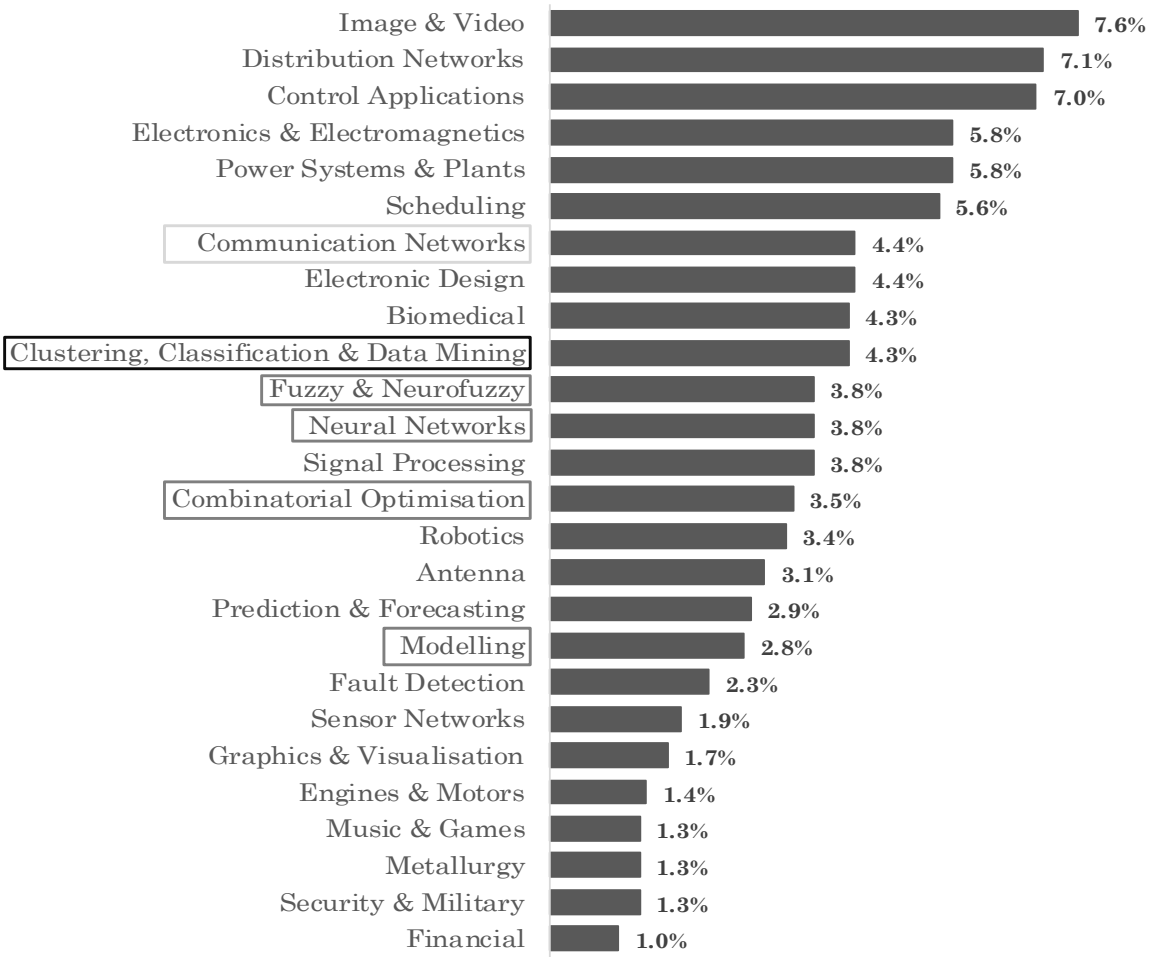


Figure 2.4: PSO applications, based on total publications (Poli, 2008)

With reference to Figure 2.4, research on the use of PSO for business customer segmentation is not as prolific as expected, although some research does exist on the use of PSO in telecommunications. Most applications of PSO seem to be in engineering or related technical fields. A few researchers

have expounded on how to use PSO to enhance existing models, such as clustering and optimisation models. In adapted form, these can be applied to business segmentation in a telecom environment.

Below are short descriptions of some relevant applications.

- **Predator-prey optimisation**

  In a very interesting and alternative discussion, Silva *et al.* (2002) presented the results of experimentally comparing the performance of several variants of the standard swarm particle optimiser as a new approach to swarm-based optimisation. The new algorithm, called predator-prey optimiser, combines the ideas of PSO with a predator-prey-inspired strategy, which is used to maintain diversity in the swarm, and prevent premature convergence to local suboptima (Silva *et al.*, 2002). With further research, this algorithm can be modified to model the targeting of customers, albeit not with the same fierce goals as predator-prey.

- **Data mining**

  Fundamentally, particle swarm optimisers are distributed algorithms where the solution for a problem emerges from the interactions between individual components of the problem, called particles. Sousa *et al.* (2004) proposed PSO as a new tool for data mining. Along with the exponential growth of information technology (IT), there has also been a huge increase in databases. It is fairly easy to create and customise a database, but with the growth in records, it is not easy to retrieve high-level knowledge from the same database.

  There are not many off-the-shelf solutions for data analysis (data mining), compared to those used for database creation and management. Data mining techniques seek out, identify, validate and predict structural patterns in data. These actions can be grouped in five categories: decision trees, classification rules, association rules, clustering and numeric prediction (Sousa *et al.*, 2004). Particle swarm algorithms have proved

competitive enough in optimisation areas to be used for the necessary pattern recognition and to satisfy, among others, the classification rules of data mining.

- **Pattern recognition and image processing**

  Although not immediately apparent, Omran (2006) demonstrated an application of PSO similar to customer segmentation. This application, pattern recognition, also has as its objective to classify objects into different categories and classes. Omran investigated the application of partitional clustering algorithms (of which KMC is one) to the problem of unsupervised classification and segmentation of images. These partitional clustering techniques are essentially based on the minimisation of a square-error function (Omran, 2006). However, the minimisation problems involved are generally considered NP-hard and combinatorial (Leung *et al.,* 2000). NP-hard (or nondeterministic polynomial time hard) problems belong to a class of computational decision problems which are at least as hard as the hardest problems for which a given solution can be verified in polynomial time (Weisstein, 1999). Polynomial time is the number of steps required to complete an algorithm for a given input.

  Experimental results for pattern recognition show that the PSO clustering algorithm performs better than current clustering algorithms (Omran, 2006). PSO proved therefore to be an efficient optimisation algorithm to generate synthetic images, measure the quality of a clustering algorithm, compare different clustering algorithms and create benchmarks. A number of revolutionary methods for image processing consist of clustering first, as input to a collective, iterative method, with emphasis on co-operation between particles, as stated in later works on multiple swarm optimisation (Clerc, 2006).

- **Travelling salesperson problem (TSP)**

  McCullock (2012) used PSO to demonstrate a simplified form of the TSP. In brief, the shortest tour needs to be found through a number of cities, without visiting the same one twice. This may be useful for salespersons in a telecom environment visiting potential enterprise customers. The algorithm used by McCullock calculates the minimum Cartesian distance through eight cities (McCullock, 2012). As another example for an extention on the TSP in three dimensions, the problem was solved by a PSO algorithm for randomly placed points on a sphere (Mijwel, 2016). Here the arc length between points were calculated as opposed to the normal Eucledian distance in two dimensions.

- **Customer segmentation**

  Traditionally, most companies use marketing campaigns to recruit new customers, or retain existing ones. As mentioned before, customer segmentation is an important technique in targeting the right customers during a marketing campaign. Most previous clustering algorithms have drawbacks, such as being stuck at local minima. Dhandayudam and Krishnamurthi (2012) started to propose a consumer segmentation model using PSO, followed by Chan *et al.* (2016), highlighting the advantage of using fewer parameters to reach a global optimal solution. The studies calculate the Recency, Frequency and Monetary values (RFM)[3] from a dataset, into value-based information. Based on this value-based information, the PSO algorithm is able to cluster consumers to find those likely to be the most profitable and valuable.

[3] The most well-known model in sales and marketing, used to compute customer lifetime value (Fader *et al.* 2005)

- **Customer portfolio model**

  Enterprise decision-makers constantly have to trade off between returns and risks. Customer portfolio paradigm is a favourable tool for increasing the value of the customer base and reducing its risks. After analysing the similarities and differences of the financial assets and customer equity, Yun and Yan (2013) proposed a customer portfolio optimisation model. In this model, semi-variance is employed as the risk metrics, considering the risk characteristics of the customer equity; and a multiphase marketing strategy is its output because of the liquidity characteristics of the customer equity. A PSO algorithm is proposed to find the solution for the NP-hard computational feature of customer portfolio model. The empirical tests from a futures company show that, given the company's risk tolerance, a multiphase acquisition and retention strategy may maximise the long-term revenue of the customer equity with this model and its PSO algorithm (Yun and Yan, 2013).

- **Multiple criteria green supplier segmentation**

  Supplier segmentation is an important strategic activity for companies. Given the increased importance of sustainable and green supply chains, this points to a large gap in the literature. Therefore, Bai *et al.* (2017) proposed a green supplier segmentation model. Given the multicriteria nature of this problem, a novel hybrid multicriteria methodology, incorporating SI, is used to evaluate the problem (Bai *et al.*, 2017).

- **Customer churn prediction**

  Churn prediction in telecommunications has gained huge prominence in recent times, due to the extensive interest exhibited by stakeholders and a large number of competitors, and the huge revenue losses incurred, owing to churn (Vijaya and Sivasankar, 2017). Predicting telecom churn is challenging because of the large volumes and sparse nature of the data. Vijaya and Sivasankar presented a technique for telecom churn prediction that employs PSO. They also proposed three variants of PSO for churn

prediction, namely PSO incorporated with feature selection as its pre-processing mechanism, PSO embedded with simulated annealing[4] and, finally, PSO with a combination of both feature selection and simulated annealing.

- **Communications area coverage**

  Recent research by Anicho *et al*. (2019) applied SI, in combination with reinforcement learning, to coordinating multiple high-altitude platform stations (HAPS). Typically, HAPS is defined by the International Telecommunications Union as, "a station located on an object at an altitude of 20 to 50 km and at a specified, nominal, fixed point relative to the earth" (Anicho *et al.,* 2019). The problem is, therefore, coordinating a swarm of unmanned aerial vehicles (commonly known as drones), to act as HAPS, in order to maximise area coverage for a set of mobile users. It was found that the SI approach led to higher convergence.

[4] Simulated annealing is a method for solving unconstrained and bound-constrained optimisation problems. The method models the physical process of heating a material and then slowly lowering the temperature to decrease defects, thus minimizing the system energy (Mohamed, 2018).

3. LITERATURE REVIEW: DEPENDENCE ANALYSIS METHODS

Supervised segmentation refers to a family of pattern analysis methods. These segmentation algorithms use a priori knowledge, as well as machine learning (ML) approaches (e.g. neural networks) involving the ground truth of a training set (Law *et al.,* 2017).

Most of these algorithms result in tree structured outputs. These are useful, as they deliver a visual, graphic representation of segments that assists with validation and explanation of these methods to non-technical stakeholders (Chulis, 2012). One key difference from an interdependence approach is that these models require a dependent variable; hence, the term 'dependence analysis' is used for supervised analysis methods.

In addition to groupings in tree format as an outcome, these dependence models generate associated probability and propensity metrics in their output. The main output of dependence segmentations is groupings of similar customers that can be further profiled. These tailored strategies may be applied to reduce churn, encourage increased spending behaviour, or introduce risk-intervention strategies, prior to approaching failure to pay (Chulis, 2012).

Critics argue that the subsequent set of algorithms is actually a predictive model, rather than a segmentation model, because of the probability or prospect prediction output. Chulis (2012) concluded that the distinction may lie in the use of the model.

Segmentation is the classification of customer bases into different groups based on multidimensional data, and this classification is used to suggest an actionable roadmap to plan relevant marketing strategies, design products and produce customer service strategies at a segment level that will drive desired business outcomes. Predictive modelling is forecasting specific consumer behaviour at individual level (Chulis, 2012).

The term, 'supervised', refers to specific data mining (or data science) techniques, such as decision trees, random forests, gradient boosting or neural networks (Grover, 2016). In the following sections, reviews are presented of a decision tree analysis method called CHAID, and a pattern analysis method using artificial neural networks, or ANNs.

3.1 Chi-square automatic interaction detection (CHAID)

CHAID is a technique that may be used for prediction, classification and the detection of complex interaction between variables, as well as displaying the modelling results in an easy-to-interpret tree diagram. The technique involves the performing of statistical tests in a similar fashion as regression analysis on numerical values (Kass, 1980). Seen as a major area of ongoing research and application in ML techniques, CHAID has grown to be a more practical alternative. It is therefore not with surprise that various commercially viable applications have been developed in CHAID. The basics of CHAID is indeed described by one such commercial company. According to the website for the data mining and ML consulting company SmartDrill, the base of the resulting decision tree represents the total modelling dataset (Taves, 2010). CHAID then creates a first layer of branches by displaying values of the strongest predictor of the dependent variable.

It continues this branching procedure until the final branches of the tree have been generated. If CHAID is being used to generate a market segmentation model, then these final or terminal branches are the final market segment. A typical CHAID model may have about a dozen terminal segments. However, a model with many more segments are sometimes built, especially to identify and understand some smaller niche segments that may represent either a significant problem or an unusually good opportunity (Taves, 2010).

For the CHAID analysis, target data for variables that are nominal, categorical or ordinal were selected. Continuous variables were grouped into categories (Kass, 1980). Missing values were not replaced; the data was used

in raw format, except for the transformed categories. When the dependent variable has only two values, e.g. direct sales or indirect sales, the result is called a nominal CHAID model. In such a model, the proportion of each market segment is shown as cases in the desired category of the dependent variable, e.g. indirect sales.

When the dependent variable is at least ordinal, i.e. the values can be arranged in some meaningful order, an ordinal CHAID model can be generated. Revenue in dollars is an example of an ordinal dependent variable. In an ordinal model, each segment is assigned an average value on the dependent variable, e.g. average revenue, and this may be shown in both the tree diagram and the contingency and gains tables (Taves, 2010)

The segments are depicted in the tree diagram, as well as being ranked in a special type of contingency table to show the frequency distribution of categories (nodes) against dependent variables (Kass, 1980).

CHAID is particularly useful for generating market segmentation models for planning purposes. In addition to its effectiveness as a predictive model, the resulting tree structure provides a valuable top-down view of the target market configuration, showing the groupings of predictors which lead to any given segment (Taves, 2010). This can be very helpful to sales executives who want to visualise and define clear market segments.

CHAID automatically determines how to group the values of predictor variables into a manageable number of categories. For example, there could be ten categories for company revenue based on ranges of values. CHAID might collapse these ten categories to only four or five statistically significant different groups. For example, in terms of revenue, these groups or buckets may be simplified to terms such as top, high, middle, and low.

One important advantage of CHAID over alternatives such as multiple regression, is that it is non-parametric, that is, the data used for CHAID is not required to fit any normal or other distribution (Kass, 1980).

However, because it uses multiple splits in different ways by default, quite large sample sizes are needed for it to work well. With small sample sizes, the respondent groups can quickly become too small for a dependable breakdown. The algorithm originally proposed by Kass (1980) only accepts nominal or ordinal (i.e. the values can be arranged in some meaningful order) categorical predictors. Before using the algorithm, continuous predictors are therefore transformed into ordinal predictors.

Figures 3.1a and 3.1b portray the CHAID algorithm, which is a revised version from the IBM documentation for the statistics platform called SPSS[5], developed originally by Nie, Bent and Hull (1970). This CHAID algorithm involved three steps, namely merging, splitting and stopping. A tree is developed by repeatedly using these three steps on each node, starting from the root node (Nie *et al.*, 1970). The probability that a variate would assume a value greater than, or equal to the observed value strictly by chance is known as the **p**-value. The **p**-value and ***Bonferroni*** adjustments are described later in chapters 4 (Methodology) and 5 (Analysis and Results).

The following variables and notations are used (IBM, 2012):

Y	The dependent or target variable, which can be *ordinal categorical*, *nominal categorical* or *continuous*.
$X_m, m = 1, \dots, M$	The set of all predictor variables. A predictor can be *ordinal categorical*, *nominal categorical* or *continuous*
$h = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$	The whole learning sample
w_n	The case weight associated with n
f_n	The frequency weight associated with case n. A non-integral positive value is rounded to its nearest integer.

[5] SPSS Statistics is a software package used for interactive or batched statistical analysis. Long produced by SPSS Inc., it was acquired by IBM in 2009. The current versions are called IBM SPSS Statistics (IBM, 2020).

The CHAID Algorithm

Step 1 – Merging

For each predictor variable X, non-significant categories should be merged. Each final category of X will result in one child node, if X is used to split the node. The merging step also calculates the adjusted **p**-value that will be used in the splitting step.

1. If X has one category only, stop and set the adjusted **p**-value to 1.
2. If X has two categories, go to step 8.
3. Otherwise, find the permissible pair of categories of X (a permissible pair of categories for an ordinal predictor is two adjacent categories and, for a nominal predictor, any two categories) that is least significantly different, i.e. most alike. The most similar pair is the pair where the test statistic gives the largest **p**-value, with respect to the dependent variable Y.
4. For the pair having the largest **p**-value, check if its **p**-value is larger than a user-specified alpha-level α_{merge} (alpha_merge). If it is, this pair is merged into a single compound category. Then a new set of categories of X is formed. If it is not, then go to step 7.
5. (Optional) If the newly formed compound category consists of three or more original categories, then find the best binary split within the compound category where **p**-value is the smallest. Perform this binary split if its **p**-value is not larger than an alpha-level $\alpha_{\text{split-merge}}$ (alpha_split-merge).
6. Go to step 2.
7. (Optional) Any category having too few observations (as compared to a user-specified minimum segment size) is merged with the most similar other category as measured by the largest of the **p**-values.
8. The adjusted **p**-value is computed for the merged categories by applying ***Bonferroni*** adjustments.

Figure 3.1a: The CHAID algorithm for merging (IBM, 2012)

The CHAID Algorithm (continued)

Step 2 — Splitting

The best split for each predictor is found in the merging step. The splitting step selects which predictor to use to best split the node. Selection is done by comparing the adjusted **p**-value associated with each predictor. The adjusted **p**-value is attained in the merging step.

1. Select the predictor that has the smallest adjusted **p**-value (that is, the most significant predictor).
2. If this adjusted **p**-value is less than, or equal to a user-specified alpha-level split α_{split} (alpha_split), split the node using this predictor. Otherwise, do not split and the node is considered a terminal node.

Step 3 — Stopping

The stopping step checks whether the tree-growing process should be stopped, according to the following stopping rules.

1. If a node becomes pure, i.e. all the cases in the node have identical values of the dependent variable, the node will not be split.
2. If all cases in a node have identical values for each predictor, the node will not be split.
3. If the current tree depth reaches the user-specified maximum tree-depth limit value, the tree-growing process will stop.
4. If the size of a node is smaller than the user-specified minimum node-size value, the node will not be split.
5. If the split of a node results in a child node where the node size is smaller than the user-specified minimum child node-size value, child nodes that have too few cases, compared to this minimum, will merge with the most similar child node as measured by the largest of the **p**-values. However, if the resulting number of child nodes is 1, the node will not be split.

Figure 3.1b: The CHAID algorithm, splitting and stopping (IBM, 2012)

3.1.1 Background

Prof Gordon V. Kass, currently a visiting associate professor at Wits School of Statistics and Actuarial Science, first published the CHAID technique (Kass, 1980), although the technique was already defined earlier as part of his PhD thesis (Kass, 1975a). The use of significance testing, based on the works on association measures for cross-classifications (Goodman and Kruskal, 1959), forms part of the CHAID method. CHAID is a decision tree technique based on adjusted significance testing. The technique is an improved extension of the Automatic Interaction Detection (AID) technique, where adjusted significance testing is used with the χ^2 statistic.

Belson (1957) offered a similar idea to the Automatic Interaction Detection (AID) technique, in an earlier work on biological classification (Kass, 1975b). The procedure for AID is described in (Morgan and Sonquist, 1963a) as a technique for analysing dependencies in multivariate data. With the availability of electronic computer programs, the AID technique gained popularity. Morgan and Sonquist (1963b) specifically described a computer program (called the *automatic interaction detector*) for the technique, with details on interpretation of the results. This was heavily used in a published research project (Morgan *et al.*, 1966). Sonquist *et al.* (1971) developed an evolution of this program, called AID-III, which is both more flexible and more powerful (Kass, 1975b). Another improved version of this program was called Search (Sonquist *et al.*, 1973). A few years Fielding described this program to explore data structures (Fielding and O'Muircheartaigh, 1977).

Messenger and Mandell (1972) proposed the Theta Automatic Interaction Detection (THAID) algorithm as a further development of AID. It similarly bisects data, but based on a different statistic. This statistic, which they call *theta* (θ), is related to the proportional reduction in misclassification errors. In the same year, Kass did work on significance testing in AID. Based on the likelihood ratio test of the null hypothesis of Scott and Knott (1974), Kass later published some extensions of AID and THAID (Kass, 1975a).

In the case of the dependent variable being categorically unrelated, Goodman (1979) proposed a null hypothesis test of independence to the categories, building on earlier studies of association measures for cross-classifications.

Further improving on this independence testing, Kass (1980) developed and published a technique to investigate large quantities of data with a categorical response or target. He called this technique CHAID. A Bonferroni correction for the multiple comparisons carried out in the CHAID algorithm formed the basis for the testing in this technique.

In a similar fashion to regression analysis, Kass and Hawkins (1982) developed an extended version of CHAID, known as XAID, which also used statistical tests, but for continuous or numeric responses. The original CHAID algorithm described by Kass and published in 1980, proved to be the basis for an *exhaustive* CHAID algorithm, published just over a decade later by Biggs, De Ville and Suen (1991).

3.1.2 Related research

Telecommunication companies around the world face escalating competition, which forces them to aggressively market special pricing programmes, aimed at retaining existing customers and attracting new ones. It is, therefore, essential to exercise proper CRM or customer relationship management (Bain & Company, 2018). Most CHAID applications are in the domain of market strategy and, specifically, market segmentation. This is also true of telecom businesses, but their marketing tactics are performed via the CRM approach. The little research conducted on CHAID for market segmentation in telecom companies, is mostly directed towards finding a way to exercise better CRM. A few references to CHAID or related techniques in telecom target marketing are mentioned here.

- **Knowledge discovery in telecommunications**

  Rygielski *et al*. (2002) mentioned two areas where knowledge discovery is conducted in the telecommunications industry. These are not directly related to CHAID, but are seen as opportunities for applying CHAID.

  - *Call-detail record analysis*: By identifying customer segments with similar use patterns, companies can develop attractive pricing and feature promotions.
  - *Customer loyalty:* Telecommunication companies can use data mining (specifically with CHAID) to identify the characteristics of customers who are likely to remain loyal, once they switch service providers, thereby enabling the companies to target their spending toward customers who will produce the most profit.

- **Telecom market segmentation model**

  Ratner (2003) is known for applying polytomous logistic regression (PLR) by means of CHAID to do market segmentation. The term is derived from the words polychotomous variable, meaning a variable that has multiple (more than two) categories (Weiss, 1995). The PLR/CHAID analysis technique was used initially to segment the cellular phone market into four groups and then build a model for classifying cellular users into one of the four groups, as a model for CRM strategy (Ratner, 2003).

- **Telecom services marketing**

  Strouse (2004) mentioned CHAID for analysing distribution channels, and churn for a more customer-centred focus. The importance of matching the pricing strategy to the marketing efforts (Strouse, 2004) could be another area where CHAID can flourish.

- **Machine learning for dynamic customer segmentation**
  Dullaghan and Rozaki (2017) investigated classification tree methods and Bayesian modelling to segment mobile customers in Ireland. The use of machines made it possible to test various classifiers for categorising a Customer's Age Group, VIP Status, Spend Status and Customer Length of Service (Dullaghan and Rozaki, 2017). This contributed to improved churn prediction, and is a major focus for CHAID research.

During the literature study, it became clear that CHAID is used commercially for target market strategies and CRM. It was, however, difficult to find applications of CHAID specifically suited to a telecom marketing strategy, and even more difficult, to B2B markets. The reasons could be (Taves, 2010):

- Proprietary software with CHAID embedded does not share the techniques used, due to the danger of intellectual property infringements.
- The CHAID technique is often used in combination with other techniques (such as *k*-means clustering) to enhance a solution, and research focus up to now has rather been on other techniques.
- No need for specific analysis in a B2B telecom environment was identified until recently.

Nevertheless, most of the literature shows that further research is needed for the most effective use of CHAID as a predictive multivariate classification technique.

3.2 Artificial Neural Networks (ANN)

In general, the term neural networks is used, but in IT, a neural network is referred to as an artificial neural network or ANN. There is no single universally accepted definition of ANN. Rouse (2018) defined it as a system of hardware and/or software patterned after the operation of neurons in the human brain. Where the basic processing unit in the human nervous system is the neuron, ANNs rely on a type of artificial neuron, called a perceptron.

The definition is expanded by defining deep learning (DL) technology as a variety of ANNs, which also falls under the umbrella of artificial intelligence (AI). According to Nicholson (2019), ANNs are a set of algorithms, loosely modelled after the human brain to recognise patterns. By interpreting sensory data through machine perception, raw input is labelled or clustered (Nicholson, 2019). In general, an ANN is initially trained, or supplied with a large amount of data and rules about data relationships (Rouse, 2018). ANNs may be considered as clustering and classification layers on top of the managed data. These algorithms assist with grouping unlabelled data, according to similarities in the input data, and to classify data from labelled datasets used for training (Nicholson, 2019). ANNs may also be used for extracting features that are then fed to other algorithms for clustering and classification. Deep neural networks (DNNs) are components of DL involving algorithms for reinforcement learning, classification and regression. A DNN is an ANN with multiple layers between the input and output layers (Bengio, 2009).

In this study, the focus is on supervised learning, specifically feedforward ANNs. In feedforward networks, data flows from the input to the output neurons – strictly forward (Fick, 2006). This multiple layered network, also called a multilayer perceptron (MLP), have greater processing power than a perceptron with one layer (Brownlee, 2016).

Backpropagation (BP) is a widely used algorithm in training feedforward neural networks for supervised learning (Goodfellow *et al.*, 2016).

According to Guresen and Kayakutlu (2011), an ANN An ANN may be represented in terms of a directed graph[7] with the following features:
The ANN directed graph has at least one start node (or Start Element; SE), one end node (or End Element; EE) and at least one Processing Element (PE). All the nodes used should be PEs, except start nodes and end nodes. A state variable n_i is associated with each node i. A real valued weight w_{ki} is associated with each link (ki) from node k to node i, and a real valued bias b_i is associated with each node i. At least two of the multiple PEs are connected in parallel. A learning algorithm helps to model the desired output for given input. There is a flow on each link (ki), from node k to node i, which carries exactly the same flow that equals n_k, caused by the output of node k. Each start node is connected to at least one end node, and each end node is connected to at least one start node. No parallel arcs (each link (ki) from node k to node i is unique).

[7] In graph theory, a directed graph (or digraph) is a graph that is made up of a set of vertices (V) connected by arcs (A), where the arcs have a direction associated with them (Bang-Jensen and Gutin, 2009).

The ANN illustrated as a directed graph is depicted in Figure 3.2 with reference to the above.

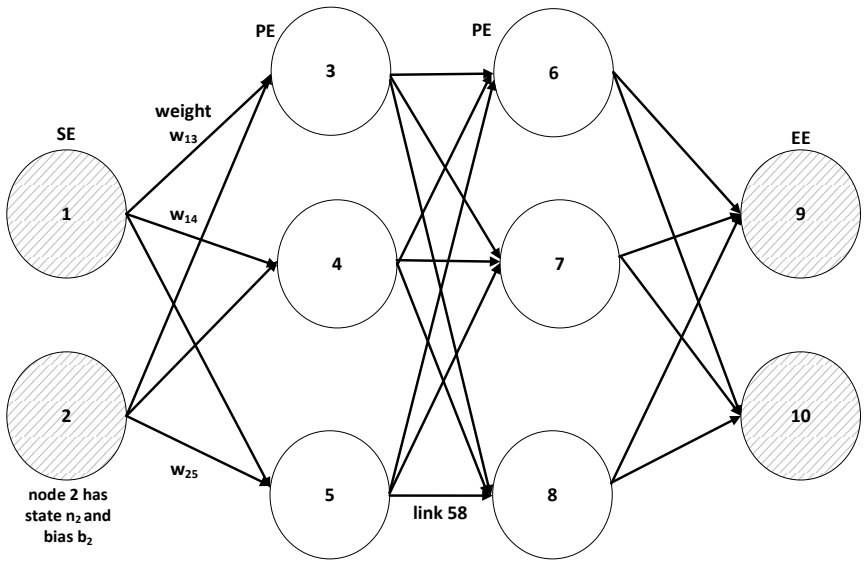


Figure 3.2: An ANN illustrated as a directed graph

Figures 3.3a and 3.3b each depict an algorithm illustrating the neural network process. The first one is a combination of the simple perceptron algorithm for pattern recognition, as described by (Fausett, 1994), and the perceptron convergence algorithm, as summarised by (Haykin, 2009). The second algorithm is a high-level version of the backpropagation algorithm developed by Hertz *et al.* (1991), as described by Hagan *et al*. (1996) and Fick (2006).

The ANN perceptron algorithm

Step 1 – **Initialisation**

For simplicity: set bias b = 0, and weights, $W = [0]$ $(w_{ij} = 0)$, and set learning rate, $\alpha = 1$ $(0 < \alpha \leq 1)$. While stop condition is false, do steps 2 to 5.

Step 2 – **Activation**

For i = 1 to n: set $x_i = s_i$, for each of the training pair **s**:**t**. Continue to steps 3 to 5.

Step 3 – **Compute actual response of output unit**

Compute the net input: $y\_in_j = b_j + \sum_i^n x_i w_{ij}\,, i = 1, \dots n, j = 1, \dots, m$

Compute the activation function:

$$y_j = \begin{cases} 1, & if\ y\_in_j > \theta_j \\ 0, & if\ -\theta_j \leq y\_in_j \leq \theta_j \\ -1, & if\ y\_in_j < -\theta_j \end{cases}$$

Step 4 – **Update weights and bias**

If an error occurred for the pattern, $\mathbf{e} = \mathbf{t} - \mathbf{y}$ or if the output is not correct, that is,

if $\mathbf{y} \neq \mathbf{t}$, then:

for i = 1 to n : set $w_{ij}(new) = w_{ij}(old) + \alpha\, t_j\, x_{ij}$, for $j = 1, \dots, m$

for $= 1, \dots, m$: set $b_j(new) = b_j(old) + \alpha\, t_j$

else, for $\mathbf{y} = \mathbf{t}$, then:

for i = 1 to n : set $w_{ij} = w_{ij}$, for $j = 1, \dots, m$

for $j = 1, \dots, m$: set $b_j(new) = b_j(old)$

Step 5 – **Stop condition and convergence**

Stop the algorithm:

- when no weight is changed in the current cycle of the training dataset (epoch), or

- when a pre-determined number of epochs is reached.

Convergence:

Another stop condition is if the error vector $(\mathbf{t} - \mathbf{y})$ is close to the **0** vector. This is clear when the weight is calculated with error instead of target (Haykin, 2009):

$w_{ij}(new) = w_{ij}(old) + \alpha(\, t_{j\_} y_j)\, x_{ij}$, for i = 1 to n, $j = 1, \dots, m$

Figure 3.3a: The ANN perceptron algorithm (Fausett, 1994)

The ANN backpropagation algorithm

The backpropagation algorithm is a generalisation of the least mean square (LMS) algorithm. In multilayer networks, a layer number is shown as a superscript, e.g. $\mathbf{a}^2$, as output of layer 2. During training, a set of input patterns and corresponding targets are presented:

$\{\mathbf{p}_1, \mathbf{t}_1\}, \ldots, \{\mathbf{p}_q, \mathbf{p}_q\}$, where $\mathbf{p}_q$ is an input pattern vector and $\mathbf{t}_q$ is the target output.

The steps below are repeated for each pattern

Step 1 – **Initialise**

Small random values are usually chosen as initial values for weights and biases

Step 2 – **Propagate input forward**

For multilayer networks, the output of one layer becomes the input to the following:

$$\mathbf{a}^{m+1} = f^{m+1}(W^{m+1}\mathbf{a}^m + \mathbf{b}^{m+1}), \text{for } m = 0,1,\ldots,M-1, \quad \text{with M number of layers.}$$

At each input to the network, the actual output $\mathbf{a}$ is compared with the target. Let initial actual output be $\mathbf{a}^0 = \mathbf{p}$. Outputs of neurons in the last layer are considered as the network outputs: $\mathbf{a} = \mathbf{a}^M$. Initial random values chosen for output to be around 0.5.

Step 3 – **Output errors**

The algorithm adjusts the network parameters, so that the mean squared error (MSE) is minimised. As with the LMS algorithm, the MSE is approximated by:

$\widehat{\boldsymbol{F}}(\mathbf{x}) = (\mathbf{t}(k) - \mathbf{a}(k))^T (\mathbf{t}(k) - \mathbf{a}(k)) = \mathbf{e}^T\mathbf{e}(k)$, for the squared error at iteration k.

Step 4– **Propagate sensitivities backward**

The sensitivity of $\widehat{\boldsymbol{F}}$ to changes in the ith element of the input layer m is simplified, in terms of weighting and bias, to: $\Delta w_{ij}^m = \frac{\partial \hat{\mathbf{F}}}{w_{ij}^m} = \mathbf{s}_i^m \mathbf{a}_j^m$, $\Delta b_i^m = \frac{\partial \hat{\mathbf{F}}}{b_i^m} = \mathbf{s}_i^m$

By using the chain rule of calculus and multiplying the Jacobian matrix, the sensitivity of $\widehat{\boldsymbol{F}}$ for ith element is (Fick, 2006): $\mathbf{s}^m = \frac{\partial \hat{\mathbf{F}}}{\partial \mathrm{n}_i^m} = \dot{\mathbf{F}}^m[\mathbf{n}^m]\left[\mathbf{W}^{m+1}\right]^T \mathbf{s}^{m+1}$.

Sensitivities are propagated backward: $\mathbf{s}^M \rightarrow \mathbf{s}^{M-1} \rightarrow \cdots \rightarrow \mathbf{s}^2 \rightarrow \mathbf{s}^1$.

The sensitivity of the final layer serves as starting point: $\mathbf{s}^M = -2\dot{\mathbf{F}}^m[\mathbf{n}^m][\mathbf{t} - \mathbf{a}]$.

Step 5 – **Adjust weights and biases**

Finally, weights and biases are adjusted using the steepest descent rule:

a. $\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m[\mathbf{a}^{m-1}]^T$, α is the learning rate.

b. $\mathbf{b}^m(k+1) = \mathbf{W}\mathbf{b}^m(k) - \alpha \mathbf{s}^m$, α is the learning rate.

Figure 3.3b: ANN backpropagation algorithm (Hagan *et al.*, 1996)

The following variables and notations, based on Fausett (1994) and Fick (2006), are used for the algorithms and in the sections that follow:

x_i, y_i	Activations of neuron units X_i and Y_j, respectively, where, for input units X_i : $x_i = \text{input signal}$; and for other units Y_j : $y_j = f(y\_in_j)$, with y_in_j as described below.
w_{ij}	Weight on connection from unit X_i to unit Y_j. Note: Some authors use opposite convention, with w_{ji} denoting the weight from Y_j to unit X_i. Compare to the node weight $w_{ki}\ of$ Guresen and Kayakutlu (2011).
b_j	Bias on unit Y_j. A bias acts like a weight on a connection from a unit with a constant activation of 1.
y_in_j	Net input to unit Y_j with $y\_in_j = b_j + \sum_i^n x_i w_{ij}$, for m number of Y_j units, where n is number of X_i units.
W	Weight matrix: $W = [w_{ij}]$.
$\mathbf{w}_{.j}$	Vector of weights: $w_{.j} = [w_{1j}, w_{2j}, \dots, w_{nj}]$, given for the jth column of the weight matrix, W, with n weights.
$\|\mathbf{x}\|$	Norm or magnitude of vector **x**.
θ_j	Threshold for activation of neuron Y_j: A stepwise function sets the activation of a neuron to 1 when its net input is greater than the specified threshold value θ_j, otherwise the activation is 0.
s	Training input vector: $\mathbf{s} = [s_1, \dots, s_i \dots, s_n]$, for n number of X_i units.
t	Training (or target) output vector: $\mathbf{t} = [t_1, \dots, t_j, \dots, t_m]$, for m number of Y_j units.
x	Input vector (to classify via the net or to get response): $\mathbf{x} = [x_1, \dots, x_i, \dots, x_n]$, for n number of X_i units.
y	Output vector (for the current output of the network):

	$\mathbf{y} = [y_1, \dots, y_j, \dots, y_m]$, for m number of Y_j units.
Δw_{ij}	Change in w_{ij}: $\Delta w_{ij} = w_{ij}$(new)-$w_{ij}(old)$.
α	Learning rate: used to control the amount of weight adjustment at each training step.

(Fausett, 1994)

p	Input pattern vector for a multilayer network.
a	Output vector for the actual output of the network.
e	Error vector: difference between the target and current output $\mathbf{e} = \mathbf{t} - \mathbf{a}$.

(Fick, 2006)

Some definitions promote the understanding of the parameters that are set as controls for the neural network and algorithm (Sharma, 2019b):

- Epoch: An epoch refers to one cycle through the full training dataset (DeepAI, 2019). For backpropagation, one epoch is when the entire training dataset is passed forward and backward through the neural network once.
- Batch: Should the entire training dataset be too big to feed the network at once, it is divided into smaller batches. For purposes of this research, a training dataset can also be divided into batches as part of the validation process (see section 4.3.2).
- Batch size: The total number of training examples, or pattern inputs, or records in the training dataset is used for the batch size.
- Iterations: The number of batches needed to complete one epoch is an iteration. This is not the same as 'iterations' of unsupervised learning methods, such as *k*-means clustering (see section 4.4.1)
- Number of batches: Not to be confused with the batch size, the number of batches is the total of all the batches used across the entire range of training data. For large training datasets, the number of batches is the same as the number of iterations.

To implement an ANN algorithm successfully, the learning process need to be controlled by additional parameters, called hyper-parameters. The following algorithm hyper-parameters are specified for the learning process:

- **Learning rate** is used to specify how fast the backpropagation algorithm performs gradient descent. A lower learning rate makes the network train faster but might result in missing the minimum of the loss function (MissingLink.ai, 2018). However, a learning rate that is too high may cause the gradient descent to increase rather than decrease the training error (Goodfellow *et al*., 2016).
- **Momentum** works by waiting after a weight is updated, and updating it a second time, using a delta amount (Missinglink.ai, 2018). The influence of past cycles is incorporated into the present weight update (Moreira and Fiesler, 1995). The momentum is a value between 0 and 1 showing how much the influence the previous weight has on the current weight calculation. This variant of the stochastic gradient descent algorithm is used to reduce oscillation (MissingLink.ai, 2018) or prevent the vanishing gradient problem (see section 3.2.1).
- **Validation** is done after a number of cycles. This number were specified before the first validation cycle was done. The number of cycles per validation test is also specified. Should there not be a validation test dataset, a number of examples can be randomly specified from the training dataset to use for validation. The validation results are rounded to the nearest decimal before it is tested for correctness.
- **Stopping criteria** is set, in terms of error threshold, validation or fixed number of cycles. The algorithm for one training dataset batch can be set to stop when an average error, or all the errors for each cycle, is below a certain value. Alternatively, the algorithm stops when a portion of the validation data output is within a close region of the desired output, or exactly the same after rounding.

For gradient descent algorithms, a neuron is needed that produces a differentiable function of its inputs, called the activation function. If the purpose of the network is classification, and output labels are mutually exclusive and each input has one label, the output activation function to use is softmax (Media, 2018). For the softmax activation function, a form of logistic regression normalises an input value into a vector of values with a probability distribution that sums up to one (Mahmood, 2019). If the inputs for the classification have multiple labels and the output classes are not mutually exclusive, a sigmoid function is suggested for each output (Media, 2018).

The sigmoid neuron unit is closely related to a perceptron, but with a curved, differentiable threshold function (Mitchell, 1997). Like a perceptron, the sigmoid unit first computes a linear combination of the inputs, or sum of the weighted inputs, as shown by *net* in Figure 3.4. Then a threshold is applied to the result. This threshold output for a sigmoid unit is a continuous function of its input.

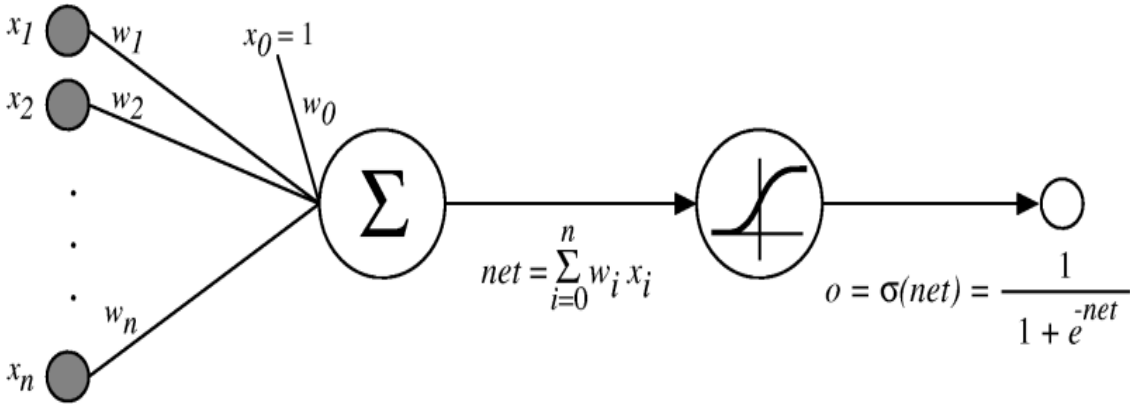


Figure 3.4: The sigmoid threshold neuron unit (Mitchell, 1997)

As i**llustrated** in Figure 3.4, the sigmoid unit computes its output o as

$o = \sigma(\vec{w}.\vec{x})$,

where

$$\sigma(y) = \frac{1}{1 + e^{-y}} \tag{3.1}$$

The function for σ is called the sigmoid, or logistic function (Mitchell, 1997).

Its output ranges continuously between 0 and 1, increasing monotonically (without decreasing at any point) with its input (Mitchell, 1997). Because it maps a very large input to a small range of outputs, this function is often referred to as *squashing* the network input (Mitchell, 1997).

The derivative of the sigmoid function is easily derived from its output, as (Maladkar, 2018):

$$\frac{d\sigma(y)}{dy} = -\frac{1}{(1+e^{-y})^2}(-e^{-y}) = \frac{e^{-y}}{(1+e^{-y})^2},$$

$$= \frac{1}{1+e^{-y}}\left(1-\frac{1}{1+e^{-y}}\right),$$

$$\therefore \frac{d\sigma(y)}{dy} = \sigma(y).\left(1-\sigma(y)\right). \quad (3.2)$$

The ANN analysis in subsection 4.4.4. was done according to the following steps, as proposed by Chauhan (2019), Mazur (2015) and Mitchell (1997):

(1) Define independent variables and dependent variable (refer to section 5.5.1)
(2) Define hyper-parameters
(3) Define the activation function and its derivative
(4) Train the model:
   a. The number of epochs (cycles) is defined
   b. Design a feed forward network with input, hidden and output units
   c. Initilise all network weights to random values between -0.5 and 0.5
   d. *Propagate the input forward*: Each instance $\vec{x}$ in the form of a training dataset record is input to the network and the output of each neuron in the network is computed.
   e. *Propagate the errors backward*: For each network output, its error term is calculated accordingly (refer to equation 4.15 of subsection 4.3.2d). Then, for each hidden neuron, the error term is calculated. Each network weight (w_{ji}) is updated.
   f. *Find the change in total error*: To find how much the total error changed, with respect to the output, the partial derivative of the

activation function is calculated (see equation 3.2). The quantity becomes *zero* once the total error is calculated close to the output, as the output does not affect the error anymore (Mazur, 2015).

(5) Applying the model: As there are many regression and pattern recognition applications for ANN, this part is also known as pattern fitting, or making predictions. Here, the full set of application data (or analysis data) is fed into the neural network to find the outputs in the desired format; in this case, classifications.

In a back-propagation network (as used in this research), the training error of the neural network is better propagated when using multiple neurons in the output layer because each neuron can be adjusted individually (Agarwal, 2019). Using a single neuron with a sigmoid activation function would not be a better alternative, as the sigmoid function saturates values close to 0 and 1 (Aldridge, 2020).

For classification, the recommendation is that the number of neurons in the output layer should be equal to the number of classes (Aldridge, 2020). For regression, a single neuron can predict the final outcome. If there were only two classes, or binary classification, a single neuron could still be used with sigmoid activation (Agarwal, 2019).

3.2.1 Background

McCulloch and Pitts (1943) produced a model of the neuron still being used today in ANN. This model is divided into two parts: a summation over weighted inputs and an output function of the sum. In 1949, Donald Hebb published 'The Organization of Behavior', which outlines a law for synaptic neuron learning (Hebb, 1949). This law, later known as Hebbian Learning, in honour of the author, is one of the simplest and most straightforward learning rules for ANNs.

Farley and Clark (1954) first used computational machines, then called 'calculators', to simulate a Hebbian network. A year after John von Neumann's death, *The Computer and the Brain* (Von Neumann, 1958) was

published. The original paper was included in the Silliman Lectures, with a preface by Klara, his wife at the time (Von Neumann, 1957). From a more recently edited publication (Von Neumann, 2000), it can be seen that Von Neumann proposed many radical changes to the way in which researchers had been modelling the brain. These still apply today

The Mark I Perceptron was created by Frank Rosenblatt at Cornell University, in the same year as the Silliman Lectures. The Mark I Perceptron was an attempt to use neural network techniques for character recognition (Rosenblatt, 1958). Despite the early success of the Mark I Perceptron in ANN research, there were many who felt that these techniques were limited. Among the critics were Marvin Minsky and Seymore Papert, who discovered that basic perceptrons were incapable of processing the Exclusive-OR (XOR) circuit and that computers lacked sufficient power to process useful neural networks. Their book on perceptrons was used to discourage ANN research and focus attention on the apparent limitations of ANN work (Minsky and Papert, 1969).

One of the limitations was that the perceptron worked only on linearly separable classification problems in the input space (Rajasekaran and Vijayalakshmi Pai, 2003). For example, the perceptrons proposed by Rosenblatt (1958) could not learn the simple boolean function, XOR, because it is not linearly separable. In their analysis of perceptrons, Minsky and Papert (1969) specifically argued that computing XOR had to be performed with multiple layers of perceptrons, called multilayer neural nets (Kurenkov, 2015). The book by Minsky and Papert (1969) introduced the first AI winter, a period of about 40 years where funding spent on AI research was reduced. Even so, the first functional networks with many layers were published by Ivakhnenko and Lapa in 1967, as the 'Group Method of Data Handling' (Schmidhuber, 2015). However, it was Werbos's backpropagation algorithm that enabled practical training of multi-layer networks (Werbos, 1974). Later Werbos (1982) applied Linnainmaa's automatic differentiation (AD) method

to neural networks and this approach became widely used. After the silence created by Minsky and Papert (1969), the backpropagation algorithm was rediscovered by Rumelhart et al. (1986). Schmidhuber adopted a multilevel hierarchy of networks, with each level individually being pre-trained by unsupervised learning and fine-tuned by backpropagation (Schmidhuber, 1992). This followed the introduction of max-pooling to aid 3D-object recognition (Weng *et al*., 1992).

Hinton et al. (2006) proposed learning a high-level interpretation using successive layers of latent variables with binary or real values. Each layer is modelled using a restricted Boltzmann machine (a generative stochastic ANN). This stochastic recurrent neural network, popularised *inter alia* by the work of Hinton (2007a), is named after the Boltzmann distribution (Hinton, 2007b). A few years later Le et al. *(*2012) created a network that learned to recognize high-level concepts, such as cats or faces, only from watching unlabelled images.

Increased computing power from GPUs and distributed computing allowed the use of larger networks, particularly in image and visual recognition problems, which forms part of the DL domain. Graphics processing units (GPUs) make backpropagation feasible, despite the vanishing gradient problem[9] (Cireşan *et al.,* 2010). This is true of multi-layered feedforward neural networks (Scherer *et al.,* 2010). For the first time, pattern recognition systems could now be built to achieve human-competitive performance on certain benchmarks (Cireşan *et al.,* 2012).

The German traffic sign benchmark is a multi-class, single-image classification challenge held annually at the International Joint Conference on Neural Networks (Schmidhuber, 2018). Using this benchmark, the

[9] In ML, the vanishing gradient problem is found in training ANNs with gradient-based learning methods and backpropagation. The problem is that in some cases, the gradient will converge to almost zero (vanishingly small), effectively preventing the weight from changing its value. In the worst case, this may completely stop the neural network from further training (Hochreiter, 1991).

increase in successful image classifications could be monitored. Between 2009 and 2012, ANNs began approaching human-level performance in various tasks, initially in pattern recognition and ML (Markoff, 2012). Two examples in use today are Apple's Siri virtual personal assistant (Apple Support, 2020), which is based on speech recognition, and Google's Street View (App Store, 2020), which uses machine vision to identify specific addresses are.

3.2.2 Related research

The area of decision-making in a company, called strategic intelligence, covers business intelligence, competitive intelligence and knowledge management (Liebowitz, 2006). The telecom sector, specifically, relies heavily on these three areas for speedy decision-making in the dynamic arena of information and communications technology (ICT). Related research may, therefore, be found in journals that primarily deal with business and management issues. However, the topic of intelligent systems through applying AI, the broader field that ANN belongs to, received scarce attention in these journals, until recently.

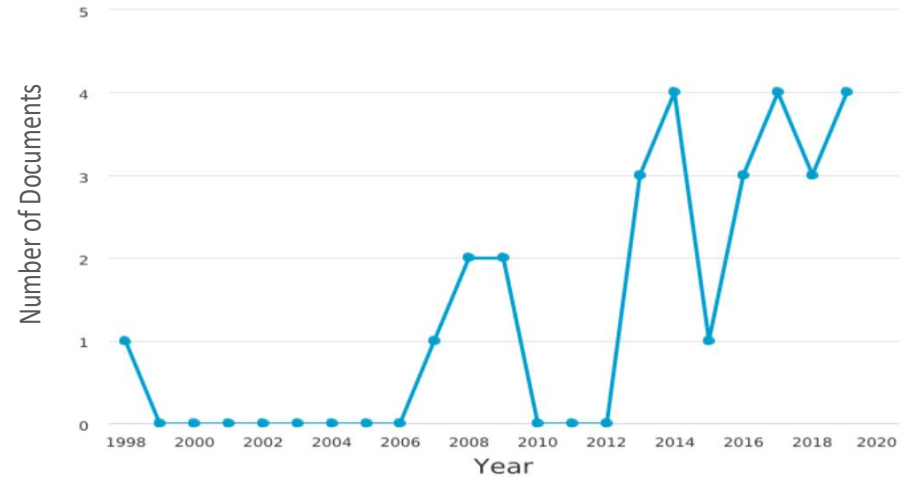
Martínez-López and Casillas (2013) conducted a basic search in Scopus[10], to find the total number of published papers from any source, on business topics including AI, intelligent systems or ML topics. This number proved to be less than 150 at the time. Of particular interest for the present study, is the segmentation and targeting of telecom business markets. Therefore, searches were conducted for research publications on the exact limitations of scope in this dissertation (refer to section 1.4.5). As expected, the searches yielded a much lower number of documents.

Figure 3.5 shows the historical progression in two searches for documents related to the topic of the present study:

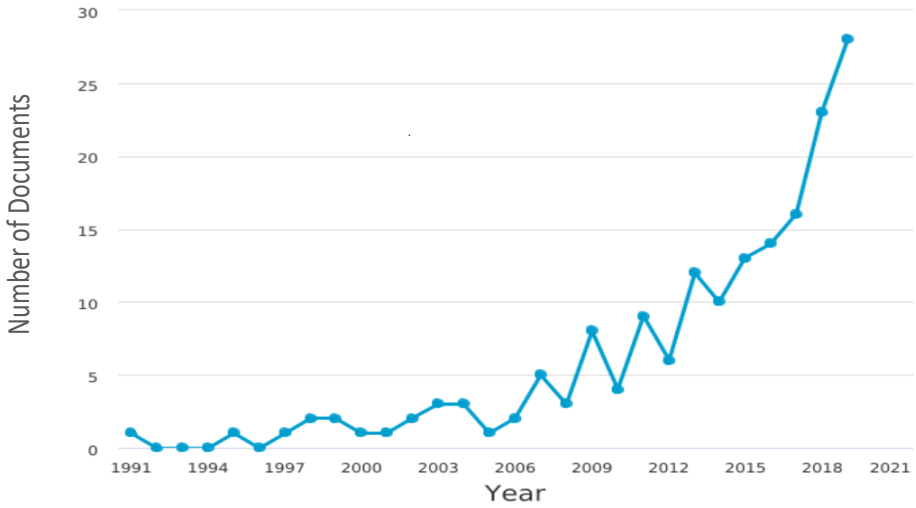
[10] Scopus is a source-neutral abstract and citation database, curated by independent subject matter experts (CIKD, 2019)

a. ANNs and *specific telecom business segmentation* and targeting, sales or marketing
b. ANNs and *general business segmentation* and targeting, sales or marketing.

It is shown that research already started in 1998, albeit with only one document published in that year. The silence in ANN research since the seventies (refer to subsection 3.2.1) was mirrored in this search. It was not until Hinton et al. (2006) introduced a high-level interpretation, using multiple ANN layers, that research started increasing again.



a. Telecom business segmentation and targeting, sales or marketing with ANN methods



b. General business segmentation and targeting, sales or marketing with ANN methods

Figure 3.5: Documents on ANN used for segmentation (Scopus, 2020)

Specific research on the application of ANNs in telecom target market segmentation was still low, however. The few research papers for specifically telecom business segmentation relate to churn prediction, mobile customer behaviour and the visualisation of market segments. Some of the related documents are mentioned below:

- **Telecommunication churn prediction using ANN**

  Based on research by Khan et al. (2019), ANNs may be utilised for the prediction of customers intending to switch over to other operators in the Pakistan telecommunications industry. In contrast to other prediction techniques, the results from an ANN approach could predict telecom churn with an accuracy of 79%.

- **Mobile customer behaviour using self organising ANN**

  In this study, Ghnemat and Jaser (2015) used self-organising maps (SOMs) to detect different usage patterns of mobile users. A large sample of customer data from a major mobile operator in Jordan was used. The study detected different behavioural segments in this market and highlights the role of data users in modern mobile markets.

- **Visualisation of market segmentation**

  Hanafizadeh and Mirzazadeh (2011) applied the integration of Delphi Fuzzy methods, Kohonen algorithms and a visualisation technique, to a market segmentation problem for visualisation and knowledge acquisition. This research was found during the business segmentation search and not the telecommunications search, although it also includes a case study of an Iranian telecommunications company.

For ANN in business segmentation and sales or marketing, a few related documents, described below, were found:

- **Mobile carrier choice behaviour analysis**

  The study of Inoue et al. (2017) proposed a new model, using ANN, to classify the future demand for mobile carriers. Research was conducted comparing three major carriers in Japan to new mobile virtual network operators.

- **Manufacturing sales prediction**

  Wang et al. (2019) used correlation analysis to verify variables affecting plastic-injection machine sales against prediction results for injection-molding machine sales at each level. ANNs were applied to obtain the prediction. Seven key external economic factors were identified to predict accurate changes in a company's annual sales prediction.

- **Quantitative assessment of taxi industry**

  This study by Zhang et al. (2017) proposed the concept of taxi industry health gradation. A four-layer criteria set was developed and weights were determined through the Analytic Hierarchy Process (AHP). Then a fuzzy evaluation set with five categories, is used to evaluate daily performance of a taxi industry.

  To simulate the ambiguous process of decision making, a three-layer feedforward neural network with BP was constructed. The result was compared with the result of fuzzy comprehensive evaluation through sensitivity analysis. The aim was to develop a more structured and efficient evaluation tool. With the GPS and Taximeter dataset of taxicabs in Wuxi, China, the model was applied to empirical studies (Zhang *et al.,* 2017).

4. METHODOLOGY

Before the data can be prepared and the hypotheses set up for the evaluation criteria, the framework in which the analysis takes place needs to be established. This is described first, in the form of segmentation schemes in order to analyse the quantitative segmentation methods described in the literature study. The data sourcing and preparation of the data used in the research are explained in this section. Parameters used for analysis are important factors to consider and are described before the evaluation criteria. Lastly, the method of analysis, including the computing environment, is described in more detail.

4.1 Segmentation Schemes

In the field of marketing, a segmentation scheme is defined as a process used whereby customers are separated into groups by some differentiating factor so that the user is able to target them with relevant messaging (May and Smith, 2012). The differentiating factor mentioned here is fundamental to the segmentation scheme selected.

There are four factors or bases of market segmentation for consumer markets, namely geographic, demographic, psychographic and behavioural. A segmentation scheme exists for each of these factors, for example, a scheme may be called segmentation by demographics or geographic segmentation.

Unlike consumer markets, B2B target markets may have up to six standard segmentation schemes (Nguyen, 2012), namely:

- Segmentation by geographic base / reach;
- Segmentation by industry / sub-industry / industry served / customer served;
- Segmentation by product class / product usage;
- Segmentation by organisation size (measured by, for example, revenue or number of employees);

- Segmentation by product delivery model / product format / packaging format / special technology / process methodology; and
- Segmentation by special use/needs.

In other words, the B2B target market may be divided in distinct groups based on (Willan, 2014):

- Company features, characteristics or who they are;
- Behaviour, the way business is done or what they do;
- Needs, requirements or what they want; and
- Attitudes, sentiment (towards a product), values or what they think.

A scheme based on attitudes was not considered for this research, as this would have required a sentiment analysis of consumers within different companies, instead of focussing on the companies themselves.

The segmentation scheme used for this research is based on the remaining three groups and may be summarised as follows:

1) It divides a target market into distinct groups based on features of the company, buying behaviour of employees of the company and needs of customers in terms of the type of products.
2) It uses a combination of standard B2B segmentation schemes, excluding the following schemes described by Nguyen (2012): product delivery, format, packaging, special technology and process methodology.
3) Segmentation of the enterprise target market was defined as firmographics (geographic information, industry sector and organisation size) and product class, evaluating the buying power of existing customers within this market.

4.2 Approach

When considering a segmentation scheme, it is important to first decide on which approach will be used for segmentation. Three major approaches were found in literature, corresponding to three types of segmentation schemes, as mentioned in section 4.1. Each approach defines the measures used for

each segmentation scheme to the to segmentation in literature (Nguyen, 2012):

- **A priori segmentation** (customer sizing), the simplest approach, currently popular in some telecommunication companies. This classification scheme is based on publicly available data – for example industry and company size – to create distinct groups of customers within a market.
- **Value-based segmentation** differentiates customers according to their buying and usage behaviour, or economic value. For example, it can be based on turnover, ICT expenditure and share of wallet (% expenditure into the selling company). Customers with the same value level are grouped into individual segments that can be distinctly targeted.
- **Needs-based segmentation** is based on different needs that customers express for a specific product or service being offered, or a desired customer relationship. The needs are discovered and verified through primary market research.

Using only the easiest approach, namely a priori segmentation, may not always be valid, since companies in the same industry and of the same size may have very different needs. It is far more powerful to segment according to customer behaviour and needs as well.

Unfortunately, it is not easy to identify the needs-based segment. Efforts need to be invested into fully understanding customers and developing a comprehensive segmentation model (Willan, 2014). Understanding customers requires considerable primary research and data collection to fully discover and correlate customer needs with their characteristics. Nguyen (2012) asserted that such research is particularly difficult for B2B technology-enabled services, such as those provided by a telecommunication company, because the buyers are typically complex, multitiered organisations that cannot easily be modelled, or predicted like consumers.

It is important to note that because the whole market or TAM it is not being segmented, the research will not aid to fully divide its total market into distinct, needs-based, homogenous segments. Nevertheless, the research can be used to identify homogenous groups of prospects that are most likely to become profitable customers. In a lesser sense, the research results may assist a company with identifying and prioritising previously unknown target market segments. The scheme followed by this research is to combine a priori (or feature-based) and value-based approaches to segmentation (Nguyen, 2018). This will allow the telecommunication company to clearly define and target its best prospects. The aim is to satisfy most of the segmentation needs without becoming too time- and resources consuming.

4.2.1 Data sourcing

From personal experience, the information needed to analyse potential markets requires significant resources that many developing B2B companies do not have. This has also been confirmed by Nguyen (2018) and is true for most B2B companies in Africa.

Consequently, the target market for this research is sourced from secondary data for potential companies in Tanzania. This country was chosen due to the distinctive challenges in marketing that exist there. It is not an extensively researched market, such as South Africa, and the researcher is familiar with the competitive environment of the telecommunication industry in that country.

It is very important to note that firmographics for companies in Tanzania is very difficult to obtain. An initial list of companies were obtained from Rasello research (2017). The list was matched with customer data from a major telecommunication provider, where some of the research was conducted. Existing customers on the initial list of companies were identified in this way. The two datasets used were:

- A target market dataset comprising 3 362 distinctive companies with 827 of these companies having subscribers with the telecom provider.

This represents 95 110 subscribers out of a total of 2 062 466 company employees.

- A customer base of 8 292 companies with 281 772 subscribers.

It was, however, necessary to source a combination of data providers to enhance missing values. The data providers used were:

- Rasello research (2017) – provides company data for marketing in selected countries that form part of Eastern Africa, e.g. Kenya, Tanzania, Uganda, Ethiopia, Rwanda and Somalia. Access was provided by the researched telecommunication provider.
- Dow Jones Factiva (2016) – listed company metrics, with business information provided per country globally. Access was provided by the researched telecommunication provider.
- Who Owns Whom research, (2015) – independently researched business metrics for a large collection of holding companies with a footprint in Africa. Access was provided by the researched telecommunication provider.

The following data sources were used to verify the validity of market data for the geographic and industry information of companies:

- Business Monitor Online (BMI, 2018) – global company data and economic indicators. Access was provided by the University of the Witwatersrand (Wits) library.
- IRESS (2019) – online marketing and financial information on JSE listed companies. Access was provided by the Wits library.
- Dun & Bradstreet (2019) – worldwide company / economic information on corporate businesses, originator of the DUNS[11] number. Specific company searches were conducted on a publicly available website.

[11] The Dun & Bradstreet D-U-N-S Number is a distinctive, nine-digit identifier for businesses. The D-U-N-S Number is used as the starting point for any company's Live Business Identity, the most comprehensive and continually updated view of any company in the D&B Data Cloud (Dun & Bradstreet, 2019).

In a number of cases, the metrics were reliant on country specific information about Tanzania, such as gross domestic product (GDP) and population statistics. These were obtained from:

- Trading Economics (2017) – country economic indicators and forecasts, publicly available.
- World Bank (2017) – demographic and economic indicators on countries worldwide. Access was provided by the Wits library.

Input data without enough company-related information were excluded. Customer data for the identified enterprise businesses was provided by the Tanzania office of the researched telecommunication service provider.

To test the various segmentation applications, the multivariate Iris flower dataset or Fisher's Iris dataset was used initially (Fisher, 1956). This dataset was introduced by the British statistician and biologist Ronald Fisher in 1936. The dataset is sometimes called Anderson's Iris dataset because Edgar Anderson collected the data to quantify the variation of related Iris flower species (Anderson, 1935). The dataset consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). It is frequently used as benchmark for classification (Eberhart *et al.,* 1996). The dataset is available at the University of California, Irvine data repository (Dua and Graff, 2019).

4.2.2 Ethical considerations

Permission was obtained from the University of Witwatersrand ethical committee before starting with data sourcing. This permission was reliant on legal approval, as well as approval from the enterprise sales and human resources departments at the telecommunication service provider where customer data was obtained. Clearance was subject to the anonymity of customer data and non-disclosure of competitive information being guaranteed.

4.2.3 Data preparation

The customer segmentation process followed in the research analysis used a representative sample of the target market. A preliminary analysis was done on the company data from the target market and the telecom service provider customer base, to match data. MS Excel 2016 was used to obtain summary statistics.

From the two datasets there seem to be related variables having different trends. This is indicative of the independent nature of the values. For example, the regional distribution of the target market vs the customer base were compared. The number of companies, number of employees, number of customers and number of subscribers were counted. The regional distribution was then calculated as the number of entities per region in proportion to the total entities in each of the entity types mentioned here.

The regional distribution is depicted in Table 4.1

Table 4.1: Regional distribution of target market

Region	Companies	Employees	Customers	Subscribers
Central	2.3%	1.4%	0.1%	0.0%
Coast	2.7%	2.4%		
Dar Es Salaam	77.8%	82.1%	**87.3%**	* 76.5%
Lake District	4.8%	5.9%	**5.2%**	1.6%
North	11.6%	7.3%	7.3%	**21.8%**
South West	0.9%	1.0%	0.1%	0.03%

***** This sales region includes Coast and Dar Es Salam

The shaded percentages indicate where the proportion of number of entities per region to total entities are more in the customer base than in the market data. A conclusion can be made that the number of customers in Dar Es Salaam and the Lake District is proportionaly higher than expected, when compared to the target market proportions in these regions.

The market distribution on the left in Table 4.1 looks more consistent than on the right, with a similar trend in number of employees per region than in the number of companies per region (Pearson's r value = 0.99).

It is interesting to note that there is a greater difference between number of subscribers and number of customers per region (Pearson's r value = 0.97). This is due to subscribers not being signed up for telecommunication services as consistently as employees in companies, per region.

It seemed that a more complete picture of the target market data could be obtained by matching the customer base to the target market. Where the target market base overlaps with the telecommunication provider's customer base, customer parameters can be included in the target market base. Parameters such as number of subscribers, revenue and average revenue per unit (ARPU) can be added where customers and target market companies are the same.

The parameters from the customer base cover data relevant to the telecommunications measurement of performance over a period of three months, namely January to March 2019. The average over the three months was used for analysis, e.g. the value for the estimated number of subscribers is the average of the subscribers per month, taken over three months. The input data from the target market dataset consisted of variables suitable for analysis, stored in an analysis file designated for this research.

Before any segmentation method can be tested, the raw data was evaluated by way of descriptive statistics during the exploratory data analysis. The raw input data from the analysis file was assessed in detail by means of graphs and scatter plots, and by considering statistics such as mean values, variances, correlations and quantiles. The results show whether variables are suitable for analysis, and whether natural clusters exist in the dataset. This stage also indicates whether the data needs to be transformed before clustering, or can be used as is. The results of these are shown in section 5.1. Detailed results of descriptive statistics and test runs are presented in Appendix B.

4.2.4 Analysis variables

Extracts from the input target data and variable descriptions are depicted in Appendix A. Section A.2 of Appendix A contains a list of the variables from the analysis file. In this file only the features applicable to the specific approach used for a test, were selected.

The nominal and ordinal categorised variables for location priority and SIC code were included as per the company preferences. Both contain numeric codes which can be evaluated for clustering or as categories.

In telecommunications, expenditure on ICT is frequently used as a measure of a potential customer's buying power. This feature is called ICT spend.

4.3 Evaluation criteria

To compare the segmentation methods used on the company data, the emphasis is primarily on fit for purpose and the quality of the solution. These are explained below. In addition, the processing time for each method is also measured as a secondary objective. In order for the algorithm to end before the number of clusters or cluster quality declines, it is also necessary to apply stopping criteria.

4.3.1 Fit for purpose

The results are measured against two hypotheses to assess the suitability of the method used.

a. Segmentation hypotheses

In order to utilise a customer segmentation process, with the best method chosen, it is important to establish clear hypotheses that will serve as the foundation of the analysis. The segmentation hypotheses should be formed around company characteristics or features that allow the division of targeted companies into distinct value-based segments. These are not exhaustive and should be treated as a quick way to test the method's segmentation capability.

After consultation with the telecom service provider where research was conducted, the author formulated the following sample hypotheses to be tested:

1. Companies with high ICT expenditure will be in a different segment than those with low ICT expenditure.
2. Manufacturing and financial companies will be in different segments.
3. Companies with high smartphone usage will be in the same segment as financial or technological companies.
4. Medium to Small companies (with less than 100 employees) will have fewer or more other devices than mobile phones, depending on the outcome of the analysis.

b. Research hypotheses and questions

The research hypotheses as given in the problem statement for this study are used as the second fit for purpose criteria to evaluate each segmentation method. The aim is to assess the practicality of the research, given the segmentation capability. Through these hypotheses the capability of the researched methods to be applied to business is evaluated. A percentage rating was assigned to hypothesis for each segmentation method, after evaluating the output of the test runs. As the ability to apply the research to business, and not a business itself is evaluated these criteria are not called business hypotheses, to ensure impartiality.

The research question from the problem statement for this study was broken down into separate, measurable research questions. These were used to evaluate how the results of the research relate to implementing the method. In Table 4.2., the hypotheses and research questions are labelled. Each analysis method was evaluated on how close it answers each research question.

Table 4.2: Hypotheses and research question labels

Label	Segmentation Hypotheses
Segment H1	1. Companies with high ICT expenditure will be in a different segment than those with low ICT expenditure.
Segment H2	2. Manufacturing and financial companies will be in different segments.
Segment H3	3. Companies with high smartphone usage will be in the same segment as financial or technological companies.
Segment H4	4. Smaller companies will have less or more other devices than mobile phones, depending on the outcome of analysis.

Label	Research Hypotheses
Research H1	1. The method for classifying the business as part of a group of businesses, is quantitative, repeatable and may be used to reduce or eliminate manual intervention in classifying businesses.
Research H2	2. The chosen method is practically useful in a specific company in the telecoms industry for targeting enterprise customers.
Research H3	3. Each business with similar properties (or in the same segment) may be targeted through the appropriate sales channels.

Label	Research Questions
Research Q1	1. How robust and repeatable are the results based on input parameters identified?
Research Q2	2. How clearly understandable for business is the method used for segmentation?
Research Q3	3. How well will the segmentation process be managed to ensure implementation of the chosen method for segmentation?
Research Q4	4. How much buy-in from sales management will there be?
Research Q5	5. How well will the results translate to the appropriate sales channel used to enable an end-to-end sales process?
Research Q6	6. How effectively can rules be formulated, as a result of the analysis?
Research Q7	7. How easily can adjustments be made to the business rules for segmentation or future enhancements?

4.3.2 Measuring the process

Applying an algorithm of a quantitatve segmentation method on a specific target dataset is defined in this study as a test run. One test run contains a number of iterations, which can be specified to reach a limit as per subsection 4.3.4b. The algorithm is executed through a process followed by the specific tool used in the test runs (see section 4.4). While the number of iterations may be an indication of the effectiveness of the algorithm, the efficiency of the process is measured by the processing time.

a. Iterations

According to the free dictionary (Collins, 2014) an iteration is a computational procedure in which a cycle of operations is repeated, often to approximate the desired result more closely. In the case of ANNs, an

iteration according to this definition aligns more to an epoch (the same as a cycle), as defined in section 3.2.

b. Processing time

Processing time (CPU time) is the time taken for a method to be processed until a solution is provided. The same computing environment was used to do the test runs in each case so that the processing times are comparable. Note that a segmentation method or algorithm was executed a couple of times to compare outputs and the number of parameters needed. The processing time was only taken once appropriate parameters for the method had been established and the optimal grouping found for a specific method. The test runs were conducted by running the specific software tool on an Intel(R) Core(TM) i5-8350U CPU@1.70GHz (1.90 GHz processing) computer with 16GB RAM and 64-bit processor, using a Windows 10 operating system.

c. Reverse scaling

For the processing time and iterations, the measurements were reverse scaled into a score to align them with the quality measurement scale. In short, this meant that higher iterations or processing time should have lower scores and vice versa. There are different ways to perform this scaling, such as taking the reciprocal value, or subtracting the value from an upper limit and transforming it. Equations generated by the author for transforming the processing time and number of iterations (or nodes) to a scale [0, 1] in reverse order, are shown below.

$$Ts(r) = \frac{(\max_i Tp(i) - (Tp(r) - \min_i Tp(i))}{\max_i Tp(i)} \times w \qquad (4.22)$$

In the above equation, $Ts(r)$ is the processing time score, $Tp(i)$ is the processing time for each run from the set of runs and $Tp(r)$ is the processing time for the current run. A built in weighting, w based on minute and second conversions, was applied to the calculation to ensure the time score was not reported as hh:mm:ss, but numerically.

$$Is(r) = \frac{(\max_i In(i) - (In(r) - \min_i In(i))}{\max_i In(i)} \tag{4.23}$$

In equation (4.23), $Is(r)$ is the iterations score, $In(i)$ is the number of iterations for each run from the set of runs and $In(r)$ is the number of iterations for the current run. Instead of a weighting, as no conversion is needed, the score is given in proportion to the maximum number of iterations or nodes; hence, the denominator $\max_i In(i)$ above. In both equations, $i \in \{1, \dots, \max(r)\}$, that is, the index for test runs or member of the set of test runs ranges from 1 to the index of the run with the maximum number of iterations.

4.3.3 Quality of output

To validate the reliability of clusters certain performance measures are used for each method. Not all performance measures (or metrics) are suitable for the analysis. For example, indices that may be used for testing the validity of clusters are the Dunn index (Dunn, 1974) and the Davies–Bouldin index (Davies and Bouldin, 1979). However, both these techniques have shortcomings. As the number of clusters and parameters of the data increase, the iterations and, therefore, computations for the Dunn index (with high value being better) also increase. A good value reported by the Davies–Bouldin index, being a low value, does not always reflect how well the clustering has been performed. For clustering it is, therefore, better to start with the silhouette as index or measurement. In similar vein performance measures most applicable to the other segmentation methods were used, as described next.

a. Silhouette index

The silhouette value is a measure of the similarity of an object within a cluster (tightness), compared to other clusters (separation). It may be used to study the separation distance between the resulting clusters and provides a way of assessing parameters. According to Rousseeuw (1987), the originator of this index, the average silhouette width provides an evaluation

of clustering validity, and might be used to select an appropriate number of clusters. The silhouette validation technique calculates the silhouette index for each sample, average silhouette index for each cluster and overall average silhouette index for a dataset. If the silhouette index value is high, the object is well matched to its own cluster and poorly matched to neighbouring clusters. Figure 4.1 gives an illustration of the construction of silhouette indices described by Rousseeuw (1987).

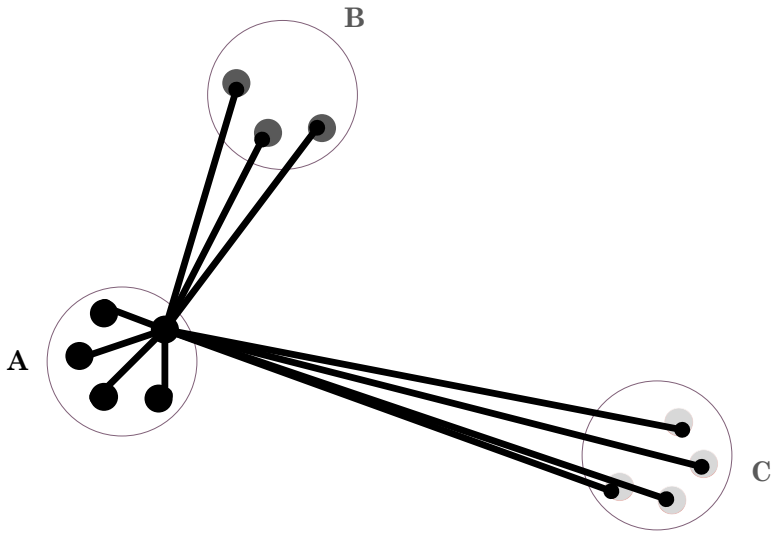


Figure 4.1: Constructing a silhouette index (Rousseeuw, 1987)

Take any object i in the dataset, and denote by A the cluster to which it has been assigned. When cluster A contains other objects apart from i, then the average difference between i and all other objects of A can be calculated as $a(i)$.

In Figure 4.1, this is the average length of all lines within A. Consider now any cluster C which is different from A, and calculate $d(i, C)$ as the average difference between i and all objects of C. In Figure 4.1, this is the average length of all lines going from i to C. After computing $d(i, C)$ for all clusters $C \neq A$, select the smallest of those numbers and denote it by $b(i) = \min_{C \neq A} d(i, C)$. The cluster B now contains objects, with $b(i)$ the smallest mean distance between i and all points in any other cluster of which i is not a member. This cluster is called the neigbour of object i, as it is the next best

fit cluster for point i. It is very useful to know the neighbour of each object in the dataset. Note that the construction of $b(i)$ depends on the availability of other clusters apart from A. Therefore the number of clusters k need to be more than one. The silhouette index is obtained by combining $a(i)$ and $b(i)$:

$$s(i) = \begin{cases} 1 - \dfrac{a(i)}{b(i)}, & if\ a(i) < b(i) \\ 0, & if\ a(i) = b(i) \\ \dfrac{a(i)}{b(i)} - 1, & if\ a(i) > b(i) \end{cases}, \tag{4.1}$$

which yields the assertion

$-1 \leq s(i) \leq 1$,

for each object i, with $s(i) = 0$ if A contains only 1 object ($i = 1$ only). Equation (4.1) may be written as one formula:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad if\ |A| > 1 \tag{4.2}$$

Interpretation of the silhouette index follows (Rousseeuw, 1987):

- If $s(i) \to 1$, the object i is 'well clustered', and the 'within' distance $a(i)$ is much smaller than the 'between' distance $b(i)$.
- If $s(i) \to 0$, the object i could be assigned to another cluster closest to it and lies equidistant from both clusters A and B. This indicates overlapping clusters, and the object i is considered 'intermediate'.
- If $s(i) \to -1$, the object i lies on average much closer to B than to A, as $a(i)$ is much larger than $b(i)$. Then i is 'misclassified' and merely placed somewhere in between the clusters.

The silhouette is more likely to be maximised (closest to the upper bound of 1) at the correct number of clusters. The number of clusters k for the clustering algorithm may be determined before running the process. A few calculations of the average silhouette for each value k will show the k which has the maximum value for $s(i)$. This k is chosen as the optimal number of clusters.

b. Density measurement

For PSO, the density of a swarm cluster is measured. The density can be tested with a measurement of the smallest error difference within a cluster, or the fitness value. The fitness function in PSO is the objective function calculated at each iteration. In their paper on data clustering using PSO, Van der Merwe and Engelbrecht (2003) described a swarm as a number of potential solutions to the optimisation problem, and a particle is seen as a potential solution. The aim is to find the potential solution (the position of a particle) that results in the best evaluation of a given objective (or fitness) function.

The fitness function may be specified as a minimisation function, defined as the within-cluster sum of squares, or Euclidian distance from a cluster centre. It may also be a function to maximise, for example, the inter-cluster distance, i.e. the distance between the centroids of the clusters. Minimisation functions that may be used as fitness measurement are the sphere and Rosenbrock functions (Jamil and Yang, 2013), and the Rastrigin function (Dieterich and Hartke, 2012). The equations for these functions are shown next.

$$\textbf{\textit{Sphere function}}: \quad f(\mathbf{x}) = \sum_{i=1}^{d} x_i^2 \ , \quad with\ x_i \in [-5.12, 5.12]. \tag{4.3}$$

$$\textbf{\textit{Rosenbrock function}}: \quad f(\mathbf{x}) = \sum_{i=1}^{d-1} (a - x_i)^2 + b(x_{i+1}^2 - x_i)^2 \ , \quad with\ a = 1\ and\ b = 100. \tag{4.4}$$

$$\textbf{\textit{Rastrigrin function}}: \quad f(\mathbf{x}) = Ad + \sum_{i=1}^{d} b(x_i^2 - A\cos(2\pi x_i))^2 \ , \quad with\ A = 1\ and\ x_i \in [-5.12, 5.12]. \tag{4.5}$$

Taking guidance from Van der Merwe and Engelbrecht (2003), the quantisation error is used as a fitness function for this project. The quantisation error is described as least square quantisation, showing that the origin of this objective function may be traced back to Lloyd (1982).

In the context of PSO clustering, a single particle represents the N_c cluster centroid vectors. Here, N_c denotes the number of cluster centroids provided as input to the PSO algorithm, i.e. the number of clusters to be generated. Each particle described in this section is then presented, in terms of the position vector:

$$\boldsymbol{x}_i = \left(\boldsymbol{m}_{i1}, \dots, \boldsymbol{m}_{ij}, \dots, \boldsymbol{m}_{iN_C}\right) , \tag{4.6}$$

where $\boldsymbol{m}_{ij}$ refers to the j^{th} cluster centroid vector of the i^{th} particle in cluster C_{ij}. Therefore, according to Van der Merwe and Engelbrecht (2003), a swarm represents a number of candidate clusterings for the *current* data vectors.

The fitness of particles for the p^{th} data vector $\mathbf{z}_\text{p}$ is measured as the quantisation error:

$$J_e = \frac{\sum_{j=1}^{N_c}\left[\sum_{\forall \mathbf{z}_\text{p}} \mathbf{d}(\mathbf{z}_\text{p}, \boldsymbol{m}_j)/\left|C_{ij}\right|\right]}{N_c} , \tag{4.7}$$

where $\left|C_{ij}\right|$ is the number of data vectors belonging to cluster C_{ij}, the subset of data vectors forming cluster j of particle i, and

where **d** is the distance of the data vector to the centroid.

The distance vector **d**, is defined as:

$$\mathbf{d}(\mathbf{z}_\text{p}, \boldsymbol{m}_j) = \sqrt{\sum_{\boldsymbol{k=1}}^{\boldsymbol{N_C}}\left(\mathbf{z}_{\boldsymbol{pk}} - \boldsymbol{m}_{\boldsymbol{jk}}\right)^{\boldsymbol{2}}} , \tag{4.8}$$

where k is a subscript for each dimension. is minimised. In the algorithm that Ballardini (2018b) applied, the fitness measure J_e, or swarm fitness, starts with a default limit of ∞, that becomes smaller progressively while it is minimised.

c. Null hypothesis testing

For CHAID, the χ^2 hypothesis testing technique is built into the method, and the test against the null hypothesis is used as a measurement to ensure pure classification structures (through decision tree nodes). According to Kass (1980), if the dependent variable is continuous, the F-test is used and, if the dependent variable is nominal or categorical, the χ^2 test for independence is used.

Each pair of predictor categories is assessed to determine which is least significantly different, with regard to the dependent variable.

The null hypothesis and alternate hypothesis used here may be stated as:

H_o: Predictor variable A and predictor variable B are independent.

H_{a}: Predictor variable A and predictor variable B are *not* independent.

Due to these steps of merging (refer to Figure 3.1a), a Bonferroni adjusted **p**-value is calculated for the merged cross tabulation (Kass, 1980).

In this study, all continuous variables used in the CHAID analysis were transformed to categorical variables. Therefore, only χ^2 adjusted **p**-values were used. The measurement formulas below, are given in terms of the χ^2 independence test described by (Statistics Solutions, 2020).

Degrees of freedom: The degree of freedom (DF) is calculated as:

$$DF = (r - 1) \times (c - 1), \tag{4.9}$$

where DF is the degree of freedom,

r = number of rows,

c = number of columns.

In equation (4.9), r is the number of levels for one categorical variable (A), and c is the number of levels for the other categorical variable (B).

Expected values: For the expected value of the two categorical variables, frequency counts are computed separately for each level of categorical variable A at each level of categorical variable B (Kass, 1975a).

$$E_{ij} = \frac{\sum_{k=1}^{r} O_{kj} \sum_{k=1}^{c} O_{ij}}{N} \quad or \quad E_{rc} = \frac{(\mathrm{n_r} \times \mathrm{n_c})}{N}, \tag{4.10}$$

where E_{ij} is the expected value,

$\mathrm{n_r} = \sum_{k=1}^{r} O_{kj}$ is the total k^{th} row observations (A),

$\mathrm{n_c} = \sum_{k=1}^{c} O_{ij}$ is the total i^{th} column observations (B),

$N =$ total observations or sample size.

Test statistic: The test statistic is a chi-square test variable (χ^2) defined by the following equation.

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{\left(O_{ij} - E_{ij}\right)^2}{E_{ij}} \tag{4.11}$$

where χ^2 is the chi– square test of independence,

$O_{ij} = t$he observed value of variable A at row i and variable B at column j,

$E_{ij} =$ the expected value of variable A at row i and variable B at column j.

Probability for null hypothesis: The **p**-value is the probability of observing a sample statistic as extreme as the test statistic, where a test statistic is computed and compared to a critical value. The critical value for the χ^2 statistic is determined by the level of significance (typically 0.05) and the degree of freedom. If the observed χ^2test statistic is greater than the critical value, the null hypothesis may be rejected (Kass, 1975b).

With the critical value (CV) in terms of the degree of freedom (DF) and level of significance (alpha-level or α-level), the null hypothesis probability is shown to be $\boldsymbol{p} = P(\chi^2 > CV)$, and the hypothesis tests will be satisfied, if

$$H_o: \boldsymbol{p} > \alpha \text{ , or for } n \text{ comparisons, with Bonferroni correction, } \boldsymbol{p} > \frac{\alpha}{n}, \tag{4.12}$$

else

$$H_a: \boldsymbol{p} \leq \alpha \text{ , or for } n \text{ comparisons, with Bonferroni correction, } \boldsymbol{p} \leq \frac{\alpha}{n}. \tag{4.13}$$

This **p**-value is a measure of the dependence between predictors, and is calculated during the algorithm for each pairwise comparison. The most similar pair will have the largest **p**-value, with respect to the dependent variable.

d. Cross-validation

Cross-validation is a technique used to test the effectiveness of ANN models, especially on limited data (Sanjay.M, 2018). Sometimes called rotation estimation, cross-validation is not a technique by itself, but the use of a combination of similar model validation techniques. The aim is to assess how closely the results of a model generalise to an independent dataset (Kohavi, 1995). In supervised learning applications, the generalisation error is a measure of how accurately a model is able to predict outcome values of previously unseen data. The objective of a supervised ANN model is to 'generalise' successfully, and using cross-validation together with error metrics to estimate the 'generalisation error' is, therefore, recommended. The generalisation error is the difference between the 'true error' and the 'apparent error'. According to Weiss and Kulikowski (1991), the true error is determined from a large number of new data points that converge asymptotically to the actual population distribution (as cited by Twomey and Smith, 1995). The true error is the difference between the actual value and the approximate value from a model (Kaw, 2012). The true error (E_t) is distinguished from the 'apparent error' (E_a), being the error when validating a dataset used to construct the ANN model, or training dataset (Efron, 1983) as cited by Twomey and Smith (1995). The testing error (T_e), also different from E_t, is the error of the neural network when validating a dataset *not* used to construct the ANN model, or test dataset. Since true error needs an approximation of the actual population, it can never be determined from sample training or test datasets. Hence, the true error is estimated from the apparent and/or testing error, using a positive bias term (Twomey and Smith, 1995):

$$E_t = E_a + b \tag{4.14}$$

The true error, in relation to the apparent error is shown in the diagram below, illustrating the generalisation gap.

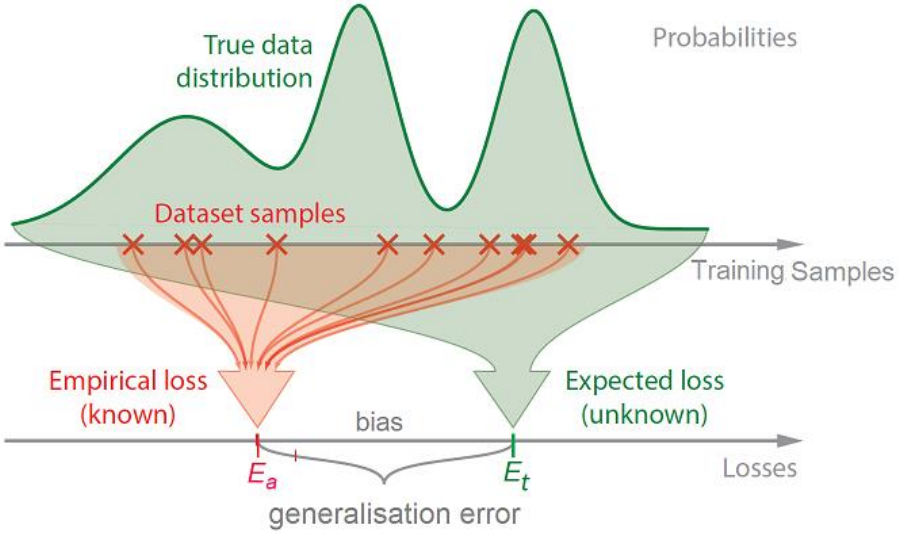


Figure 4.2: True error with generalisation gap (Cluzeau *et al.*, 2020)

The above diagram was adapted from a standard definition for the 'generalisation gap', as given in the public report for 'Concepts of Design Assurance for Neural Networks (CoDANN)'. The report was created jointly by the European Union Aviation Safety Agency (EASA), and Daedalean AG (sUAS News Press, 2020), as part of new regulations (EASA Regulations, 2020). Daedalean is a swiss technology company building autonomous piloting software systems for civil aircraft and advanced aerial mobility (Daedalean AG, 2020).

Usually, a neural network model validation is based on some specified network performance measure on test data, or data that was not used in constructing the model. There are four frequently reported performance measures or error metrics (Twomey and Smith, 1995):

1) Mean absolute error (MAE),
2) Root mean squared error (RMSE),
3) Mean squared error (MSE) and
4) Percent good (PG) classification.

The first two error metrics, MAE and RMSE, are more appropriate for networks with smooth, analogue or continuous output targets. Specifically, RMSE is an expansion on MSE, and is often used for pattern classification networks, with continuous actual outputs and discrete output targets.

For the purpose of analysis, where output targets are discrete on discrete classes, MSE was chosen as an adequate error measure. The definition of MSE is given below (Twomey and Smith, 1995):

$$MSE = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}(o_{ij}-t_{ij})^2}{n} \quad , \tag{4.15}$$

where n is the number of patterns in the validation set; m is the number of components in the output vector; o is the output of a single neuron j; t is the target for the single neuron j; and each input pattern is denoted by vector **i** (Twomey and Smith, 1995).

In pattern recognition, interpretive schedules or rules are employed for classifying a correct response to a given input vector, **i** (Twomey and Smith, 1995). The last performance measure listed above, PG, is used in pattern classification, as a measure of the number of correct classifications over all n patterns, using one of these interpretive schedules. For the segmentation of enterprise target datasets, more related to regression analysis, PG would not necessarily be appropriate. Instead, an analysis of residuals and output plots is conducted. The residual, or error r_p, for a pattern or validation dataset, p, is merely the difference between the target output (t_p) and the network output (o_p):

$$r_p = o_p - t_p \tag{4.16}$$

Both the 'generalisation error' and 'residual' were used in this study. These were calculated during the use of a cross-validation technique for training and testing the network model. The verification techniques for cross

validation considered in this study, and the accompanying sampling methods are described below (Twomey and Smith, 1995):

- **Train / test split validation**

  The complete training dataset can be split into a training set and a test set, for validation. Usually the split is done in 70:30 or 80:20 proportions (Sanjay.M, 2018). With limited data there is a likelihood of high bias, as some information about data not used for training might be missed. True error (E_t) is estimated directly as the testing set error (T_e), and bias could be calculated by subtracting the apparent error or training set error (E_a), from the testing set error (Twomey and Smith, 1995). From equation (4.14), it is derived as:

$For\ E_t \approx T_e,$

$$T_e = E_a + b,$$

$$\therefore\ b = T_e - E_a. \tag{4.17}$$

  For the analysis, this train/test split validation was used in only two test-run batches to illustrate the process of train-and-test.

- **K-folds cross validation**

  For a less biased model for validation, the k-folds or grouped validation divides the available training data into k groups, subsets or folds. Each subset uses $k-1$ data groups for model construction, and the hold out group for kth model validation (Twomey and Smith, 1995). Then the model is fitted using the $k-1$ subsets and the remaining k^{th} subset is used as a test set to validate the model (Sanjay.M, 2018). The error values or scores are recorded, and the process is repeated until every k subset has been used as a test set. The average of the recorded scores is taken, which serves as the performance metric for the model (Sanjay, 2018). Bias is estimated by subtracting E_a of the application network from the estimate of E_t (Twomey and Smith, 1995), derived from equation (4.14). A modified version of k-folds cross validation was followed for the analysis under discussion, as demonstrated in Figure 4.3, adapted from Gufosowa (2019).

Figure 4.3: Adapted k-fold cross validation

- **Bootstrap**

  Random samples are drawn with replacement from the original dataset of n observations (Efron, 1982). This sampling method is known as bootstrap training. During bootstrap testing, measures of accuracy such as bias, variance and error are assigned to sample estimates (Efron, 1982).

  The bias of each bootstrapped network is estimated by subtracting the training set error (E_a) of that sample network from the error of the network evaluated on the original total dataset (E_o). This is repeated r times, each with a different randomly drawn dataset. The *overall bias estimate* (B_a) is obtained by averaging over the r estimates of bias. An estimate of true error (E_t) is obtained by adding the *bias estimate* to the *apparent error* of the application model. Note that in this case, the average of all E_a, over r estimates of bias, is used as the *apparent error* of the application model. Similarly, the average of all E_t is used as an estimate of the *true error* of the model.

  One way the original error (E_o) can be estimated is by calculating the MSE, as in equation (4.15), using residuals as in equation (4.16). A

mathematical interpretation of the above statement of Twomey and Smith (1995) was made, using n as the number of patterns in the validation set, m as the number of components in the output vector, and b_k as the bias estimate for sample network k:

$$E_o = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} {r_{ij}}^2}{n}, \tag{4.18}$$

$$b_k = E_o - E_{ak}, \qquad k = 1, \ldots, r$$

$$B_a = \left(\sum_{k=1}^{r} b_k\right)/r, \tag{4.19}$$

$$E_a = \left(\sum_{k=1}^{r} E_{ak}\right)/r, \tag{4.20}$$

$$E_t = E_a + B_a. \tag{4.21}$$

The final application model is constructed using all of the data. Therefore, $r+1$ models are constructed or $r+1$ neural networks are trained (Twomey and Smith, 1995). Bootstrap is derived from jackknife sampling. The jacknife sampling technique systematically leaves out each observation from a dataset, calculating the estimate each time, and then finding the average of these calculations (Quenouille, 1949). The bootstrap is generally noted to be less variable than the grouped cross validation or grouped jackknife. However, it is downward biased, always estimating lower errors than what is possibly the reality.

4.3.4 Stopping criteria

In measuring the quality of output, there are limits to how far the process should run before the value of output cannot improve any further. Stopping criteria were applied to all the segmentation methods used.

a. Number of segments

According to Wyman, (2005), a good rule of thumb is that the number of segments should reach a limit, if:

1) The larger segments stabilise and/or

2) New segments pull from too many prior segments and/or

3) Some segments begin to substantially grow in size

b. Number of iterations

The maximum number of iterations is an easy stopping criteria. However, specifying an upper limit for the number of iterations does not guarantee that an optimal solution, or feasible number of clusters have been reached. It is always a good practice to also set the performance measures, e.g. measure of tightness or fitness, to a lower limit.

c. Specific boundaries

In each of the segmentation methods or algorithms, there are unique criteria within which the analyses needs to be performed. For example, CHAID settings might include the following: the smallest node is 5% of the total nodes; the decision tree is allowed to grow to four or five branches; and set $\propto = 0.05$. This is explained further in the next section.

4.4 Method of Analysis

Each of the methods used in practice was tested individually for its capability as a technique for segmenting company data. The tests follow different algorithms, which are compared with each other. Different variations of the same method were also tested.

Before testing the algorithms, the most suitable software tools for academic or analytical purposes were selected. The following tools, altered to fit the target data, were used for the test runs:

- K-means clustering (KMC tool) – an Excel-based tool for cluster analysis, developed by Prof Otto Rauh for the Hochschule, Heilbron, (Rauh, 2013a). This tool concentrates only on the standard KMC algorithm, ensuring that no adjustment to a generic algorithm is needed.
- PSO-clustering algorithm – MATLAB code for clustering data, written by Dr Augusto L Ballardini for Cornell University (Ballardini, 2018a). The code is publicly available under the GPL-v2 licence[12]. The sample input data used in the code, the Fisher's Iris dataset (Dua and Graff, 2019), was replaced by the target dataset in this study.
- Easy CHAID – an SPSS-derived web-based application of the CHAID algorithm. The implementation was thoroughly tested against IBM SPSS native code (IBM, 2020), reaching the exact same results. The author is Rafael R. Troiani, whose applications are used mostly in the banking industry, (Troiani, 2016).
- JustNN – a Simple and easy to use neural network application developed for Neural Planner Software. The latest version (version 4.0b) was used, as developed by by Stephen Wolstenholme (2016). The

[12] The GNU General Public Licence (GNU GPL or GPL) is a widely used free software licence that guarantees end users the freedom to run, study, share, and modify the software (Stallman, 1991).

software is free and can build networks with no node or connection limits. A licensed alternative, Easy NN, is available at a fee.

Extracts of the source code or functions behind each of the tools used for analysing the methods are included in Appendix C. The methodologies used to test each of the segmentation methods are described in more detail below. Results of the test runs are discussed in Chapter 5.

4.4.1 KMC clustering

a. Process

In running the KMC algorithm for *k*-means clustering, the same process was followed every time (Rauh, 2013b):

1) **Provide the data.** The company dataset was linked to the KMC algorithm tool in Excel as input, and a name given for the test run.
2) **Explore the data.** In the evaluation of the KMC method, test runs were performed with raw and transformed data, using the silhouette index to determine the need for transformations in the final run.
3) **Transform the data.** Necessary transformations, such as standardisation or the replacement of missing values, were then performed. Note that clustering was not run on raw data, if there were missing values.
4) **Cluster.** Using the KMC algorithm, several solutions with different numbers of clusters and different numbers of iterations were generated.
5) **Validate and choose a solution.** The best solution was found by ranking quality indices (silhouettes and weights) and comparing scatter plots of the solutions, as well as validating the suitability of the segments found.

b. Data transformations

Among the many tranformations available for the tests in this study, the researcher opted for those most commonly used (Rauh, 2013b):

- Replacing missing values

- Standardising or normalising values.

In most cases, missing values were replaced with average values (means). In special cases, replacement with medians or modes was considered. However, the company dataset was not suitable for mode values, as these were not available for all parameters. The median value was not suited as replacement of null values because this might skew clustering around a middle point and not where most data lies. The medians and modes of the data are shown in the next chapter, in the section on exploratory data analysis.

In the analysis, the terms, standardise and normalise are not distinguished and the researcher refers to standardisation for both. There are various techniques for standardisation, with the best two chosen for the test analysis:

- Standardise to the range [0,…,1]: more flexible and can be applied to all kinds of numerical values.
- The z-transformation, well known for changing the values of a variable in such a manner that the mean is 0 and the standard deviation is 1.

c. Initialisation

The following initialisation values are set in the KMC tool (Rauh, 2013a), every time the algorithm runs :

- An arbitrary number of clusters k;
- Maximum limit for the number of necessary iterations; and
- A random centre for each cluster.

d. Feasible clusters

The number of clusters may range from two to the number of objects in the dataset. For this analysis, the limit for the number of clusters was set at ten, being a reasonable maximum number used in practice. A few preliminary iterations were run with different k numbers of clusters as input to determine the optimal number of clusters before running tests on the algorithm. Iterations were run for k values in the set {3,4,5,7,10}.

In theory, the k value with the maximum silhouette index is then chosen. However, in most cases, the k with the maximum silhouette contained clusters with very few objects. It was not practically feasible to target these clusters as segments. A method was then developed by the author to determine the number of feasible clusters. A feasible cluster value was identified by comparing the number of objects, n, for every cluster in the run, to the average number of objects for the largest k that could be chosen (in this case, where $k = 10$). In other words, a feasible cluster at iteration j was identified as one where the number of objects, n, satisfy the condition:

$$n_j >= \frac{\sum_{i=1}^{k} n_i}{\max_{k \in \{3,4,5,7,10\}} k} \quad \text{and } j = 1, \dots, k \tag{4.24}$$

The right side of the condition in equation (4.24) is reduced to a division of the total number of observations, or the total records in the company dataset, 3362. This is divided into the maximum $k = 10$. The minimum objects a cluster should have to be feasible, is then 336. The test runs showed that the number of feasible clusters c, those with an adequate number of objects, was, in most cases, smaller than the input k values. The silhouette $s(i)$ for these is then much lower than the maximum silhouette, indicating overlapping clusters.

Nevertheless, the interpretation for $s(i) \to 0$ based on equation (4.1) allows for the allocation of objects from clusters that are not feasible to the closest feasible cluster, as $s(i) > 0$ still. With the $s(i)$ found to be in the region of 0.4 or more, it may be inferred that the objects are at least 40% tight for both feasible and non-feasible clusters. If the number of clusters were reduced, these objects could be added to any of the closest clusters.

The logical choice would be to reduce the number of clusters to $k = c$ and use the feasible number of clusters to find the solution. But not all c are the same, and the maximum c does not necessarily provide tight enough clusters. Again, the author designed a method whereby $s(r)$ was taken into

account with c, by means of a weighting $w(r)$, with

$$w(r) = s(r) * \frac{c_r}{\bar{c}}, \quad r = 1, \dots, r_{max\,k} \tag{4.25}$$

where c_r is the feasible number of clusters at test run r, and $\bar{c}$ the mean of all the test feasible clusters. An interesting result is that the test run with the largest weighting has the same number of feasible clusters c as input number of clusters k. In cases where a run r has $k_r > c_r$, it was found that the c_r is the same as the c_s for the run s with the largest w_s. The feasible number of clusters c_s for the run with the largest value for w was then used as the number of clusters used to test the method.

e. Iterations

Iterations for the KMC algorithm continue until a stable balance was reached. The number of necessary iterations was usually rather small. However, the algorithm can move away from, or never reach a stable equilibrium. The number of iterations was, therefore, set to a limit before the algorithm starts to run. It was found that 1 000 or more iterations do not improve the solution significantly. For the testing, 400 to 1 000 iterations (with increments of 100) were specified each time to find the optimal number of iterations. To determine the number of iterations, the average silhouette at each run for each number of iterations is used. The run with the maximum value for the average of $s(i)$, will have the number of iterations to set as a limit.

f. Cluster centres

The KMC algorithm begins with creating a pseudorandom centre for each cluster, relying on an initial random seed value. This seed value can be manually changed during the start of an iteration to create more random centres. This was only done after the first iteration, where the solutions obtained did not make sense, or to confirm that the algorithm was not influenced by different random starting points. The possibility of finding a better solution by changing the seed was rather small.

g. Test runs

Test runs were executed, using the KMC algorithm, to:

1) Determine the transformations to be applied.
2) Find the optimal number of clusters.
3) Decide on the best number of iterations to use.
4) Apply these settings and transformations and find an optimal solution.
5) Verify the solution through a few additional test runs.

Each time a test run was executed, the KMC algorithm as described in Figure 2.1 of the literature review was performed. In addition, scatter plots and statistics were generated**.**

4.4.2 PSO evaluation

a. Process

In running the PSO algorithm, it was decided to follow the same process as for the KMC algorithm, as both methods are used for clustering. Output plots and values for fitness were generated from the test runs. Data exploration was performed, initially as a separate analysis, and the results were applied for all methods.

b. Transformations

The standard PSO algorithm was tested, as well as two hybrid PSO test runs. The first hybrid tests were run with initial values based on recalculated centroids using *k*-means procedures in the MATLAB code of (Ballardini, 2018a). The second hybrid test runs were initialised with *k*-means centres from KMC test run results. To ensure the KMC centroids or centres were on the same scale for the PSO algorithm, the following transformations were performed on the data, using guidance from Rauh, (2013b):

- Replaced missing values with average values (means).
- Standardised values to the range [0,…,1].
- Applied a z-transformation on values to create a distribution around 0, with standard deviation of 1.

c. Initial values

The following initial values were set every time the algorithm ran (Ballardini, 2018a):

- Number of centroids;
- Number of necessary iterations;
- Number of particles;
- Number of dimensions;
- Intitial position for each particle;
- Initial swarm best position;
- Initial swarm velocity;
- Ranges-to-scale initial values; and
- Intial fitness value.

For all the PSO analyses, the same number of centroids was set as the maximum feasible number of clusters for the KMC algorithm. The groundwork was done during the KMC testing, so now a fixed number of clusters was all that was needed.

The number of necessary iterations is much smaller than for KMC. It was found that a number of 50 iterations or more does not improve the solution significantly. To determine the number of iterations, different runs were performed, until the measure of density (fitness) was repeated or below an acceptable limit.

Using variations of iterations, particles and dimensions, two groups of datasets were used for the test runs. There was no need for additional tests to determine the number of clusters or combination of variables, as these were analysed during data exploration and KMC analysis.

d. PSO algorithm

Each time a test run, as described in the introduction of this section, was performed, a variation of the algorithm described in Figure 2.2 of the literature review was executed based on the work of Van der Merwe and Engelbrecht (2003). Due to variations in the PSO algorithm, it might seem

that the original PSO algorithm should differ from the MATLAB code. However, the contrary was shown to be true, as the MATLAB code was developed with the original algorithm in mind (Ballardini, 2018a).

e. PSO evaluation

In every iteration, the fitness value was calculated according to Ballardini (2018a), and based on the calculated quantisation error of Van der Merwe and Engelbrecht (2003). The following values were calculated:

- Local fitness: After checking for at least one element inside cluster C belonging to the current centroid, the local fitness is defined as the mean of all distances between the points belonging to each centroid.
- Average fitness: The local fitness is added to the average fitness and the average calculated over the number of initial centroids
- Swarm fitness: If the average fitness of a particle is less than the swarm fitness of the particle, then the swarm fitness is assigned the average fitness. Note that the average fitness is assigned at least once, as swarm fitness is initiated with an infinite value (∞).
- Global fitness: After evaluation of the whole set of local fitnesses, the global fitness is evaluated as the minimum of the set of swarm fitnesses per particle. The global fitness is reported at each iteration.

4.4.3 CHAID analysis

a. Preparation

The following node options were set every time before the algorithm runs (Troiani, 2016):

- Significance level (α-level) for merging nodes or α_{merge}, used for variables with more than two categories, for deciding to merge two or more categories based on their similarity against the dependent variable. Larger **p**-values generate broader trees.

- Significance level (α-level) for splitting nodes or α_{split}, based on the predictor variable's differences, in terms of the dependent variable. This is the maximum level the **p**-value can have to split a node into two.
- Minimum node size, m. If the splitting of a parent node created a child node smaller than m, the algorithm would try to merge it with the other most similar child node, until the resulting merged child node is larger than the node size specified, m.
- Dependent variable, or which variable to use as measurement per category, for example, target market count for prospects and customers or a frequency count per company size classification, e.g. Large, Medium or Small.

b. Splitting and merging

The algorithm proposed by Kass (1980) was performed during the analysis. At each test run, the probabilities for the test statistic used for merging or splitting pairwise nodes are:

- If the largest **p**-value is larger than a user-specified alpha-level, α_{merge}, this pair is merged into a single, compound category.
- If the the smallest adjusted **p**-value (the most significant predictor) is less than, or equal to, a user-specified alpha-level, α_{split}, the node is split.

Example. For two categorical variables A and B, with A having three categories (rows) and B having four (columns), the degree of freedom is $2 \times 3 = 6$. For a test statistic (χ^2) and CV of 15, $\boldsymbol{p} = P(\chi^2 > 15) = 0.02$. For a significance level of $\alpha = 0.05$, $\boldsymbol{p} < \alpha$, and the null hypothesis (H_o) cannot be accepted. If $\boldsymbol{p}$ is the smallest adjusted **p**-value for a group of pairwise nodes, then the node with $\boldsymbol{p}$ is split (Troiani, 2016).

c. Growing the tree

During each run of the algorithm, there were no iterations, as a tree structure is built with nodes being merged or split, according to the

categories for each variable. The root node contains the dependent or target variable. The algorithm split the target variable into two or more categories, called the parent node or initial node. Independent variable categories which came below the parent categories in the CHAID analysis tree were called the child nodes. In this analysis tree, the category that is a major influence on the dependent variable comes first, and the least important category, called the terminal node, comes last. A node becomes terminal, if it cannot split any further. Growing the tree structure may end, if the number of levels (tree-depth) reaches a specified value.

d. Stopping criteria

The following situations caused the nodes not to split anymore.This is similar to the items identified in Figure 3.1b (IBM, 2012) and listed here for completeness:

- All categories in the node have identical values of the dependent variable;
- All cases in a node have identical values for each predictor;
- The size of a node is smaller than the user-specified minimum node size value; and
- The resulting number of child nodes is 1.

e. Contingency tables

When nodes were split, instead of merged, or at each successful step of the tree-growing process, a contingency table (or crosstab or two-way table) was used to show the frequency distribution of categories (nodes) against dependent variables. This provided a view of the interrelation and interactions between the categories into which nodes are split, as well as the dependent values. The distribution of probability could also be derived from the contingency table.

f. The gains table

The gains table is useful for assessing the levels of expenditure that may be expected from prospective customers for different segments of the target

data (Taves, 2010). This is invaluable for planning sales campaigns and calculating possible revenue as a portion of the prospective customer's ICT expenditure. Gains tables for customers, prospects and total target market are created to show the cumulative frequency and weighted averages of raw data behind the dependent variable. Here the original values from the decision tree are used for weighted average calculations. Statistics produced by the gains table make it easy to determine how far to proceed with selecting predictions representing a given level of performance, such as revenue, response rate and number of employees.

As example, a gains table from Taves (2010) is shown in Table 4.3. The table was created to illustrate the total buying power, of existing and potential customers, using ICT spend as measurement. The Segment IDs referred to are the result of a decision tree showing categories of ICT spend, where segments have been assigned to the terminal nodes. The weighted average mentioned was adapted to illustrate how results would look like for data in this research. Actual results from test runs are discussed in subsection 4.5.4b.

Table 4.3: Gains table example (Taves, 2010)

(1) Segment ID	(2) Segment Count	(3) Percent of Total	(4) Average ICT Spend	(5) Segment Index	(6) Cumulative Count	(7) Cumulative Percent	(8) Total ICT Spend per Segment	(9) Cumulative Average ICT Spend	(10) Cumulative Index
3	196	0.2%	R 3 580	224	196	0.2%	701 680	R 3 580	224
4	58 603	66.8%	R 2 250	141	58 799	67.1%	131 856 750	R 2 254	141
2	72	0.1%	R 1 170	73	58 871	67.1%	84 240	R 2 253	141
1	622	0.7%	R 820	51	59 493	67.9%	510 040	R 2 238	140
5	448	0.5%	R 760	48	59 941	68.4%	340 480	R 2 227	139
6	27 734	31.6%	R 240	15	87 675	100.0%	6 656 160	R 1 599	100

1) The first column shows the segment IDs (from 1 to 6), arranged from the highest ICT spend to the lowest.
2) The second column shows the total count (N) per segment.
3) The third column shows what percentage of the total target base falls in each segment.
4) The fourth column shows the *weighted* average ICT expenditure per company, per segment. This is calculated for each segment as:

$$\frac{\left(N_c \times \text{ICT}_{\text{Spend}_c}\right)\left(N_p \times \text{ICT}_{\text{Spend}_p}\right)}{N} \tag{5.1}$$

where N_c and $\text{ICT}_{\text{Spend}_c}$ are the number of customers and average ICT spend per customer, respectively, for the segment. Then N_p and $\text{ICT}_{\text{Spend}_p}$ are the number of prospects and average ICT spend per prospect, respectively, for the segment. The total number of customers and prospects is given by N.

5) The fifth column shows the ICT spend as a relative index, where the *weighted* average of the whole target base is set at 100. This is calculated for each segment as:

$$\frac{Avg\ \text{ICT}_{\text{Spend}_s}}{Avg\ \text{ICT}_{\text{Spend}_T}} \times 100 \tag{4.27}$$

where $Avg\ \mathrm{ICT}_{\mathrm{Spend}_s}$ is the *weighted* average ICT spend per segment and $Avg\ \mathrm{ICT}_{\mathrm{Spend}_T}$ is the *weighted* average ICT spend for the total target base.

6) The sixth column shows the cumulative count of companies per segment.
7) The seventh column shows the cumulative count as a percentage of the total count.
8) The eighth column is the *Average ICT Spend* multiplied by the count per segment, resulting in the *Total ICT Spend* per segment.
9) The ninth column shows the cumulative ICT spend value as the *Total ICT Spend* from 8) divided by the total count per segment, or:

$$\frac{\sum_{i=1}^{s} N_i \times \mathrm{ICT}_{\mathrm{Spend}_i}}{\sum_{i=1}^{s} N_i} \tag{4.28}$$

where N_i is the total count per segment i as given in 2), $\mathrm{ICT}_{\mathrm{Spend}_i}$ is the *weighted* average ICT spend per segment i as given in 4),
and s is the latest segment, with $i, s \in \{1, \dots, 15\}$

10) The tenth column is the cumulative ICT spend from 8) *divided* by $Avg\ \mathrm{ICT}_{\mathrm{Spend}_T}$, the *weighted* average ICT spend for the total target base, *times* 100.

4.4.4 ANN classification

a. Preparation

The network for an ANN model represents the learning process that training data undergoes. A feedforward neural network was chosen for each test run, using backpropagation. A gradient descent algorithm is used to find the best fit (minima of a curve) of the cost function (error function) of the model. The algorithm is iterative, in order to converge an under-fitted graph (not representing the underlying data structure) to a graph that optimally fits the data (Sharma, 2019a).

For the ANN model to be constructed within certain boundaries, parameters have been set up to control the network. Internal model parameters, such as neuron weights and biases, are estimated from training examples (Missinglink.ai, 2018). External parameters are set by the operator of the neural network to control the learning process. These hyper-parameters can be applied to network model structure or to the algorithm followed in the learning process (Missinglink.ai, 2018). In preparation for the analysis of the ANN models in this research, a decision was made to run learning processes on different training data examples.

The parameters used for each of these test runs are described in more detail in the sections that follow.

b. Model parameters

The internal model parameters used by the software for the ANN learning were based on the training dataset provided. For each neuron the following parameters were calculated (Wolstenholme, 2016):

- Number of neurons in the input layer;
- Net input from all neurons in the previous layer;
- The bias for the current neuron;
- The error value at the current neuron;
- The activation function, used as net input to the next layer of neurons; and
- Number of neurons in the output layer.

c. Network hyper-parameters

The topology and size of the neural network was defined for each test training run. The neural network topology was in the form of a directed graph, as defined in section 3.2. The size of the ANN for each training batch is given in terms of the number of hidden layers, and the number of neurons in each layer. In addition, the growth rate, in terms of the number of cycles or time (seconds) is specified. A new network was produced when the cycles or seconds elapse, until the optimum network was found (Wolstenholme, 2016).

d. Algorithm hyper-parameters

The algorithm hyper-parameters are known as controls in the JustNN software used for test runs (Wolstenholme, 2016). The hyper-parameters below, as described in section 3.2, were specified for the learning process.

- Learning rate
- Momentum
- Validation
- Stopping criteria

e. Activation function

Input values for training process can have multiple labels, with output values not necessarily mutually exclusive (refer to subsection 5.5.5). In addition, due to the fixed output range, and being differentiable over different values of y (Mahmood, 2019), the JustNN software uses a standard sigmoid activation function (Vozhehova *et al.*, 2019).

5. ANALYSIS AND RESULTS

The results of applying the methodology to test the quantitative segmentation methods described in the literature study are provided in this chapter. The solutions derived from the different segmentation methods are clarified using tables and bivariate scatter plots. In the plots, the observations in the population dataset are represented by different symbols and colours. Each symbol and colour indicates the cluster to which the observation belongs. The evaluation criteria in the previous chapter were used to compare the clustering methods. The comparison results across segmentation methods are encapsulated at the end of this chapter. By way of exploratory data analysis the raw data is evaluated first.

5.1. Exploratory Data Analysis

Summary statistics and results are provided here in the form of graphs and plots, without compromising the companies' privacy. For more descriptive statistics behind these results please refer to Appendix B.

5.1.1 Location priority

The locations of companies were frequently identified in relation to landmarks, instead of physical address. These landmarks proved to have a certain pattern, depending on the type of company. For example, factories were in industrial areas or close to bus stops, due to factory worker transport. An attempt was made to prioritise the locations, where 1 is the highest priority and 43 the lowest, in terms of ease to find, making this an ordinal scale variable. Refer to section A.3 in the appendices for a list of these priorities. Figure 5.1 depicts a summary of top five industries where companies operate, with a count of companies per landmark.

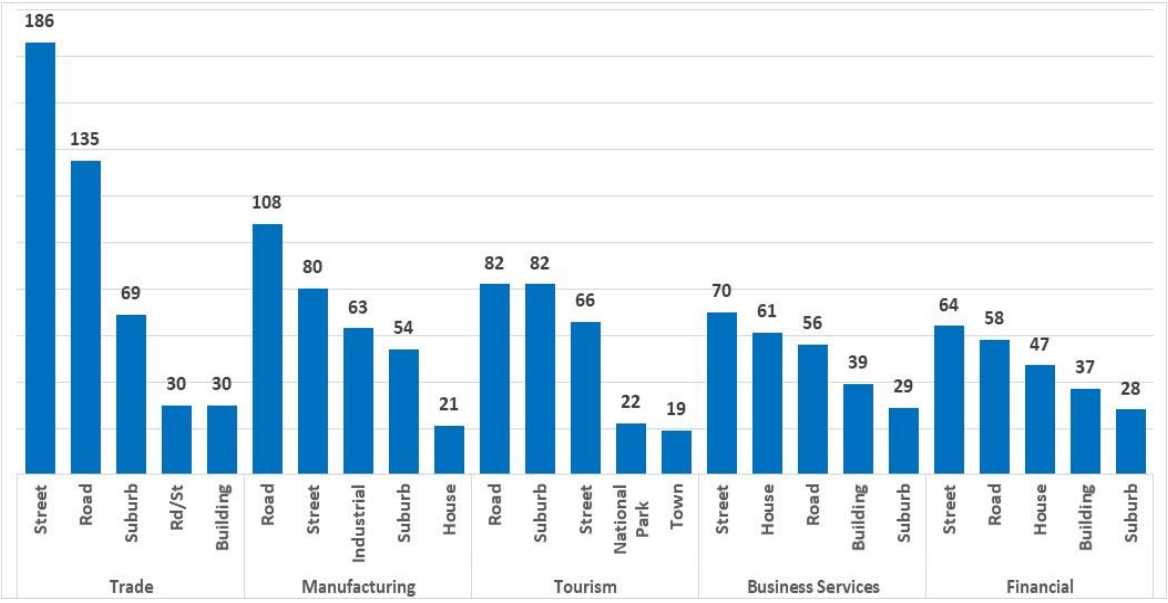


Figure 5.1: Number of companies found at industry locations

Additional analysis on the location priorities, outside the scope of this research, would reveal more appropriate information. An interesting outcome is the fact that ICT companies, financial institutions and business services tend to be located close to city centres; well-known buildings (or towers); shopping centres or plazas; or smaller buildings specific to the company, called houses, e.g. ATC House, Arcade House.

This location priority was analysed using the same method as for Standard Industrial Classification (SIC) codes, but the mean per location type or feature showed no difference, as there were no subgroups per priority. A few runs were performed during the analysis, with and without location priorities. The effect did not seem to influence the numeric parameters. For SIC codes, no correlation with location priority was found and the Pearson's r value was -0.095.

5.1.2 Standard industry codes

One approach to describing a business is by considering its economic activity. It is important to have a more homogeneous unit than an enterprise description to perform an analysis of economic activities. This unit is an establishment, defined as an enterprise or part of an enterprise situated in

one location and engaged in mainly one type of principal activity, and may also be engaged in secondary activities which generate a minor part of its production output (South African Reserve Bank, 2011). An enterprise may be comprised of more than one establishment, but an establishment may belong to only one enterprise.

Based on the type of activity described, establishments are grouped together into industries which are classified according to the SIC of all Economic Activities.

In 1937, the Central Statistics Board of the United States first published the SIC System (US Census Bureau, 1938). In 1997, this was replaced by the North American Industry Classification System (NAICS), due to rapid changes in the United States and world economy (US Office of Management and Budget, 2007). Since 1948, the world has adopted the International Standard Industrial Classification (ISIC) as the more appropriate classification standard for countries outside the United States (United Nations Statistical Office, 2008).

The analysis file contains a nominal variable for the SIC, better described as Standard Industrial Classification (Statistics South Africa, 1993). The publication by Statistics South Africa is derived from the 1990 edition of the United Nations (UN) International Standard Industrial Classification (United Nations Statistical Office, 1990), but adjusted for South African conditions (South African Reserve Bank, 2011). The major economic activities and divisions into which institutional sectors or units are classified, according to the SIC, are presented in section A.3 of Appendix A.

In order to be able to use the SIC for segmentation purposes, it is necessary to treat this nominal variable as a numeric variable in some cases. This is especially true in the case of clustering methods such as *k*-means. As part of the algorithm, the observation points or particles are centred around the middle point. The mean of a group of points is needed in various dimensions in this calculation. The SIC code variable was, therefore, evaluated to see

whether a calculation such as the mean of a group of SICs would skew the clustering significantly. Figure 5.2 portrays a comparison between SIC codes and the average of SIC codes per industry. Note that Sub Industry are economic activities defined by the researched telecommunication provider based on the SIC codes. These sub industries names can be seen as more detailed as a division and less generalised as a major division as described in section A.3.3 of Appendix A.3.

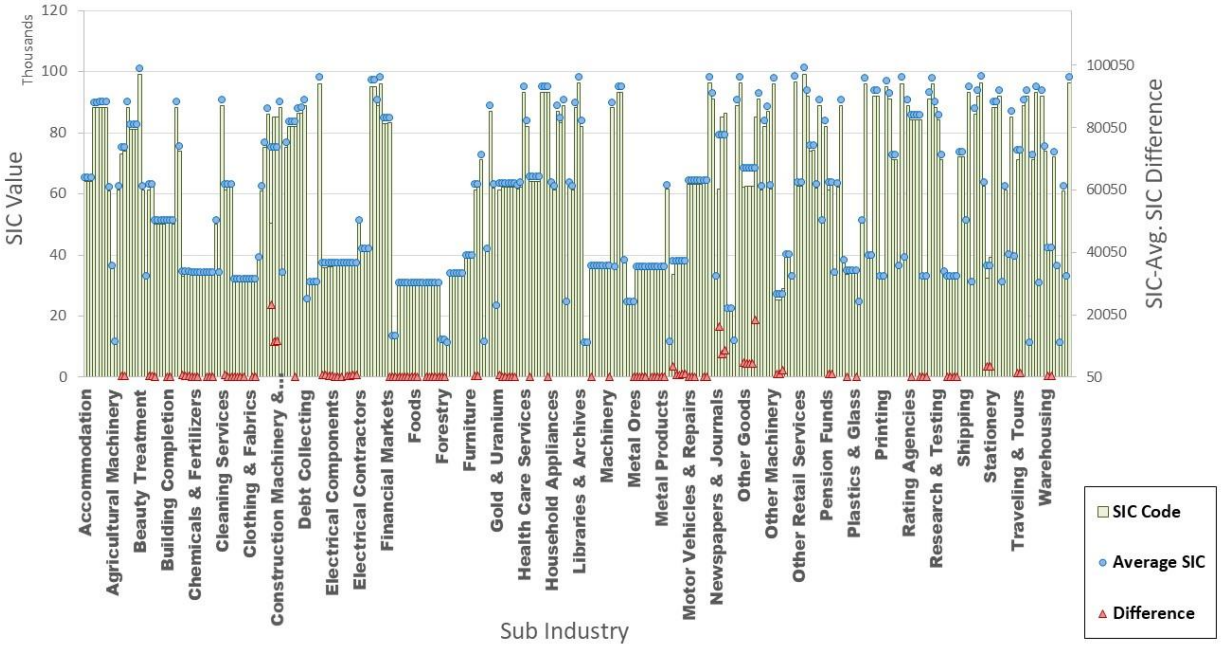


Figure 5.2: Comparing SIC codes and averages

The graph plots the SIC code for all industries, the average SIC per industry, and the difference between the SICs and average SICs.

Although the correlation between SIC and the average SIC in a reference list is high (Pearson's r value = 0.995), the correlation goes down when SIC codes and average SIC codes are evaluated by means of the input data (Pearson's r value = 0.945). It is, however, still high enough for using average SIC in groups, but the confidence level for SIC codes needs to be very high, as they are discrete values and need to estimate a specific discrete level. So, for 90% confidence on continuous values, the SIC code could be used as a metric for clustering, but not for 99% confidence for the target market input data. A

few runs were performed for clustering with and without SIC codes, which proves the effect to be acceptable, as long as the other variables in the analysis are fit for purpose.

5.1.3 Numerical variables

The numerical variables used for analysis of segmentation methods were inspected for central tendency and distribution of data values. The frequency and position were also evaluated with quantiles of the data.

Descriptive statistical values are tabulated in section B.1 of Appendix B.

i. **Central tendency**

Histograms indicating the mode, median and mean of the numeric variables are shown in Figures 5.2 and 5.3. Customer base statistics are shown on the right and target market (company) statistics are shown on the left. Note that the graphs do not include axis values as values were to far apart to be displayed meaningfully. The main purpose of Figure 5.3 and Figure 5.4 is to graphicaly show the distances between values. It is clear that the centre of the values and the most frequent values are far from the mean, in each case, especially in variables describing the the company (Company Employees, Company Turnover, ICT Spend).

It may, therefore, be an indication that no inherent clustering around a mean value exists in the data. However, the distribution may still lean towards another area of the dataset.

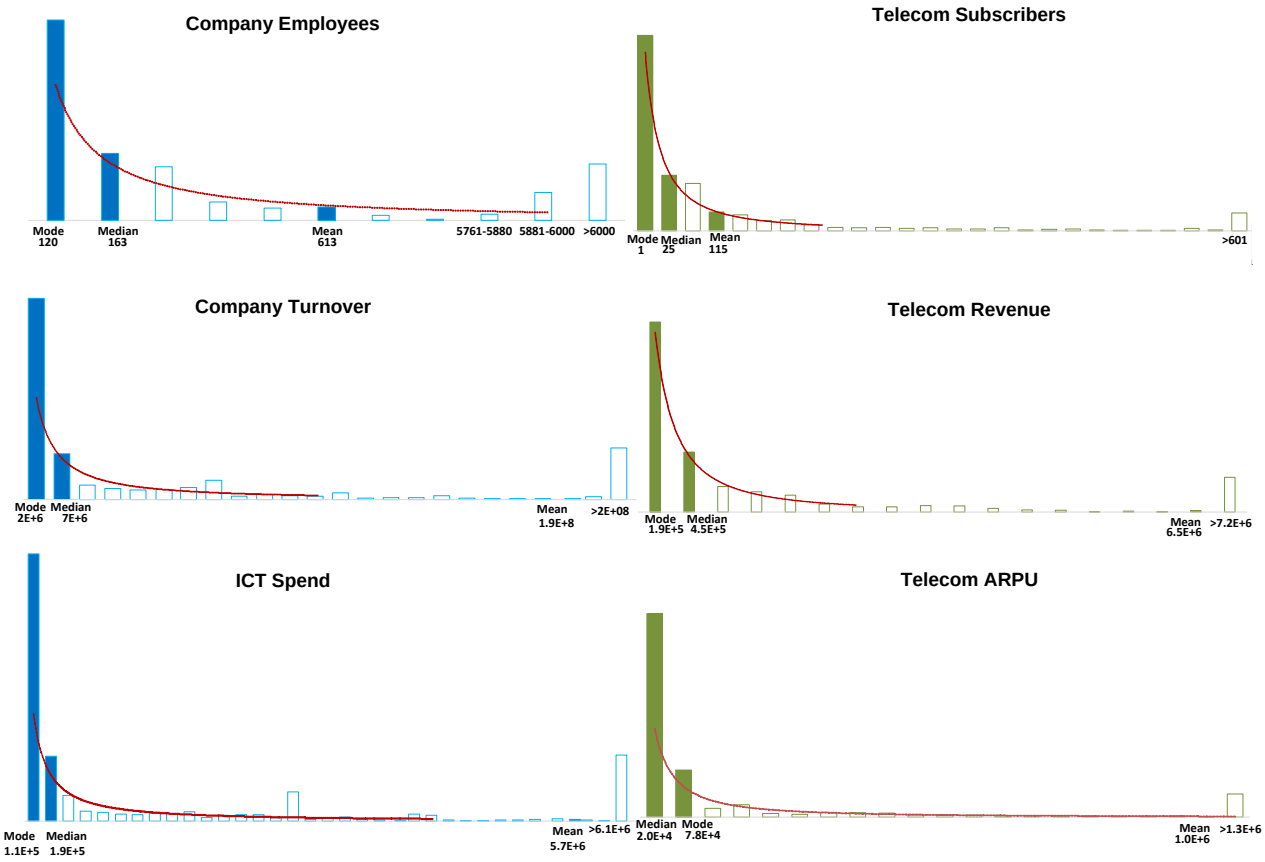


Figure 5.3: Target data measures of central tendencies

The solutions, products and devices most used are measured in terms of the number of lines, also called connections or number of MSISDNs (mobile station international subscriber directory numbers). Figure 5.4 shows the central tendencies, in terms of statistical values around the mean.

For all these items, the number of lines most encountered (mode) and midpoint of the number of lines (median) are constant. However, the mean value is very high, in relation to the mode and median. The extreme maximum values in the range of each of these measurements confirm that there are a number of outliers. The maximum number of lines for the solutions, products and devices most used are, respectively, 158630, 158629 and 13933.

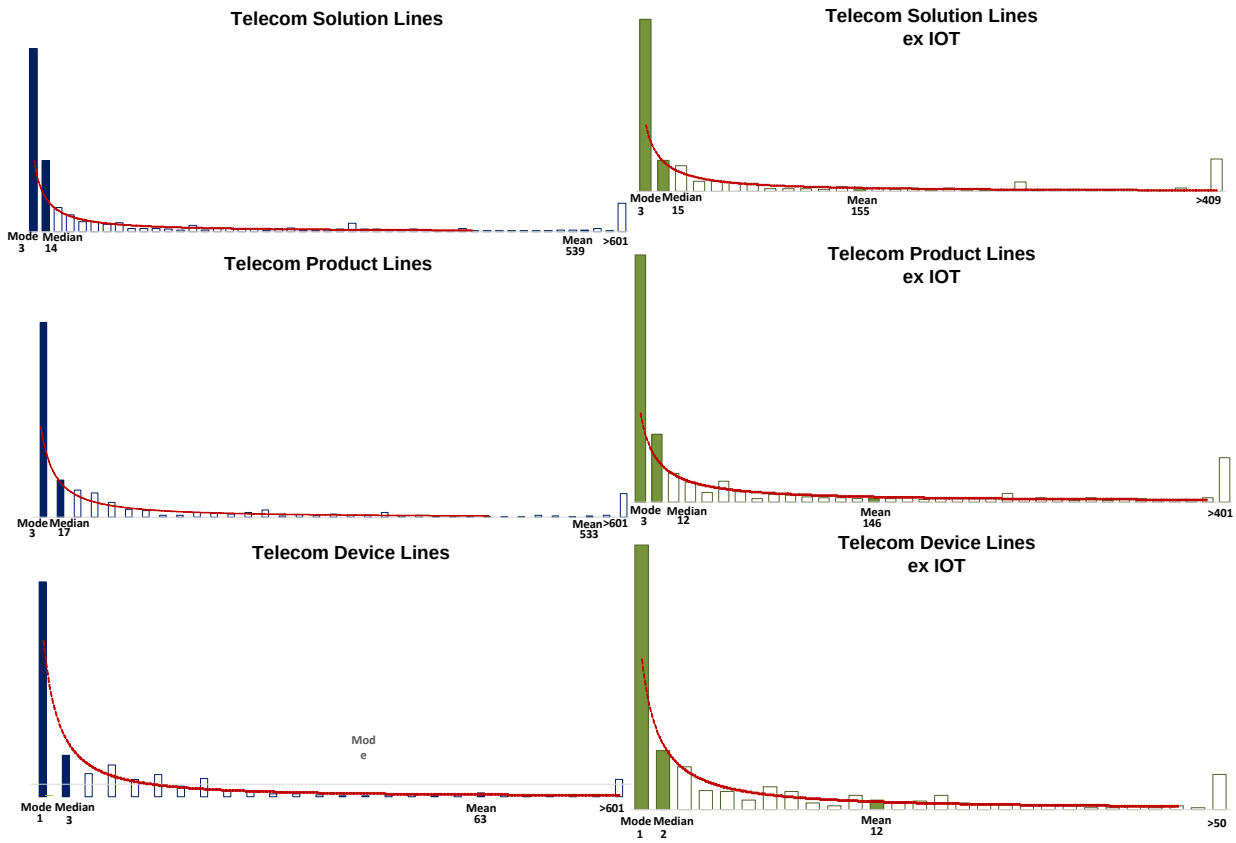


Figure 5.4: Customer lines around the mean

On further review, it was found that a few of the products most used are for machine-to-machine (M2M) applications. The M2M products form a subset of internet of things (IOT). Where M2M is the connection of devices and appliances, e.g. for vehicle tracking, IOT is a broader definition of all interconnected devices, cloud data, internet and humans via technology. The transaction volume is much higher for M2M than for mobile usage, as devices are used to monitor processes or activities at regular short intervals (ranging from a few seconds to every few days). The right hand histograms in Figure 5.4 show the reduction in the means when IOT solution types, or M2M products are excluded from the observations. The maximum number of lines are then reduced to 4315, 4152 and 213 respectively.

A decision had to be made, therefore, whether to include IOT and M2M for the segment analysis. If only other solutions such as mobile or fixed line solutions were being offered, then it would make sense. The Tanzanian

market tends to move more towards automated device monitoring and control, especially for security reasons and as an alternative where there is a lack of resources. Consequently, it was decided to keep all the observations in further analysis.

ii. **Variability**

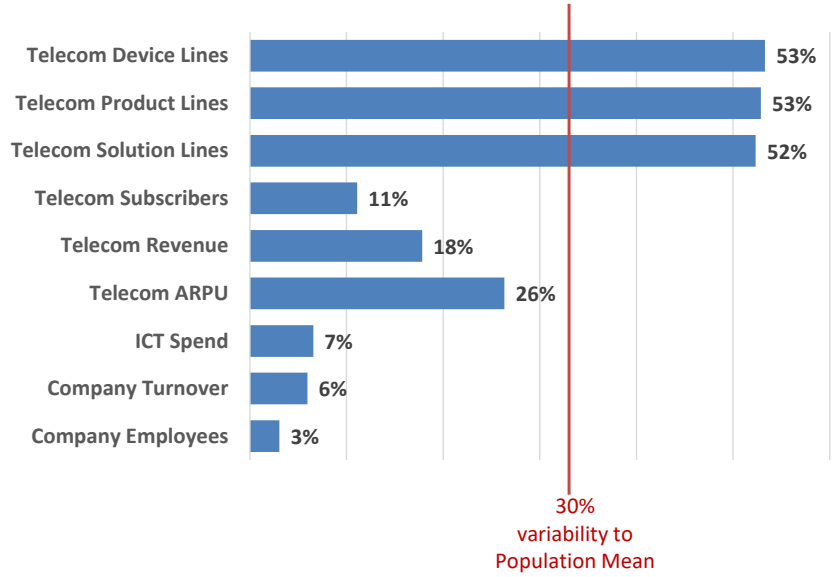
In order to confirm the tendency to move away from the mean, the spread or distibution of the data is interpreted. Firstly, the relative standard error is measured as a coefficient of variation to confirm that the target market is represented by the dataset. When the coefficient is about 30% or less, it shows a small enough variation from the mean for the dataset, and the variable may be used in further analysis to represent the population or target market (Klein *et al.,* 2002).

Then the skewness and kurtosis are measured. When the skewness is greater than two in absolute value, the variable is considered to be asymmetrical around its mean. When the kurtosis is greater than or equal to three, then the variable's distribution is significantly different from a normal distribution in its tendency to produce outliers (Westfall and Henning, 2013). Figure 5.5 shows the coefficient of variation, skewness and kurtosis measured for each numeric variable.

Considering all factors, it appears from Figure 5.5 that the largest variance lies with the three measures for items most used by subscribers of a customer, namely solution, product and device line count.

All numeric variables are asymmetric, but these three are significant enough not to be used as representative of actual subscriber needs. Even so, standardised data can be used for clustering, as all variables are then distributed according to a uniform scale.

a. Coefficient of variation for dataset variables



b. Skewness and kurtosis for dataset variables

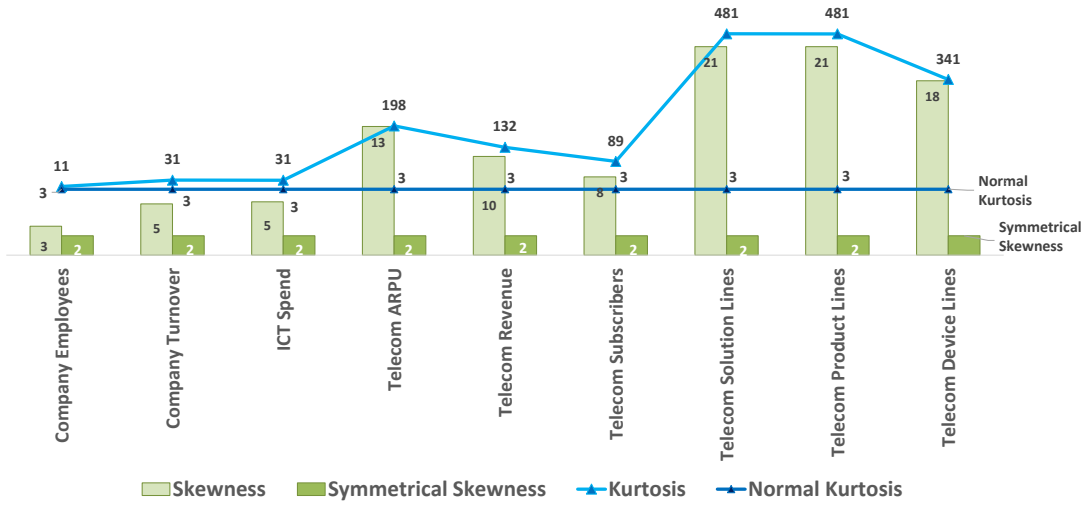


Figure 5.5: Target data measure of variability

5.1.4 Ratio scale variables

The company's expenditure on ICT, in relation to the countrywide expenditure and the population of ICT users, was measured for two variables. This expenditure was evaluated for variability only, as both variables rely on the ICT Spend and Company Employees variables, which were evaluated for central tendency.

The two variables evaluated were:

- **ICT as % of country ICT**: ICT spend as a percentage of the country ICT spend.
- **ICT as % of usage population**: Number of employees as a percentage of total population in the same industry, using ICT.

Due to company data being compared to a whole industry or the country, the proportions were very low, as can be seen in the summary of measures in Table .

Table 5.1: Ratio scale measures

Measure	ICT as % of country ICT	ICT as % of usage population
min	0.0000%	0.000010%
max	1.7127%	0.046400%
mean	0.0409%	0.002254%
median	0.0013%	0.000678%
mode	0.0008%	0.000390%
standard deviation	0.1558%	0.003913%
standard error	0.0027%	0.000067%
skewness	5.47	2.99
kurtosis	30.78	11.33

Both ICT as % of country ICT and ICT % of usage population had very low coefficients of variation (7% and 3%, respectively). Even though the values themselves are very small, the descriptive statistics show that, with transformation on a consistent scale, these variables can be used for further analysis.

Observations for ICT as % of country ICT prove to be more widespread, with most of the values lying below a much higher value than the median (0.0013%).

On the other hand, ICT % of usage population is more evenly distributed, with most values below the median (0.00068%). The quantiles displayed in Figure 5.6 are an illustration of this.

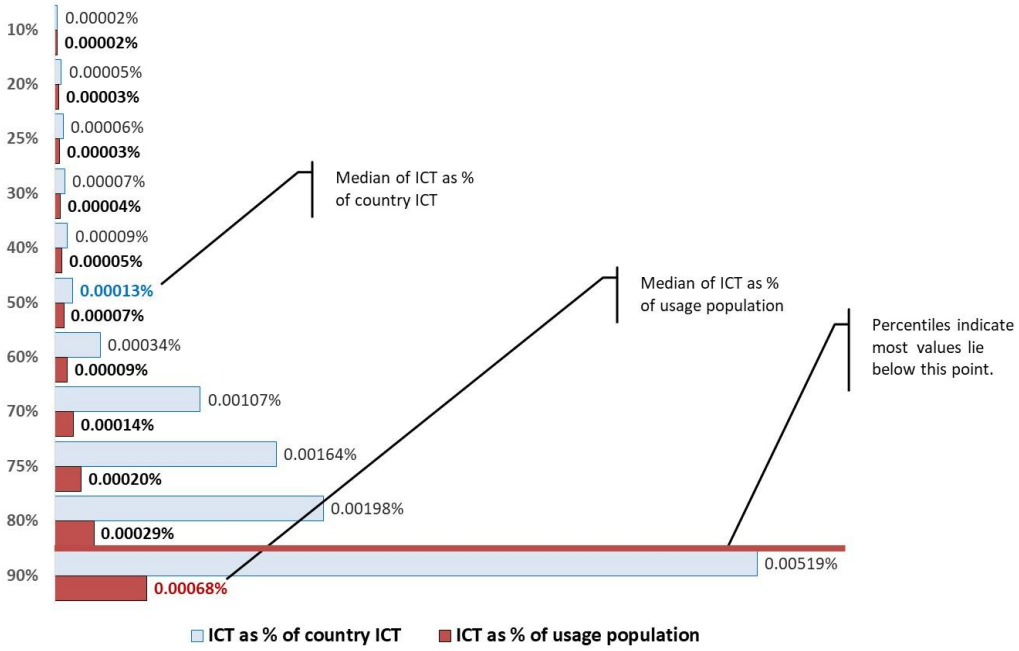


Figure 5.6: Percentiles for target data ratio variables

5.2 K-means clustering

5.2.1 Input data

Different combinations of input variables were used for each of the five groups of test runs on the KMC algorithm tool (Rauh, 2013a). The names of each test run group and variables used are indicated in subsection A.2.1 of Appendix A.

The methodology used relied on the combination of input data. For each of the KMC test runs, the input segmentation scheme and approach were:

- KMC1: Features, a priori approach, using company firmographics;
- KMC2: Behaviour, value approach, using expenditure and sales numbers;
- KMC3: Needs, value approach, using customer subscriber totals;
- KMC4: Behaviour and Needs, value-based approach, using customer subscriber totals, expenditure and sales numbers; and
- KMC5: Features, Behaviour and Needs, a priori and value-based approach. The a priori approach used company firmographics, customer subscriber totals, expenditure and sales numbers.

5.2.2 Data transformations

As explained earlier regarding the method of analysis, clustering was not run on raw data, due to missing values for companies with no customer metrics. Missing values were replaced by the average (mean) of the observations before further standardisation of the data.

When running cluster iterations on the raw data, with missing values replaced, the silhouette index was very high ($s(i)$ ranged between 0.88 and 0.94). However, only one feasible cluster could be found ($c = 1$).

All the variables were then transformed to the same scale and a number of runs performed, specifying various numbers of clusters k. By applying both a [0, 1] transformation and the standardised z-transformation to different runs in succession, an optimal solution was found. The best solution was provided when the z-transformation was applied first, and, subsequently, a [0, 1] transformation was done to the standardised z-transformation. For all further test runs, the standardised z-transformation and the [0, 1] transformation were thus first applied to the data.

The silhouette ranged from 0.40 to 0.48, and weighting for the most feasible number of clusters ranged from 0.37 to 0.47, with the feasible number of clusters ranging from 2 to 4. The number of iterations used was between 300 and 700.

5.2.3 Number of clusters

After a number of test runs, results for silhouette and weight were found for $k \in \{3,4,5,7,10\}$. The weighting and silhouette for all input numbers of clusters are shown in Figure 5.7. The most feasible number of clusters for the dataset was found to be $c = 4$.

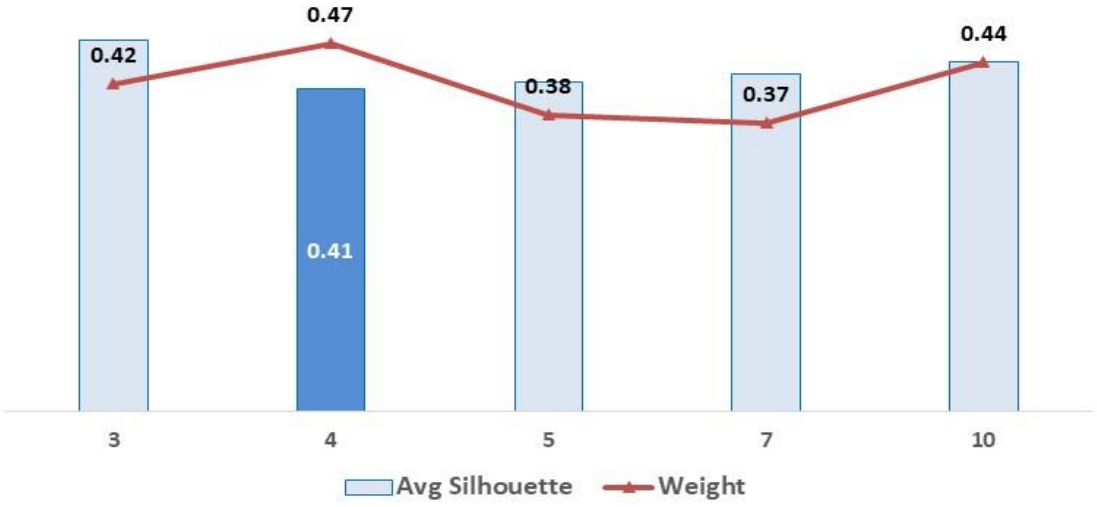


Figure 5.7: Feasible number of cluster tests

5.2.4 Number of iterations

For the test runs, 100 to 1 000 iterations (with increments of 100) were specified each time to find the optimal number of iterations. With each test run, however, the number of feasible clusters changed from the input number ($k = 4$). This was taken into account and weights were calculated again. First, the average of all the feasible numbers of clusters was used, then the maximum of these and, finally, using the most feasible number of clusters, as determined in the previous section, a decision was made.

According to the outline of these steps in Figure 5.8, the limit for the necessary number of iterations was set at 500 ($j = 1,..,500$). If a solution was not feasible, further runs were made with at least 600 iterations.

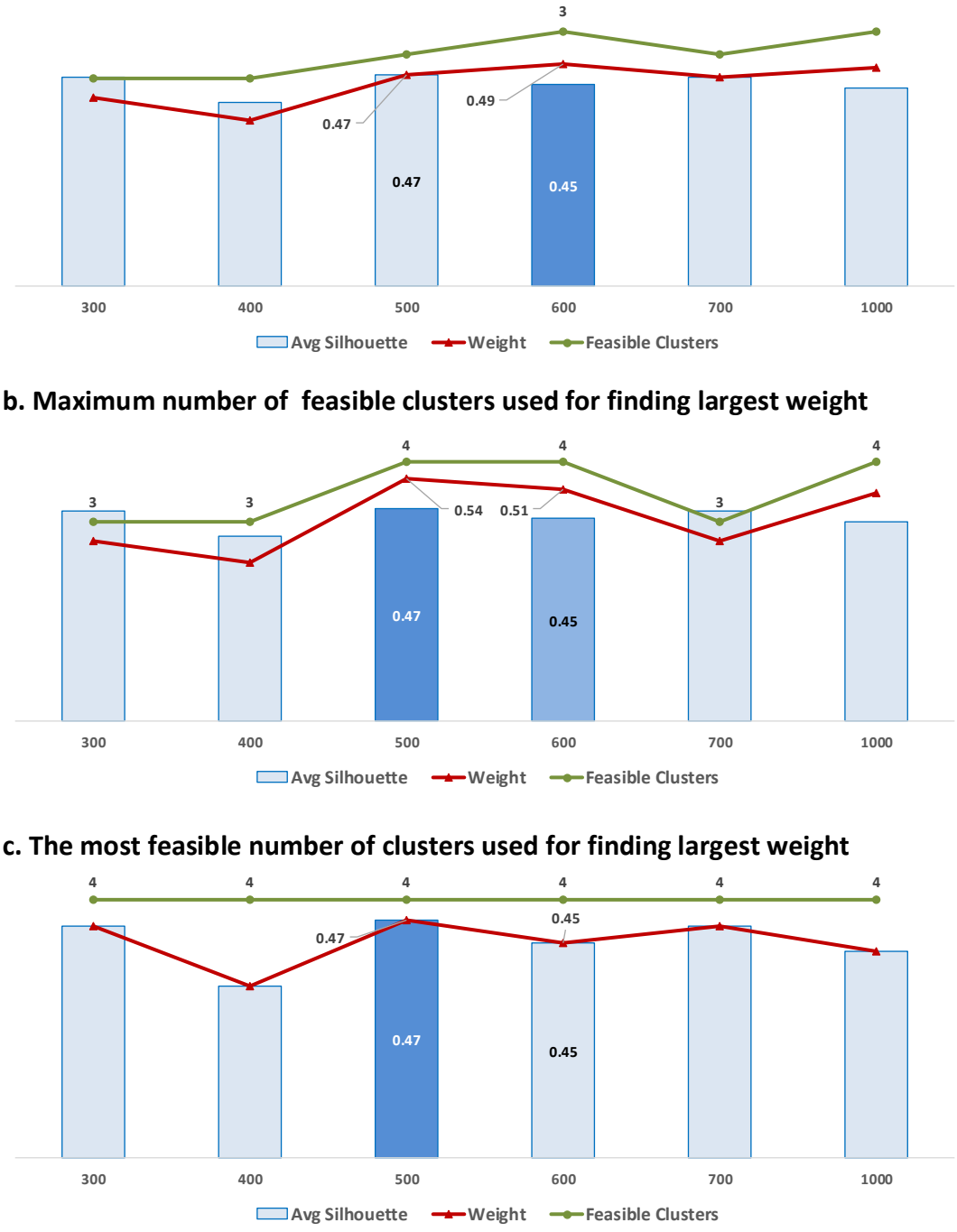


Figure 5.8: Finding the limit for number of iterations

5.2.5 KMC tests

Referring to subsection A.2.1 of Appendix A, the five groups of test runs were performed, each with the following input parameters: feasible number of clusters, $c = 4$, iteration limit, $j = 500$ and with a random centre value for each cluster.

a. Feature-based approach (test KMC1)

Tests were run 11 times before a solution was found containing an adequate number of feasible clusters ($c = 4$). The first run was a test on the raw data, with missing values replaced by the average, and clusters set at $k = 5$ and maximum number of iterations at $j = 500$. The variables used with their centres per cluster are shown in section B.2 of Appendix B for the first and last run. Note that iterations were set at $j = 600$ from the *seventh* run.

For validation, bivariate scatter plots were created on some variable pairs before and after clustering. The variable pair of Location priority vs. SIC code gives the clearest depiction of the clusters, as seen in Figure 5.9. Note that the variable names as used in the KMC tool are plotted, these correspond to variables used in the study.

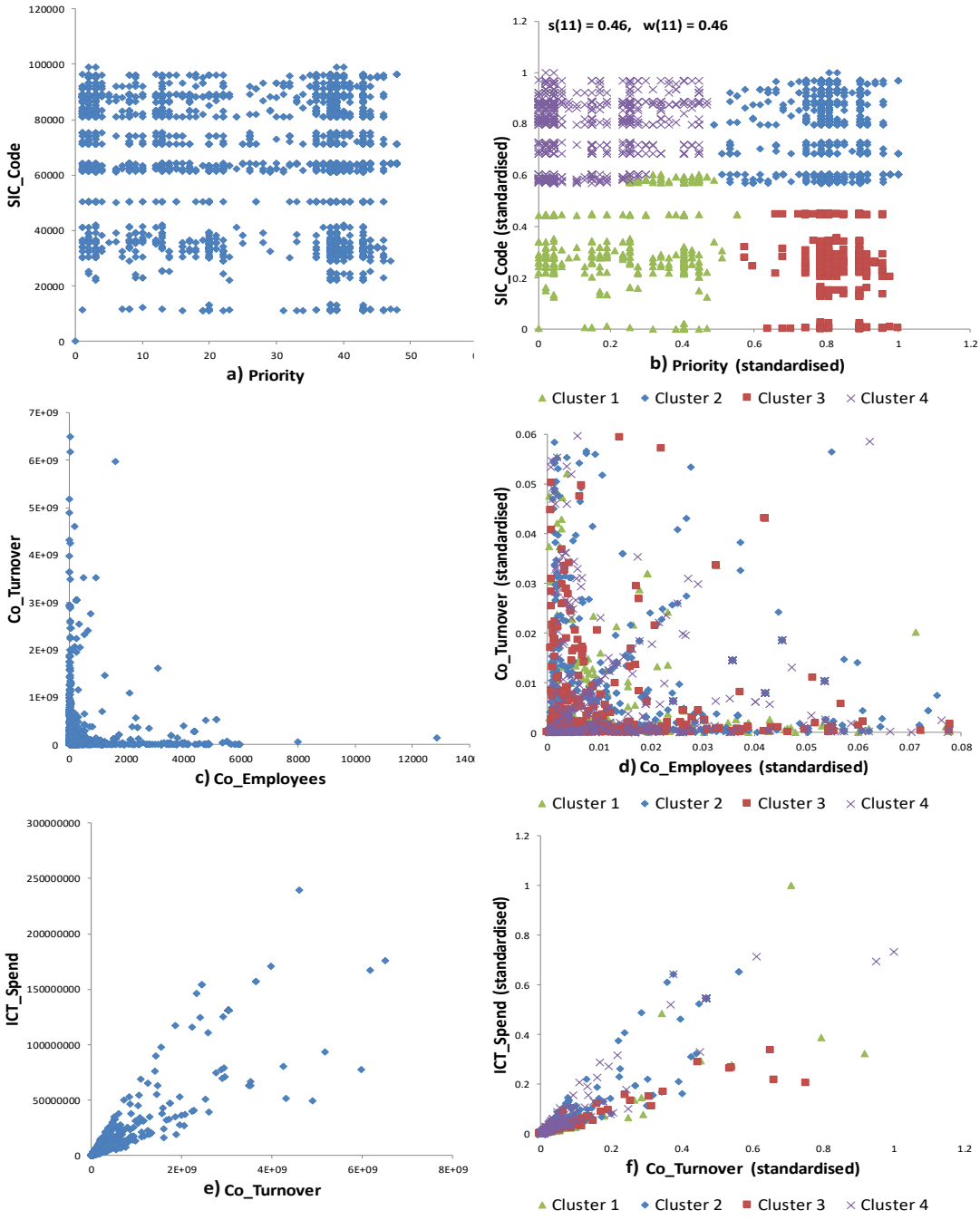


Figure 5.9: Test KMC1, last run scatter plots for validation

b. Value-based approach (tests KMC2 – KMC4)

Between three and seven iterations were performed for each test before a solution was found. Even though the initial number of clusters was specified as four, the feasible number of clusters ranged between two and four. In test KMC4, more test iterations were made after the third, with number of

iterations being 600 and 700 (from iteration ten onwards). There was no improvement in the solutions, but a move away from a solution, until run 15, when the same solution was given as in run 3. The variables used for each test, with their cluster centres are shown in section B.2 of Appendix B for each of the tests.

Using weight (w) as a measure, KMC3 was found to be the test with the best solution for value-based clustering, with $w = 0.54$ after five runs. The best before and after-bivariate plots in Figure 5.10 show that the clusters are clearest where $s(r)$ is higher, with r the run number where feasible clusters were found. However, the number of feasible clusters is then below what is needed. Note that all clusters are plotted in Figure 5.10, but in KMC2 and KMC4 tests, there are only two feasible clusters. Even if some objects overlap, the middle plot, for test KMC3, shows the best fit for the maximum number of feasible clusters, being four. Note again the variable names as used in the KMC tool. These correspond to variables used in the study.

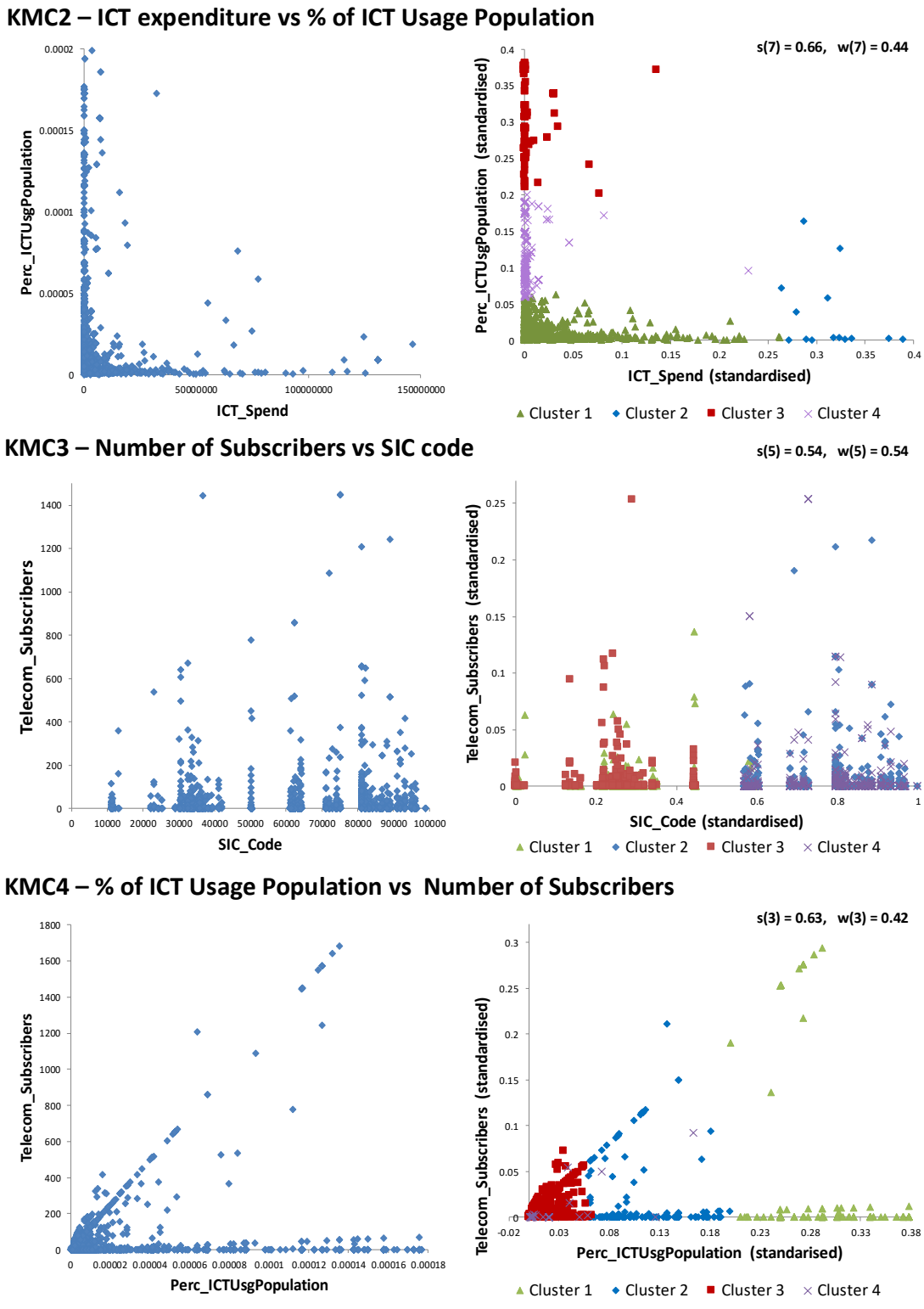


Figure 5.10: Tests KMC2 to KMC4, best bivariate plots

c. Feature and value-based approach (test KMC5)

Tests were iterated nine times before a solution was found containing an adequate number of feasible clusters ($c = 4$). The first run was a test on the raw data, with missing values replaced by the average, and clusters set at $k = 4$ and maximum number of iterations set at $j = 500$. The variables used for the first and last runs, with their cluster representation, are shown in section B.2 of Appendix B for the first and last run. Note that the variable names used in the KMC tool are shown. These correspond to variables used in the study.

More runs were made after run 9 with iterations, $j = 600$ (from run 10 onwards). There was no improvement in the solutions, but a move away from a solution, until run 13, when the closest solution was given, but with $c = 3$. The best bivariate plot for test KMC5, is displayed in Figure 5.11. There are two distinct groups of clusters, due to the nature of the nominal variable SIC code. It may be reasoned that the discrete values for industries (SIC code) are the major contributing factor to the cluster pattern.

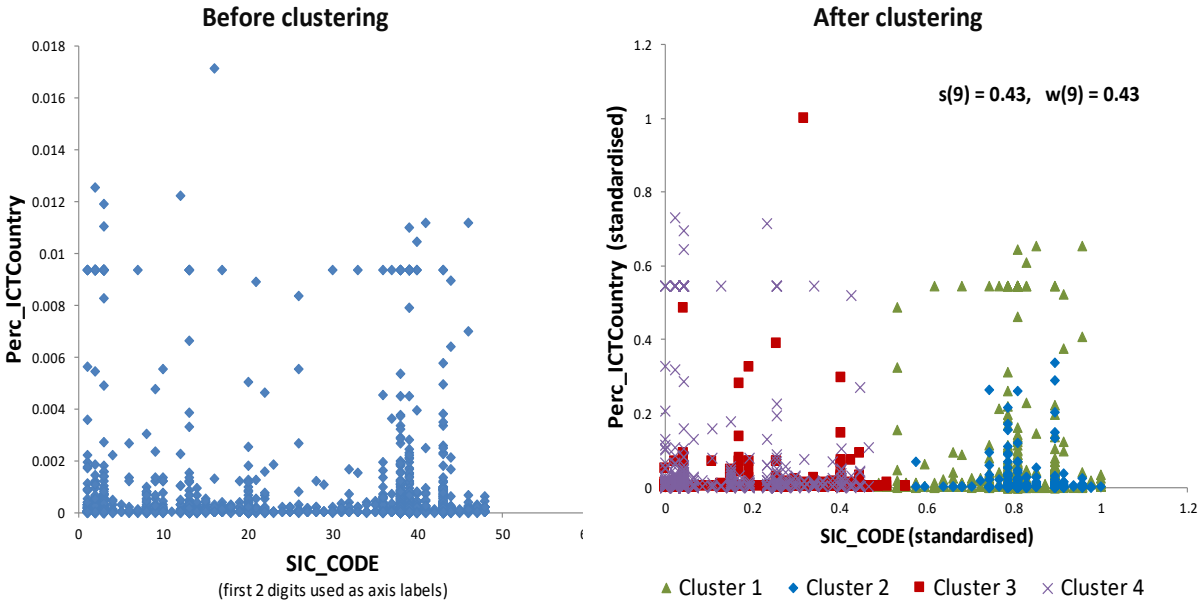


Figure 5.11: Test KMC5 best bivariate plot

d. Processing time

Processing time was measured in CPU time for each of the tests, and for each standardisation: no transformation, [0-1] transformation, and z-transformation. Standard input values were used according to the input variables in section A.2 of Appendix A. Results for each of the tests, with CPU time in minutes and seconds, are shown in section B.2 of Appendix B.

The number of input clusters (k) were slightly more than the optimal number determined, to ensure objectivity. Iterations were kept at the minimum desired level of 500. The resulting feasible clusters (c) varied for each transformation. Figure 5.12 shows a plot for these run times in seconds (displayed as mm:ss), per transformation, in order to see any dependence. An additional plot of average time and average number of feasible clusters per test is also shown.

From the graph, it is clear that the processing time increased from test KMC2 onwards, regardless of the number of feasible clusters. The same applies to the silhouette and weights, as is reflected in the final decision in the next chapter. The only change with every test is the number of input variables. Section A.2 of Appendix A shows the input variables with every test. Counting the variables shows an increase in number of variables with nearly every test (KMC1 has 5, KMC2 has 5, KMC3 has 6, KMC4 has 9 and KMC5 has 13 variables).

The processing time for KMC is highly reliant on the number of input variables. If processing time is an important factor, the number of variables will need to be kept below a certain limit.

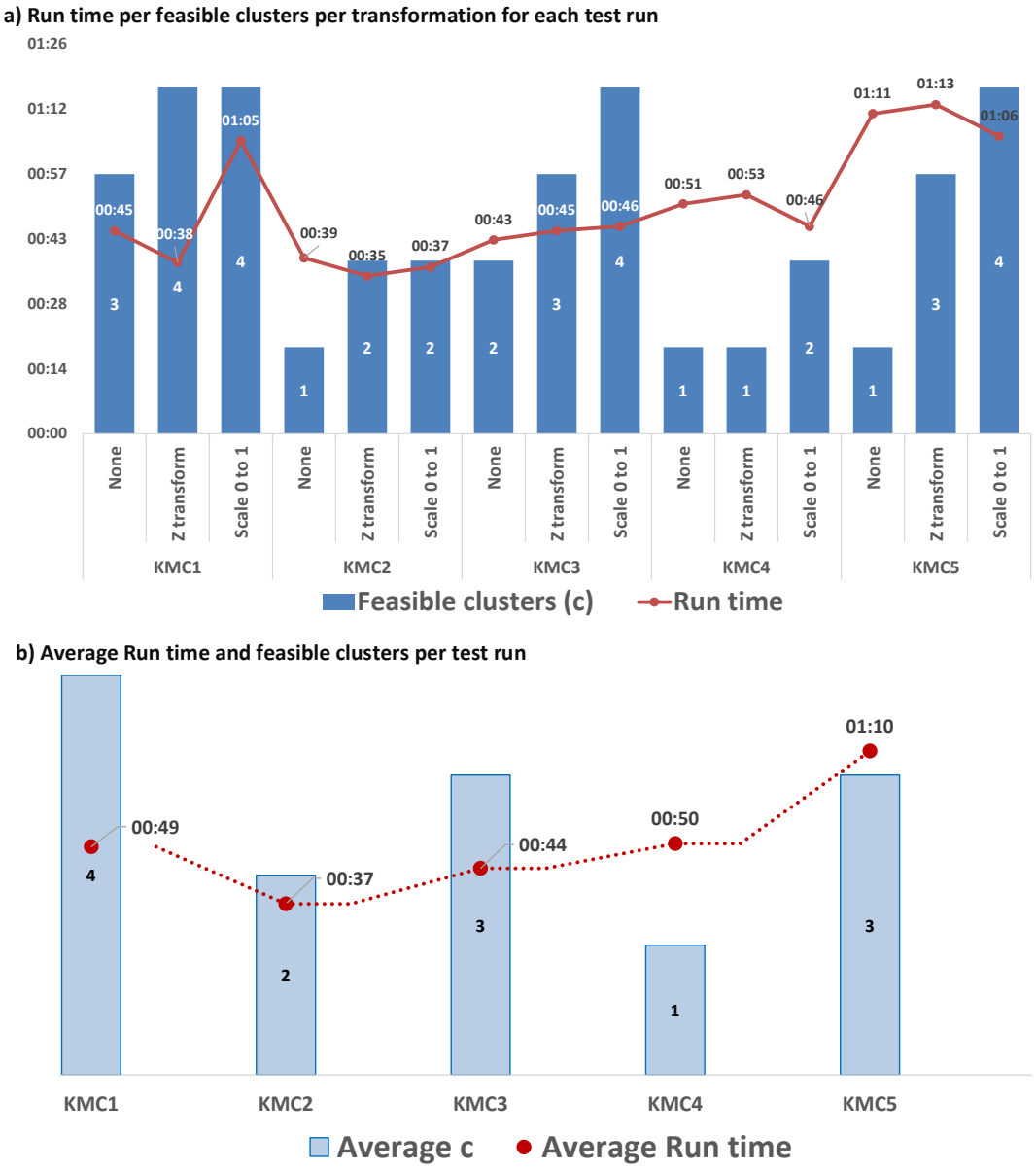


Figure 5.12: Run time (mm:ss) versus number of feasible clusters (*c*)

For example, a criterion might be that the processing time for a large company dataset of about 200k records should be under an hour. It can then be inferred that the processing time for the target market dataset of 3 362 records needs to be a maximum of one minute (01:00). The number of input variables should then be smaller than or equal to that for KMC4 tests, which is nine variables.

5.2.6 Final KMC Solution

Taking all the output metrics, according to Appendix B (section B.2) and Figure 5.12, into account, the best KMC test runs are summarised below.

KMC1:

- A feature-based approach is used, employing company firmographics.
- This will produce four feasible clusters.
- At the very least, data can be transformed via the z-transformation.
- Variables on which segmentation into clusters can be done are:
  - Location priority
  - SIC Code
  - Company Employees
  - Company Turnover
  - ICT Spend

KMC3:

- A value-based approach is used, employing customer subscriber totals.
- This will produce three feasible clusters, if a z-transformation is performed.
- By applying a further [0, 1] transformation to the data, four feasible clusters will be produced.
- Variables on which segmentation into clusters can be performed are:
  - Location priority
  - SIC code
  - Telecom ARPU
  - Telecom revenue
  - Telecom Subscribers
  - Telecom solution line counts (optional)
  - Telecom product line counts (optional)
  - Telecom device line counts (optional)

- Note that the last three variables are optional, as they are not representative, according to the data exploration. If more data sourcing is performed for these variables, these may be included.

The decision to choose the above two tests as a starting point for a KMC solution was based on the quality metrics illustrated in Figure 5.13. As explained in the section on data transformations, the z-transformation was performed first, followed by the [0, 1] scale transformation. In the first graph below, this transformation shows the best *s* and *w* values for each test.

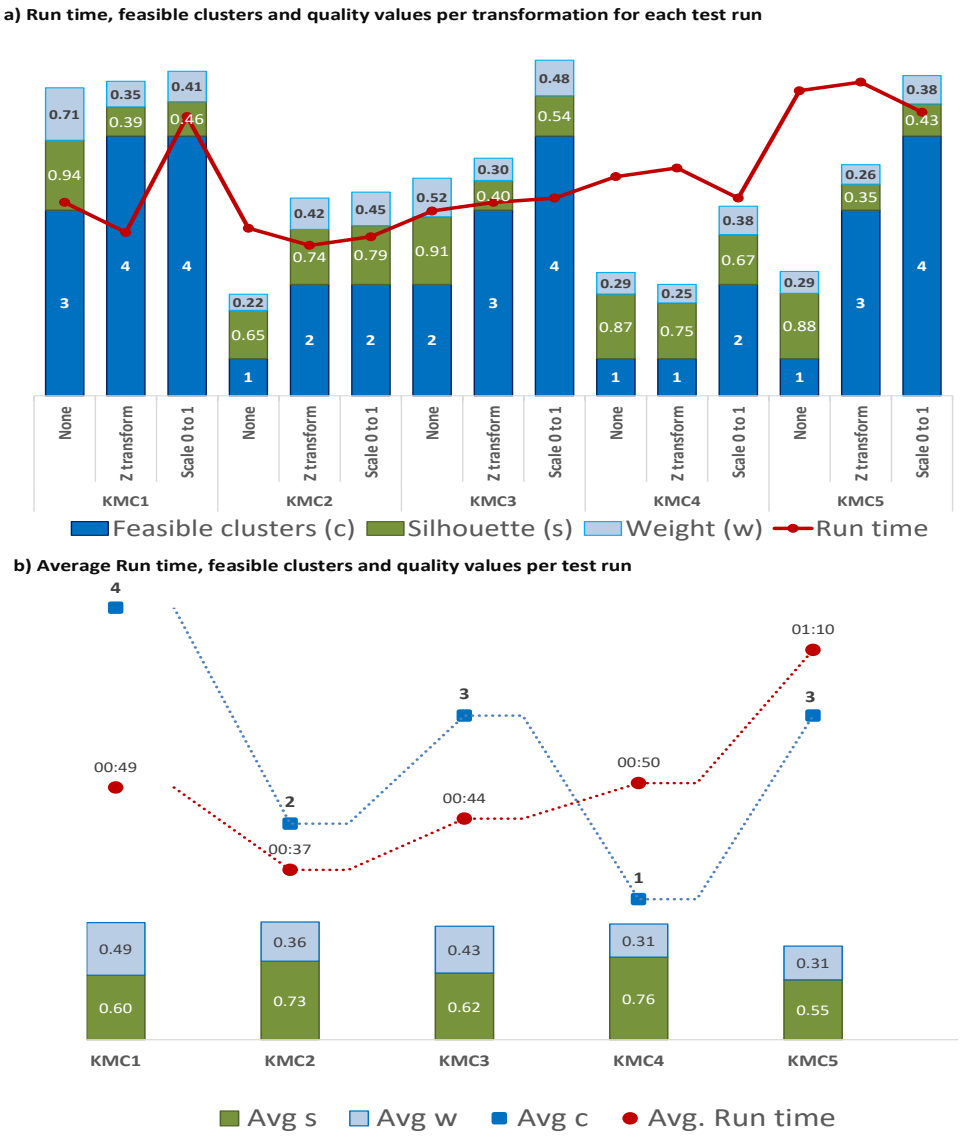


Figure 5.13: Selecting clustering solution from quality metrics

The maximum number of feasible clusters are obtained after at least one data transformation. On summarising *s* and *w* values for each test, it became apparent which test produced the highest and second highest number of feasible clusters, respectively. The best *s* and *w* combination is found in tests KMC1 and KMC3, with KMC5 a possible second candidate. Test KMC3 also satisfies the processing time constraint, with run time for each transformation being below one minute.

5.3 Particle Swarm Optimisation

5.3.1 Input data

As with the KMC algorithm test runs, different combinations of input variables were used for each of the two groups of test runs on the PSO algorithm tool. The names of each test run group and variables used are indicated in A.2.2 of Appendix A. Where applicable, these two groups represent input variables and cluster centres from the two most feasible KMC runs, KMC1 and KMC3.

The test runs were performed for the following algorithms:

- Standard PSO algorithm
- Hybrid PSO algorithms:
  - MATLAB generated *k*-means cluster centroids as input
  - KMC analysis results for *k*-means cluster centres as input

For the second hybrid PSO tests, in Appendix B, the cluster centres used as input are given in Table B.5 for KMC1 and for KMC3 in Table B.6.

5.3.2 Data transformations

For the first group, PSO1, there were no missing values. For PSO2 average values were used to fill gaps. The data for both PSO1 and PSO2 was standardised to the range [0,…,1], after which a z-transformation was applied. Test runs on original values, with only the gaps filled by the average, show no optimal solution.

5.3.3 Initial values

Below is a list of initial values set for each standard PSO algorithm as well as hybrid PSO test runs (Ballardini, 2018a).

a. Number of centroids

The number of feasible clusters in the KMC runs, being four, was used as the number of centroids for all runs.

b. Number of necessary iterations

Test runs were performed with 3, 15 and 35 iterations. For the hybrid PSO runs, 50 iterations were run in a few cases as confirmation. The fitness value mostly reached an acceptable level by iteration 15.

c. Number of particles

A constant value of 3362 was set for the number of particles, being the same as the number of records in the input dataset.

d. Number of dimensions

The number of dimensions was set according to the number of input variables in the dataset. For test run PSO1, the number of dimensions was set at five, and for test run PSO2, it was set at 8.

e. Initial position for each particle

The initial local position for each particle is an initial centroid of the swarm to which that particle will belong. For the standard PSO runs, random centroid positions were generated for the number of centroids, dimensions and particles specified. This corresponds to the algorithm described in Figure 2.2 with random vector $x_i \sim U(b_{lo}, b_{up})$, and lower and upper bounds for the particle b. The initial positions for hybrid PSO runs were set using either MATLAB-generated *k*-means centroids or KMC-generated *k*-means cluster centres.

f. Initial best position and velocity

Initial values for the best position were randomly assigned based on the number of centroids and dimensions. During the algorithm, the initial

swarm best position was replaced with the **lbest** position, which is the particle with the best local position at a point in the algorithm.

Random values were generated for the number of centroids, dimensions and particles specified. A scalar value of 0.1 was then multiplied with this random number vector to represent initial velocity for each particle.

g. Ranges to scale initial values

The range values were used to scale the initial values, in order to be within the range of the input data. The minimum of all input data was subtracted from the maximum input data per dimension (or variable).

h. Initial fitness value

The swarm fitness value was initially set at infinity. This value became smaller with each iteration, minimising the fitness function.

5.3.4 PSO clustering tests

Referring to table A.8 in Appendix A, the two groups of test runs (PSO1 and PSO2) were performed using the above initialisation values. Note that the particle position, best position and velocity relies on the initial random values when a test run is performed. These random values start with a seed value that is kept for any subsequent runs. Some of the test runs did not converge to a solution, unless the memory was refreshed to create a new seed. For these cases, the MATLAB code was executed and refreshed for every run. After each run, scatter plots of the original plot were created, with the swarm clusters overlaid on the same plot for further review. Tables of iterations, fitness values and limits are shown under section B.3 in Appendix B.

Runs for standard PSO, as well as the two hybrid PSO tests, were performed in each of the test run groups, PSO1 and PSO2.

a. Feature-based dimensions (test PSO1)

Tests were run on the same input dataset used for the feature-based *k*-means tests (KMC1). Iteration limits of 3, 15 and 35 were used for each test. Fitness values below a limit of 0.1 were deemed to be sufficient for measure of density.

i. Standard PSO1 algorithm

Figure 5.14 depicts fitness values and clusters per iteration limit. In figures 5.14b and 5.14c it is clear that the fitness level reaches a sufficient level after 3 to 10 iterations. Any iteration after this may be considered redundant and only result in clusters being so dense that they approach one single location on the plane. Clusters are much tighter than for the KMC algorithm. The graph in Figure 5.14a shows how clusters start to form after 3 iterations. In Figure 5.14c it shows how too many iterations result in clusters centring around one point, forming one cluster.

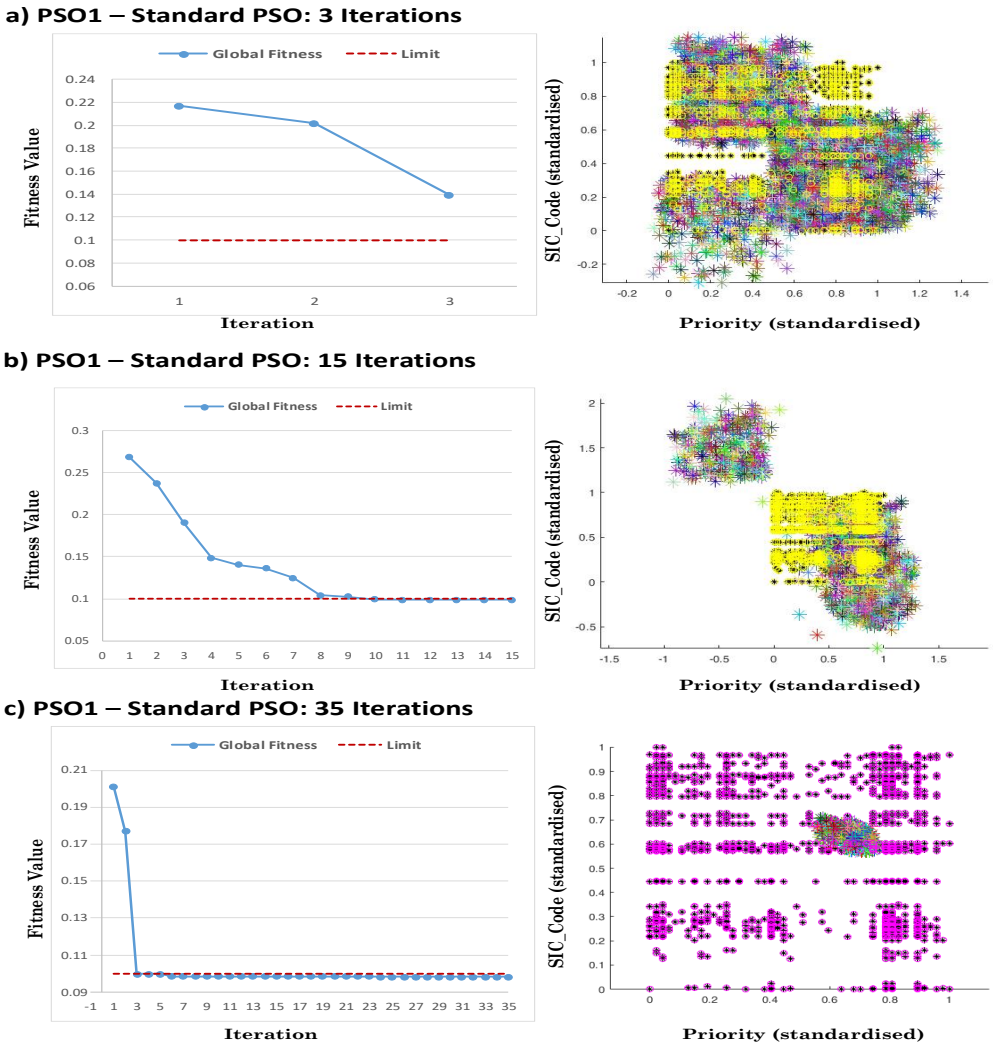
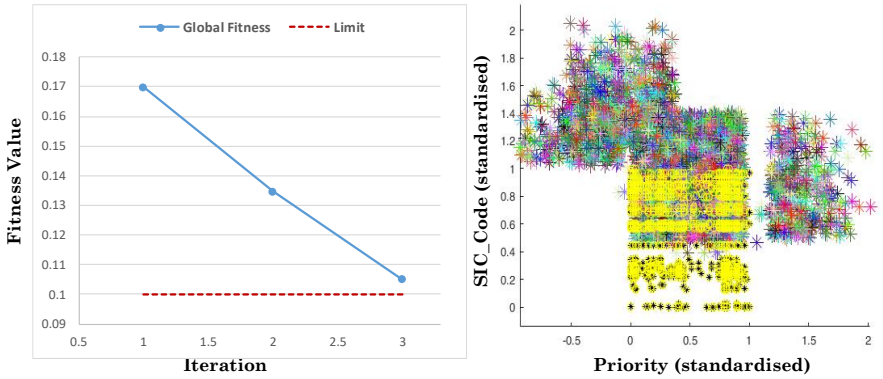


Figure 5.14: PSO1 plots for standard PSO fitness and clusters

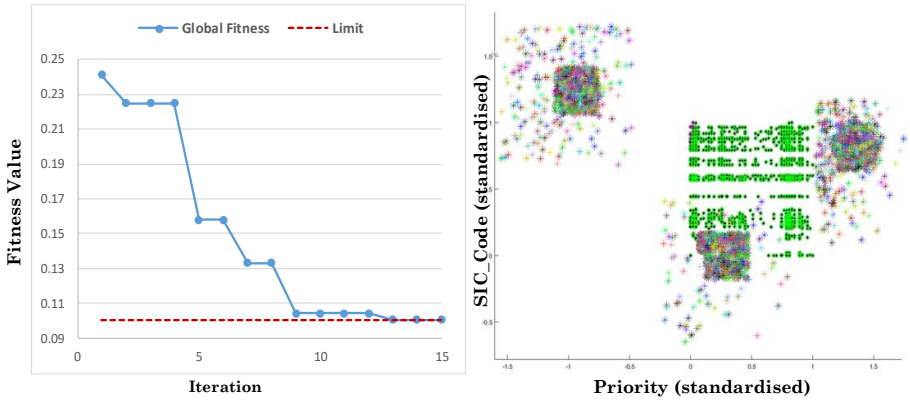
ii. Hybrid PSO1 – MATLAB *k*-means centroids

Figure 5.15 depicts fitness values and clusters per iteration limit. Here, the fitness values reach a sufficient level after 10 to 13 iterations. Any iteration after this may be considered redundant and only result in clusters being so dense that they approach one single location on the plane. Cluster density seems to improve from the standard PSO algorithm.

a) PSO1 – Hybrid PSO (MATLAB k-means centroids): 3 Iterations



b) PSO1 – Hybrid PSO (MATLAB k-means centroids): 15 Iterations



c) PSO1 – Hybrid PSO (MATLAB k-means centroids): 35 Iterations

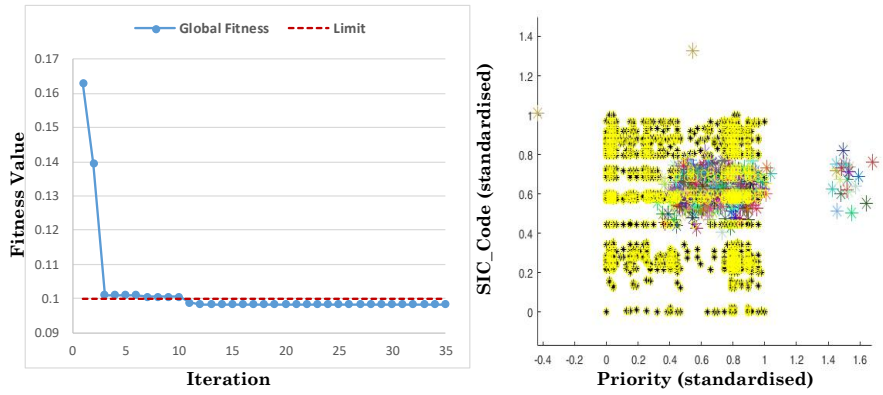


Figure 5.15: PSO1 plots for hybrid MATLAB *k*-means centroids

iii. Hybrid PSO1 – KMC *k*-means centres

Figure 5.16 depicts fitness values and clusters per iteration limit. The fitness level reaches a sufficient limit after 6 iterations. Cluster density seems to be better at lower iterations than for MATLAB-generated *k*-means centroids (MathWorks, 2019). When higher iteration limits are used, the fitness value seems to stabilise at one point. At this point, the fitness value is about 0.2. The clusters still seem to be feasible, although not as dense as required. As with the feasible clusters in the *k*-means tests, a decision may be made on whether the clusters are practical to use or not.

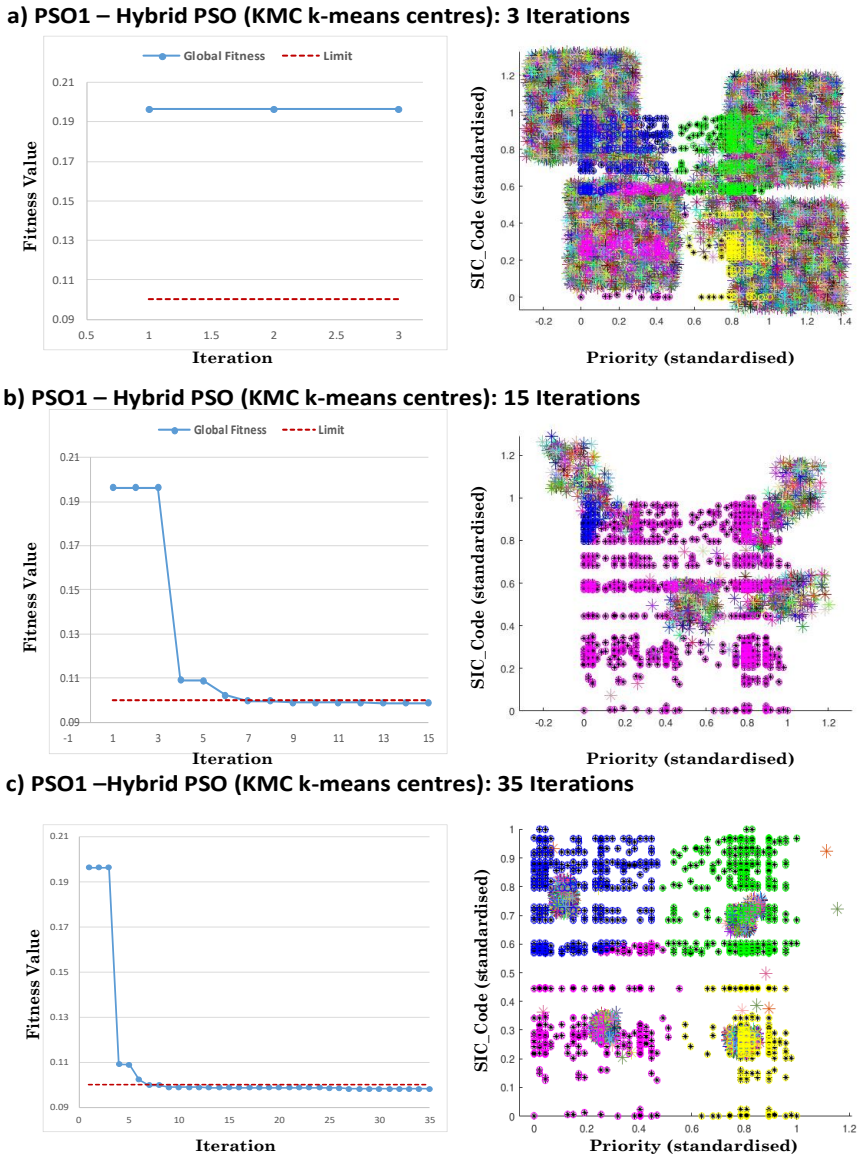


Figure 5.16: PSO1 plots for hybrid KMC *k*-means centres

b. Value-based dimensions (tests PSO2)

Tests were run on the same input dataset used for the best value-based *k*-means test (KMC3). Iteration limits of 3, 15, 35 and 50 were used for each test, although 50 iterations proved unnecessary, as this number did not improve the solution significantly. Fitness values below a limit of 0.5 were deemed to be sufficient for measuring density. Scatter plots for standard PSO and the two hybrid PSO test runs were created once again for further review.

i. Standard PSO2 algorithm

Figure 5.17 depicts fitness values and clusters per iteration limit. The fitness level reached a sufficient level only after 16 iterations. Clusters are less tight than corresponding PSO1 standard clusters.

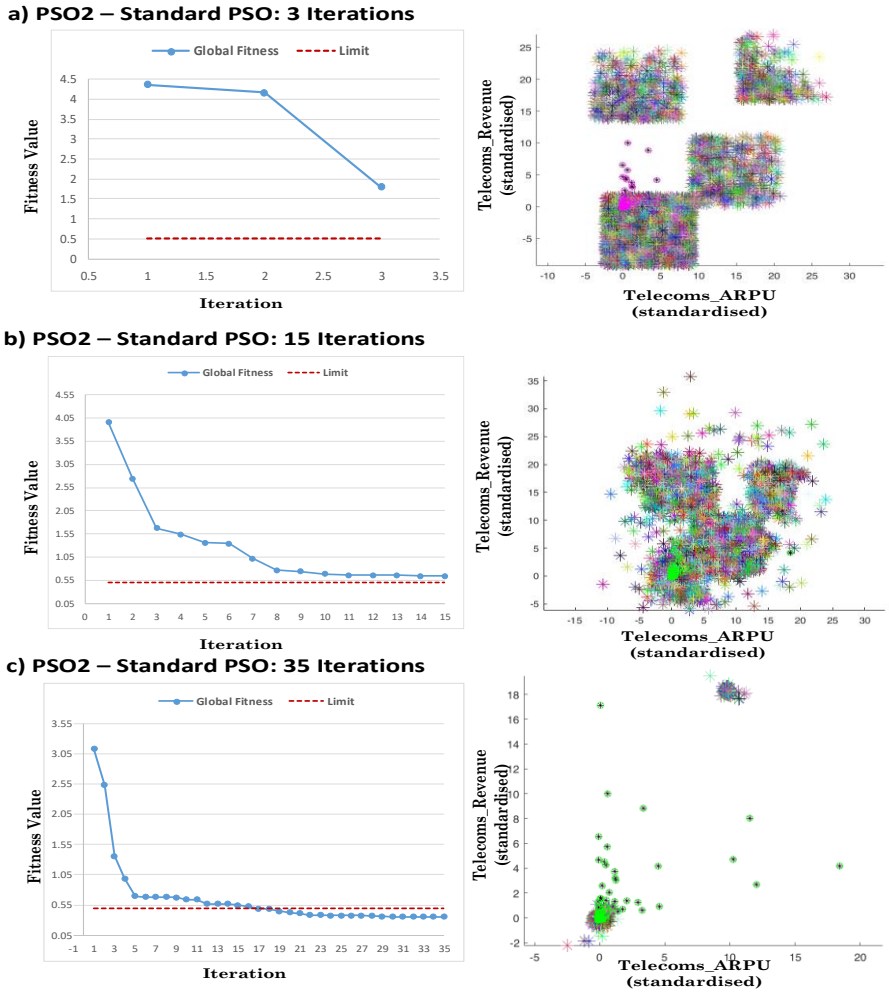


Figure 5.17: PSO2 plots for standard PSO fitness and clusters

ii. Hybrid PSO2 – MATLAB *k*-means centroids

The seeding for random initial values had to be refreshed at each run, otherwise the number of iterations for a feasible solution was very high, with fitness values staying constant. With a new seed at every run for the initial velocity position, and best position per cluster, the test run results proved to be much more favourable. According to Figure 5.18, the PSO algorithm shows the best fitness function value after 5 PSO iterations.

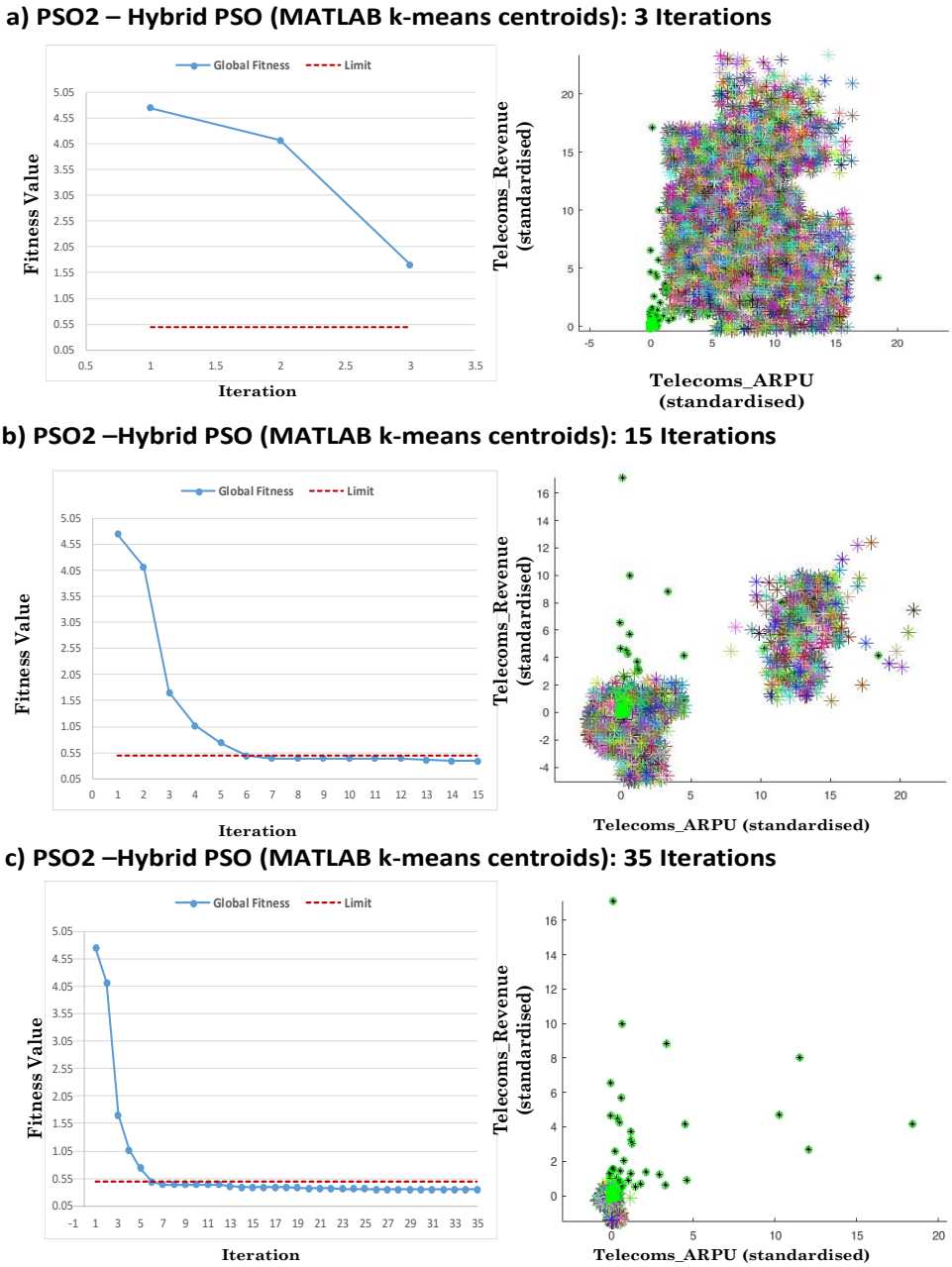


Figure 5.18: PSO2 plots for hybrid MATLAB *k*-means centroids

iii. Hybrid PSO2 algorithm – KMC k-means centres

Figure 5.19 depicts fitness values and clusters per iteration limit. The fitness level reaches a sufficient limit after 3 to 6 iterations. Cluster density seems to be tighter and reaches lower iterations than for MATLAB generated *k*-means centroids.

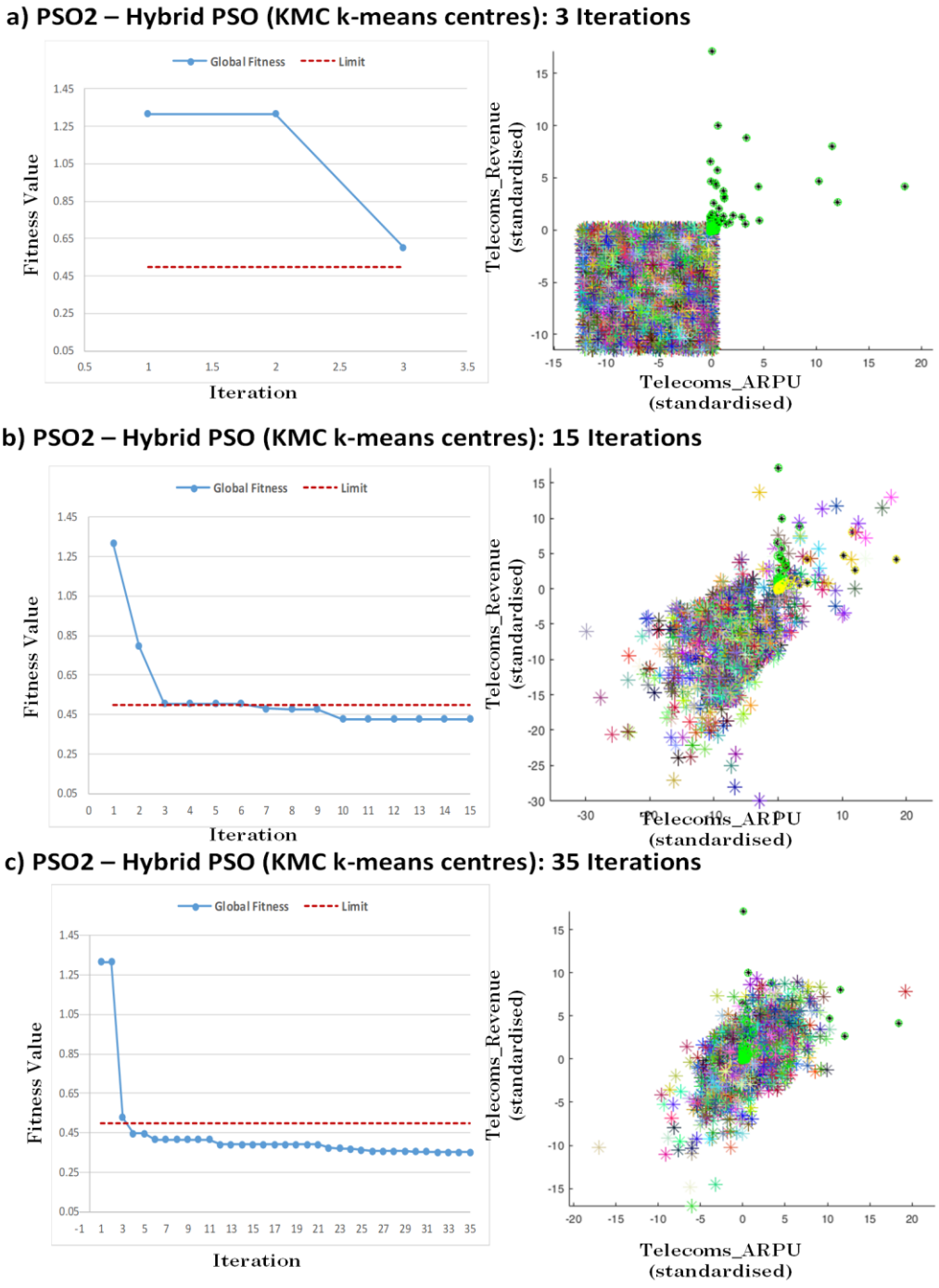


Figure 5.19: PSO2 plots for hybrid KMC *k*-means centres

c. Processing time

The cost in processing time for the PSO algorithm is quite high, mainly due to the three-way calculation per particle (velocity, position, best position) for every iteration. In addition, every swarm is evaluated virtually at the same time (each particle for each swarm is evaluated in nested loops). What PSO makes up for in the number of iterations (very few needed), is paid for in processing time, as may be seen in Table 5.2 below, where CPU processing times were measured as run times in hours and minutes.

Table 5.2: PSO test run processing times (hh:mm)

Test	PSO	Run	Dimensions	Particles	Run time per No of Iterations				Run time	Avg. Run
(t)	Algorithm	(r)	(d)	(i)	3	15	35	50	Averages	time
PSO1	Standard	1	5	3362	00:03	00:18	00:43		00:22	00:32
	Hybrid MATLAB	2	5	3362	00:07	00:37	01:27		00:44	
	Hybrid KMC	3	5	3362	00:05	00:26	01:01		00:30	
PSO2	Standard	1	8	3362	00:06	00:30	01:10	01:40	00:51	01:15
	Hybrid MATLAB	2	8	3362	00:12	01:00	02:20	03:20	01:43	
	Hybrid KMC	3	8	3362	00:08	00:42	01:38	02:20	01:12	

The processing time for the PSO test runs were long, with PSO1 runs taking on average 32 minutes and the PSO2 runs taking on average 1 hour 15 minutes. Additional tests were done for PSO2 with 50 iterations to confirm the runtimes.

As in the case of KMC, the processing time is very dependent on the number of input dimensions. As PSO results in much better clustering solutions, it may be worthwhile to implement PSO as a batch run after hours in a business environment. However, then the input data variables should be kept at between five and eight features/dimension variables, and the number of records (particles) at no more than about 80 000.

5.3.5 Final PSO solution

Taking all the output metrics depicted in figures 5.13 to 5.18 into account, the most feasible PSO cluster run from each group was chosen. As clusters are n-dimensional, one can plot results as three-dimensional, instead of bivariate plots. Figure 5.20 below depicts the 3D plots for the two test runs

found to be closest to an optimal solution. Note that only the fitness value at the stage when a feasible solution was found is used as indication of feasibility for the PSO cluster iteration.

PSO1 – Initialised with KMC cluster centres

When using the three variables, Location priority, SIC Code and ICT Spend, in standardised format, a 3D plot clearly shows the four clusters superimposed on the original transformed data. Clusters reach an optimal state after iteration 10, with a final fitness value of 0.0984 nearing a constant value for the last few iterations.

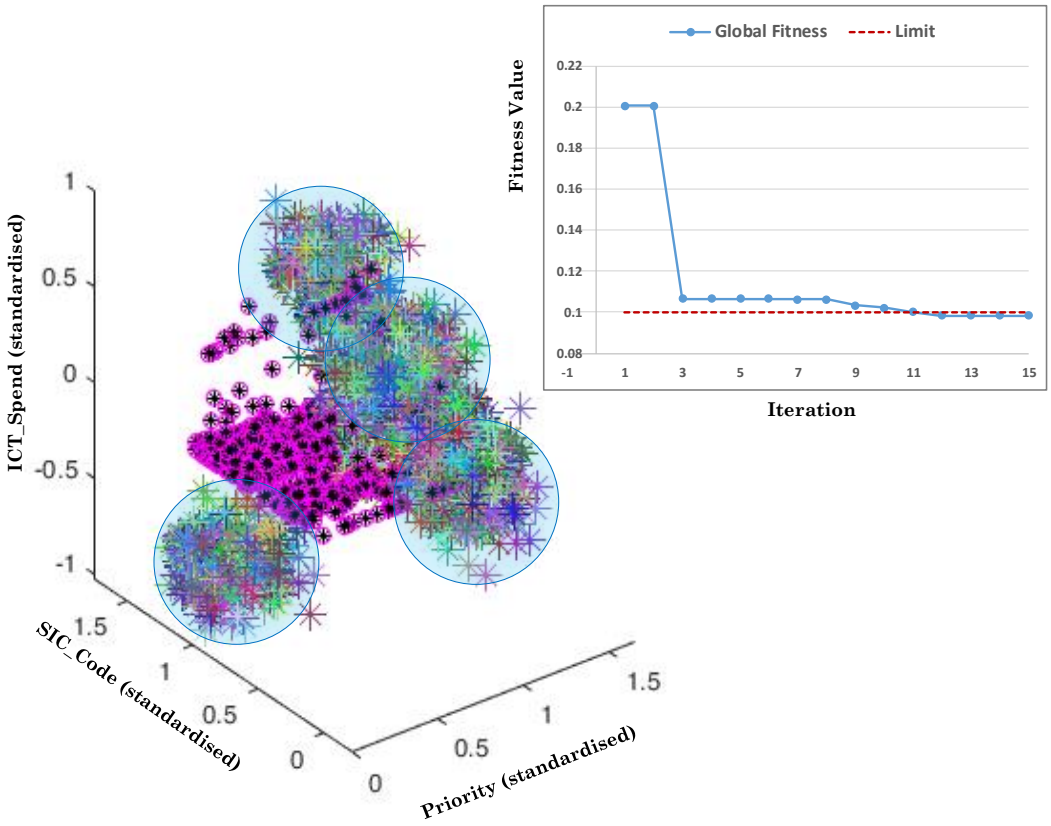


Figure 5.20: PSO1 with KMC *k*-means centres – 3D plot

PSO2 – Initialised with MATLAB cluster centroids

The 3D plot does not show clusters as clearly for standardised variables Telecom ARPU, Telecom revenue and Telecom subscribers. The clusters are localised around the origin. The zoomed view at the bottom right of Figure 5.21 shows a likeness of clusters, but it is not very distinct.

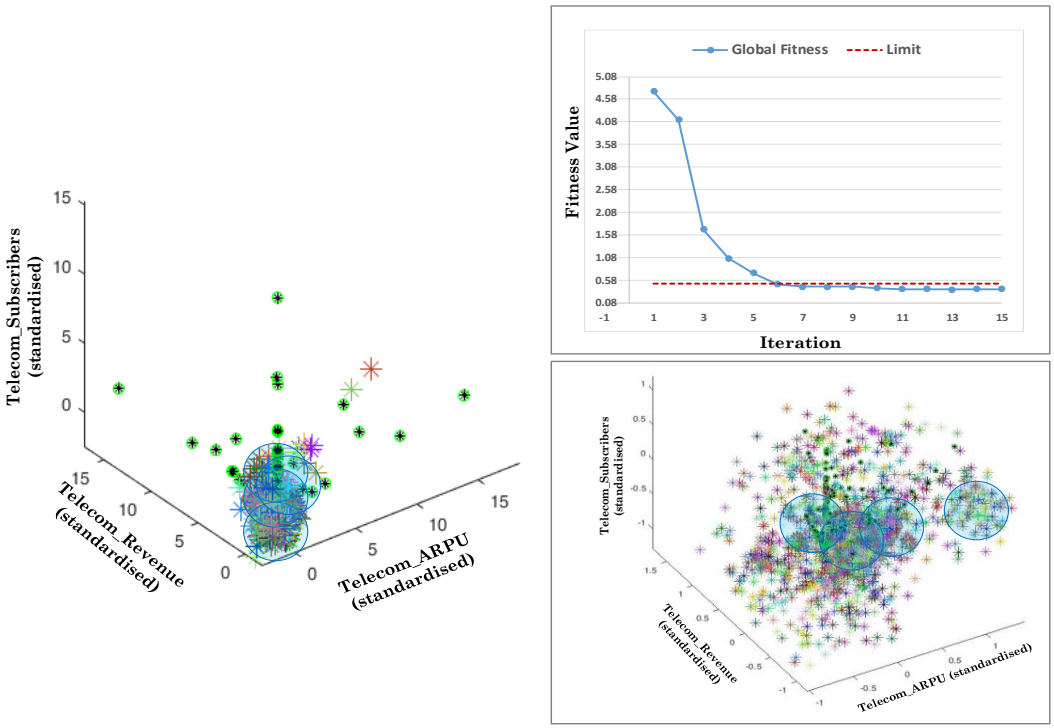


Figure 5.21: PSO2 with MATLAB *k*-means centroids – 3D plot

The lack of distinct clusters might be expected, as variables in the PSO2 run tend to be less independent of each other. Clusters reach an optimal state after iteration 5, with a final fitness value of 0.4464.

5.4 Chi-square Automatic Interaction Detection

5.4.1 Input data

In subsection A.2.3 of Appendix A the different combination of variables for the CHAID test runs are shown. The purpose of each test may be summed up as follows:

CHAID1 – The distribution of customers and prospects per sales region.

CHAID2 – Customers grouped per company size, number of employees and main solution sold.

CHAID3 – Customers grouped according to industry and main product subscribed to.

CHAID4 – The final CHAID solution: Customers, prospects and total target base grouped per number of employees and sales region using gains tables for evaluation.

5.4.2 Data transformation

Continuous variables were transformed to ordinal variables. This was done based on the groupings of the data, resulting in the ordinal categories below.

Table 5.3: Ordinal categories for CHAID analysis

ICT Spend	Telecom_ARPU	Telecom_Revenue	Co_Employees	Telecom_Subscribers
< $1M	< $50k	< $1M	24-Jan	1-4
$1M-$9M	$50k-$199k	$1M-$3M	25-49	5-9
>= $10M	$200k-$999k	> $3M	50-99	10-24
	>= $1M		100-499	25-49
			>= 500	>= 50

5.4.3 Node Options

The options described in section 4.4.3a for the nodes splitting and merging are used in the test runs for CHAID in this study, as summarised below.

Table 5.4: Significance levels and options for CHAID test runs

Test	Target	α_{merge}	α_{split}	m	*Dependent Variable*
CHAID1	Total base	1	0.8	5	Target_Flag
CHAID2	Customers	1	0.8	10	Classification
CHAID3	Customers	0.5	0.8	30	Telecom_Revenue
CHAID4	Customers	0.8	0.5	5	Target_Flag

Note that the α_{merge} and α_{split} values for significance levels are higher than the standard of a 0.05 level. This is due to a certain homogeneity in distribution of the dependent variables for predictors such as industry and region. The consequence is that low α-levels result in the impractical merging of categories. For example, the merging of categories for the variables, industries and regions, is performed purely based on the dependent variable (counts, revenue, ICT spend), where the categories are demographically far apart in practice and, therefore, need to be kept distinct. Note, however, that the α-levels may be reduced for Bonferroni corrections for n comparisons. In some cases, the merging of demographic variables would be more acceptable, as the focus is more on other variables, such as Product Type.

It was also found that the minimum node size (m in the above table) needed to be reduced in various instances. This is especially true where splits are beneficial or need to be increased. In some cases m need to be increased where splits are to be reduced and merges still result in acceptable tree structures.

5.4.4 CHAID classification tests

The tests were performed after determining the best fit for predictor and dependent variables, as shown in subsection A.2.3 of Appendix A, and representing the result in an appropriate tree structure. After a few tests were run, a decision tree was created for each scenario. Probabilities of the test statistic were calculated at each node creation step as a test for the merging or splitting of nodes.

The decision trees and detailed steps for node creation, including contingency tables, are presented in section B.4 of Appendix B. Note that tests CHAID1 and CHAID3 include additional tree structures and steps not described here, but they are included for completeness in Appendix B.

a. Customer and prospect base distribution (test CHAID1)

The merging and splitting steps for each node are shown in Table 5.5 and Table 5.6 as pairwise category comparisons and contingency tables. The **p**-values in bold show where the maximum value was, compared to the significance level α_{merge}. In none of the cases, the predictor variable categories were merged.

An attempted merge was performed for Region categories where the node size is smaller than the minimum node size ($m = 5$). The pairing of Central and Coast for Node 2 has a node size of four. However, the maximum **p**-value of 0.59 indicates that the categories are not similar enough.

Table 5.5: Merging attempts for first CHAID1 test run

Merging							
Node	**Predictor**	**Category Pairs**		**p-value**	α_{merge}	*m*	**Result**
Node 0	ICT_Spend	0	'$1M-$9M' x '>= $10M'	**0.94**	**1.0**	**5**	*Merging stopped*
		1	'>= $10M' x '< $1M'	0.13			
	Region	0	'Central' x 'Coast'	0.13	**1.0**	**5**	*Merging stopped, categories not similar*
		1	'Coast' x 'Dar Es Salaam'	0.06			
		2	'Dar Es Salaam' x 'Lake District'	**0.99**			
		3	'Lake District' x 'North'	0.46			
		4	'North' x 'South West'	0.05			
Node 1	Region	0	'Central' x 'Coast'	0.54	**1.0**	**5**	*Merging stopped, categories not similar*
		1	'Coast' x 'Dar Es Salaam'	**1.00**			
		2	'Dar Es Salaam' x 'Lake District'	0.78			
		3	'Lake District' x 'North'	0.25			
		4	'North' x 'South West'	0.01			
Node 2	Region	0	'Central' x 'Coast'	0.29	**1.0**	**5**	*Attempt merge: Pair 0 node size (4) < m*
		1	'Coast' x 'Dar Es Salaam'	0.20			
		2	'Dar Es Salaam' x 'Lake District'	**0.59**			
		3	'Lake District' x 'North'	0.12			
	Region	0	'Central' 'Coast' x 'Dar Es Salaam'	0.31	**1.0**	**5**	*No merging, categories not similar*
		1	'Dar Es Salaam' x 'Lake District'	**0.59**			
		2	'Lake District' x 'North'	0.12			
Node 3	Region	0	'Central' x 'Coast'	0.26	**1.0**	**5**	*Merging stopped, categories not similar*
		1	'Coast' x 'Dar Es Salaam'	0.03			
		2	'Dar Es Salaam' x 'Lake District'	0.70			
		3	'Lake District' x 'North'	0.78			
		4	'North' x 'South West'	**0.89**			

The splitting of nodes was also attempted, and contingency tables were created, as evident from the table shown next. The dependent variable called Target flag gives a grouping indicator of the base data, in terms of the two categories of Customer and Prospect.

Table 5.6: Evaluation for splitting of first CHAID1 test run

Splitting								
Node	**Contingency Tables**			**Result**	χ^2	**p-value**	α_{split}	*m*
	ICT_Spend	**Customers**	**Prospects**	*Split into:*				
	$1M-$9M	233	625	*Node 1*	**6.28**	**0.04**		
	>= $10M	74	196	*Node 2*				
	< $1M	520	1714	*Node 3*				
	Region	**Customers**	**Prospects**	*No action:*				
Node 0	*Central*	20	58				**0.8**	**5**
	Coast	15	77					
	Dar Es Salaam	655	1959	not the smallest p-value	7.85	0.16		
	Lake District	40	120					
	North	86	303					
	South West	11	18					
	Region	**Customers**	**Prospects**	*Split into:*				
	Central	11	21	*Node 4*				
	Coast	9	24	*Node 5*				
Node 1	*Dar Es Salaam*	167	445	*Node 6*	**8.05**	**0.15**	**0.8**	**5**
	Lake District	14	34	*Node 7*				
	North	25	95	*Node 8*				
	South West	7	6	*Node 9*				
	Region	**Customers**	**Prospects**	*No split:*		p = 0.27		
	Central, Coast	1	7					
Node 2	*Dar Es Salaam*	68	168	*will create group sizes < m*	3.94	Bonf. Adj. = 1.07	**0.8**	**5**
	Lake District	3	5					
	North	2	16					
	Region	**Customers**	**Prospects**	*Split into:*				
	Central	8	34	*Node 10*				
	Coast	6	49	*Node 11*				
Node 3	*Dar Es Salaam*	420	1346	*Node 12*	**5.50**	**0.36**	**0.8**	**5**
	Lake District	23	81	*Node 13*				
	North	59	192	*Node 14*				
	South West	4	12	*Node 15*				

The resultant decision tree has ICT spend categories separately at level 1 (node 0 split), and distinct Region categories at level 3 (node 1 and node 3

split). Note that a count is given for each node, in terms of the dependent variable categories. Detailed values were found in the gains tables. These could not be displayed clearly in Figure 5.22, being for illustrative purposes only, showing the node splits.

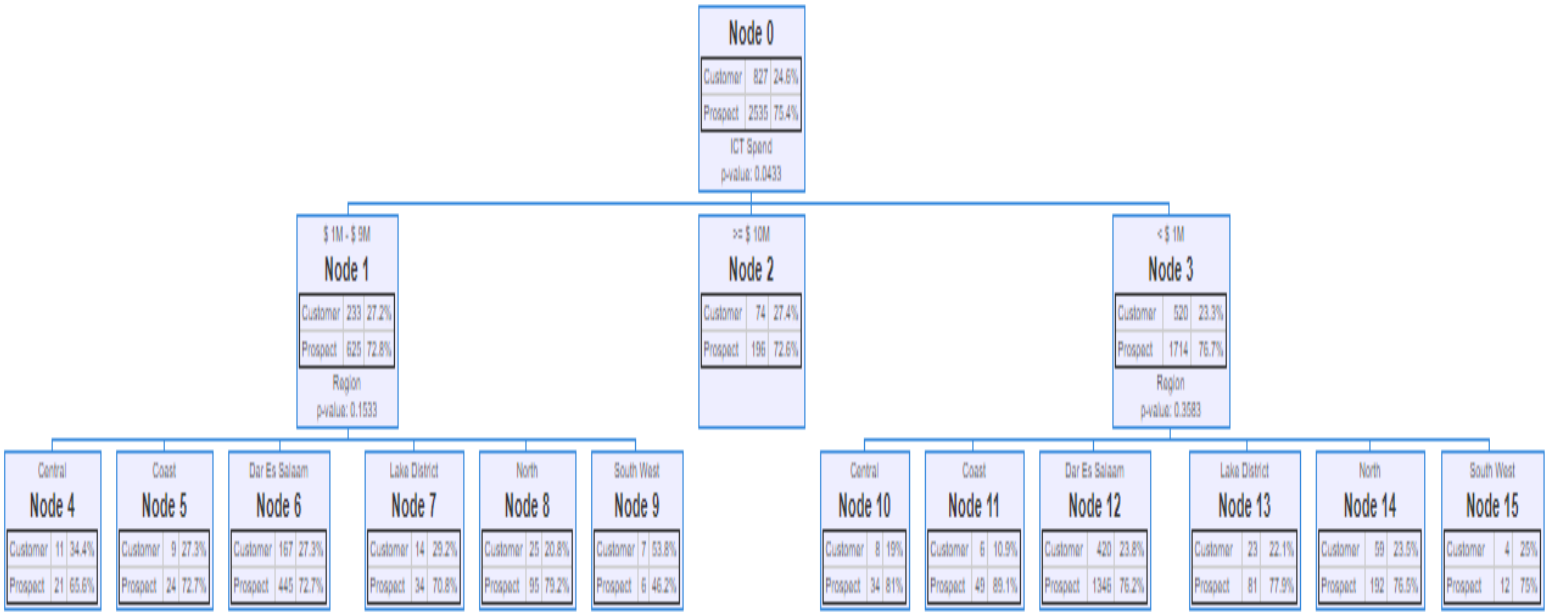


Figure 5.22: Test CHAID1 decision tree

For node 2, the Region categories were too small to split, but the categories were also too different to merge in pairs. Therefore, this node includes all categories as a single node.

b. Customer classification per number of employees (test CHAID2)

In this test, the count of records is given per company size. The dependent variable, Classification, gives the size of a company, in terms of the five categories of Large, Medium, Small, Very Small and Micro.

As an example, the **p**-value plots for merging, and contingency tables for splitting, are shown below for node 5 (where Company employee count is '50–99') and node 9 (where Telecom ARPU is < $50k). A visual plot of **p**-values for node 5 predictor variable categories shows how the significance level is not met for merging.

Figure 5.23: Test CHAID2 plots for Node 5 **p**-values

The maximum **p**-value in each plot does not reach 0.8, which is still much smaller than the significance level of 1.0. Therefore, no merges were performed on any predictor variable categories. As is evident from the contingency table for node 5 below, no splits were performed for Region; hence, no nodes for Region were created.

Table 5.7: Splitting of Node 5 for CHAID2 test run

	Contingency Tables						Result	χ^2	p-value	α_{split}	m
	Region	Large	Medium	Micro	Small	Very Small	*No split*				
	Central	3	1	0	0	0	$p - value < \alpha_{split}$ BUT total counts (node size) per 'Region' in contingency table is less than m in most cases	**14.65**	**0.15**		
	Coast	3	2	0	0	0					
	Dar Es Salaam	50	58	0	8	0					
	Lake District	9	1	0	1	0					
	North	7	9	0	0	0					
	South West	4	0	0	0	0					
	Telecom_ARPU	Large	Medium	Micro	Small	Very Small	*Split into:*				
Node 5	*< $50k*	55	39	0	4	0	*Node 10*			**0.8**	**10**
	$50k - $199k	10	19	0	1	0	*Node 11*	**12.31**	**0.06**		
	$200k - $999k	5	7	0	1	0	*Node 12*				
	>= $1M	6	6	0	3	0	*Node 13*				
	Telecom_MainSolution	Large	Medium	Micro	Small	Very Small	*No split*				
	Fixed	4	1	0	0	0	$p - value < \alpha_{split}$ BUT total node size per 'Telecom_MainSolution' is less than m in some cases	**57.13**	**1.70E-09**		
	IOT	32	0	0	0	0					
	MPESA	8	6	0	1	0					
	Mobile	24	39	0	2	0					
	Unknown	8	25	0	6	0					

Similar to Region, no split was performed on the categories for Telecom main solution. In node 9, however, a split was performed on these categories, as seen below. A split could be performed from node 5 into nodes 10 to 13 for categories of Telecom ARPU, as the **p**-value < α_{split} significance level, and all totals per category in the contingency table are greater than the minimum node size, m.

The visual plot below of **p**-values against the α_{merge} significance level shows that the significance limit was reached, but not exceeded by the **p**-values in a few cases and, therefore, no merges were performed.

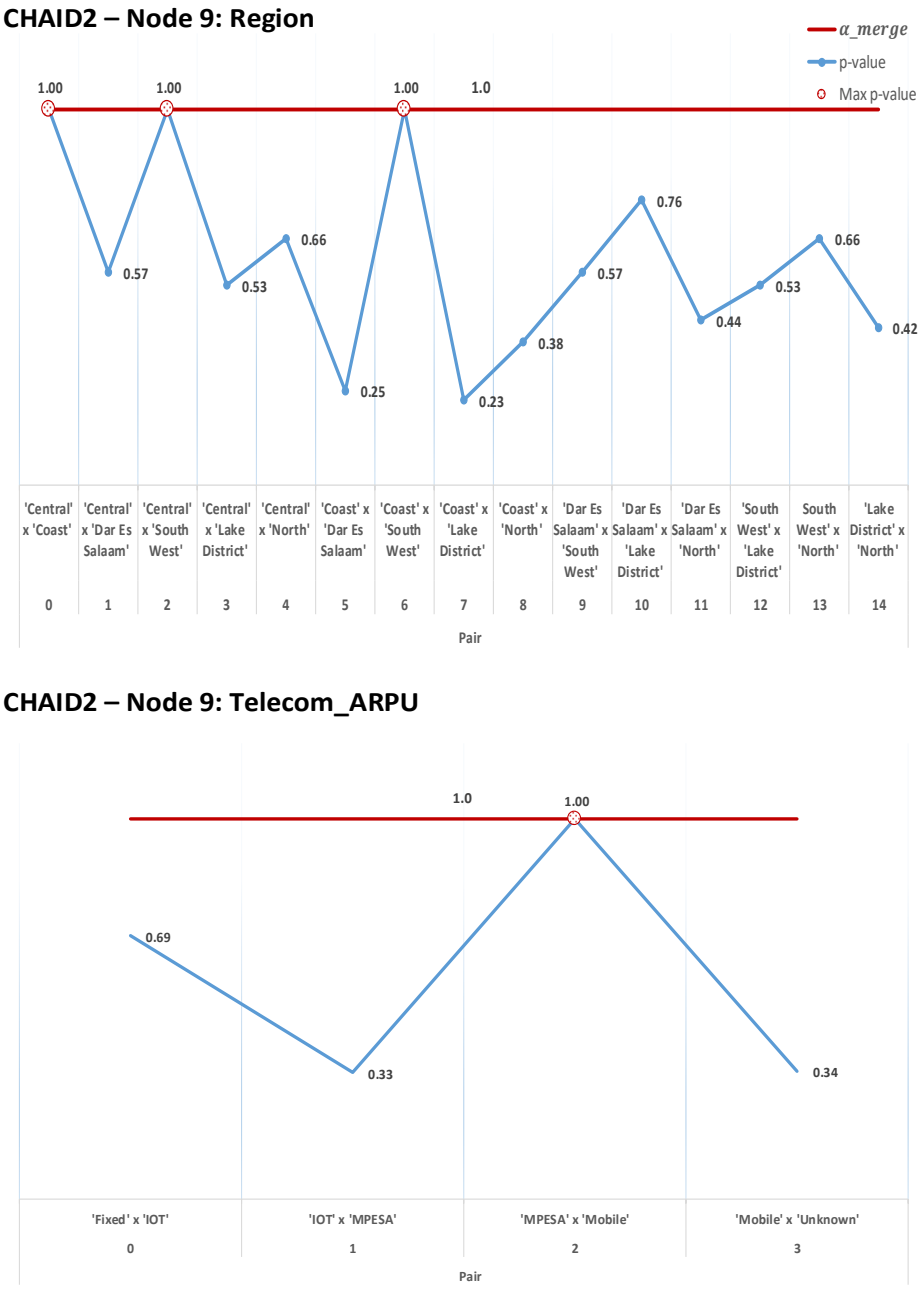


Figure 5.24: Test CHAID2 plots for node 9 **p**-values

The categories for Telecom main solution need to be compared for splitting, as shown in Table 5.8 below, before any further merging attempts can be made. The total counts for Region is mostly below *m* and, therefore, a split of the categories was not possible here, and merging did not satisfy the significance level either.

Table 5.8: Splitting of node 9 for CHAID2 test run

	Contingency Tables						Result	χ^2	p-value	α_{split}	m
Node 9	**Region**	**Large**	**Medium**	**Micro**	**Small**	**Very Small**	*No split* p-value $<$ α_{split} BUT total counts (node size) per 'Region' in contingency table is less than m in most cases	**2.67**	**0.75**	**0.8**	**10**
	Central	1	0	0	0	0					
	Coast	4	0	0	0	0					
	Dar Es Salaam	139	46	0	0	0					
	Lake District	10	4	0	0	0					
	North	15	3	0	0	0					
	South West	1	0	0	0	0					
	Telecom_MainSolution	**Large**	**Medium**	**Micro**	**Small**	**Very Small**	*Split into:*	**5.59**	**0.23**		
	Fixed	16	1	0	0	0	*Node 14*				
	IOT	9	1	0	0	0	*Node 15*				
	MPESA	15	5	0	0	0	*Node 16*				
	Mobile	117	39	0	0	0	*Node 17*				
	Unknown	13	7	0	0	0	*Node 18*				

The categories for Telecom main solution may be split at nodes 14 to 18 as significance level, if node-size criteria are satisfied. Nodes 5 and 9 and their sub-nodes are shown on the right of the decision tree for CHAID2 below. Note that output of the Easy CHAID application are web-based and could not be displayed clearly in Figure 5.25. It is therefore shown for illustrative purposes only, showing the node splits.

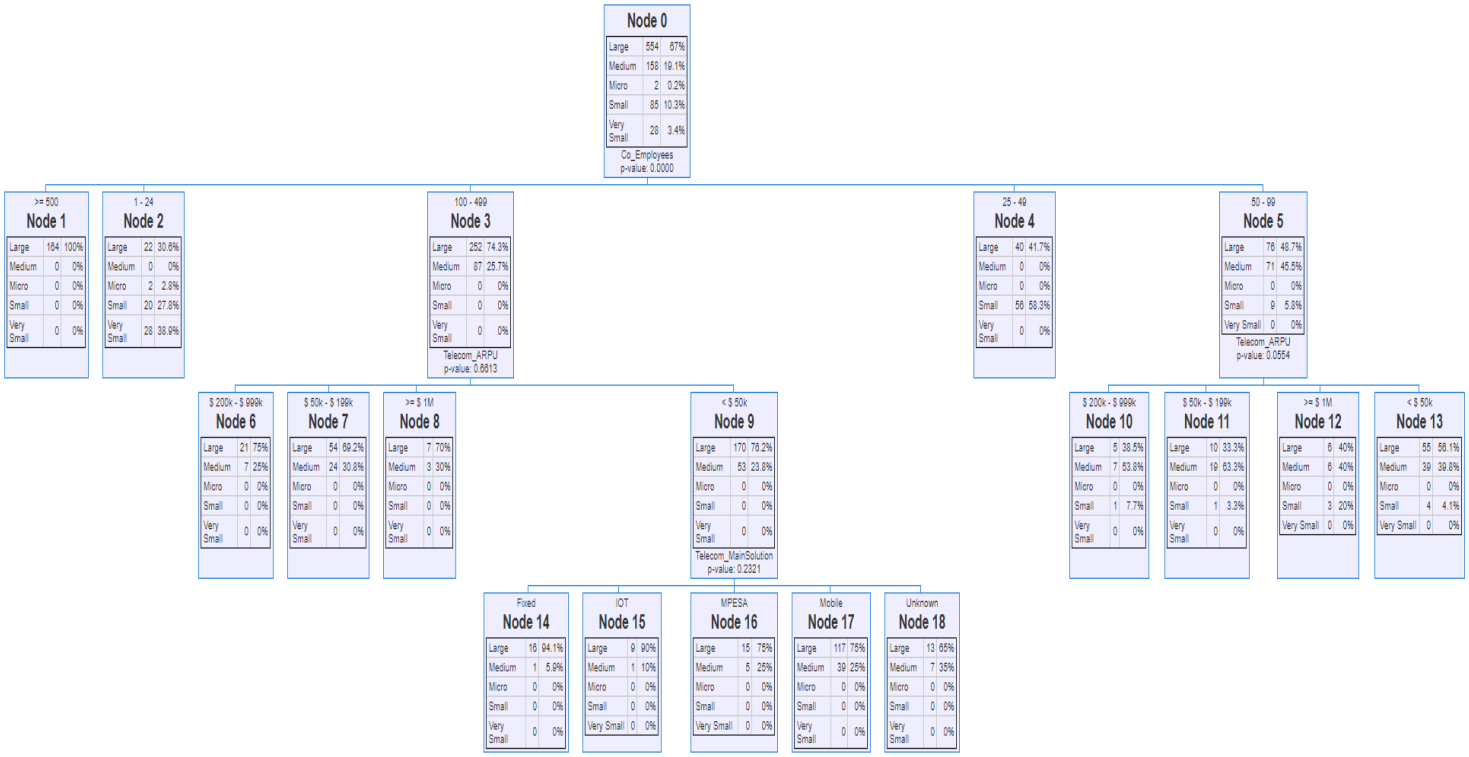


Figure 5.25: Test CHAID2 decision tree

c. Customers per industry and main product (test CHAID3)

Here, the categories of Industry were found to have the most influence on the dependent variable, and were merged and split on the first level. Further merges and splits were performed on Telecom product and Telecom subscribers. This comparing process is briefly described for node 3, the node with the largest node size. Node 3 contains the merged Industry categories of Community and Personal Services, Financial, Health and Retail, Wholesale, Trade. The most significant split for this category is on Telecom main product. The counts are shown below for the categories of the dependent variable Telecom revenue.

Table 5.9: Telecom Main product counts for Node 3 of CHAID3

Telecom_MainProduct	$1M-$3M	<$1M	>$3M	Total Count
Data MPLS/VPN	3	10	5	**18**
M2M	10	5	4	**19**
Mobile Data		6	2	**8**
Mobile Integrated	7	35	6	**48**
Mobile Voice	1	7	1	**9**
Mpesa	12	22	5	**39**
Top-Up	16	81	12	**109**
Unknown	17	53	11	**81**
Total	**66**	**219**	**46**	**331**

The shaded areas in the table show where the criteria for minimum node size were not met (m = 30, according to section 5.4.3). After a number of merging steps, the final split was performed, as shown below.

Table 5.10: CHAID3 final split of Node3

	Contingency Tables					Result	χ^2	p-value	α_{split}	α_{merge}	m
	Telecom_MainProduct	$1M-$3M	>$3M	<$1M	Total	*Split into:*					
Node 3	*Data MPLS/VPN, M2M*	13	9	15	**37**	*Node 7*	**19.53**	**0.01**	**0.80**	**0.5**	**30**
	Mobile Data, Mobile Integrated, Mobile Voice	8	9	48	**65**	*Node 8*					
	MPESA	12	5	22	**39**	*Node 9*					
	Top-Up	16	12	81	**109**	*Node 10*					
	Unknown	17	11	53	**81**	*Node 11*					

The **p**-value decreased significantly to a low 0.01, showing that an acceptable split was performed. It is interesting to note that most counts for this node are for Top Up, which is valuable for market planning in the combined industries (Community and Personal Services, Financial, Health and Retail, Wholesale, Trade) represented by node 3.

The category for node 11 is shown as Unknown, and this is further split on Industry, after testing against the null hypothesis. Since Industry in node 11 has more than two categories, attempts were made to merge similar categories. The pairs of categories for Industry, with their **p**-values, are:

- Pair 0: Community and Personal Services and Financial, with **p**-value: 0.86
- Pair 1: Community and Personal Services and Retail, Wholesale, Trade, with **p**-value: 0.35
- Pair 2: Financial and Retail, Wholesale, Trade, with p-value: 0.12.

The highest **p**-value is 0.86 for pair 0. This leaves Retail, Wholesale, Trade, which is already a distinct category. In similar vein, the categories for Telecom subscribers are split into three sub-nodes of node 6. The contingency tables for Telecom subscribers and Telecom revenue are then as shown below.

Table 5.11: CHAID3 final split of node6 and node 11

	Telecom_Subscribers	$1M-$3M	>$3M	<$1M	Sub-nodes	Total Count
Node 6	*>=50*	16	15	40	Node 12	**71**
	1-4	1	5	42	Node 13	**48**
	10-24, 25-49, 5-9	16	23	53	Node 14	**92**
	Total	**33**	**43**	**135**		**211**

	Industry	$1M-$3M	>$3M	<$1M	Sub-nodes	Total Count
Node 11	*Community & Personal Services, Financial*	9	1	22	Node 15	**32**
	Retail, Wholesale, Trade	8	10	31	Node 16	**49**
	Total	**17**	**11**	**53**		**81**

The above result shows that predictor variables can be revisited to further split their categories on levels further down the decision tree. The resultant decision tree has ICT spend as dependent variable, and merged Industry and Telecom main product with a split of Telecom subscribers for one of the nodes. The decision tree output from Easy CHAID is shown in Figure 5.26 for illustrative purposes.

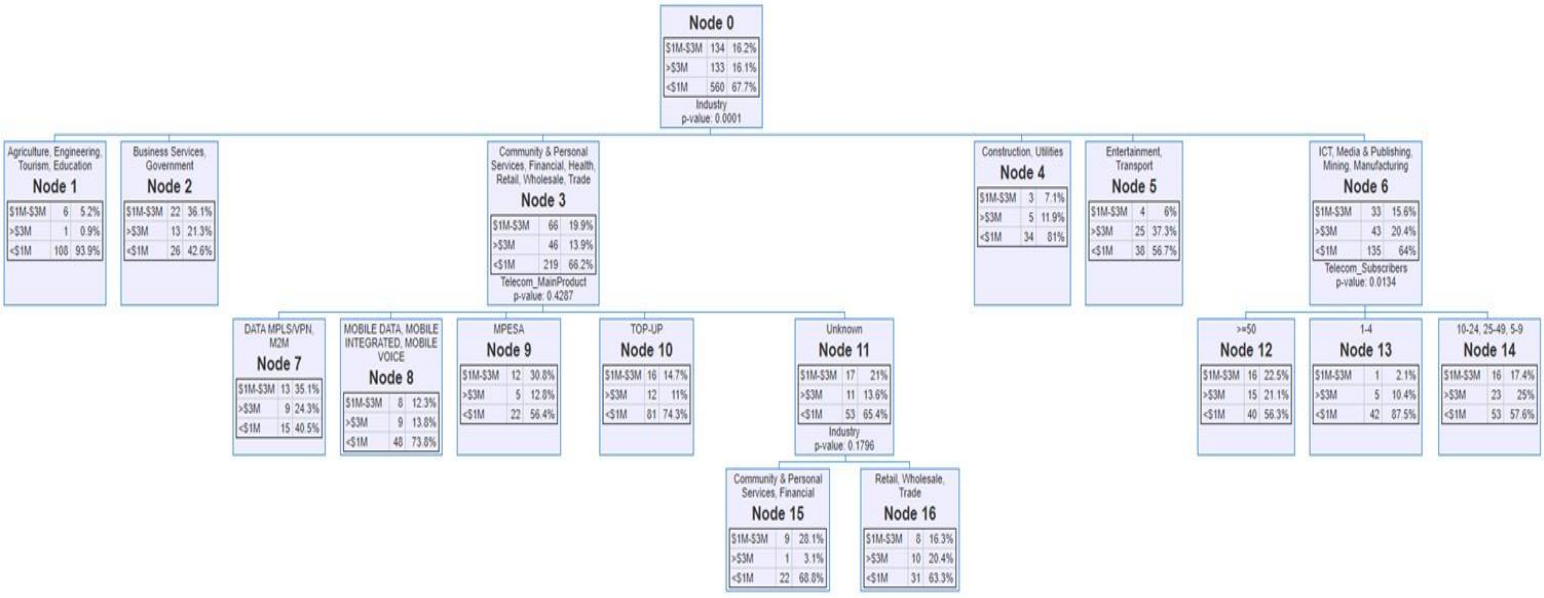


Figure 5.26: Test CHAID3 decision tree

d. Processing time

The Easy CHAID web-based application was used to set up a decision tree with results, where no multiple cycles or iterations were executed. As processing were done almost instantaneously through the tree diagram, CPU time could not be measured. The processing time was estimated based on the relative iteration times of the KMC tests.

A factor were derived from the KMC run times, where the average run time was taken as 50 seconds. Conservatively, should a CHAID run be one second, then KMC runs would be 50 times slower. Therefore every second were divided by 50. This value was used as the unit to calculate run times based on the number of predictor nodes in the decision tree. The results were slightly adjusted with a factor to make provision for the pairwise calculations of the **p**-value. The estimated processing times for CHAID runs are given as units of 1/60th of a second, for example 00:00:30 is the format for 30 units of 1/60th second, or half a second. In Table 5.12 below the estimated times are shown in the format *mm:ss:tt* (t for the unit of 1/60th second). These run times, together with the number of nodes and **p**-value, are shown for each CHAID run.

Table 5.12: CHAID tests estimated run times

Test (t)	Run (r)	Nodes	Segments	p-value	Run time (mm:ss:tt)	Avg. No. of Nodes	Avg. Run time
CHAID1a	1	15	12	0.43	00:00:28		
CHAID1b	2	14	11	0.26	00:00:32	17	00:00:33
CHAID1c	3	23	16	0.09	00:00:39		
CHAID2	1	18	15	0.36	00:00:43	18	00:00:43
CHAID3a	1	16	13	0.22	00:00:35	15	00:00:33
CHAID3b	2	14	11	0.2	00:00:30		
CHAID4	1	19	15	0.32	00:00:26	19	00:00:26

5.4.5 Final CHAID Solution

Any of the decision trees may be used as a solution for a CHAID test run, depending on the objective and ease of implementation. In this section, a manual implementation of a solution for the CHAID4 test run is demonstrated. The steps to create the tree structure of CHAID4 is given in subsection B.4.2 of Appendix B.

a. Decision tree diagram

The first step towards implementing the final solution is to use the last nodes (terminal nodes or branches) in the decision tree as classification groups or segments. A typical CHAID model may have about twelve terminal nodes resulting in segments. By changing the node options at the start of the tree-growing process, a model could be built with more segments. In the solution for the CHAID4 test, there are fifteen segments, in order to cover as much separate categories possible for predictors ICT spend and Region.

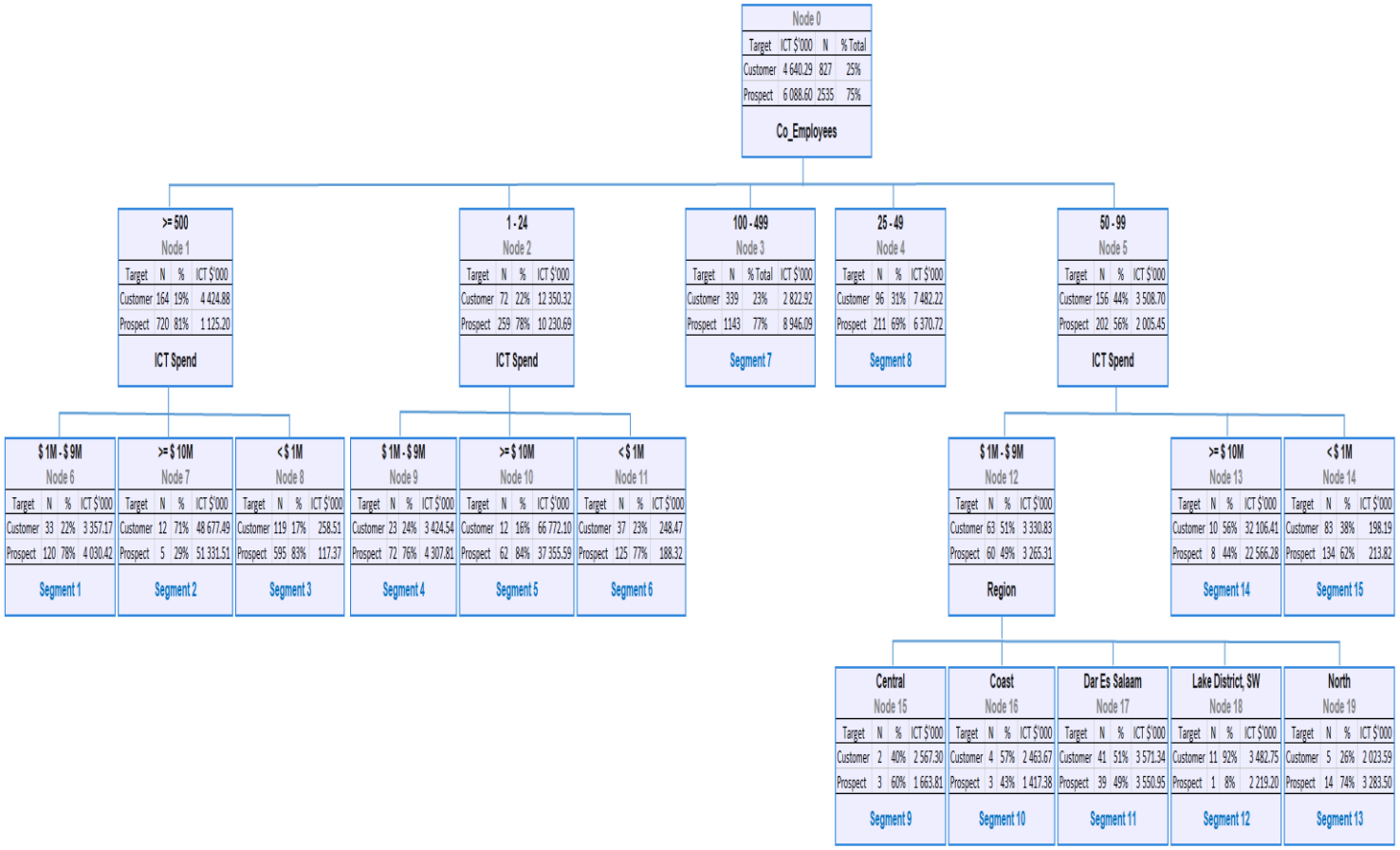


Figure 5.27: Test CHAID4 decision tree with segment nodes

b. Gains table

The segments depicted in the tree diagram are also ranked in a gains table (Table 5.13), which provides quantitative detail that is useful for market and sales planning. Statistics produced by the gains table make it easy to select prospects representing a given level of above-average performance. An example would be: telecommunication revenue; subscribers; companies per category and telecommunication expenditure. Financial assumptions may also be gathered as part of the predictive CHAID model to generate various estimates, such as profit gains. In CHAID4, the dependent variable has two values, namely customer and prospect. Therefore, a nominal CHAID model is generated. The gains table below is based on the total target base for CHAID4. Separate gains tables for prospects and customers are shown in subsection B.4.2 of Appendix B.

Table 5.13: CHAID4 gains table for the total target base

Segment ID	Segment Count	Percent of Total	Average ICT Spend ($ '000)	Segment Index	Cum. Count	Cum. Percent	Cum. $ ICT Spend	Cum. Index
2	17	0.5%	49 458.09	863	17	0.5%	49 458.06	863
5	74	2.2%	42 125.83	735	91	2.7%	43 495.59	759
14	18	0.5%	27 866.35	486	109	3.2%	40 914.61	714
7	1 482	44.1%	7 545.45	132	1 591	47.3%	9 831.58	172
8	307	9.1%	6 718.29	117	1 898	56.5%	9 328.01	163
4	95	2.8%	4 093.97	71	1 993	59.3%	9 078.52	158
1	153	4.6%	3 885.21	68	2 146	63.8%	8 708.26	152
11	80	2.4%	3 561.40	62	2 226	66.2%	8 523.29	149
12	12	0.4%	3 377.45	59	2 238	66.6%	8 495.70	148
13	19	0.6%	2 951.95	51	2 257	67.1%	8 449.03	147
9	5	0.1%	2 025.20	35	2 262	67.3%	8 434.83	147
10	7	0.2%	2 015.26	35	2 269	67.5%	8 415.02	147
6	162	4.8%	202.06	4	2 431	72.3%	7 867.72	137
15	217	6.5%	207.85	4	2 648	78.8%	7 240.00	126
3	714	21.2%	140.89	2	3 362	100.0%	5 732.34	100

The top ±3% of the total base, contributing about 75% of the ICT spend value, is shown in the darkest shading. The next highest ± 50% of the base, contributing about 10% of the ICT spend, is shown in a lighter shading. The shades gradually become lighter until the lightest shade is found at the bottom, for the smallest contributor, namely the bottom ±30% of the base contributing to about 0.35% of the ICT spend.

c. Further analysis

On inspection of the final gains table (Table 5.13), some very useful deductions was made. For example, the best segment has an index of 863, which means it has the capacity to spend 8.63 times more than the average of the whole target base, and at least 430 times more than the worst segment (with segment index of 2).

From the cumulative values, the gains table shows that the best three segments (segments 2, 5 and 14) represent 3.2% of the total base, have an average ICT spend of $40 914.61 and, therefore, spend 7.14 times as much as the average of the whole target base.

In section B.4 of Appendix B the gains tables for customers and prospects are shown separately. Looking at both sides of the market may indicate more possibilities for cross selling (amongst customers) or up-selling (targeting prospective new customers from the prospects).

5.5 Artificial Neural Networks (ANN)

5.5.1 Input data

Input variables and clusters from the two most feasible *k*-means runs, KMC1 and KMC3, were used as input for training and testing the ANN model. The names of each test run group and the variables used are indicated in section A.2.4 of Appendix A.

The training and test runs were performed for the following two groups:

- ANN1: Feature-based, including company firmographics of prospects
- ANN2: Value-based, using expenditure and sales numbers of customers

5.5.2 Data transformations

The blanks in the training datasets for group ANN2 were not replaced with zeros or averages as the model is not a cluster model, and training was done on original data, as far as possible. A few of the batches of training data were normalised to ascertain whether transformation of the data would result in changes in network performance or different outputs. This was done by fitting the data to the range [0,…,1], after which a z-transformation was applied.

5.5.3 Training, validation and application

Some model parameters for batches of training and validation datasets are shown in Table 5.14. Note that the analysis was performed purely for classification and not for pattern recognition; therefore, smaller batch sizes were needed.

Table 5.14: ANN batch training and validation sets

Training/Validation Batch	Transformation	Example Records	Output Variables	Output Type
Training Set 1.1a	Normalised	889	1	Numerical
Training Set 1.1b	Normalised	889	4	Binary
Training Set 1.1c	None	889	1	Numerical
Training Set 1.1d	None	889	4	Binary
Training Set 1.2	None	889	1	Numerical
Validation Set 1.2a	None	5	1	Numerical
Validation Set 1.2b	None	10	1	Numerical
Validation Set 1.2c	None	40	1	Numerical
Validation Set 1.2d	None	206	1	Numerical
Validation Set 1.2e	None	405	1	Numerical
Training Set 1.3	None	889	1	Numerical
Validation Set 1.3a	None	7	1	Numerical
Validation Set 1.3b	None	15	1	Numerical
Validation Set 1.3c	None	30	1	Numerical
Validation Set 1.3d	None	105	1	Numerical
Validation Set 1.3e	None	302	1	Numerical
Training Set 1.4a	Normalised	500	1	Numerical
Training Set 1.4b	None	500	1	Numerical
Training Set 2.1	Normalised	300	1	Numerical
Training Set 2.2	None	300	1	Numerical
Training Set 2.3	None	300	1	Numerical

In a broader sense, ANN analysis is carried out in three phases (Shankar, 2015):

(1) Training phase: The learning process for fitting the best model

(2) Validation or Test phase: The performance (validation) and accuracy (testing) of the selected model are calculated

(3) Application phase: The final model is applied to obtain the outcome (classes, patterns, predictions).

In the analysis of this ANN model, the phases were slightly different, as the batch analyses did not always have the same purpose. The main phases of the ANN analysis in this study are, therefore:

(1) Testing: The initial batches (training sets 1.1a – 1.1e) were used to perform tests on the type of input and output to be used in further batch runs. These batches were taken from test run group ANN1.

(2) Training: All batched training datasets were trained through a number of learning cycles. After the initial testing, the training was performed on the chosen input (no transformation/normalised) and output (one variable with numerical values).

(3) Validation: After initial batches were run, training was performed on two different random batches (training datasets 1.2 and 1.3), followed by validation batches with varying numbers of examples for each of these batches.

(4) Application: After training, and validation in the case of training datasets 1.2 and 1.3, the analysis dataset was run through the optimal network chosen in each batch run to produce class values as output each time. For ANN2 application runs, the analysis dataset was used, excluding the training batch dataset every time.

(5) Verification: After training, randomly sampled batch training sets in each group (ANN1 and ANN2) were run on normalised training data and original data (not transformed). For ANN1, training batches 1.4a and 1.4b, and for ANN2, training batches 2.1, 2.2 and 2.3 were used for training. The whole analysis data was then processed as verification. Note that ANN2 applies to customers only; therefore, the target market dataset used had many blank values. No validation was, therefore, performed for test run group ANN2.

5.5.4 Hyper-parameter values

The default values and control settings below were used by the JustNN software (Wolstenholme, 2015) as hyper-parameter values in the analysis.

a. Growth rate

The growth rate was set at ten cycles or every five seconds.

b. Hidden layers

As a rule, not more than one hidden layer was specified for each batch training iteration, with not more than three neurons (or nodes) in the hidden layer. Two of the batches (training sets 1.1d and 1.1e) were used

to demonstrate the outcome of more than one hidden layers with one to five neurons per layer.

c. Learning rate

After a few test runs it was found that a default learning rate or step size of between 0.2 and 0.6 would result in an optimum solution more rapidly, without increased training error.

d. Momentum term

Although momentum is independent of learning rate, a low learning rate is assisted by a higher momentum and vice versa. For the average learning rate of 0.6, the momentum value of 0.8 was considered appropriate. For batch training runs with more hidden layers (see paragraph b.), smaller momentum values, in the range of 0.4 were used.

e. Validation cycles

The number of cycles (epochs) before the first validation cycle was set at 100, and the cycles per validation were also set at 100.

f. Validation examples

No random examples from the training dataset were selected, as separate validation datasets were used. Rather, various validation examples for ANN1 were used during batch training sets 1.2 and 1.3 (refer to Table 5.14).

g. Target error stops

Learning is set to stop when the average error of all cycles is below 0.01, as opposed to all errors being below a certain limit.

h. Validating stops

The validation cycles could be set to stop when all of the validating examples were within 10% of the desired outputs. In this case, the validation stop was made stricter by specifying a stop when the validating examples were correct after rounding.

i. Fixed period stops

Additional stopping criteria were set, in terms of the number of cycles reached, instead of time (number of seconds reached). Even though convergence to an optimal solution took place long before 1000 cycles, in some cases, the number of cycles where learning is stopped was set at 3000. This was done to illustrate that no oscillation or vanishing gradient had occurred.

5.5.5 ANN classification tests

Training batches, validation tests and final application, called queries in the JustNN software (Wolstenholme, 2016), were run on ANN1 for feature-based classification, and on ANN2 for value-based classification.

The two groups were analysed using the hyper-parameters specified in the previous section. Note that the initial values for the weights and bias terms for the activation functions were small random values to put the first output of each neuron at around 0.5. After each training batch run the network diagram and learning curve were plotted. These diagrams and graphs are shown in section B.5 of Appendix B per batch run. Based on final classification results, scatter plots of the application runs were created.

a. Feature-based classification (test ANN1)

Training was conducted on examples sampled from the feature-based *k*-means test datasets (KMC1). Different batches with 889 examples each were trained on neural networks. If the average training error across all cycles or epochs within an iteration was below a limit of 0.01, it was deemed enough for training to be completed. Even if the average training error was below the limit after a few cycles, the maximum error could be higher, or spike in cycles that followed. The maximum cycles were, therefore, kept at a constant 3000, to pick up any cases of spiking training errors, oscillation or vanishing gradient indicators. Initially one hidden layer and learning rate of 0.6 and momentum of 0.8 were used. These were adjusted as different control settings were tested.

i. Learning rate, transformation and output neurons

First, transformed dataset examples were used (training batches 1.1a and 1.1b). Subsequently, learning examples with standard data (without transformation) were used for training batches 1.1c and 1.1d. The results proved similar and the number of hidden layers were, therefore, adjusted to measure any improvements. Various control settings were tested until a learning rate of 0.4 and momentum of 0.2 were shown to result in the lowest training error. Further training was needed for batch 1.1d, with weights randomly reseeded and initialised again to values between -0.5 and +0.5.

The ANN training was conducted for one output and four outputs, signifying the classes or segments (the same number of classes as the feasible clusters in the KMC method; see subsection 5.2.3). The results in the four outputs were binary, showing either a 1 or a 0, depending on which class had been activated as output. For one output, discrete numerical values between 1 and 4 were used to identify the classes. Network diagrams and learning curves are shown in B.5.1 of Appendix B. For each batch, iteration scatter plots of the application runs were created. These are shown below for comparison.

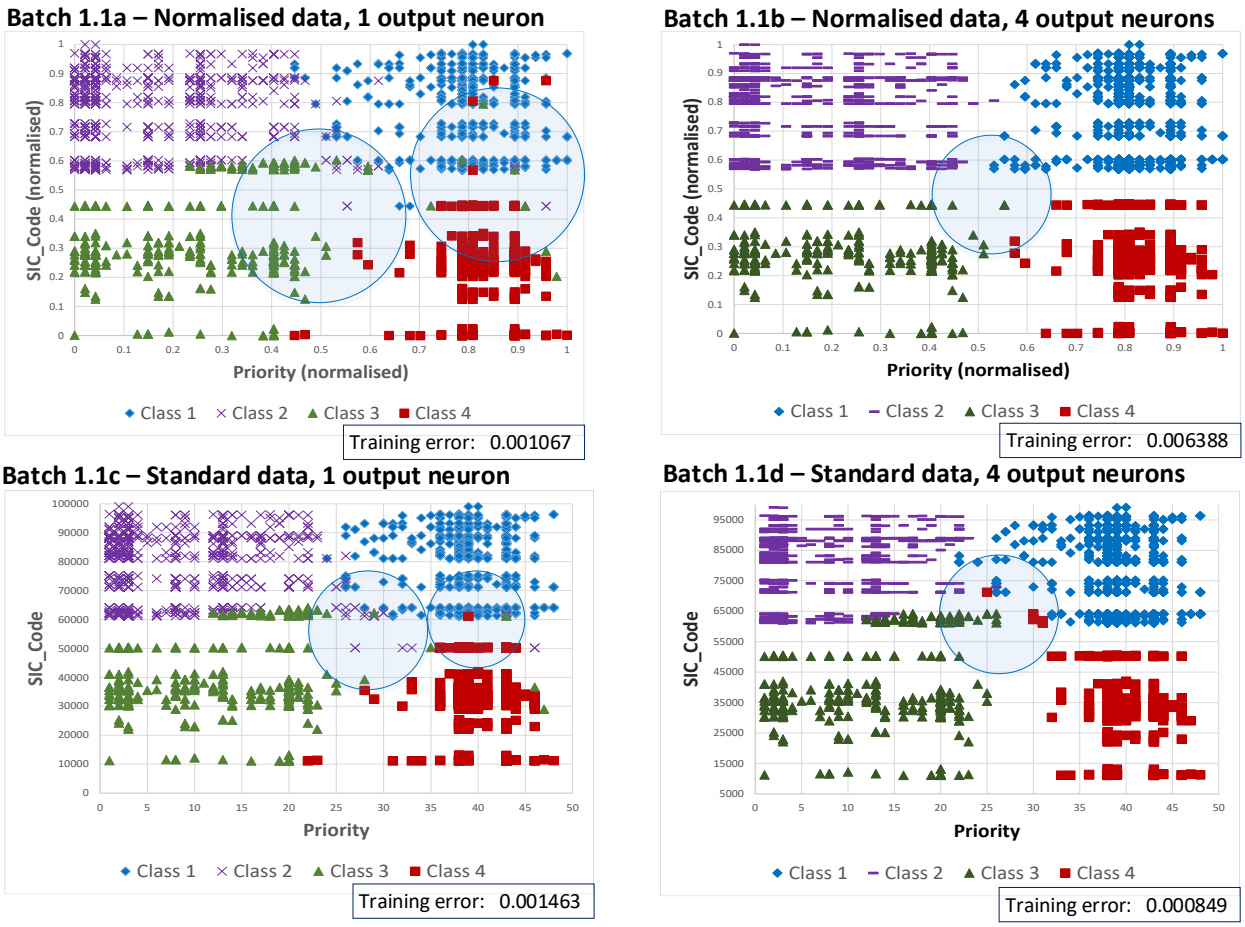


Figure 5.28: ANN1 batches 1.1a–1.1d Location priority and SIC codes

Note that there is less overlap of classes for four output neurons than for one output neuron (indicated with circles in Figure 5.28). The ANN networks using one output neuron resulted in classes with continuous numerical values, for batches 1.1a and 1.1c. These values were rounded off to discrete integer values between 1 and 4. This may explain the bigger overlap of classes. The results above confirm the thinking of current AI practitioners, cited.

Further test runs were performed on ANNs, using one neuron in the output layer and the sigmoid activation function. The motivation was to find an easier network implementation, and to verify certain statements about the number of outputs and the activation function to use. The aim was to establish at which point this configuration could present a result as good as the alternative.

ii. Training and validation test batches

After testing combinations of hidden layers, the number of neurons, learning rates, momentum and weight initialisations, batches 1.2 and 1.3 were run as validation. Different numbers of validation examples were used during training. Additional iterations were run for validation, but as part of training batch runs with the same control settings as in the training runs. The actual output for validation runs was taken from one of the most feasible *k*-means runs, KMC1 (refer to subsection 5.2.6). As a control measure, validation examples with random, incorrect output values were also used. The learning curve plots for the batch runs with their different validation results are shown in subsection B.5.2 of Appendix B. The validation results are summarised in Table 5.15 for clarity.

Table 5.15: ANN actual and random output validation

Training Batch 1.2 Validation results							
Actual Output				Random Output			
Validation examples	Examples correct	Validation result	Validation error	Validation examples	Examples correct	Validation result	Validation error
5	5	100%	0.0062	5	1	20%	0.2810
10	10	100%	0.0066	10	3	30%	0.1685
40	35	88%	0.0140	40	13	33%	0.2372
206	197	96%	0.0082	206	47	23%	0.3212
425	407	96%	0.0078	425	110	26%	0.3074
Training Batch 1.3 Validation results							
Actual Output				Random Output			
Validation examples	Examples correct	Validation result	Validation error	Validation examples	Examples correct	Validation result	Validation error
7	7	100%	0.0114	7	2	29%	0.1382
15	14	93%	0.0083	15	3	20%	0.2547
30	29	97%	0.0076	30	7	23%	0.3061
105	99	94%	0.0193	105	28	27%	0.2552
302	290	96%	0.0091	302	91	30%	0.3074

It may be concluded that the actual output values and distribution differ significantly from the random outputs for both batch runs. This is confirmed by the following:

- T-test probability results on Validation Error results are 0.00038 for training batch 1.2 and 0.00070 for training batch 1.3. this is far below the standard significance level indicating different means.
- F-test values on 'Examples correct' of 0.025 for training batch 1.2 and 0.045 for training batch 1.3 are below standard significance level and, therefore, variance is slightly different.

For each training batch, scatter plots of the solutions were created. The whole dataset, including training examples, was used to find the output class values. In order to have a discrete number for each class (1 to 4), the output values had to be rounded off to integers. The plots for the most important input variables for classes 1 to 4 are demonstrated below.

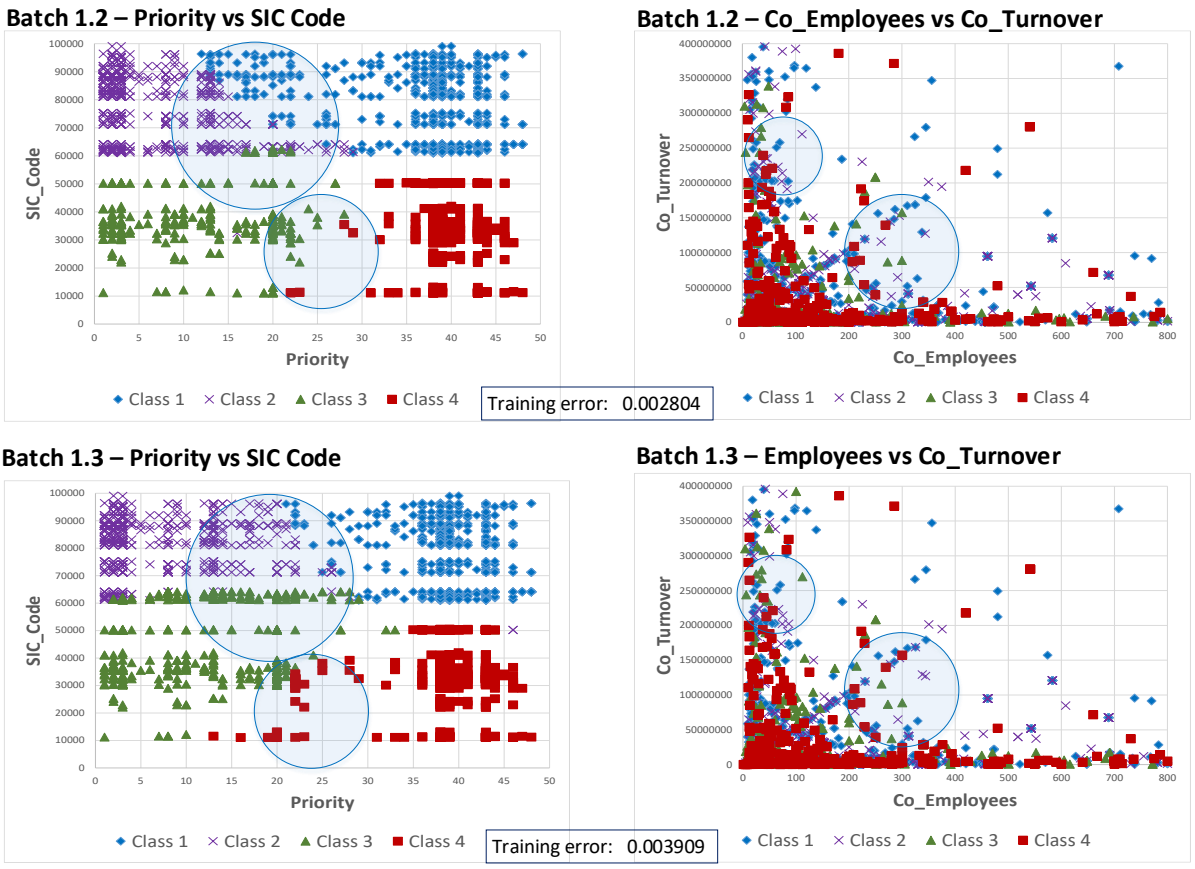


Figure 5.29: ANN1 batches 1.2 and 1.3 comparison plots

In order to clarify test results, the output classes and plots were used to evaluate the following hypotheses based on statements in the previous section:

- Using a single neuron with a sigmoid activation function, results in saturation of values closest to the lowest and the highest output value, (Aldridge, 2020)
- The inputs for the classification have multiple labels and the output classes are not mutually exclusive. Therefore, the sigmoid activation function is applicable (Media, 2018).

An overlap of classes is evident from the circled areas in Figure 5.29. This is more apparent in the Priority vs. SIC code plots. In batch 1.2, the lowest value (class 1) is more prevalent and overlaps with class 2.

The highest value output (class 4) is shown more predominantly for batch 1.3. This indicates that the first hypothesis above seems to hold true. While input variables have continuous values (Company turnover) and discrete values (Priority and SIC code), it can be accepted that input values have multiple labels, for example, different SIC codes. From the plots on the right of Figure 5.29, it is evident that output classes could not be proven to be mutually exclusive. To find the best configuration, while staying with the current topology, further batches were trained using normalised and standard values.

iii. Final application with transformed and standard data

As a final evaluation, two iterations were done on different, random sample batches of 500 examples each. The first iteration was on transformed data (batch 1.4a), and the last without any transformation (batch 1.4b). The training error for the transformed data converged to the maximum limit of 0.01, while standard training data resulted in a training error that is fairly low, until it spikes temporarily during the last few cycles. These training error plots are shown with the networks in B.5.1 of Appendix B. The application runs were performed on target data without including the training batch data. The scatter plots below, of the solutions, show a few small differences between transformed (normalised) and standard training data.

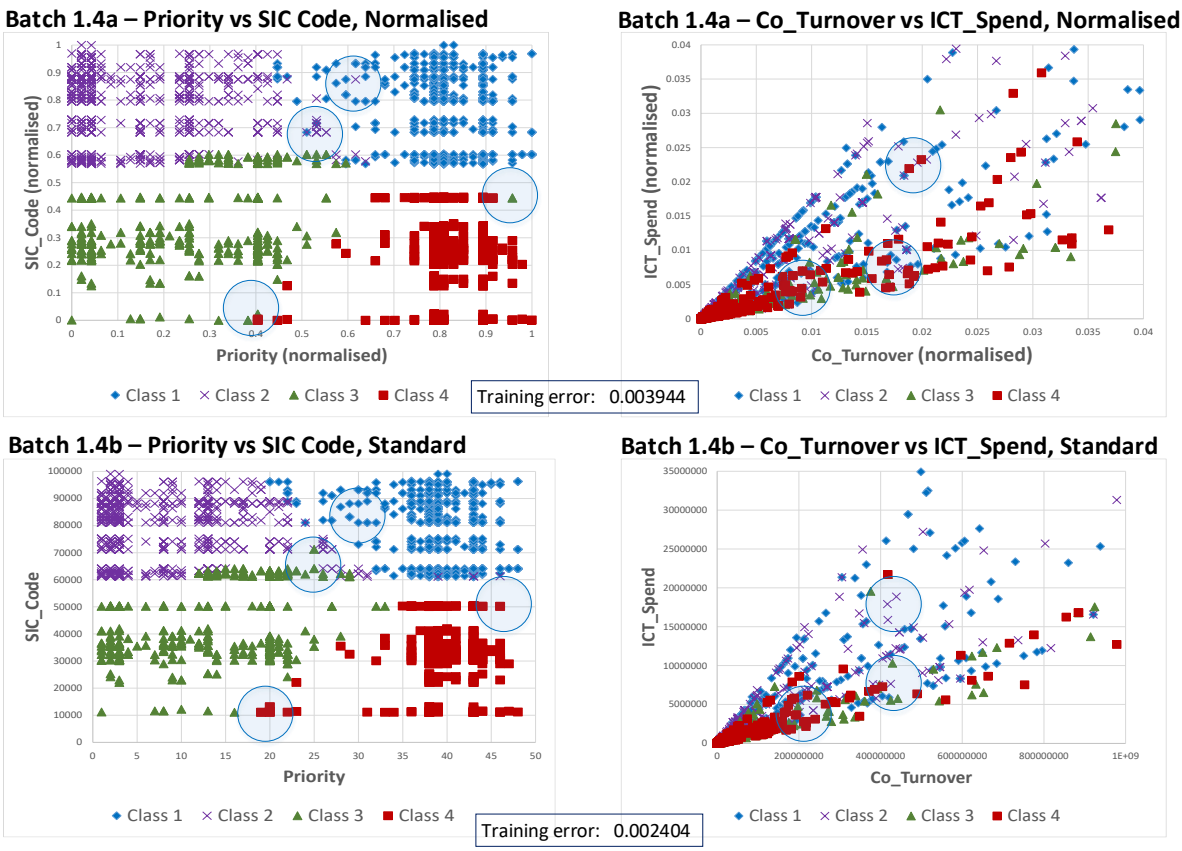


Figure 5.30: ANN1 batch 1.4 normalised vs standard plots

It is evident from the encircled points that the overlap of classes is greater than with the previous normalised/standard training conducted on more examples (training batches 1.1a to 1.1d). This could prove that classes are not mutually exclusive, and the sigmoid activation function is appropriate, as long as the training dataset is big enough. To conclusively find practical solutions, the final test runs were performed on different training data, focusing on customer value metrics.

b. Value-based classification (test ANN2)

Value metrics for existing customers was used as input, with two market-related parameters from the previous runs still included (Location Priority SIC value). The target dataset contained 828 records for customers, from a total of 3362 records. The training batches contained 300 random examples every time. These random examples were chosen from records containing

customer data only. This example represents 36% of the customer data (9% of the total data).

Training set 2.1 (the first batch-training dataset) used normalised values for the learning cycles. The two subsequent batches (training sets 2.2 and 2.3) used standard data for training. Each training iteration used one hidden layer and one output neuron with numerical values as output, representing the four classes. There was an initial learning rate of 0.6, and momentum of 0.8 for each training batch iteration. To improve on the training error, one additional iteration was run for each batch-training set (normalised and standard training data). After the first iteration, the weights were initialised again (without seeding) to values between -0.5 and +0.5. The number of neurons in the hidden layer was increased in the additional iteration for each training run. This proved to lower the average training error slightly. The maximum cycles were kept at a constant 3000 to pick up any anomalies, after the feasible training error level was reached. After the final iteration, scatter plots were created on the output, first with the training data included, then without the training data. Differences between the outputs of each training batch were found to be negligible.

Plots for the market-related variables (Priority and SIC codes) are shown below for comparison, with different pairwise plots of some value-based variables (Telecom revenue, Telecom subscribers and Telecom ARPU).

The plots were developed from the solution outputs of the second iteration for each training batch. Note where the circles depict small differences. The final iteration performed the best with a marginally smaller overlap of classes. Referring to B.5.3 of Appendix B, the hidden layer nodes were increased from 3 to 6 for the second iteration. This increase proved that a larger hidden layer can present fewer training errors, within limits.

Figure 5.31: ANN2 batches 2.1–2.3 plots after two iterations

Pairwise plots of Priority, SIC codes, Telecom ARPU, Telecom revenue, Telecom subscribers, Telecom Device Lines, Telecom product line count, and Telecom solution line count were developed. The plots for these other customer metrics are shown below.

Figure 5.32: ANN2 batches 2.1–2.3 plots of other customer variables

c. Processing time

Processing time was measured for each of the training batches until the maximum of 3000 cycles was reached. As training was conducted rapidly, processing time is shown in terms of seconds.

Below is a plot for these run times in seconds, in the format *ss:tt*. The number of hidden layers and the node count per layer are also shown against each time. Note that the darker-shaded bars for input nodes indicate where input-training data was normalised.

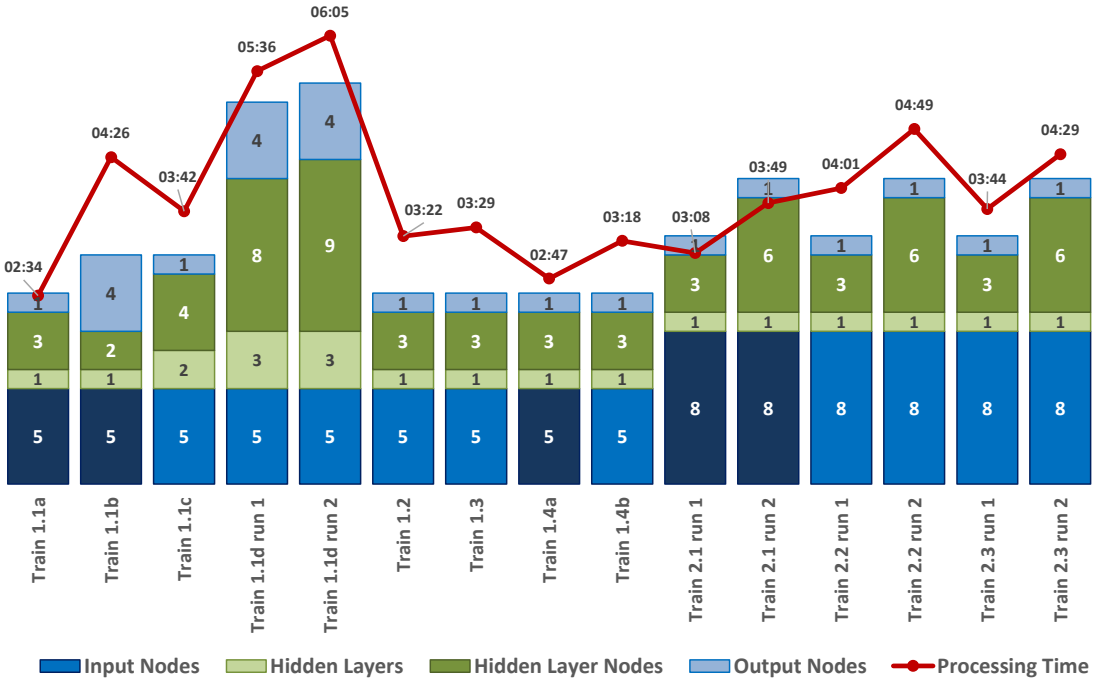


Figure 5.33: ANN node count and processing time per training batch

The number of hidden layers, hidden layer nodes and outputs seems to have the greatest influence on processing time. Although the number of input variables or nodes contributes to the processing time, no major difference was found in the processing time for different sample sizes (300 to 889 examples). The influence of the number of examples is apparent in larger datasets, 10 times or more the size of the target dataset used here.

Another measure of time that could be used indirectly is the time when an optimal solution was reached. This could not be measured as time, but as the

number of cycles when a solution was reached. Through inspection of the learning curve, the number of cycles is noted at the point where the graph dips below the 0.01 error limit. Refer to the figures in section B.5 in Appendix B for plots of the minimum, average and maximum training errors. The number of cycles, now called solution cycles, is shown below together with the number of hidden layers and the node count per layer (darker bars for input nodes indicate normalised data).

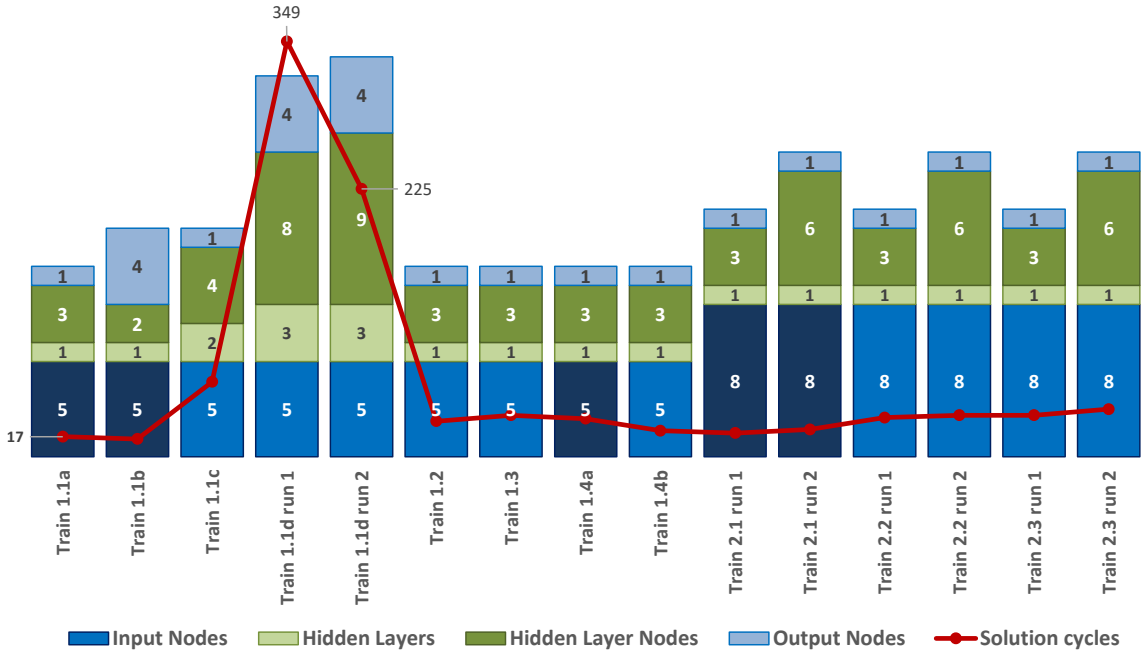


Figure 5.34: ANN node count and solution cycles per training batch

The influence of hidden layers and the number of nodes is now more pronounced. This could indicate that the number of hidden layers and nodes does have an effect on the number of cycles needed to reach an optimal solution. The deeper the network, the longer it takes to reach the solution, due to fewer cycles. One hidden layer with fewer nodes, not more than the input nodes, would be sufficient to reach a solution in a short time, with fewer cycles to run through.

5.5.6 Final ANN solution

Besides evaluating the usefulness of the single output network, with sigmoid activation function, the following was confirmed by means of test runs on the two groups of ANN training batches:

- Output classes are not necessarily mutually exclusive.
- The sigmoid function saturates values close to class values 1 and 4.

Taking all the results into account, the most feasible ANN network topology could be identified, as shown per training batch. A measure of reliability of the network is the training error, also called the apparent error, as this is an estimate with bias of the true error, as described in subsection 4.3.2d for cross-validation.

A comparison of the training errors per network layer, and number of nodes, is depicted below for each batch run, showing normalised training batches as darker input node bars.

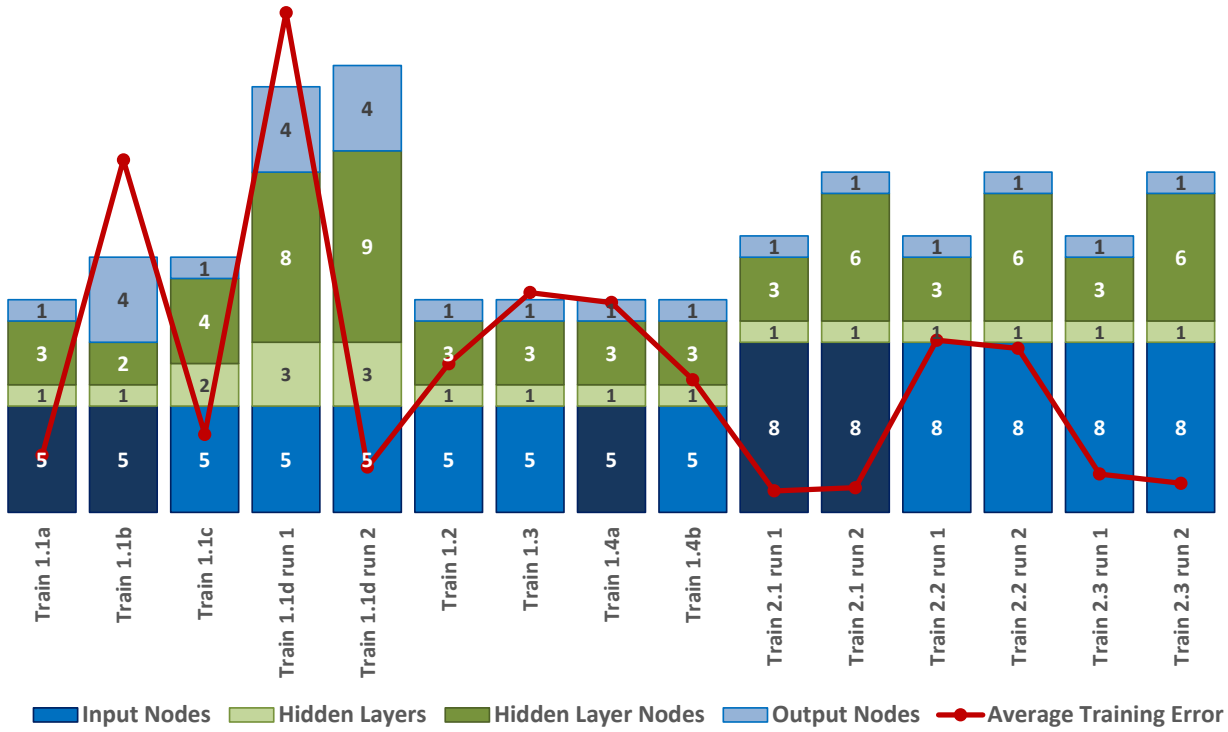


Figure 5.35: ANN training error for different nodes and layer count

Combining the processing time stipulated in Figure 5.33, and the training error results in Figure 5.35, the following training batch networks were identified as the most feasible, or as having the quickest processes and lowest number of training errors.

Table 5.16: ANN most feasible training batch networks

Test Group	Training Batch	Transformation	Examples	Input Nodes	Hidden Layers	Hidden Layer Nodes	Output Nodes	Solution Cycles	Average Training Error	Processing Time
ANN1	Train 1.1a	Normalised	889	5	1	3	1	17	0.001073	02:34
	Train 1.1c	None	889	5	2	4	1	63	0.001465	03:42
ANN2	Train 2.1 run 1	Normalised	300	8	1	3	1	20	0.000403	03:08
	Train 2.3 run 1	None	300	8	1	3	1	35	0.000719	03:44

In both groups, ANN1 for target market features and ANN2 for customer values, the normalised training data gave the best results. Hidden layers can be increased to more than one for lower training errors. However, based on the results, it would not be necessary to have more than two hidden layers. For some of the training batches, weights were reset and second training runs performed. This proved to decrease the training error considerably, in some cases. However, the lower training error has a cost, in relation to processing time and training cycles. Table 5.17 below is a representation of the results of the retraining runs.

Table 5.17: ANN batch retraining results

Test Group	Training Batch	Transformation	Reset Actions	Hidden Layers	Hidden Layer Nodes	Output Nodes	Solution Cycles	Average Training Error	Processing Time
ANN1	Train 1.1d run 2	None	Reseed & re-initialise weights	3	9	4	225	0.000849	06:05
ANN2	Train 2.1 run 2	Normalised	Re-Initialise weights	1	6	1	23	0.000462	03:49
	Train 2.2 run 2	None	Re-Initialise weights	1	6	1	35	0.003083	04:49
	Train 2.3 run 2	None	Re-Initialise weights	1	6	1	40	0.000544	04:29

Having one hidden layer and one output node may, therefore, be enough to create a network with a reasonably low training error and processing time. This was demonstrated in the first test group with target market data. Below

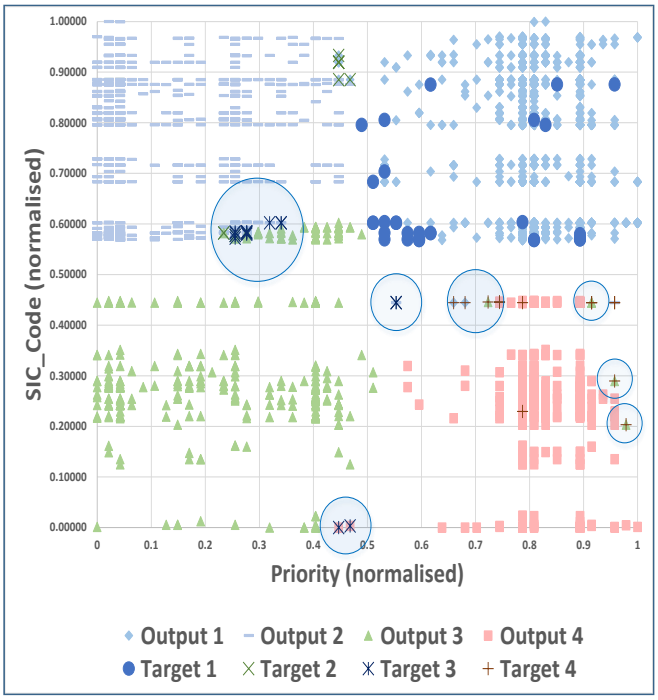
are the training batches for two different sample sizes (number of examples) with results.

Table 5.18: ANN batch results for one hidden layer and one output

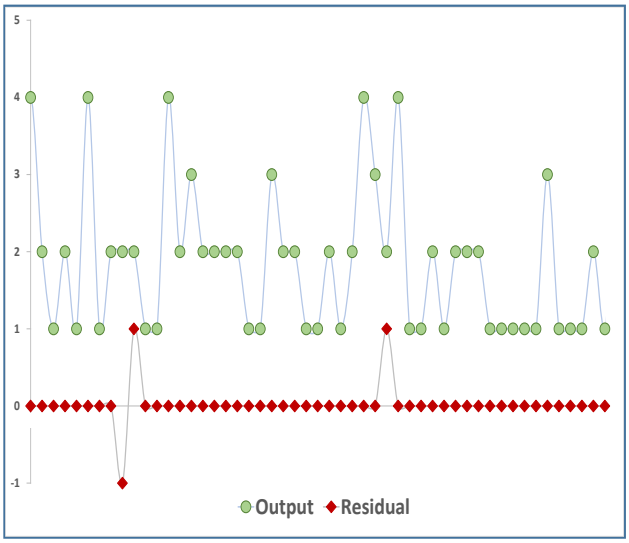
Test Group	Training Batch	Transformation	Examples	Input Nodes	Hidden Layers	Hidden Layer Nodes	Output Nodes	Solution Cycles	Average Training Error	Processing Time
ANN1	**Train 1.2**	None	889	5	1	3	1	30	0.002793	03:22
	Train 1.3	None	889	5	1	3	1	35	0.004131	03:29
	Train 1.4a	Normalised	500	5	1	3	1	32	0.003944	02:47
	Train 1.4b	None	500	5	1	3	1	22	0.002492	03:18

Validation runs were performed on the first two training batches shown in Table 5.18, proving results with low generalisation errors. This indicates that the neural networks for training batches 1.2 and 1.3 work well on data not used in training, or real world data.

The true error was, however, not evaluated and, therefore, bias could still be present, as derived from equation (4.14) of subsection 4.3.2d. Bias is a systematic distribution of the residuals for patterns (Twomey and Smith, 1995), or examples, in this case. Taking the training error as the apparent error, residuals were calculated according to equation (4.16). Here, the target values were taken from the results of the two most feasible *k*-means runs, KMC1 and KMC3, as determined in section 5.2. The residual can be visually represented by output vs. target plots, as well as output vs. residuals. Below is an example using training batch 1.1a output and target in a plot of normalised input parameters Priority and SIC code. In addition, a portion of the output vs. residuals is shown as illustration.



Priority and SIC_Code with Output vs Target plot



Output vs Residual scatter plot

Figure 5.36: ANN batch 1.1a output vs target and residual plots

Only the targets that differed from the output values are plotted in the left graph of Figure 5.36, in order to depict the differences more clearly.

Some of the target values could align with other output values. The cases where the output class falls in a different target class are circled. In the residual plot, the majority of residuals are in a straight line around zero, indicating a low presence of bias.

Without having to plot each batch to compare residuals, it is sufficient to look at the error metrics per training batch iteration. For this, equation (4.15) for the MSE was used to calculate the residual error. Residual errors were below the standard acceptable level of 0.1 for ANN1 training, but for ANN2 training, residual errors were slightly more than this level. Below is a summary of the residual error, compared to the training error for each batch.

Training Batch	Transformation	Examples	Average Training Error	Residual Error
1.1a	Normalised	889	0.0011	0.0434
1.1b	Normalised	889	0.0066	0.0345
1.1c	None	889	0.0015	0.0229
1.1d	None	889	0.0008	0.0550
1.2	None	889	0.0028	0.0547
1.3	None	889	0.0041	0.0521
1.4a	Normalised	500	0.0039	0.0256
1.4b	None	500	0.0025	0.0318
2.1	Normalised	300	0.0004	0.1175
2.1*	Normalised	300	0.0005	0.1290
2.2	None	300	0.0032	0.1157
2.2*	None	300	0.0031	0.1257
2.3	None	300	0.0007	0.1014
2.3*	None	300	0.0005	0.1110

∗ Training batch excluded from the application run

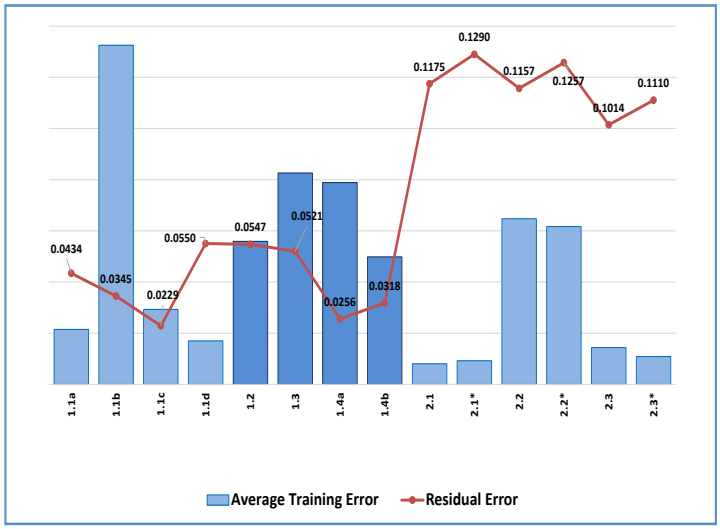


Figure 5.37: ANN residual error vs training error per batch run

The higher residual error values for training batches in group ANN2 could be attributed to a few factors. One contributing factor could be that the number of examples was lower than in the previous training iterations. Another factor is that the training data used was sparser. Input nodes were related to customer values, which constituted only a portion of the total target market data. Even after a second training run or iteration, the residual errors did not improve. The training errors improved slightly, while staying much lower than the training error limit of 0.01, indicating that the learning had reached saturation point. This difference in training results, compared to actual results, indicated by the residual error, shows that bias is still largely present.

The shaded areas in Figure 5.37 show where a reasonable good fit of the network model may be expected. The residual error is fairly in line with the training error for these batches, indicating a low bias. When looking for a practical implementation of a neural network, this topology of one hidden layer and one output layer, coupled with a learning rate of 0.6 and momentum of 0.8, could be used.

5.6 Discussion

The comparative analysis of the test runs in this chapter shows how promising the algorithms may be as potential segmentation methods. The test runs of each quantitative method are compared to ascertain which of them yield the most achievable results. The point of departure is the evaluation criteria defined in section 4.3.

Depending on the segmentation method, the performance measures used for the comparison of test runs evaluated the cluster tightness, swarm density, correctness of structure and a model's generalisation capability. These performance measures were translated into a score and used to measure the quality of the method. For KMC and PSO methods, the silhouette and fitness metrics were used directly. For CHAID, the last probability that values were split by chance was used. A well-fitted diagram will end with low p-values. Therefore, this value was subtracted from 1 to obtain a score. In the ANN method, combined values for the residual and training errors were subtracted from 1.0 to obtain a score.

In addition, the number of iterations was measured. The term 'iteration' is used collectively, although each method does not have the same connotation. In the KMC and PSO methods, the term is used for one set of calculations through all the input records, called objects (KMC) or particles (PSO). For CHAID, one iteration is the pairwise comparison through all predictor variables, to create a node. For ANN, the term should not be confused with the definition for 'iteration' used in neural networks, as it has two totally different meanings. The iterations referred to here for a performance measure, correspond to a 'cycle' in ANN, which is one set of network computations on all records in a training batch.

The time taken for a test run, until a feasible solution was found, was also measured. In the case of CHAID tests and some of the ANN training runs, this processing time was estimated (see subsections 5.4.4d and 5.5.4d). Together with the number of iterations, the processing time values indirectly

evaluate the algorithm used in processing a specific method. This evaluation is represented as a processing score. This score is the average of the processing time score and the iteration score. In this way, the combined effect of these two measures can be evaluated.

In this chapter, the qualitative evaluation of suitability of a method was not measured as an integral part of the testing. These fit-for-purpose measurements rely on a collection of hypotheses, which include the answering of research questions. Current knowledge of the enterprise business environment in a major South African telecommunications company was used to evaluate hypotheses and answer research questions. These results were given a percentage, transformed into a scale from 0 to 1, to align with the measurement scale for quality mentioned here. Note that percentage values were based on the potential value of the analysis methods, not the actual value, as the segmentation methods in the research have not been implemented in a telecom business environment. These values were given for segment hypotheses, research hypotheses and research questions, and transformed to a hypothesis score.

In Figure 5.38, the results per test run are shown in sequence from best to worst for the hypotheses, quality and processing scores.

Note the processing time and iteration scores at the bottom are shown separately, even though they are collectively referred to as process scores.

Figure 5.38: Test run hypotheses, quality and processing scores

According to Figure 5.38, the KMC test run scores are safe, stable and middle-of-the-range fit for purpose, through the hypotheses and quality output. The processing time and number of iterations tend to be a limitation for KMC tests.

The PSO test run scores indicate the least ability to be implemented in practice, and processing time is considerably longer. Conversely, PSO test runs demonstrate best quality of output for the least number of iterations.

The CHAID test runs have most of the highest scores. Quality of output and scores for the number of iterations are a bit less. The results here depend largely on the type of data, more so in CHAID's case. The final number of segments cannot be predetermined, even though categories for all variables are defined beforehand. This puts CHAID in a slightly different category than the other methods, where there is classification or clustering into a fixed number of classes.

The ANN test runs show the best scores for quality and practical implementation, specifically for feature-based data (ANN1). The next best result for both feature and value-based ANN test runs is for processing time. When CHAID test runs are excluded, only methods for segmentation into a specified number of classes remain. Of these, feature-based tests ANN1 and KMC1 are the most practical to implement, according to the hypotheses scores. The quality of output shows best on feature-based data with ANN1 and PSO1 test runs. Feature and value-based tests for ANN and behaviour tests on spending, KMC1, perform the best on processing time when not compared to CHAID tests. Both feature and value-based tests for PSO resulted in highest quality output, regardless.

To characterise each test run better, the evaluation criteria are best measured together. By evaluating combinations of test criteria, the capability of a method is exposed more concisely. The combined score of two or three evaluation criteria was calculated by taking the average of each criteria score. A depiction of each test run in sequence of importance is shown below for each combined evaluation.

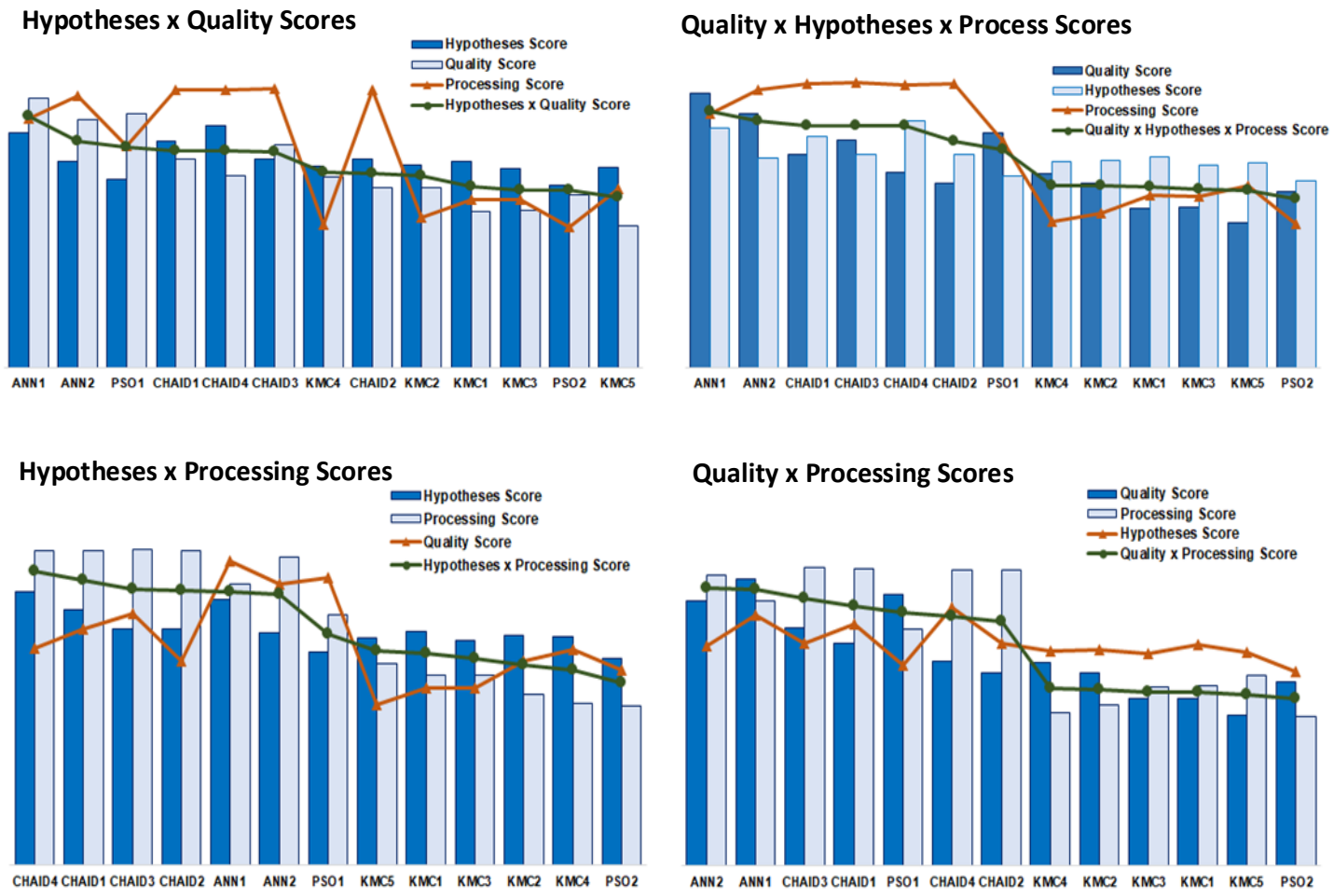


Figure 5.39: Test run combined scores

In Figure 5.39, the KMC4 value-based test runs, including customer data, show reasonable results for practicality and quality of output. However, feature-based PSO1 test runs are more promising when only the practicality (hypotheses score) and quality of test runs are taken into account. Looking at the combination of quality of output, hypotheses in practice and processing of the algorithm, both ANN test runs appear the best. For quality and processing, the ANN test runs were performing the best, followed by CHAID3 tests on customers per industry and CHAID1 tests on customers and prospects per region. However, regarding the hypotheses and processing scores alone, ANN lags behind all the CHAID tests, which might be an unfair comparison, due to the unique nature of CHAID. When leaving CHAID out of the comparison, both ANN test runs on feature-based and customer value-

based data perform best overall. This is followed by PSO1 feature-based tests for all combined criteria. Interesting to note is that KMC4 value-testing performed on customer data would be the third choice on nearly all the combined evaluation criteria. Where practicality and processing are considered, KMC5 tests on firmographic and customer value data are an alternative.

On a basic level, the test runs on the target data were evaluated and ranked according to evaluation criteria in this chapter. These results may be beneficial for tactical purposes where specific types of enterprise market data are classified.

The objectives of this study were, however, on a more strategic level. In a more general sense, quantitative methods used during the test runs were assessed for their suitability to perform B2B target market segmentation.

Final conclusions regarding the analysis of each quantitative segmentation method are explained in the next chapter.

6. CONCLUSIONS

6.1 Recommending a Quantitative Segmentation Method

The problem statement refers to two MVA categories, namely interdependence and dependence analysis. All the variables in the test runs were chosen with the above in mind, and the methods were discussed within each of these categories.

Deciding on only one best analysis method is probably not the best route to take, depending on the questions to consider and the objectives of each analytic approach. Each method does, however, have merit, as long as the objective is met. The best all-round approach may well be to use a clustering method for desired outcomes on interdependent data and a classification method where the outcome requires dependence analysis.

The evaluation criteria defined in section 4.3 were used again for comparing the quantitative segmentation methods. In this chapter, the results of the evaluation criteria as applied to the test runs are summarised by taking the average of the test run scores for each segmentation method.

A gains table as described in section 5.4.5 of the previous chapter was used to compare the evaluation criteria scores and rank segmentation methods. This proved very useful to ascertain the levels at which the evaluation criteria are met, using an index that is calculated per criteria score. Detailed scores and indices used in comparing the methods are given in section B.6 of Appendix B. A summary of the combined scores used for evaluation are shown in Figure 6.1. Note that the combined scores for quality x hypotheses, quality x processing and hypotheses x processing are included in the combined graph at the top right for all the criteria. Each criteria score per method is shown separately in the graph at the top left. At the bottom, the hypotheses and quality scores are shown against the processing scores for comparison.

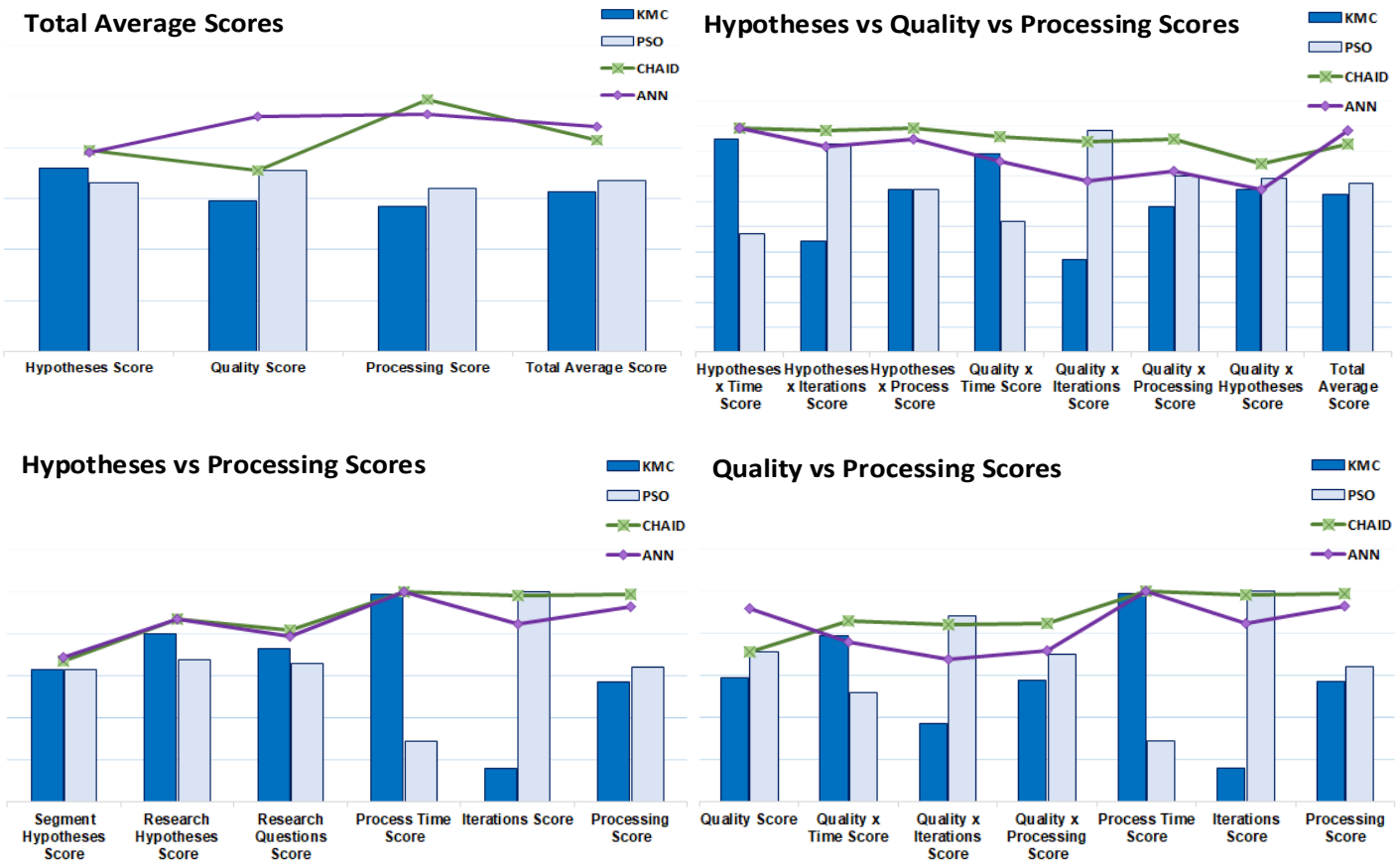


Figure 6.1: Analysis method criteria scores

A summary of the most suitable methods per criteria, according to Figure 6.1, follows.

Fit for purpose (hypotheses criteria):

- CHAID (79%)
- ANN (78%)

Quality of output (quality criteria):

- ANN (92%)
- CHAID (71%) or PSO (71%)

Processing (time and iterations criteria):

- CHAID (99%)
- ANN (93%)

Applicability (all criteria combined):

- ANN (88%)
- CHAID (83%)

The criteria score was translated to a percentage to show the degree to which each method meets the evaluation criteria. The CHAID and ANN methods

override all the criteria, due to the very high quality of output and low processing time. As CHAID follows a slightly different process, with manual intervention, it can be accepted that this process is applicable, in general. For a system-driven process, with a predetermined number of segments, the following methods are most suitable for each of the criteria:

Fit for purpose (hypotheses criteria):

- ANN (78%)
- KMC (72%)

Quality of output (quality criteria):

- ANN (92%)
- PSO (71%)

Processing (time and iterations criteria):

- ANN (93%)
- PSO (64%)

Applicability (all criteria combined):

- ANN (88%)
- PSO (67%)

It is interesting to note that at least one of the suppositions made before this study proved to be inconclusive. The opinion in practice was to use KMC, as it has always been a successful method for market research and planning. However, the target in this case is the consumer market, with specific consumer-related answers to market research questions. For a B2B environment, KMC is still a highly appropriate method (72% suitable) based purely on hypotheses. However the method is superseded by other methods in other areas. Should the KMC method still be the preferred choice, it will be high on processing resources, and the quality of output will not be optimal. It is still better than the PSO method, however, which, although high on quality output, remains very high on processing time, with an average of 50 minutes for one test run on the analysis data. With much larger datasets, this could prove a problem, if not scheduled to run after hours, with a limit

to the number of variables. Regarding ANN, residual errors were based on target output values that were generated using KMC solutions. The danger is that the bias may be larger than estimated. If this is a concern, ANN methods will need larger training data, with *k*-folds cross-validation (see subsection 4.3.2d).

As mentioned before, the CHAID method can be implemented as a general classification method, more suitable for high-level investigative classification. For grouping enterprise data into predefined classes, the combined suitability of remaining methods is summarised as:

Fit for purpose and speed (hypotheses x processing time):

- ANN (89%)
- KMC (85%)

Fit for purpose and algorithm integrity (hypotheses x iterations):

- PSO (83%)
- ANN (82%)

Quality output and fit for purpose (quality x hypotheses):

- PSO (69%)
- KMC (65%) or ANN (65%)

Quality and speed of output (quality x processing time):

- KMC (79%)
- ANN (76%)

Quality output and algorithm integrity (quality x iterations):

- PSO (88%)
- ANN (68%)

Here it becomes clearer for what purpose each method is most suitable. The ANN method is suitably fit for purpose at a low processing speed (incidentally, to the same degree as CHAID, at 89% suitability). The PSO method is an average of 80% suitable to be implemented (fit for purpose) due to the high quality output and integrity of the algorithm. KMC is the most suitable (85%) as an unsupervised clustering method, fit for purpose with

speedy execution. Overall, the KMC method is more known (at 79% suitability) for its quality of output, with good processing speed.

The hypotheses and research questions were also individually evaluated for each segmentation method to relate the summarised criteria scores to the business environment in more detail.

Refer to section B.6 of Appendix for a graph with the scoring results of these labels for each method.

The best overall hypotheses, quality and processing features per segmentation method are itemised below.

K-means clustering (KMC) with **82%** suitability**:**

- The method is suitable for implementation with fast processing (85%).
- The method executes at reasonable speed, with average quality (79%)
- Specific hypotheses and research questions resolved:
    - Research H1 (85%) – The method is quantitative, repeatable and able to eliminate manual intervention, from where the results can be used directly or as input for further clustering.
    - Research H2 (79%) – The method is practical enough to be useful in a specific company in the telecommunication industry.
    - Research Q1 (78%) – The results are robust and repeatable.
    - Research Q4 (78%) – Buy-in from sales management is possible.
    - Research Q2 (76%) – The method is clearly understandable by business.
    - Research Q3 (76%) – Segmentation process can be well managed, to ensure proper implementation.

Particle Swarm Optimisation (PSO) with **87%** suitability**:**

- The algorithm is extremely effective, with very few iterations (99%)
- Quality is above average due to effective algorithm (88%)
- The method is implementable, given a low number of iterations (83%)
- Specific hypotheses and research questions resolved:

  - Research H1 (83%) – The method is quantitative and repeatable, and may eliminate manual intervention up to a point, from where the results can be used for further clustering.
  - Research Q4 (83%) – Buy-in from sales management will be very high if the algorithm is explained.

Chi-square Interaction Detection (CHAID) with **86%** suitability**:**

- Highly efficient algorithm with split-second execution (99%)
- Very implementable, satisfying majority of hypotheses (79%)
- Reasonable quality output, combined with an effective algorithm (85%)
- Specific hypotheses and research questions resolved:
  - Research Q2 (90%) – The method is highly understandable by business, and the calculations can be demonstrated manually.
  - Research H3 (89%) – Businesses in the same segment can be properly targeted through the right sales channel.
  - Research H2 (88%) – The method is highly practical and will be very beneficial for specific companies in the telecommunication industry.
  - Research H1 (85%) – The method is quantitative and repeatable with partial, but effective manual intervention, and reliable calculations.
  - Research Q1 (85%) – Based on clear categories for predictors and dependent variables, the results are very robust and repeatable.
  - Research Q6 (84%) – Effective rule formulation is possible because of proper analysis of segment categories.
  - Research Q4 (83%) – Major buy-in and support from sales management is possible, due to the stability of calculations.
  - Segment H1 (79%) – Outputs are unique per segment (dependent variable), e.g. companies with high ICT expenditure will be in a different segment than those with low ICT expenditure.
  - Research Q7 (79%) – Adjustments to business rules can be made fairly easily and future enhancements are easily applied.

- Research Q3 (78%) – The segmentation method process can be well managed, and implementation is reasonably easy.
- Research Q5 (78%) – Segments are set up in categories based on sales metrics to easily translate to the right sales channel.

Artificial Neural Networks (ANN) with **86%** suitability**:**

- Highly efficient algorithm with short execution time (93%)
- Output of very high quality due to proper training beforehand (92%)
- Wide variety of practical application as confirmed by hypotheses (78%)
- Specific hypotheses and research questions resolved:
  - Segment H2 (95%) – Classifications are highly distinct; for example, for an industry parameter, the manufacturing and financial companies will be in different segments.
  - Research H3 (89%) – Businesses in the same segment can be properly targeted through the right sales channel.
  - Research H1 (87%) – The method is quantitative and repeatable, and can be used to entirely eliminate manual intervention in classifying business.
  - Research H2 (85%) – The method is very practical and will be highly beneficial for specific applications in telecom industry businesses.
  - Research Q1 (85%) – With well-defined hyper-parameters, and large enough training data, the results will be very robust and repeatable.
  - Research Q4 (85%) – As the method is very relevant in today's technology, buy-in and support from sales management will be huge.
  - Research Q5 (85%) – Segments from proper training data can be easily translated to the right sales channel.
  - Research Q2 (80%) – The method is known by business, and can be explained on a high level, with results that will draw attention.

The conclusion may be drawn that the segmentation method used is reliant on the objective. For example, to ensure strong hypotheses where answers are needed about the target market, it is advisable to consider a CHAID

segmentation method. However, when taking quality of output into consideration, CHAID has proved that it does not answer the hypotheses successfully as a customer segmentation method with the aim to identify segments for solutions offered to customers. The results of test runs, where CHAID2 was lower than the other CHAID runs for hypotheses and quality combined, were proof of this. Here, the ANN method (specifically ANN2), aimed at customer solutions, would serve better.

6.2 Additional Comments

Throughout the analysis of segmentation methods, more emphasis was placed on the quality of the solution and its practicality, than on running the process. Nevertheless, it was found that the number of iterations and processing time (translated from CPU time) may influence practical implementation on large datasets.

Fortunately, saving on processing time and attaining the highest quality are not always critical requirements for segmentation in a business environment. This is because the optimal solution might then not be practically viable. Considerations of targeting the segments and having sales channels available for follow-up are more important. Statistical rigour might be sacrificed in favour of a feasible outcome by applying less rigid input criteria. This was evident in the test runs for each method in the study. A feasible cluster and weighting were introduced (refer to section 4.4.1d) for KMC. In the case of PSO, the fitness value limit was adapted for test runs using customer data, as there were many gaps in the data. During CHAID analysis, the α_{merge} and α_{split} values were either increased or decreased, depending on the data. Lastly, with ANN training, the learning rate, momentum and number of hidden layers and nodes were adjusted.

Analysis is, however, more important than these criteria levels. No segmentation can be performed without proper analysis of the data. This fact was repeatedly confirmed during the research. Patterns and interactions emerging in the data contributed to the final solution. Ignoring them would

have proved detrimental to the segmentation model. In section 4.1 on segmentation schemes, mention was made of the difference between B2B market segmentation and consumer market segmentation.

The research in this study confirmed more differences in B2B to the consumer market, such as:

- There are usually fewer segments. Having clusters of four segments, or classification into 12 segments has been proven to be in line with the norm for B2B segmentation.
- A small number of large customers contributed to most of the revenue, satisfying the Pareto[13] principle. This is very true of a B2B market.
- Although B2B segmentation appears less complex than consumer market segmentation, the study analysis proves that some complexities do exist. For example, the choice of the type of industry, combined with product, device or solution may result in various combinations of segments.

A further outcome of the analysis is a clear distinction between segmenting prospects and existing customers. The segment analysis regularly points to a dual purpose, e.g. prospects vs. customers, strategic vs. tactical objectives, independent vs. dependent variables and clustering vs. classification. This study shows that the outcome of test runs naturally tends to highlight differences in approach.

The need for segmentation is the only constant in the ever-changing field of marketing, especially in the area of B2B marketing, where statistical, albeit quantitative, segmentation methods are yet to become the norm.

[13] The Pareto principle (also known as the 80/20 rule, the law of the vital few, or the principle of factor sparsity) states that, for many events, roughly 80% of the effects come from 20% of the causes. Management consultant Joseph M. Juran suggested the principle and named it after Italian economist Vilfredo Pareto (Bunkley, 2008), who noted the 80/20 connection while at the University of Lausanne in 1896, as published in his first work, Cours d'économie politique (Flux, 1897), in which Pareto showed that about 80% of the land in Italy was owned by 20% of the population (Wikipedia contributors, 2020a)

Further financial calculations, such as the size of one or more segments multiplied by the average ICT spend, can help to improve planning for communications/marketing budgets. By using these calculations, the segments with the most spending potential can be analysed further to identify those companies that are below average for their segment. Marketing campaigns may be aimed at these companies to promote spending in a more effective way, or to provide the best telecommunication solution for them. Many other decisions may be made with the help of tools used in the analysis. For example, the use of the gains table from CHAID and segmentation rules based on KMC may give guidance on how to conduct market planning among underperforming segments.

These type of decision making has important implications for managerial applications. In a sales environment, decisions on resource allocation is important. Here a technique like CHAID can assist to allocate the appropriate sales staff to B2B customers. Apart from sales, managerial decision making rely on other resource parameters like working hours, type of equipment, office locations, to name a few. Any of the quantitative methods could apply here, where grouping of parameters will assist managers for making decisions.

6.3 Opportunities for Future Research

Regarding current methods used in this research for B2B segmentation, the test run results show that the processing time and number of iterations may influence the quality of a segmentation algorithm and, henceforth, impact the practicality of the segmentation method. An analysis of the number of iterations and the processing time before an optimal solution is found, may prove beneficial. This applies to areas where the segmented solutions change frequently in a short period, such as marketing automation.

A predefined list of emails sent to business is considered as marketing automation. Although sending the e-mails may not be automation in the true sense, the analytics and marketing processes can be automated (Koshy,

2018). In a blog on improving business productivity (Pick, 2019), Riya Sander indicated that marketing automation requires the extensive use of digital marketing to move prospects through a process to customers (Sander, 2019). Digital marketing also entails the use of search engine optimisation (SEO). This is a combination of tactics that help improve the ranking of a website on search engines, so that it appears among the top results in relevant web searches (Sander, 2019). Having the segmented enterprise target market linked to the best SEO strategy will be essential in the future of B2B marketing and further research is required in this area.

Matching the proper sales channel to the prospect will produce an increase in customer satisfaction, and prolong the customer's relationship with the telecommunication solution provider, which is an essential component of enterprise sales. A sales channel may be direct, with a salesperson acting as the customer account manager during the customer's lifecycle. Alternatively, a sales channel may be indirect, with a third party assigned to manage all customer requirements. Some business customers have a low ICT budget and may not spend a great deal on telecommunication solutions. Such customers may be assigned remote telephone account managers, or they could be managed via regular e-mail correspondence. Research on the correlation between the needs-based company segments and the proper sales channel will apply here.

Lead management plays an essential role in moving prospects to customers. Gartner (2019) defined CRM lead management as the process whereby leads are captured; their activities and behaviour tracked, thereby qualifying them and giving them constant attention to make them sales-ready, to be passed on to the sales team. More research may be conducted on the market segments of the prospective enterprise customers to find the best marketing channel to use. The more frequently used B2B marketing channels are emails to key decision-makers; content marketing or blogs; organic searches

via SEO; identifying social media groups per industry (or other analysed segment); and video distribution online (Ayyar, 2015).

In an ever-changing business environment where the global economy does affect local businesses, it will become essential to keep up to date with events that might affect the buying power of customers. This is especially true of business customers where decisions on spending are made for the whole enterprise, affecting more than a few subscribers at a time. Linking news events that affect the economy to company segments is, therefore, fertile ground for research.

Major global events may also influence businesses in various industries.

One current major event on a global scale is the coronavirus or COVID-19 outbreak. Not only the health industry, but also most other industries, such as tourism, technology, food services, entertainment, transport and retail, is affected by this disease. Technology companies, airlines, hotels and other businesses are being stifled by the restrictions imposed to curb the spread of the virus. Some industries such as the property sector are, however, benefiting from the lower interest rates, owing to the economic downturn. Incidentally, a technology company from China, where the virus originated, has provided recommendations to help companies manage the coronavirus crisis (Appier blog writers, 2020).

The recommendations include:

(1) Show support where the need is.

(2) Use relevant topics and keywords for precise targeting.

(3) Increase online content.

(4) Protect brands by using contextual targeting.

(5) Focus on multichannel communication.

(6) Target high-value customers.

(7) Include keywords and visuals relevant to the crisis.

The company supplies AI platforms to help enterprises solve challenging business problems. Therefore, the recommendations frequently mention

target markets; segmentation techniques; using AI tools; and deep learning for analysing data.

An often neglected area of research is the origins of a quantitative algorithm. An investigative research study will reveal the reasoning behind formulas and progression of thought behind the mathematical equations used in segmentation methods. Applications of these methods, long overlooked over time, may be shown as being very relevant to solve certain challenges today.

Alternatively, a next-generation MVA technique to be explored further, is structural equation modelling (SEM), which has become a quasi-standard in marketing and management research, especially when it comes to analysing the cause-effect relations between latent constructs (Hair *et al*. 2011). While marketing researchers have a basic understanding of covariance-based SEM (CB-SEM), most of them are only barely familiar with the other useful approach to SEM, namely partial least squares SEM (PLS-SEM), as researched by Esposito Vinzi *et al.* (2008) and Hair *et al*. (2011). The conclusion was that PLS-SEM path modelling, if appropriately applied, is indeed a silver bullet for estimating underlying models in many theoretical and observed data conditions (Hair *et al.,* 2011). Considering the ever-increasing importance of understanding latent phenomena, such as consumer perceptions, expectations, attitudes and intentions, and their influence on organisational performance measures, e.g. stock prices, it is not surprising that SEM has become one of the most prominent statistical analysis techniques today (Hair *et al*., 2011). It is therefore well worth comparing to the methods in this dissertation as further research.

The techniques mentioned in many of the recommendations are for pattern recognition, applicable to business segmentation in the form of quantitative segmentation methods. The results of this study reveal the vast opportunities for applying these methods in B2B marketing.

REFERENCES

Agarwal, P. (2019) *Neural networks – One vs multiple output neurons*, *Artificial Intelligence Stack Exchange*. Available at: https://ai.stackexchange.com/questions/13944/one-vs-multiple-output-neurons (Accessed: 25 August 2020).

Aldridge, T. (2020) *How many neurons are in the output layer?*, *Quora*. Available at: https://www.quora.com/How-many-neurons-are-in-the-output-layer (Accessed: 25 August 2020).

Anderson, E. (1935) 'The Irises of the Gaspe peninsula', *Bulletin of the American iris Society*, 59, pp. 2–5.

Anicho, O. *et al.* (2019) 'Comparative Study for Coordinating Multiple Unmanned HAPS for Communications Area Coverage', in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. Atlanta, GA, USA: IEEE, pp. 467–474. doi: 10.1109/ICUAS.2019.8797881.

App Store (2020) *Google Street View*, *App Store*. Available at: https://apps.apple.com/us/app/google-street-view/id904418768 (Accessed: 9 December 2020).

Appier blog writers (2020) '7 Things to Help Brands Manage the Coronavirus Crisis', *Appier blog*, 20 February. Available at: https://www.appier.com/blog/7-things-to-help-brands-manage-the-coronavirus-crisis/ (Accessed: 12 April 2020).

Apple Support (2020) *Use Siri on all your Apple devices*, *Apple Support*. Available at: https://support.apple.com/en-us/HT204389 (Accessed: 9 December 2020).

Ayyar, R. (2015) 'The top 5 B2B marketing channels, and how to ace them', *Memeburn*, 27 October. Available at: https://memeburn.com/2015/10/the-top-5-b2b-marketing-channels-and-how-to-ace-them/ (Accessed: 10 October 2019).

Bai, C. *et al.* (2017) 'Multicriteria Green Supplier Segmentation', *IEEE Transactions on Engineering Management*, 64(4), pp. 515–528. doi: 10.1109/TEM.2017.2723639.

Bain & Company (2018) *Customer Relationship Management*, *Management Tools*. Available at: https://www.bain.com/insights/management-tools-customer-relationship-management/ (Accessed: 1 December 2019).

Baldock, A. (2005) 'Marketing investment planning – B2B catches up', p. 3.

Ballardini, A. L. (2018b) 'A tutorial on Particle Swarm Optimization Clustering', p. 14.

Ballardini, A. L. (2018a) *PSO-Clustering algorithm [Matlab code]*. Iralab. Available at: https://github.com/iralabdisco/pso-clustering (Accessed: 25 August 2020).

Bang-Jensen, J. and Gutin, G. Z. (2009) *Digraphs: Theory, Algorithms and Applications*. 2nd edn. London: Springer-Verlag (Springer Monographs in Mathematics). doi: 10.1007/978-1-84800-998-1.

Belson, W. A. (1957) 'The Effects Of Television', *The Australian Quarterly*, 29(4), pp. 59–70.

Bengio, Y. (2009) 'Learning Deep Architectures for AI', *Foundations and Trends® in Machine Learning*, 2(1), pp. 1–127. doi: 10.1561/2200000006.

Biggs, D., De Ville, B. and Suen, E. (1991) 'A method of choosing multiway partitions for classification and decision trees', *Journal of Applied Statistics*, 18(1), pp. 49–62. doi: 10.1080/02664769100000005.

Blank, S. G. and Dorf, B. (2012) *The startup owner's manual: the step-by-step guide for building a great company*. Pescadero, California: K & S Ranch Inc. Available at: http://amzn.com/B009UMTMKS (Accessed: 10 September 2020).

BMI (2018) *Valued Insights Partner*, *BMI Research*. Available at: https://www.bmi.co.za/ (Accessed: 2 October 2018).

Bonabeau, E. *et al.* (1999) *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, USA.

Bratton, D. and Kennedy, J. (2007) 'Defining a Standard for Particle Swarm Optimization', in *2007 IEEE Swarm Intelligence Symposium*. *2007 IEEE Swarm Intelligence Symposium*, Honolulu, HI, USA: IEEE, pp. 120–127. doi: 10.1109/SIS.2007.368035.

Brownlee, J. (2016) 'Crash Course On Multi-Layer Perceptron Neural Networks', *Machine Learning Mastery*, 16 May. Available at: https://machinelearningmastery.com/neural-networks-crash-course/ (Accessed: 2 November 2020).

Bührmann, J. H. (2016) *The Effects Of Clustering On The Medium And Large-Scale Capacitated Location-Routing Problem*. Ph.D Thesis. University of Witwatersrand.

Bunkley, N. (2008) 'Joseph Juran, 103, Pioneer in Quality Control, Dies (Published 2008)', *The New York Times*, 3 March. Available at: https://www.nytimes.com/2008/03/03/business/03juran.html (Accessed: 7 October 2020).

Chan, C. C. H. *et al.* (2016) 'Marketing segmentation using the particle swarm optimization algorithm: a case study', *Journal of Ambient Intelligence and Humanized Computing*, 7(6), pp. 855–863. doi: 10.1007/s12652-016-0389-9.

Chauhan, N. S. (2019) *Build an Artificial Neural Network(ANN) from scratch: Part-1*, *Medium*. Available at: https://towardsdatascience.com/build-an-artificial-neural-network-ann-from-scratch-part-1-a21988497962 (Accessed: 11 August 2020).

Chen, Y. *et al.* (2007) 'Customer segmentation based on survival character', *Journal of Intelligent Manufacturing*, 18(4), pp. 513–517. doi: 10.1007/s10845-007-0059-z.

Chernev, A. (2007) *Strategic Marketing Analysis*. Second. Brightstar Media Inc (Kellog School of Management).

Chulis, K. (2012) *Optimal segmentation approach and application*, *IBM Developer*. Available at: http://www.ibm.com/developerworks/library/ba-optimal-segmentation/index.html (Accessed: 6 May 2020).

CIKD (2019) 'How much do you know about Scopus?', *Canadian Institute For Knowledge Development*, 13 August. Available at: https://cikd.ca/2019/08/13/how-much-do-you-know-about-scopus/ (Accessed: 29 July 2020).

Cirera, X. *et al.* (2016) *ICT Use, Innovation, and Productivity: Evidence from Sub-Saharan Africa*. The World Bank (Policy Research Working Papers). doi: 10.1596/1813-9450-7868.

Cireşan, D. *et al.* (2012) 'Multi-column Deep Neural Networks for Image Classification', *arXiv:1202.2745 [cs]*. Available at: http://arxiv.org/abs/1202.2745 (Accessed: 22 September 2020).

Cireşan, D. C. *et al.* (2010) 'Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition', *Neural Computation*, 22(12), pp. 3207–3220. doi: 10.1162/NECO_a_00052.

Clerc, M. (2006) *Particle swarm optimization*. London ; Newport Beach: ISTE.

Clerc, M. (2012) 'Standard Particle Swarm Optimisation'. Available at: https://hal.archives-ouvertes.fr/hal-00764996 (Accessed: 19 September 2019).

Cluzeau, J. M. *et al.* (2020) *Concepts of Design Assurance for Neural Networks (CoDANN)*. Policy Document. Available at: https://www.easa.europa.eu/document-library/general-publications/concepts-design-assurance-neural-networks-codann (Accessed: 2 August 2020).

Collins (2014) 'Collins English Dictionary', *Complete and Unabridged*. 12th edn. Edited by J. M. Sinclair. Available at: https://www.thefreedictionary.com/collins (Accessed: 7 December 2020).

Daedalean AG (2020) *Daedalean / Home*. Available at: https://daedalean.ai (Accessed: 2 August 2020).

Davies, D. L. and Bouldin, D. W. (1979) 'A Cluster Separation Measure', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2), pp. 224–227. doi: 10.1109/TPAMI.1979.4766909.

DeepAI (2019) *Epoch*, *machine-learning-glossary-and-terms*. Available at: https://deepai.org/machine-learning-glossary-and-terms/epoch (Accessed: 10 August 2020).

Dhandayudam, P. and Krishnamurthi, Dr. I. (2012) 'An improved clustering algorithm for customer segmentation', *International Journal of Engineering Science and Technology*, 4, p. 8.

Dieterich, J. M. and Hartke, B. (2012) 'Empirical Review of Standard Benchmark Functions Using Evolutionary Global Optimization', *Applied Mathematics*, 03(10), pp. 1552–1564. doi: 10.4236/am.2012.330215.

Dow Jones Factiva (2016) *Factiva - Global News Monitoring & Search Engine | Dow Jones*, *Dow Jones Professional*. Available at: https://professional.dowjones.com/factiva/ (Accessed: 19 June 2018).

Dua, D. and Graff, C. (2019) 'UCI Machine Learning Repository', in *UCI Machine Learning Repository*. Irvine, CA: University of California, School of Information and Computer Science. Available at: https://archive.ics.uci.edu/ml/datasets/Iris (Accessed: 17 May 2019).

Dullaghan, C. and Rozaki, E. (2017) 'Integration of Machine Learning Techniques to Evaluate Dynamic Customer Segmentation Analysis for Mobile Customers', *International Journal of Data Mining & Knowledge Management Process*, 7(1), pp. 13–24. doi: 10.5121/ijdkp.2017.7102.

Dun & Bradstreet (2019) *Dun & Bradstreet - Accelerate Growth and Improve Business Performance*, *Drive Performance with Partnerships*. Available at: https://www.dnb.com/ (Accessed: 12 March 2020).

Dunn, J. C. (1974) 'Well-Separated Clusters and Optimal Fuzzy Partitions', *Journal of Cybernetics*, 4(1), pp. 95–104. doi: 10.1080/01969727408546059.

EASA Regulations (2020) *The European Union Authority for aviation safety*, *EASA Pro*. Available at: https://www.easa.europa.eu/home (Accessed: 2 August 2020).

Eberhart, R. C. *et al.* (1996) *Computational Intelligence PC Tools*. Boston: Academic Press.

Efron, B. (1982) *The jackknife, the bootstrap, and other resampling plans*. Philadelphia, Pa: Society for Industrial and Applied Mathematics (CBMS-NSF Regional conference series in applied mathematics, 38).

Efron, B. (1983) 'Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation', *Journal of the American Statistical Association*, 78(382), pp. 316–331. doi: 10.1080/01621459.1983.10477973.

Esposito Vinzi, V. *et al.* (2008) 'REBUS-PLS: A response-based procedure for detecting unit segments in PLS path modelling', *Applied Stochastic Models in Business and Industry*, 24(5), pp. 439–458. doi: 10.1002/asmb.728.

Fader, P. S. *et al.* (2005) 'RFM and CLV: Using Iso-Value Curves for Customer Base Analysis', *Journal of Marketing Research*, 42(4), pp. 415–430. doi: 10.1509/jmkr.2005.42.4.415.

Farley, B. and Clark, W. (1954) 'Simulation of self-organizing systems by digital computer', *Transactions of the IRE Professional Group on Information Theory*, 4(4), pp. 76–84. doi: 10.1109/TIT.1954.1057468.

Fausett, L. (1994) *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice-Hall.

Fick, M. (2006) 'Neural Networks: Only study guide for HONNNS-H'. University of South Africa.

Fielding, A. and O'Muircheartaigh, C. A. (1977) 'Binary Segmentation in Survey Analysis with Particular Reference to AID', *The Statistician*, 26(1), p. 17. doi: 10.2307/2988216.

Fisher, R. A. (1936) 'The use of multiple measurements in taxonomic problems', *Annals of Eugenics*, 7(2), pp. 179–188. doi: 10.1111/j.1469-1809.1936.tb02137.x.

Flux, A. W. (1897) 'Vilfredo Pareto. Cours d'Économie Politique', *The Economic Journal*, 7(25), pp. 91–94. doi: 10.2307/2956966.

García-Gonzalo, E. and Fernández-Martínez, J. L. (2012) 'A Brief Historical Review of Particle Swarm Optimization (PSO)', *Journal of Bioinformatics and Intelligent Control*, 1(1), pp. 3–16. doi: 10.1166/jbic.2012.1002.

Gartner (2019) *Definition of Lead Management*, *Gartner Sales Glossary*. Available at: https://www.gartner.com/en/sales/glossary/lead-management (Accessed: 2 September 2019).

Ghnemat, R. and Jaser, E. (2015) 'Classification of Mobile Customers Behavior and Usage Patterns using Self-Organizing Neural Networks', *International Journal of Interactive Mobile Technologies (iJIM)*, 9(4), p. 4. doi: 10.3991/ijim.v9i4.4392.

Goodfellow, I. *et al.* (2016) *Deep learning*. Cambridge, Massachusetts London, England: The MIT Press (Adaptive computation and machine learning).

Goodman, L. A. (1979) 'Simple Models for the Analysis of Association in Cross-Classifications having Ordered Categories', *Journal of the American Statistical Association*, 74(367), pp. 537–552.

Goodman, L. A. and Kruskal, W. H. (1959) 'Measures of Association for Cross Classifications. II: Further Discussion and References', *Journal of the American Statistical Association*, 54(285), pp. 123–163. doi: 10.1080/01621459.1959.10501503.

Grover, S. (2016) *Analytical Segmentation for Data-Driven Marketing*, *SAS Blogs*. Available at: https://blogs.sas.com/content/customeranalytics/2016/01/18/analytical-segmentation-data-driven-marketing/ (Accessed: 16 July 2020).

Gufosowa, A. (2019) *K-fold cross validation*. Available at: https://commons.wikimedia.org/wiki/File:K-fold_cross_validation_EN.svg (Accessed: 28 July 2020).

Guresen, E. and Kayakutlu, G. (2011) 'Definition of artificial neural networks with comparison to other networks', *Procedia Computer Science*, 3, pp. 426–433. doi: 10.1016/j.procs.2010.12.071.

Hagan, M. T. *et al.* (1996) *Neural network design*. Boston: PWS Pub.

Hair, J. F. *et al.* (2011) 'PLS-SEM: Indeed a Silver Bullet', *Journal of Marketing Theory and Practice*, 19(2), pp. 139–152. doi: 10.2753/MTP1069-6679190202.

Hamilton, H. (2012) *Clustering*, *Knowledge Discovery in Databases*. Available at: http://www2.cs.uregina.ca/~dbd/cs831/notes/clustering/clustering.html (Accessed: 19 December 2019).

Haykin, S. S. (2009) *Neural networks and learning machines*. 3. ed. New York: Pearson.

Hebb, D. O. (1949) *The organization of behavior; a neuropsychological theory*. Oxford, England: Wiley (The organization of behavior; a neuropsychological theory), pp. xix, 335.

Heppner, F. H. and Grenander, U. (1990) 'A Stochastic Nonlinear Model for Coordinate Bird Flocks', in Krasner, S. (ed.) *The Ubiquity of Chaos*. American Association for the Advancement of Science, pp. 233–238. Available at: https://books.google.co.za/books?id=PrxfPQAACAAJ.

Hertz, J. *et al.* (1991) *Introduction to the theory of neural computation*. Redwood City, Calif: Addison-Wesley Pub. Co (Santa Fe Institute studies in the sciences of complexity, v. 1).

Hinton, G. E. (2007a) 'Boltzmann machine', *Scholarpedia*, 2(5). doi: 10.4249/scholarpedia.

Hinton, G. E. (2007b) 'Learning multiple layers of representation', *Trends in Cognitive Sciences*, 11(10). doi: 10.1016/j.tics.

Hinton, G. E. *et al.* (2006) 'A Fast Learning Algorithm for Deep Belief Nets', *Neural Computation*, 18(7). doi: 10.1162/neco.2006.18.7.1527.

Hochreiter, S. (1991) *Untersuchungen zu dynamischen neuronalen Netzen (Investigations into dynamic neural networks)*. Diploma Thesis. Institut für Informatik, Technische Universität München.

IBM (2012) *CHAID and Exhaustive CHAID Algorithms*, *IBM Statistics Software documentation*. Available at: ftp://ftp.software.ibm.com/software/analytics/spss/support/Stats/Docs/Statistics/Algorithms/13.0/TREE-CHAID.pdf (Accessed: 20 April 2019).

IBM (2014) *Home of IBM product documentation*, *IBM Knowledge Center*. Available at: www.ibm.com/support/knowledgecenter (Accessed: 20 April 2020).

IBM (2020) *SPSS Statistics 21.0.0*. IBM. Available at: www.ibm.com/support/knowledgecenter/sslvmb_21.0.0/eos (Accessed: 5 November 2019).

Inoue, A. *et al.* (2017) 'Mobile-carrier choice behavior analysis between three major mobile-carriers and mobile virtual network operators', in *IEEE Computer Society Conference, June 26-28, 2017*, Kanazawa, Japan. Available at: http://ieeexplore.ieee.org/document/8022769/ (Accessed: 29 September 2020).

Intellectus Statistics (2020) *Statistical Analysis Software for Students, Universities, and Businesses*. Statistics Solutions (Intellectus Statistics Resources). Available at: https://www.intellectusstatistics.com/intellectus-statistics-resources/ (Accessed: 19 April 2020).

Iress (2019) *Iress Software for better performance*, *Iress.com*. Available at: https://www.iress.com/ (Accessed: 30 January 2020).

Ivakhnenko, A. G. and Lapa, V. G. (1967) *Cybernetics and forecasting techniques*. New York : American Elsevier Pub. Co. Available at: http://archive.org/details/cyberneticsforec0000ivak (Accessed: 22 September 2020).

Jain, A. K. and Dubes, R. C. (1988) *Algorithms for Clustering Data*. Prentice Hall (Advanced Reference Series).

Jamil, M. and Yang, X.-S. (2013) 'A Literature Survey of Benchmark Functions For Global Optimization Problems', *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2), p. 150. doi: 10.1504/IJMMNO.2013.055204.

Kass, G. V. (1975b) *Significance testing in, and some Extensions of Automatic Interaction Detection*. Ph.D Thesis. University of Witwatersrand.

Kass, G. V. (1975a) 'Significance Testing in Automatic Interaction Detection (A.I.D.)', *Applied Statistics*, 24(2), p. 178. doi: 10.2307/2346565.

Kass, G. V. (1980) 'An Exploratory Technique for Investigating Large Quantities of Categorical Data', *Applied Statistics*, 29(2), p. 119. doi: 10.2307/2986296.

Kaw, A. (2012) 'Chapter 01.02 Measuring Errors', *Holistic Numerical Methods*. Available at: https://nm.mathforcollege.com/chapter-01-02-measuring-errors/ (Accessed: 2 November 2020).

Kennedy, J. and Eberhart, R. (1995) 'Particle swarm optimization', in *Proceedings of ICNN'95 - International Conference on Neural Networks*, pp. 1942–1948 vol.4. doi: 10.1109/ICNN.1995.488968.

Khan, Y. *et al.* (2019) 'Customers Churn Prediction using Artificial Neural Networks (ANN) in Telecom Industry', *International Journal of Advanced Computer Science and Applications (IJACSA)*, 10(9). doi: 10.14569/IJACSA.2019.0100918.

Klein, R. J. *et al.* (2002) 'Healthy People 2010 Criteria for Data Suppression'. American Psychological Association (National Centre for Health Statistics). doi: 10.1037/e583742012-001.

Kohavi, R. (1995) 'A study of cross-validation and bootstrap for accuracy estimation and model selection', in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*. San Mateo, CA, p. 7.

Koshy, V. (2018) *How to Align Your Marketing Automation Strategy with the Buyer's Journey*, *EngageBay - All-in-one marketing, sales, and service software for growing businesses*. Available at: https://www.engagebay.com/blog/marketing-automation-strategy/ (Accessed: 10 August 2020).

Kotler, P. and Keller, K. L. (2006) *Marketing management*. Prentice Hall. Available at: http://archive.org/details/marketingmanagem00phil (Accessed: 10 September 2020).

Koza, J. R. (1992) *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press.

Kurenkov, A. (2015) *A 'Brief' History of Neural Nets and Deep Learning*, *Andrey Kurenkov's Web World*. Available at: /writing/ai/a-brief-history-of-neural-nets-and-deep-learning/ (Accessed: 22 June 2019).

Langton, C. G. (1997) *Artificial Life: An Overview*. MIT Press.

Law, M. T. *et al.* (2017) 'Efficient Multiple Instance Metric Learning Using Weakly Supervised Data', in. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, pp. 5948–5956. doi: 10.1109/CVPR.2017.630.

Le, Q. V. *et al.* (2012) 'Building high-level features using large scale unsupervised learning', in. *29 th International Conference on Machine Learning*, Edinburgh, Scotland. Available at: http://arxiv.org/abs/1112.6209 (Accessed: 12 July 2020).

Leung, Y. *et al.* (2000) 'Clustering by scale-space filtering', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12), pp. 1396–1410. doi: 10.1109/34.895974.

Liebowitz, J. (2006) *Strategic Intelligence: Business Intelligence, Competitive Intelligence, and Knowledge Management*. Auerbach Publications. doi: 10.1201/9781420013900.

Lin, Q. (2007) 'Mobile Customer Clustering Analysis Based on Call Detail Records', *Communications of the IIMA*, 7(4), p. 6.

Lloyd, S. (1982) 'Least squares quantization in PCM', *IEEE Transactions on Information Theory*, 28(2), pp. 129–137. doi: 10.1109/TIT.1982.1056489.

Mahmood, H. (2019) *Activation Functions in Neural Networks*, *Medium*. Available at: https://towardsdatascience.com/activation-functions-in-neural-networks-83ff7f46a6bd (Accessed: 11 August 2020).

Maladkar, K. (2018) 'Types of Activation Functions in Neural Networks and Rationale behind it', *Analytics India Magazine*, 11 January. Available at: https://analyticsindiamag.com/most-common-activation-functions-in-neural-networks-and-rationale-behind-it/ (Accessed: 11 August 2020).

Markoff, J. (2012) 'Scientists See Promise in Deep-Learning Programs', *The New York Times*, 23 November. Available at: https://www.nytimes.com/2012/11/24/science/scientists-see-advances-in-deep-learning-a-part-of-artificial-intelligence.html (Accessed: 22 September 2020).

Martínez-López, F. J. and Casillas, J. (2013) 'Artificial intelligence-based systems applied in industrial marketing: An historical overview, current and future insights', *Industrial Marketing Management*, 42(4), pp. 489–495.

MathWorks (2019) *MATLAB*. United States (MATLAB and Simulink). Available at: https://www.mathworks.com/products/matlab.html (Accessed: 15 May 2020).

May, M. and Smith, T. (2012) 'Battle for Value: Wargaming for Business, Non-Profit, and Government Strategy Development', in Cunha, M. M. C. (ed.) *Handbook of Research on Serious Games as Educational, Business and Research Tools*. IGI Global, pp. 578–597. Available at: https://www.igi-global.com/gateway/chapter/64274 (Accessed: 29 July 2020).

Mazur (2015) 'A Step by Step Backpropagation Example', *Matt Mazur*, 17 March. Available at: https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/ (Accessed: 6 October 2020).

McCulloch, W. S. and Pitts, W. (1943) 'A logical calculus of the ideas immanent in nervous activity', *Bulletin of Mathematical Biophysics*, 5(4), pp. 115–133. doi: 10.1007/BF02478259.

McCullock, J. (2012) *PSO Solves Travelling Salesperson Problem*, *Mnemosyne studio*. Available at: http://www.mnemstudio.org/particle-swarm-tsp-example-1.htm (Accessed: 12 July 2019).

Media (2018) 'Machine learning - How does Sigmoid activation work in multi-class classification problems', *Data Science Stack Exchange*, October. Available at: https://datascience.stackexchange.com/questions/39264/how-does-sigmoid-activation-work-in-multi-class-classification-problems (Accessed: 20 August 2020).

Messenger, R. and Mandell, L. (1972) 'A Modal Search Technique for Predictibe Nominal Scale Multivariate Analys', *Journal of the American Statistical Association*, 67(340), p. 768. doi: 10.2307/2284634.

Mijwel, M. M. (2016) 'Application of particle swarm optimization in 3-dimensional travelling salesman problem'. Unpublished. Available at: http://rgdoi.net/10.13140/RG.2.2.31886.05447 (Accessed: 20 September 2020).

Mijwel, M. M. (2018) 'Particle Swarm Optimization'. Baghdad College of Economics Sciences University.

Minsky, M. L. and Papert, S. (1969) *Perceptrons; an Introduction to Computational Geometry*. MIT Press.

MissingLink.ai (2018) *Hyperparameters: Optimization Methods and Real World Model Management*, *Neural Network Concepts*. Available at: https://missinglink.ai/guides/neural-network-concepts/hyperparameters-optimization-methods-and-real-world-model-management/ (Accessed: 10 August 2020).

Mitchell, T. M. (1997) *Machine Learning*. New York: McGraw-Hill (McGraw-Hill series in computer science).

Mohamed, K. S. (2018) 'Thermo-Inspired Machine Learning Algorithm: Simulated Annealing', in Mohamed, K. S. (ed.) *Machine Learning for Model Order Reduction*. Cham: Springer International Publishing, pp. 35–46. doi: 10.1007/978-3-319-75714-8_3.

Moreira, M. and Fiesler, E. (1995) 'Neural Networks with Adaptive Learning Rate and Momentum Terms', *IDIAP Technical report*.

Morgan, J. N. *et al.* (1966) *Productive Americans: A study of how individuals contribute to economic progress*. University of Michigan.

Morgan, J. N. and Sonquist, J. A. (1963a) 'Problems in the Analysis of Survey Data, and a Proposal', *Journal of the American Statistical Association*, 58(302), p. 21.

Morgan, J. N. and Sonquist, J. A. (1963b) 'Some results from a non-symmetrical branching process that looks for interaction effects', *Young*, 8, pp. 40–52.

Mwantimwa, K. (2019) 'ICT usage to enhance firms' business processes in Tanzania', *Journal of Global Entrepreneurship Research*, 9(1), pp. 1–23. doi: 10.1186/s40497-019-0170-6.

Nguyen, T. A. (2012) 'A Guide to Best Current B2B Customer Segmentation'. Openview Labs. Available at: https://openviewpartners.com/blog/#.X1oafmhLjcs.

Nguyen, T. A. (2018) 'Customer Segmentation: A Step by Step Guide for Growth', *Openview Partners*, 3 July. Available at: https://openviewpartners.com/blog/customer-segmentation/ (Accessed: 4 September 2020).

Nicholson, C. (2019) *A Beginner's Guide to Neural Networks and Deep Learning*, *Pathmind*. Available at: http://wiki.pathmind.com/neural-network (Accessed: 21 June 2020).

Nie, N. H. *et al.* (1970) *SPSS: Statistical Package for the Social Sciences*. IBM.

Omran, M. G. H. (2006) *Particle Swarm Optimization for Pattern Recognition and Image Processing*. Ph.D Thesis. University of Pretoria. doi: 10.1007/978-3-540-34956-3_6.

Parsopoulos, K. E. and Vrahatis, M. N. (2010) *Particle Swarm Optimization and Intelligence: Advances and Applications*. Information Science Publishing (IGI Global), Hershey, PA, U.S.A. doi: 10.13140/2.1.3681.1206.

Pick, T. (2019) *B2B Marketing Zone*, *Aggregage*. Available at: https://www.b2bmarketingzone.com/ (Accessed: 12 October 2020).

Poli, R. (2008) 'Analysis of the Publications on the Applications of Particle Swarm Optimisation', *Journal of Artificial Evolution and Applications*, 2008, pp. 1–10. doi: 10.1155/2008/685175.

Qiuru, C. *et al.* (2012) 'Telecom customer segmentation based on cluster analysis', in *2012 International Conference on Computer Science and Information Processing (CSIP)*. Xian, Shaanxi, China: IEEE, pp. 1179–1182. doi: 10.1109/CSIP.2012.6309069.

Quenouille, M. H. (1949) 'Approximate Tests of Correlation in Time-Series', *Journal of the Royal Statistical Society. Series B (Methodological)*, 11(1), pp. 68–84.

Rajasekaran, S. and Vijayalakshmi Pai, G. A. (2003) *Neural Networks, Fuzzy Logic and Genetic algorithm: Synthesis and Applications*. PHI Learning Pvt. Ltd.

Rasello research (2017) *CRM & Business Intelligence Software*, *www.rasello.com*. Available at: http://rasello.com/ (Accessed: 29 August 2018).

Ratner, B. (2003) *Market Segment Classification Modelling With Logistic Regression*. Available at: http://www.dmstat1.com/res/_5MarketSegmClassificationwithLRM.html (Accessed: 3 August 2020).

Rauh, O. (2013a) *KMC - a simple tool for k-means clustering*, *Die Informatikseite von Prof. Dr. Otto Rauh*. Available at: http://www.orauh.de/data-mining/kmc-clustering-tool/ (Accessed: 4 November 2019).

Rauh, O. (2013b) 'KMC User Guide'. Heilbronn University, Germany. Available at: https://www.orauh.de/software/kmc-clustering-tool/.

Reeves, W. T. (1983) 'Particle Systems A Technique for Modeling a Class of Fuzzy Objects', *Computer Graphics*, 17(3), pp. 359–375.

Reynolds, C. W. (1987) 'Flocks, Herds, and Schools: A Distributed Behavioral Model', in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: Association for Computing Machinery (SIGGRAPH '87), pp. 25–34. doi: 10.1145/37401.37406.

Rosenblatt, F. (1958) 'The perceptron: A probabilistic model for information storage and organization in the brain.', *Psychological Review*, 65(6), pp. 386–408. doi: 10.1037/h0042519.

Rouse, M. (2018) *What is an Artificial Neural Network (ANN)?*, *SearchEnterpriseAI*. Available at: https://searchenterpriseai.techtarget.com/definition/neural-network (Accessed: 17 June 2020).

Rousseeuw, P. J. (1987) 'Silhouettes: A graphical aid to the interpretation and validation of cluster analysis', *Journal of Computational and Applied Mathematics*, 20, pp. 53–65. doi: 10.1016/0377-0427(87)90125-7.

Rumelhart, D. E. *et al.* (1986) 'Learning representations by back-propagating errors', *Nature*, 323, pp. 533–536. doi: 10.1038/323533a0.

Rygielski, C. *et al.* (2002) 'Data mining techniques for customer relationship management', *Technology in Society*, p. 20.

Sander, R. (2019) *Improving Business Productivity with Marketing Automation*, *B2B Marketing Zone*. Available at: http://www.b2bmarketingzone.com/marketing-automation/segmentation/?open-article-id=9563087&article-title=improving-business-productivity-with-marketing-automation&blog-domain=webbiquity.com&blog-title=webbiquity (Accessed: 12 October 2019).

Sanjay.M (2018) *Why and how to Cross Validate a Model?*, *Towards Data Science*. Available at: https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f (Accessed: 31 January 2020).

Scherer, D. *et al.* (2010) 'Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition', in Diamantaras, K., Duch, W., and Iliadis, L. S. (eds) *Artificial Neural Networks – ICANN 2010*. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), pp. 92–101. doi: 10.1007/978-3-642-15825-4_10.

Schmidhuber, J. (1992) 'Learning Complex, Extended Sequences Using the Principle of History Compression', *Neural Computation*, 4(2), pp. 234–242. doi: 10.1162/neco.1992.4.2.234.

Schmidhuber, J. (2015) 'Deep Learning in Neural Networks: An Overview', *Neural Networks*, 61, pp. 85–117. doi: 10.1016/j.neunet.2014.09.003.

Schmidhuber, J. (2018) 'How bio-inspired deep learning keeps winning competitions'. Available at: https://www.kurzweilai.net/how-bio-inspired-deep-learning-keeps-winning-competitions (Accessed: 27 April 2020).

Scopus (2020) *The largest database of peer-reviewed literature*. Available at: https://www.elsevier.com/en-xm/solutions/scopus (Accessed: 29 August 2020).

Scott, A. J. and Knott, M. (1974) 'A Cluster Analysis Method for Grouping Means in the Analysis of Variance', *Biometrics*, 30(3), p. 507. doi: 10.2307/2529204.

Shankar, B. (2015) 'Machine learning - Is validation set always necessary?', *Cross Validated*, 24 May. Available at: https://stats.stackexchange.com/questions/153789/is-validation-set-always-necessary (Accessed: 22 June 2020).

Sharma, S. (2019a) *Activation Functions in Neural Networks*, *Medium*. Available at: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6 (Accessed: 11 August 2020).

Sharma, S. (2019b) *Epoch vs Batch Size vs Iterations*, *Medium*. Available at: https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9 (Accessed: 10 August 2020).

Sheng, L. and Xu, X. (2006) *A method of telecom consumer market segmentation based on the RFM model*, *Journal of Harbin Institute of Technology*. Available at: http://en.cnki.com.cn/Article_en/CJFDTOTAL-HEBX200605024.htm (Accessed: 16 September 2020).

Silva, A. *et al.* (2002) 'An Empirical Comparison of Particle Swarm and Predator Prey Optimisation', in *Artificial Intelligence and Cognitive Science. 13th Irish Conference, AICS 2002*, Limerick, Ireland: Springer (Lecture Notes in Computer Science), pp. 103–110. doi: 10.1007/3-540-45750-X_13.

Simkin, L. (2008) 'Achieving market segmentation from B2B sectorisation', *Journal of Business & Industrial Marketing*, 23(7), pp. 464–474. doi: 10.1108/08858620810901220.

Smith, T. J. (2013) *What Are Firmographics?*, *The Wiglaf Journal*. Available at: https://wiglafjournal.com/what-are-firmographics/ (Accessed: 10 September 2020).

Sonquist, J. A. *et al.* (1971) *Searching for structure (alias-AID-III);: An approach to analysis of substantial bodies of micro-data and documentation for a computer program ... the Automatic Interaction Detector Program),*. Institute for Social Research, University of Michigan.

Sonquist, J. A. *et al.* (1973) *Searching for structure: an approach to analysis of substantial bodies of micro-data and documentation for a computer program*. Rev. ed. Ann Arbor: Survey Research Center, University of Michigan.

Sousa, T. *et al.* (2004) 'Particle Swarm based Data Mining Algorithms for classification tasks', *Parallel Computing*, 30(5–6), pp. 767–783. doi: 10.1016/j.parco.2003.12.015.

South African Reserve Bank (2011) 'Institutional Sector Classification Guide for SA'. Available at: https://www.resbank.co.za/Publications/Guides/Pages/Institutional-Sector-Classification-Guide-for-SA---2011.aspx (Accessed: 7 October 2020).

Srivastava, A. and Kumbharvadiya, S. (2014) 'Developing the Code: Executing Particle Swarm Optimization in SAS®', in *SAS Global Forum 2014*. Washington, DC (2030), p. 4.

Stallman, R. (1991) *GPL-v2 licence*. Free Software Foundation (GNU General Public License). Available at: http://www.gnu.org/licenses/gpl-2.0.html (Accessed: 25 October 2019).

Statistics Solutions (2020) 'Chi-Square Test of Independence', *Statistical Consulting Blog*. Available at: http://www.statisticssolutions.com/non-parametric-analysis-chi-square/ (Accessed: 23 July 2020).

Steinhaus, H. (1956) 'Sur la division des corps matériels en parties (On the division of material bodies into parts)', *Bulletin de l'académie polonaise des sciences*, 4(12), pp. 801–804.

Strouse, K. G. (2004) *Customer-centered Telecommunications Services Marketing*. Artech House.

Stuntebeck, V. A. (2012) 'B2B customer segmentation: Important considerations when segmenting business customers'. IBM developerWorks. Available at: https://www.ibm.com/developerworks/library/ba-b2b-custseg-spss/index.html (Accessed: 10 October 2019).

sUAS News Press (2020) 'EASA and Daedalean Created Concepts of Design Assurance for Neural Networks', *The Business of Drones*, 2 April. Available at: https://www.suasnews.com/2020/04/easa-and-daedalean-created-concepts-of-design-assurance-for-neural-networks/ (Accessed: 2 October 2020).

Suh, E. H. *et al.* (1999) 'Customer list segmentation using the combined response model', 17(2), pp. 89–97.

Taves, P. (2010) *A Basic Introduction to CHAID*, *SmartDrill Data Mining*. Available at: http://smartdrill.com/Introduction-to-CHAID.html (Accessed: 12 July 2019).

Trading Economics (2017) *Tanzania - Economic Indicators*, *tradingeconomics.com*. Available at: https://tradingeconomics.com/tanzania/indicators (Accessed: 30 March 2019).

Troiani, R. (2016) *Easy CHAID*. Available at: http://www.easychaid.com/ (Accessed: 25 November 2019).

Twomey, J. M. and Smith, A. E. (1995) 'Validation and Verification', in Kartam, N., Flood, I., and Garett, J. (eds) *Artificial Neural Networks for Civil Engineers: Fundamentals and Applications (Chapter 4)*. New York: ASCE Press, pp. 1–29.

United Nations Statistical Office (1990) *International Standard Industrial Classification of All Economic Activities*. United Nations.

United Nations Statistical Office (ed.) (2008) *International Standard industrial classification of all economic activities (ISIC)*. Rev. 4. New York: United Nations (Statistical papers. Series M, no. 4, rev. 4).

US Census Bureau (1938) *Statistical Abstract of the United States: 1937*, *The United States Census Bureau*. Available at: https://www.census.gov/library/publications/1938/compendia/statab/59ed.html (Accessed: 23 October 2020).

US Office of Management and Budget (2007) 'North American Industry Classification System (NAICS)'. Executive Office of the president. Available at: https://www.census.gov/eos/www/naics/ (Accessed: 7 October 2020).

Van der Merwe, D. W. and Engelbrecht, A. P. (2003) 'Data clustering using particle swarm optimization', in *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, pp. 215-220 Vol.1. doi: 10.1109/CEC.2003.1299577.

Vijaya, J. and Sivasankar, E. (2017) 'An efficient system for customer churn prediction through particle swarm optimization based feature selection model with simulated annealing', *Cluster Computing*, 22(S5), pp. 10757–10768. doi: 10.1007/s10586-017-1172-1.

Vincent, J. F. V. *et al.* (2006) *Biomimetics: its practice and theory*, *Journal of the Royal Society, Interface*. J R Soc Interface. doi: 10.1098/rsif.2006.0127.

Von Neumann, J. (1958) *The Computer and The Brain*. First Edition. New Haven: Yale University Press (Library of Congress, 58–6542).

Von Neumann, J. (2000) *The Computer and the Brain*. New Haven, CT : Yale Nota Bene. Available at: http://archive.org/details/computerbrain0000vonn (Accessed: 2 November 2020).

Von Neumann, K. (1957) 'Preface, Von Neumann Silliman lectures', *The MacTutor History of Mathematics archive*. Available at: http://www-history.mcs.st-and.ac.uk/Extras/Von_Neumann_Silliman.html.

Vozhehova, R. A. *et al.* (2019) 'Artificial Neural Network Use For Sweet Corn Water Consumption Prediction Depending On Cultivation Technology Peculiarities', *Research Journal of Pharmaceutical Biological and Chemical Sciences*.

Wang, D. *et al.* (2018) 'Particle swarm optimization algorithm: an overview', *Soft Computing*, 22(2), pp. 387–408. doi: 10.1007/s00500-016-2474-6.

Wang, P.-H. *et al.* (2019) 'Application of Neural Networks to Explore Manufacturing Sales Prediction', *Applied Sciences*, 9(23), p. 5107. doi: 10.3390/app9235107.

Weiss, D. J. (1995) 'Polychotomous or Polytomous?', *Applied Psychological Measurement*, 19(1), pp. 4–4. doi: 10.1177/014662169501900102.

Weiss, S. M. and Kulikowski, C. A. (1991) *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems*. 1st Edition. San Mateo, CA: Morgan Kaufmann.

Weisstein, E. W. (1999) *NP-Hard Problem*. Wolfram Research, Inc. Available at: https://mathworld.wolfram.com/NP-HardProblem.html (Accessed: 22 October 2020).

Weng, J. *et al.* (1992) 'Cresceptron: a self-organizing neural network which grows adaptively', in *Proceedings of the IJCNN International Joint Conference on Neural Networks*, pp. 576–581 vol.1. doi: 10.1109/IJCNN.1992.287150.

Werbos, P. J. (1974) *Beyond regression: new tools for prediction and analysis in the behavioral sciences*. Ph.D Thesis.

Werbos, P. J. (1982) 'Applications of advances in nonlinear sensitivity analysis.', in *Lecture Notes in Control and Information Sciences*. *10th IFIP Conference*, New York: Springer-Verlag, pp. 762–770. Available at: https://jglobal.jst.go.jp/en/detail?JGLOBAL_ID=200902071980476932 (Accessed: 2 July 2019).

Westfall, P. and Henning, K. S. S. (2013) *Understanding Advanced Statistical Methods*. CRC Press.

Who Owns Whom contributors (2016c) 'Average ICT Spend as a percentage of turnover for the industry'. Who Owns Whom (Pty) Ltd. Available at: https://www.whoownswhom.co.za/.

Who Owns Whom contributors (2016a) 'Employee and Turnover estimates'. Who Owns Whom (Pty) Ltd. Available at: https://www.whoownswhom.co.za/.

Who Owns Whom contributors (2016b) 'Thresholds for the classification of enterprises by size'. Who Owns Whom (Pty) Ltd. Available at: https://www.woweb.co.za/.

Who Owns Whom research (2015) *Who Owns Whom - African Business Information*, *Woweb.co.za*. Available at: https://www.woweb.co.za/ (Accessed: 12 June 2016).

Wikipedia contributors (2020b) 'JavaScript', *Wikipedia*. Wikipedia, The Free Encyclopedia. Available at: https://en.wikipedia.org/w/index.php?title=JavaScript&oldid=981005078 (Accessed: 12 October 2020).

Wikipedia contributors (2020a) 'Pareto principle', *Wikipedia*. Wikipedia, The Free Encyclopedia. Available at: https://en.wikipedia.org/w/index.php?title=Pareto_principle&oldid=982066158 (Accessed: 9 July 2020).

Willan, D. (2014) 'B2B Market Segmentation'. Circle Research. Available at: https://savanta.com/.

Wolstenholme, S. (2002) *Neural Network Software*, *Neural Planner Software*. Available at: http://www.npsnn.com/ (Accessed: 22 August 2020).

Wolstenholme, S. (2015) 'JustNN Help User Guide'. Neural Planner Software Ltd.

Wolstenholme, S. (2016) *JustNN*. United Kingdom: Neural Planner Software.

World Bank (2017) *Tanzania | Data*, *worldbank.org*. Available at: https://data.worldbank.org/country/tanzania (Accessed: 1 June 2017).

Wyman, F. (2005) 'Best Segmentation Practices and Targeting Procedures that Provide the most Client-Actionable Strategy'. *The Market Research Event*, San Francisco, California. Available at: http://www.websm.org/db/17/11405/Events/IIR_Market_Research_Event/?menu=1&lst=&q=search_1_111111_-1&qdb=17&qsort=1.

Wyner, G. A. (1995) 'Segmentation Then and Now', *Marketing Research*, p. 40.

Yan, J. *et al.* (2015) 'Sales pipeline win propensity prediction: a regression approach', *arXiv:1502.06229 [cs]*. Available at: http://arxiv.org/abs/1502.06229 (Accessed: 10 September 2020).

Yun, C. and Yan, P. (2013) 'A customer portfolio model based on multi-phase marketing strategy and particle swarm optimization', in *2013 International Conference on Management Science and Engineering 20th Annual Conference Proceedings*. Harbin, China: IEEE, pp. 957–962. doi: 10.1109/ICMSE.2013.6586393.

Zhang, Y. *et al.* (2017) 'A data-driven quantitative assessment model for taxi industry: the scope of business ecosystem's health', *European Transport Research Review*, 9(2), p. 23. doi: 10.1007/s12544-017-0241-0.

Zhao, J. *et al.* (2010) 'Improved K-Means cluster algorithm in telecommunications enterprises customer segmentation', pp. 167–169.

Zulu, Min. L. D. (2018) 'National Small Business Amendment Schedule 1', *Government Gazette*, (No. 41970), p. 5.

Zyxo (2010) 'Mixotricha', *The Difference between Segmentation and Clustering*, 17 July. Available at: https://zyxo.wordpress.com/2010/07/17/the-difference-between-segmentation-and-clustering/ (Accessed: 4 September 2020).

Appendix A - TARGET AND REFERENCE DATA

A.1 Target dataset

A.1.1 KMC Input data

Below an extract of the data used per target flag for the KMC test runs.

Table A.1: KMC target data extract for Prospects

Co_ID	Location_Priority	SIC_Code	Co Employees	ICT Spend	Perc ICTCountry	Perc ICTUsgPopulation
1	38	88940	56	34 232	0.00000245	0.00000250
2	39	61109	50	19 501 098	0.00139293	0.00000400
3	43	35420	11	10 819	0.00000077	0.00000090
4	39	71239	103	1 304 806	0.00009320	0.00000450
5	1	30420	158	28 163	0.00000201	0.00000700
6	3	93199	330	16 057	0.00000115	0.00001190
7	38	39294	269	2 653 215	0.00018951	0.00000970
8	38	71239	103	1 304 806	0.00009320	0.00000450
9	38	63319	43	19 709	0.00000141	0.00000350
10	39	83190	120	497 370	0.00003553	0.00000530
11	39	71222	71	164 242	0.00001173	0.00000310
12	39	88940	1118	77 886	0.00000556	0.00004040
14	43	38309	29	7 286 911	0.00052049	0.00000230
15	22	61393	772	8 944	0.00000064	0.00002790
16	39	83190	120	497 370	0.00003553	0.00000530
17	22	93191	364	167 396	0.00001196	0.00001320
20	38	83190	120	497 370	0.00003553	0.00000530
21	6	62310	72	24 439	0.00000175	0.00000320
22	43	32520	281	54 889	0.00000392	0.00001020
23	38	83200	3747	92 605	0.00000662	0.00013540
24	39	63319	278	106 499	0.00000761	0.00001000
25	3	62602	204	4 050	0.00000029	0.00000740
26	39	61909	109	138 623	0.00000990	0.00000390
27	39	62330	4858	27 884	0.00000199	0.00017550
28	1	61909	109	138 623	0.00000990	0.00000390
29	43	88993	1142	105 079	0.00000751	0.00004130
30	10	88140	461	4 081 293	0.00029152	0.00001670
31	40	32520	281	54 889	0.00000392	0.00001020
32	38	83200	185	167 570	0.00001197	0.00000820
33	39	62310	56	901 190	0.00006437	0.00000250
34	40	92002	456	24 830	0.00000177	0.00001650
35	39	71222	20	207 653	0.00001483	0.00000160
36	2	88110	258	130 855 989	0.00934678	0.00000930
38	43	61109	3945	247 162	0.00001765	0.00014250
39	27	71222	120	185 082	0.00001322	0.00000530
40	3	88110	6	2 825	0.00000020	0.00000050
41	38	71222	39	194 457	0.00001389	0.00000310
42	39	64103	107	2 777 284	0.00019838	0.00000390
43	39	88920	7	3 679	0.00000026	0.00000050
44	39	71229	4000	88 691	0.00000634	0.00014450
45	38	61102	828	112 500	0.00000804	0.00002990
46	43	50230	123	100 869	0.00000721	0.00000540
47	10	50230	123	100 869	0.00000721	0.00000540
48	39	50220	17	5 571 880	0.00039799	0.00000130
49	38	30491	352	63 105	0.00000451	0.00001270
50	39	36600	3215	89 392	0.00000639	0.00011620

Co_ID	Location_Priority	SIC_Code	Co Employees	ICT Spend	Perc ICTCountry	Perc ICTUsgPopulation
2005	13	74140	219	12 578	0.00000090	0.00000790
2006	44	92003	112	2 732 640	0.00019519	0.00000490
2009	38	32430	1058	285 480	0.00002039	0.00003820
2010	40	33800	49	1 067 035	0.00007622	0.00000400
2012	36	88993	1142	105 079	0.00000751	0.00004130
2013	43	71222	120	185 082	0.00001322	0.00000530
2014	8	88993	1142	105 079	0.00000751	0.00004130
2016	43	88993	1142	105 079	0.00000751	0.00004130
2020	38	62519	204	76 927	0.00000550	0.00000740
2022	37	64101	36	1 530 326	0.00010931	0.00000290
2023	39	61392	206	24 041	0.00000172	0.00000740
2024	13	81910	133	9 454 085	0.00067529	0.00000480
2027	39	33599	93	39 418	0.00000282	0.00000410
2028	39	88993	1142	105 079	0.00000751	0.00004130
2029	3	88993	1142	105 079	0.00000751	0.00004130
2030	20	71222	120	185 082	0.00001322	0.00000530
2031	34	71222	41	193 780	0.00001384	0.00000330
2032	22	11130	486	42 686	0.00000305	0.00001760
2033	1	41113	273	1 038 417	0.00007417	0.00000990
2034	39	64101	547	100 716	0.00000719	0.00001980
2036	48	64103	107	2 777 284	0.00019838	0.00000390
2037	2	96330	242	8 334 400	0.00059531	0.00000870
2038	38	64103	107	2 777 284	0.00019838	0.00000390
2039	43	64101	19	2 899 564	0.00020711	0.00000150
2040	48	64103	107	2 777 284	0.00019838	0.00000390
2041	39	35800	539	44 116	0.00000315	0.00001950
2042	43	83190	120	497 370	0.00003553	0.00000530
2044	19	81910	453	2 775 703	0.00019826	0.00001640
2045	43	81910	3519	2 427 467	0.00017339	0.00012710
2046	39	88940	1118	77 886	0.00000556	0.00004040
2047	38	63319	278	106 499	0.00000761	0.00001000
2048	2	71222	120	185 082	0.00001322	0.00000530
2050	43	74140	24	114 776	0.00000820	0.00000190
2052	44	62310	1912	84 729	0.00000605	0.00006910
2053	38	62399	80	2 243 939	0.00016028	0.00000350
2054	43	64103	107	2 777 284	0.00019838	0.00000390
2055	12	41111	1000	32 439	0.00000232	0.00003610
2056	39	63319	278	106 499	0.00000761	0.00001000
2058	38	62310	1912	84 729	0.00000605	0.00006910
2059	39	62519	204	76 927	0.00000550	0.00000740
2060	39	62519	204	76 927	0.00000550	0.00000740
2061	39	71222	120	185 082	0.00001322	0.00000530
2062	39	61392	206	24 041	0.00000172	0.00000740
2063	36	88110	258	130 855 989	0.00934678	0.00000930
2064	38	62110	146	804 322	0.00005745	0.00000530
2065	2	81990	689	4 256 152	0.00030401	0.00002490

Table A.2: KMC target data extract for Customers

Co_ID	Location_Priority	SIC_Code	ICT Spend	Telecom ARPU	Telecom Revenue	Telecom Subscribers
7006	38	61392	24040.67	78 221	234 663	3
7076	20	11400	33588.68	28 035	756 933	27
7076	38	33599	70844.87	50 731	1 369 729	27
7119	39	63319	106498.61	78 221	312 884	4
7124	43	61420	4168.55	95 438	95 438	1
7167	38	64103	2777283.99	396 440	792 879	2
7171	43	50220	141248.3	95 438	95 438	1
7189	38	92007	154387.67	1 149	257 335	224
7298	38	35101	994541.66	186 178	1 117 070	6
7318	38	61399	49272.66	95 438	95 438	1
7334	39	88311	139964.97	53 346	800 187	15
7346	38	61909	138622.53	78 221	156 442	2
7381	43	32510	62137.3	95 438	95 438	1
7439	39	36600	340038.06	71 662	3 296 471	46
7464	38	11110	59110.69	49 093	932 768	19
7468	2	83190	497369.62	95 438	190 876	2
7489	8	74190	2042738.89	192 943	771 772	4
7506	1	36600	1308178.63	44 305	1 041 156	47
7506	3	36600	685236.42	22 152	553 806	25
7506	20	36600	373765.32	22 152	287 979	13
7506	39	36600	186882.66	22 152	155 066	7
7506	43	36600	1059001.74	66 457	841 786	38
7506	46	36600	311471.1	22 152	243 675	11
7540	43	11220	3388.25	1 628	9 769	6
7550	14	83190	497369.62	95 438	190 876	2
7575	44	64101	860808.18	147 621	147 621	1
7647	43	64101	451571.5	3 159	173 744	55
7653	40	41111	105590.72	72 386	361 928	5
7663	39	41111	241350.21	12 530	162 894	13
7676	3	88110	130855989	9 007 746	36 030 982	4
7683	43	71222	234737.75	2 457	17 196	7
7684	36	11400	15329.27	5 441	5 441	1
7686	38	92002	146290.4	1 149	257 335	224
7688	38	93300	128473.8	6 808	428 895	63
7697	40	41200	1514811.64	17 957	2 298 537	128
7703	41	41111	530908.5	2 193	245 568	112
7705	39	91109	2418326.22	8 374	1 247 788	149
7709	38	84130	654089.31	9 507	38 028	4
7720	38	63319	106498.61	1 145	143 101	125
7725	20	13100	16074.73	1 149	183 761	160
7818	20	61430	13712627.7	1 493 643	17 923 718	12
7824	2	88140	4081292.57	157 231	1 100 619	7
7834	3	42000	611660.95	8 715	217 867	25
7838	3	81990	138513.32	4 105	49 258	12
7839	38	61430	90925.33	1 146	581 028	507

Co_ID	Location_Priority	SIC_Code	Telecom Subscribers	Telecom SolutionLines	Telecom ProductLines	Telecom DeviceLines
185239	1	81121	79	12	12	1
185239	3	81121	253	24	24	2
185239	18	81121	43	12	12	1
185239	22	81121	146	12	12	1
185239	38	81121	15	12	12	1
185239	39	81121	236	36	36	3
185239	43	81121	146	12	12	1
188786	40	81110	4			
189002	20	50219	778	369	329	2
189053	20	50220	449	261	260	11
189358	38	88920	11	53	53	4
189389	40	81121	293	263	257	3
189533	38	63121	14	44	44	2
190387	38	30521	81	81	51	13
190646	43	82130	4	3	3	1
191687	8	11600	17	11	11	1
192274	38	93300	36	34	34	
199277	2	82190	4			
201338	12	82110	649	3	3	1
205787	3	62110	4			
205788	1	62110	1			
205788	14	62110	1			
205789	38	62520	144	24	23	2
205980	24	62110	125	102	102	
208578	40	88312	73	59	56	1
211471	3	23000	8	17	17	
212598	40	33420	47	327	150	13
219118	3	32420	36	3	3	
219342	3	82190	1			
219349	2	83110	9	28	22	
219349	36	83110	9	28	22	
219351	3	81990	3	3	3	2
219554	12	81110	338	275	231	3
219627	1	81110	79	720	2315	79
221216	38	30312	4	3	3	
221242	32	74140	4	3	3	
221789	20	61410	10			
222725	41	74190	12			
224779	26	71211	4	3	3	
224779	43	71211	6	3	3	
225677	39	86200	5			
225751	41	33542	100	29	21	1
227245	39	75200	17	8	13	1
228078	43	74139	14	83	83	1
229288	20	11210	33	3	3	

A.1.2 PSO Input data

Below an extract of the data used for the PSO test runs.

Table A.3: PSO target data extract (Prospects/Customers combined)

Co_ID	Location Priority	SIC Code	Co Employees	Co Turnover	ICT Spend	Telecom ARPU	Telecom Revenue	Telecom Subscribers	Telecom SolutionLines	Telecom ProductLines	Telecom DeviceLines
2813	13	64202	22	50232546.8	9 041 878	5344258.88	21377035.53	4			
2817	43	93300	420	1094948.03	15 329	39418.56	236511.34	6			
2820	43	34240	341	3538136.82	63 686	1280.93	175487.76	137	1087	686	59
2824	13	88993	1142	4530325.41	217 456	77803.82	1244861.16	16			
2851	43	81990	420	4598781.73	289 723	78221.06	469326.36	6			
2864	38	30120	97	267406.92	3 476	70133.58	70133.58	1			
2867	38	50220	473	4657159.53	200 258	78221.06	547547.4	7			
2876	38	74139	230	3384844.09	91 391	37436.6	37436.6	1			
2879	43	71222	15	4333620	216 681						
2883	2	30330	219	36350198.48	472 553	126757.4	380272.21	3			
2928	38	30491	15	623303281.9	8 102 943	207545.72	2283002.97	11	48	35	1
2929	13	88993	9	98545.32	4 730	2332.2	4664.4	2			
2947	20	39102	200	66782131.62	1 268 861						
2956	38	71222	49	1194948.03	59 747	5146.81	5146.81	1			
2957	41	61221	46	657234634.6	9 858 520	2914907.04	8744721.13	3			
3004	20	35101	50	5621921.94	106 817	23029.53	23029.53	1			
3006	38	32430	332	13125188.84	682 510	109022.23	5233067.04	48	364	250	7
3028	20	50230	25	49627365.52	496 274	35121.26	175606.32	5			
3029	43	33100	293	1844674	33 204	1142.1	150757.56	132	3	3	
3058	1	86200	120	548928.69	38 425	95438.22	190876.44	2			
3107	43	71239	154	32322038.79	872 695						
3119	41	92007	114	5349705.52	251 436						
3197	43	71222	14	4385210.72	219 261						
3251	16	74140	27	1094948.03	29 564	14633.86	175606.32	12	34	34	
3262	39	30312	97	4178593.81	54 322	95438.16	95438.16	1			
3266	38	63400	100	22405584.65	403 301	122955.44	245910.88	2			
3309	12	50230	1254	10397383.58	103 974	35121.25	3371640.48	96			
3399	38	33520	140	23408170.7	280 898	95438.22	190876.44	2			
3417	39	33542	50	854423639	16 234 049	3999698.82	11999096.45	3			
3430	3	33230	52	2131732.44	38 371	6462.32	51698.52	8	6	6	1
3444	1	62330	4858	17478053.38	262 171	54854.73	3839830.8	70			
3523	43	93300	47	27334851.06	382 688	136126.57	1225139.1	9			
3547	20	50220	473	4657159.53	200 258						
3556	36	50220	12	183568705	7 893 497						
3562	37	41200	229	174287903.7	4 880 061	1936053.56	3872107.11	2			
3623	39	88999	300	29979782.06	1 289 131	78221.07	312884.28	4			
3635	22	71239	368	13526070.58	365 204	1147.84	189393	165	3	3	
3642	9	61909	109	9241501.73	138 623	78221.04	156442.08	2			
3681	2	83110	1762	8886920	559 876	78221.06	1955526.48	25			
3691	40	61501	20	192976912.2	2 894 654	176706.46	706825.84	4			
3726	39	63319	278	5916589.21	106 499	1144.81	143100.84	125			
3735	8	11600	193	960000	13 440	384.12	17669.29	46	3	38	1
3782	38	85290	44	905814.84	9 058	4179.31	4179.31	1			
3789	43	81121	3508	6498226.17	409 388	1146.23	1805306.4	1575	16	16	
3809	43	88999	2271	8494941.19	365 282	70677.18	2332347.07	33			
3863	39	37300	207	86681862.24	2 080 365	319792.71	959378.12	3			
8578	37	81110	73	4071065.86	256 477	63822.37	1340269.69	21	3	3	171
8924	13	83200	3747	5351455.28	171 247	1146.46	1928341.44	1682	5	5	1
8930	38	36200	84	14388046.19	345 313	2374.37	111595.57	47	40	36	5
8931	38	36400	169	64252183.78	1 542 052	967134.31	967134.31	1			
9120	20	96130	153	81725860.64	4 249 745	293129.5	586259.01	2			
9125	46	64101	45	24485210.47	1 224 261	127354.81	1146193.33	9	3	3	1
9299	3	36100	67	733615.18	17 607	1379.97	34499.16	25	11	7	
9369	2	88993	1142	4530325.41	217 456	77803.82	1244861.16	16	1	1	
9434	39	86900	543	51906216.54	2 802 936	4079.16	995315.67	244	8	8	
9463	2	30530	189	8680143.89	112 842	12194.14	2060808.94	169	2399	4152	186
9490	3	86900	543	51906216.54	2 802 936	4079.16	995315.67	244	60	57	2
9511	3	91109	583	120916310.9	2 418 326	15907.99	1256731.57	79	59	40	4
9517	13	93300	219	3284844.09	45 988	1640.35	167316.12	102	816	3	49
9521	39	81990	138	337300261.2	21 249 916	60436.18	6466670.79	107	1015	1015	187
9525	3	83190	120	7894755.82	497 370	1404.13	61781.52	44	43	43	1
9558	43	64101	34	32406896.21	1 620 345	16333.84	114336.86	7	3	3	
9563	2	75200	3226	6985078.78	265 433	1146.53	1660181.28	1448	18	18	
9672	33	64101	235	2773127.87	138 656	653.64	120922.92	185	136	85	4
9686	3	74110	306	3950540.97	106 665	1149.24	157446.12	137	3	3	
10732	1	72112	15	1157266902	31 246 206	2181.02	32715.24	15	10	4	1
10815	3	83200	65	3604418.03	115 341	1672.15	33443.04	20	127	87	1
11951	1	75200	525	417770000	15 875 260	33957.01	8013853.92	236	77	69	1
12547	38	81121	527	413594629.4	26 056 462	58793.64	7937140.96	135	762	682	99
12547	40	81110	527	2323055031	146 352 467	190218.55	16929450.58	89	762	682	99
13224	38	73000	750	2765457798	74 667 361	27372341.73	301095759.1	11			
15333	10	36200	114	429004747.1	10 296 114	5633368.1	197167883.5	35	216	138	11
15662	38	61394	1000	6707003.12	207 917	78221.06	1095094.8	14			
16411	38	32321	90	6568713.89	118 237	1157.3	46292.16	40	8	6	1
19762	40	81121	392	817172.64	51 482	3361.9	201714.12	60	523	494	15
19784	39	81110	480	212363119.6	13 378 877	12987.14	4077961.13	314	3410	3121	171
19974	40	30530	1108	6125501.35	79 632	1147.29	570203.28	497	49	27	7
19974	43	61222	800	2840657.26	42 610	1146.8	411701.04	359	49	27	7
20015	1	81110	129	66966286.92	4 218 876	160603.86	1284830.9	8	8	8	1
20015	2	81110	374	194150320.2	12 231 470	312592.96	3721579.17	24	23	23	4
20015	3	81110	295	153139958.5	9 647 817	153899.89	2924097.92	19	18	18	3
20015	8	81110	124	64370694.41	4 055 354	155065.8	1240526.39	8	8	8	1
20015	13	81110	155	80463368.01	5 069 192	155065.8	1550657.99	10	10	10	2
20015	16	81110	97	50354494.82	3 172 333	162449.88	974699.31	6	6	6	1
20015	32	81110	97	365030355	22 996 912	5864481.67	340139936.8	58	569	569	102
20015	38	81110	243	126145796.3	7 947 185	330773.47	2436748.27	15	15	15	3
20015	39	81110	280	145353180.9	9 157 250	307405.17	2791184.38	18	18	18	3
20015	43	81110	159	82539842.02	5 200 010	157281.02	1572810.25	10	10	10	2
20072	2	81110	156	12556691.76	791 072	15769.44	173463.84	11	2	3	1
20072	24	81110	85	44125072.78	2 779 880	22152.26	841785.77	38	2	3	1
20072	43	81110	85	44125072.78	2 779 880	22152.26	841785.77	38	2	3	1
20284	12	83200	120	8073045.02	258 337	980.66	61781.52	63	302	576	7

A.1.3 CHAID Input data

Below an extract of the data used per target flag for the CHAID test runs.

Table A.4: CHAID target data extract for Prospects and Customers

Co_ID	Classification	Region	Industry	Target Flag	Telecom MainSolution	Telecom MainProduct	Telecom DeviceType	Co Employees	ICT Spend	Telecom ARPU	Telecom Subscribers
918	Medium	Dar Es Salaam	Tourism	Prospect				120	185 082		
919	Large	Dar Es Salaam	Financial	Prospect				3747	92 605		
920	Large	Dar Es Salaam	Business Services	Prospect				1142	105 079		
921	Very Small	Dar Es Salaam	Construction	Prospect				14	3 384 521		
922	Large	Dar Es Salaam	Transport	Prospect				205	2 293 161		
923	Large	Dar Es Salaam	Manufacturing	Prospect				3215	89 392		
924	Medium	Dar Es Salaam	Education	Prospect				77	124 786		
925	Large	Dar Es Salaam	Business Services	Prospect				1142	105 079		
926	Small	Dar Es Salaam	Manufacturing	Prospect				25	27 785		
927	Large	Dar Es Salaam	Business Services	Prospect				258	130 855 989		
928	Large	Dar Es Salaam	Business Services	Prospect				461	4 081 293		
929	Large	Dar Es Salaam	ICT	Prospect				110	2 718 982		
930	Large	Dar Es Salaam	Retail, Wholesale, Trade	Prospect				109	138 623		
931	Large	Dar Es Salaam	ICT	Customer	Mobile	TOP-UP	Mobile Handset	783	1 534 824	945.96	87
932	Very Small	Dar Es Salaam	Business Services	Prospect				11	171 043 261		
933	Large	Dar Es Salaam	ICT	Prospect				543	2 802 936		
934	Large	Dar Es Salaam	ICT	Prospect				543	2 802 936		
935	Large	Dar Es Salaam	Business Services	Prospect				461	4 081 293		
936	Small	Dar Es Salaam	Business Services	Prospect				30	2 465 933		
937	Small	Dar Es Salaam	Business Services	Prospect				30	62 715 862		
938	Large	Dar Es Salaam	Manufacturing	Prospect				390	52 444		
939	Medium	North	Manufacturing	Prospect				132	159 277		
940	Medium	Dar Es Salaam	Education	Prospect				105	154 388		
1533	Large	Dar Es Salaam	Community & Personal Services	Customer	MPESA	MPESA C2B	M2M	292	905 656	461166.2	3
1833	Large	Dar Es Salaam	Community & Personal Services	Customer	MPESA	TOP-UP	Tablet	223	31 237	1937.68	63
2224	Large	Dar Es Salaam	Transport	Customer	IOT	M2M	Mobile Handset	1157	18 573 159	5522.24	1087
2279	Large	Dar Es Salaam	Utilities	Customer	IOT	M2M	M2M	5882	93 926	530.05	5711
2310	Large	Lake District	Community & Personal Services	Customer	IOT	M2M	M2M	3600	93 899	299	3416
3735	Large	North	Agriculture	Customer	MPESA	MOBILE INTEGRATED-HIGH	Mobile Handset	193	13 440	384.12	46
5493	Large	Dar Es Salaam	ICT	Customer	IOT	M2M	M2M	110	34 925 314	79899.65	45
7851	Very Small	North	Transport	Customer	IOT	M2M	M2M	14	1 422 489	4948549	7
8578	Large	Dar Es Salaam	Financial	Customer	MPESA	MPESA C2B	M2M	73	256 477	63822.37	21
20015	Large	Central	Financial	Customer	IOT	M2M	M2M	278	9 091 841	153835.1	18
20015	Large	Coast	Financial	Customer	IOT	M2M	M2M	97	3 172 333	162449.9	6
20015	Large	Dar Es Salaam	Financial	Customer	IOT	M2M	M2M	1435	66 755 487	7121019	143
20015	Large	North	Financial	Customer	IOT	M2M	M2M	68	2 223 904	171680	4
20015	Large	South West	Financial	Customer	IOT	M2M	M2M	75	2 452 835	150635.4	5
22162	Large	Central	Financial	Customer	MPESA	MPESA C2B	M2M	28	915 725	24009.89	13
22162	Large	Dar Es Salaam	Financial	Customer	MPESA	MPESA C2B	M2M	182	5 952 213	259843.9	81
22162	Large	Lake District	Financial	Customer	MPESA	MPESA C2B	M2M	30	981 134	24009.89	13
22162	Large	North	Financial	Customer	MPESA	MPESA C2B	M2M	16	523 271	44589.8	7
23978	Large	Dar Es Salaam	Financial	Customer	IOT	M2M	M2M	3503	1 631 062	1450.33	1243
28894	Large	North	Agriculture	Customer	MPESA	MPESA C2B	Mobile Handset	258	60 025	1144.86	116
33172	Large	Dar Es Salaam	Retail, Wholesale, Trade	Customer	IOT	M2M	Mobile Handset	166	33 689	1139.41	75
98169	Large	Dar Es Salaam	Manufacturing	Customer	IOT	M2M	M2M	86	5 825 433	3836514	41
121575	Large	Dar Es Salaam	Transport	Customer	Fixed	DEDICATED INTERNET-BIA	USB Modem	418	1 124 636	15543.97	38
179167	Large	Dar Es Salaam	Manufacturing	Customer	MPESA	MPESA C2B	Mobile Handset	165	4 169	3033.13	28
183510	Large	Central	Financial	Customer	IOT	TOP-UP	M2M	419	13 703 171	63979.99	396
183510	Large	Coast	Financial	Customer	IOT	TOP-UP	M2M	239	7 816 367	42028.08	228
183510	Large	Dar Es Salaam	Financial	Customer	IOT	TOP-UP	M2M	1462	47 813 929	177732.4	1393
183510	Large	Lake District	Financial	Customer	IOT	TOP-UP	M2M	410	13 408 831	52174.78	392
183510	Large	North	Financial	Customer	IOT	TOP-UP	M2M	342	11 184 927	63132.63	325
183510	Large	South West	Financial	Customer	IOT	TOP-UP	M2M	234	7 652 845	42228.24	223
185239	Large	Dar Es Salaam	Financial	Customer	Fixed	DATA MPLS/DATA VPN-LOW	Mobile Handset	2010	65 735 976	177417.7	903
185239	Large	North	Financial	Customer	Fixed	DATA MPLS/DATA VPN-LOW	Mobile Handset	105	669 908	20623.61	15

A.1.4 ANN Input data

Below an extract of the data used per target flag for the CHAID test runs.

Table A.5: ANN Prospect/Customer data for Training and Validation

ID	Priority	SIC Code	Co Employees	Co Turnover	ICT Spend	ClassID
50	39	36600	3215	3724678.6	89392.29	4
61	22	88920	4785	1048652.6	50335.33	2
150	43	32420	1489	5292743.2	275222.64	4
211	20	33800	150	19364712	348564.81	3
350	17	62519	204	5128439	76926.58	3
361	17	63122	77	9257507.1	166635.13	3
400	2	81990	689	67557962	4256151.63	2
461	26	71222	120	3701633.8	185081.69	1
500	21	74140	219	465857.01	12578.14	2
511	8	72112	41	503337641	13590116.32	2
561	17	88993	1142	2189141.9	105078.81	2
650	39	63319	278	5916589.2	106498.61	1
700	16	34250	79	95393982	1717091.67	3
761	20	50310	78	3284844.1	32848.44	3
911	38	74139	10	1.741E+09	47009802.19	1
950	38	88920	4785	1048652.6	50335.33	1
1100	20	33800	173	138319371	2489748.67	3
1200	43	92002	456	528290.34	24829.65	1
1250	2	33520	596	5498563.6	65982.76	3
1311	43	62330	55	164197221	2462958.31	1
1411	2	35690	4149	4806747	86521.45	3
1500	39	88993	1142	2189141.9	105078.81	1
1511	3	88930	32	1094948	19709.06	2
1600	9	30499	1888	2187307.4	28435	3
1750	8	71222	72	178854.89	8942.74	2
2061	39	71222	120	3701633.8	185081.69	1
2100	38	64101	18	61213026	3060651.31	1
2200	7	92009	320	3785384.4	177913.07	2
2250	20	34240	293	3408197.7	61347.56	3
2261	43	71222	66	100000	5000	1
2400	4	50240	27	379548.15	3795.48	3
2461	15	50220	130	16944896	728630.52	3
2561	46	88993	1142	2189141.9	105078.81	1
2600	9	39219	20	1.725E+09	32779880.15	3
2611	38	50219	340	28780322	287803.22	4
2661	3	88110	258	3.043E+09	130855989.2	2
2711	12	81990	689	67557962	4256151.63	2
2750	38	35522	146	3607817.7	155136.16	4
4861	39	50219	54	181209437	1812094.37	4
5761	38	64103	107	55545680	2777283.99	1
555	43	35419	132	8383013.5	159277.26	4
803	38	63122	86	1094948	19709.06	1
1051	33	50230	123	10086863	100868.63	4
1203	39	92002	27	7571740.7	355871.81	1
1451	36	91101	253	9703548	194070.96	1
1555	39	63319	278	5916589.2	106498.61	1
1603	38	35690	4149	5447197.4	98049.55	4
1651	16	88211	41	416579599	17912922.77	2
1755	43	50220	18	122379803	5262331.54	4
2051	41	33541	125	133083481	2528586.14	4
2155	1	75200	13	320328649	12172488.65	2
2555	39	71222	120	3701633.8	185081.69	1
2603	38	33592	1199	4411551.9	52938.62	4
3955	20	35102	283	3198702.9	60775.36	3
3955	20	35101	50	5621921.9	106816.52	3

Prospect Infographics

ID	Priority	SIC Code	Telecom ARPU	Telecom Revenue	Telecom Subscribers	Telecom SolutionLines	Telecom ProductLines	Telecom DeviceLines	ClassID
13	38	31290	1146.19	4117098.36	3592	3	3		2
18	40	82130	1141.59	119866.8	105	77	91	11	1
19	39	62393	1145.98	1879409.04	1640	1	1		1
37	43	61430	14182.5	340379.89	24	166	110	9	1
54	2	81110	35121.24	35121.24	1				3
66	38	82190	126715.32	380145.96	3	3	3		1
67	40	82190	412408.1	1237224.31	3	3	3		1
77	39	93111	5358.15	155386.28	29	27	24		1
78	38	92007	5358.15	155386.28	29	27	24		1
79	38	96130	12058.99	410005.56	34	34	34		1
100	1	83110	2781.05	27810.48	10	41	26	5	3
108	3	85220	5659.9	118857.81	21	3	3		3
120	38	13200	135954.1	271908.2	2				2
134	41	38200	83401.66	333606.65	4				2
264	20	74139	770067.62	18481622.88	24	172	155	5	3
280	38	61394	33845.22	67690.45	2				1
345	39	62393	878772.57	3515090.28	4	2	2		1
369	8	63400	935982.15	6551875.03	7	6	6		3
448	38	50230	104103.02	1561545.36	15	11	11	1	2
449	41	62519	671805.49	1343610.98	2	1	1		1
498	14	32520	1411.39	38107.44	27	10	10		4
503	43	50220	17537.78	210453.4	12	11	11	1	2
511	8	72112	523874.22	9429735.91	18	6	6	1	3
512	8	71229	1161.14	35995.2	31	6	6	1	3
523	43	62519	19198.43	38396.86	2				1
530	10	41300	314455.96	628911.91	2	2	2		4
571	38	88993	1145.65	587716.92	513	1	1		1
663	39	93300	1401.09	35027.16	25	3	3		1
684	38	63121	194989.37	2534861.87	13	11	11	1	1
690	26	71222	246775.6	740326.8	3				1
708	48	96330	37033.48	3221912.78	87	73	73	3	1
784	37	91109	363897.76	17830990.3	49	219	481	1	1
833	40	41111	619366.89	619366.89	1	1	1		2
846	38	95120	1928.29	129195.6	67	326	301	3	1
883	39	32420	1145.42	766284.96	669	3	3		2
894	38	64101	39145.09	587176.32	15	14	6	1	1
931	36	86900	945.96	82298.4	87	73	70	4	1
960	38	93192	136461.6	409384.8	3	3	3		1
964	13	96130	1301113.95	14312253.44	11	57	36	1	3
1012	2	93199	12744.58	3530249.88	277	2480	2281	61	3
1037	36	30530	571.51	23431.8	41	27	25	1	2
1043	2	61222	953306.29	4766531.47	5				3
1058	38	36600	12878.09	12878.09	1				2
1071	43	61410	1148.99	103409.28	90	14	14	1	1
1072	14	61394	47872.42	1340427.75	28	21	13	1	4
1100	20	33800	30024.17	1501208.62	50	358	197	3	4
1107	39	88220	903.87	33443.04	37	31	31		1
1158	39	50230	1406.17	63277.68	45	9	9	1	2
1168	43	11220	4682.83	70242.48	15	11	11	1	2
1173	8	83190	1404.13	61781.52	44	1	1		3
1177	43	50230	1406.17	63277.68	45				2
1185	38	34111	1145.13	295442.52	258	212	212	1	2
1187	38	62340	55125.94	937141.04	17				1
1201	20	32210	8826.64	441331.99	50	400	338	3	4
1202	20	33100	13321.81	146539.94	11	203	109	4	4

Customer Subscriber Totals

A.2 Input variables per analysis

Below is a list of the variables used to analyse each method.

Table A.6: Input variables for segmentation methods

Variable	Category	Description	KMC	PSO	CHAID	ANN
Subsidiary Flag	**Reference**	Indication if company belongs to a holding company				
Target Flag	**Nominal**	Specifying if company is a customer or a potential customer (prospect)			☑	
Classification	**Nominal**	Pre-classified company size as Large, Medium, Small, etc.			☑	
Area	**Reference**	Suburb or city or district within a region				
Region	**Nominal**	Regions where target market companies reside			☑	
Landmark	**Reference**	A geographic landmark or feature close to the company location, e.g. Bus stop, Street, etc.				
Location_Priority	**Ordinal**	Ranked priority of the landmark or feature closest to the company location, in terms of ease to find	☑	☑		☑
SIC_Code	**Ordinal**	Standard Industry Code defining the industry sector of the company, e.g. Manufacturing and related activities	☑	☑		☑
Industry	**Nominal**	The main industry sector as per high level SIC, e.g. Manufacturing			☑	
Sub_Industry	**Reference**	The sub industry as per lower level SIC, e.g. Textile manufacturing				
Co_Employees	**Numerical**	Number of employees of company	☑	☑	☑	☑
Co_Turnover	**Numerical**	Annual turnover in US Dollars	☑	☑		☑
ICT_Spend	**Numerical**	Amount spend on Information Technology and Communications in US Dollars	☑	☑	☑	☑
Perc_ICTCountry	**Ratio**	ICT Spend as a percentage of the country ICT spend	☑			
Perc_ICTUsgPopulation	**Ratio**	Number of employees as percentage of total population in the same industry using ICT	☑			
ICT_Usage_Population	**Reference**	Input value for above, population ICT users per Industry				
Telecom_ARPU	**Numerical**	Telecom provider's average revenue per unit (subscriber)	☑	☑	☑	☑
Telecom_Revenue	**Numerical**	Telecom provider's revenue from the company	☑	☑	☑	☑
Telecom_Subscribers	**Numerical**	Number of subscribers signed up to the telecoms provider	☑	☑	☑	☑
Telecom_SolutionLines	**Numerical**	Number of connections for the main solution provided by the telecoms company	☑	☑		☑
Telecom_ProductLines	**Numerical**	Number of connections for the major product class provided by the telecoms company	☑	☑		☑
Telecom_DeviceLines	**Numerical**	Number of connections on the most used device provided by the telecoms company	☑	☑		☑
Telecom_MainSolution	**Nominal**	The main solution provided to the company by the telecoms provider, e.g. mobile, fixed, IOT			☑	
Telecom_MainProduct	**Nominal**	The main product class provided to the company by the telecoms provider, e.g. Mobile voice, data, etc.			☑	
Telecom_DeviceType	**Nominal**	The type of most used device provided to the company by the telecoms provider, e.g. handset, router, tablet etc.			☑	
Telecom_DeviceManufacturer	**Reference**	The manufacturer of the most used device provided to the company by the telecoms provider, e.g. Apple, Samsung, Blackberry, Nokia, etc.				

A.2.1 KMC Input variables

Below a summary of input variables used for each KMC test run.

Table A.7: Input variables for KMC tests

Variable	KMC1	KMC2	KMC3	KMC4	KMC5
Location priority	☑	☐	☑	☐	☑
SIC code	☑	☐	☑	☐	☑
Company employee count	☑	☐	☐	☐	☑
Company turnover	☑	☐	☐	☐	☑
ICT spend	☑	☑	☐	☑	☑
ICT as % of country ICT	☐	☐	☐	☑	☑
ICT as % of usage population	☐	☑	☐	☑	☑
Telecom ARPU	☐	☑	☑	☑	☑
Telecom revenue	☐	☑	☑	☑	☑
Telecom subscribers	☐	☑	☑	☑	☑
Telecom solution line count	☐	☐	☑	☑	☑
Telecom product line count	☐	☐	☑	☑	☑
Telecom device line count	☐	☐	☑	☑	☑

A.2.2 PSO Input variables

Below a summary of input variables used for each PSO test run.

Table A.8: Input variables for PSO clustering tests

Variable	PSO1	PSO2
Location priority	☑	☑
SIC code	☑	☑
Company employee count	☑	☐
Company turnover	☑	☐
ICT spend	☑	☐
Telecom ARPU	☐	☑
Telecom revenue	☐	☑
Telecom subscribers	☐	☑
Telecom solution line count	☐	☑
Telecom product line count	☐	☑
Telecom device line count	☐	☑

A.2.3 CHAID Input variables

Below a summary of input and dependent variables used for each CHAID test run.

Table A.9: Predictor and dependent variables for CHAID tests

Variable	CHAID1	CHAID2	CHAID3	CHAID4	Predictor	Dependent
Target Flag	☑	☐	☐	☑	☐	☑
Classification	☑	☑	☐	☐	☐	☑
Region	☑	☑	☐	☑	☑	☐
Industry	☑	☐	☑	☐	☑	☐
Company employee count	☐	☑	☐	☑	☑	☐
ICT spend	☑	☐	☐	☑	☐	☑
Telecom ARPU	☐	☑	☐	☐	☑	☐
Telecom revenue	☐	☐	☑	☐	☐	☑
Telecom subscribers	☐	☐	☑	☐	☑	☐
Telecom main solution	☐	☑	☐	☐	☑	☐
Telecom main product	☐	☐	☑	☐	☑	☐
Telecom device type	☐	☐	☑	☐	☑	☐

A.2.4 ANN Input variables

Below a summary of input variables used for each PSO test run.

Table A.10: Input variables for ANN training tests

Variable	ANN1 Training	ANN2 Training	ANN1 Validation
Location priority	☑	☑	☑
SIC code	☑	☑	☑
Company employee count	☑	☐	☑
Company turnover	☑	☐	☑
ICT spend	☑	☐	☑
Telecom ARPU	☐	☑	☐
Telecom revenue	☐	☑	☐
Telecom subscribers	☐	☑	☐
Telecom solution line count	☐	☑	☐
Telecom product line count	☐	☑	☐
Telecom device line count	☐	☑	☐

A.3 Reference tables

A.3.1 Target reference file

Below an extract to serve as sample of a reference file created separately from the analysis file for information purposes only.

Table A.11: Sample extract of target reference file

Co_ID	Group_ID	DataSource_Flag	Record Created Date	Area	Region	Addr_Feature	Industry	Co Employees Source	Co Turnover Source
1		Secondary Data	2017/12/15	Ubungo	Dar Es Salaam	Sam Nujoma_Road	Business Services	FACTIVA	Estimated
2		Secondary Data	2017/12/20	Kisutu	Dar Es Salaam	Mkwepu_Street	Retail, Wholesale, Trade	FACTIVA	Estimated
9		Secondary Data	2017/12/14	Regent Estate	Dar Es Salaam	Mwai Kibaki_Road	Retail, Wholesale, Trade	FACTIVA	Estimated
11		Secondary Data	2017/12/15	Ilala	Dar Es Salaam	Zanaki/Makunganya_Street	Tourism	FACTIVA	Estimated
13	100	Secondary Data	2018/03/29	Arusha	North	Dodoma_Road	Manufacturing	FACTIVA	Estimated
14		Secondary Data	2018/03/29	Chang'Ombe	Dar Es Salaam	Chang'Ombe_Suburb	Manufacturing	FACTIVA	Estimated
41		Secondary Data	2017/12/07	Arusha	North	Engira_Road	Tourism	FACTIVA	Estimated
43		Secondary Data	2017/12/18	Kivukoni	Dar Es Salaam	Azikiwe_Street	Business Services	FACTIVA	Estimated
44	106	Secondary Data	2018/03/29	Kivukoni	Dar Es Salaam	Ohio_Street	Transport	FACTIVA	Estimated
51		Secondary Data	2018/03/29	Arusha	North	Unga Limited Area_Suburb	Manufacturing	FACTIVA	Estimated
58		Secondary Data	2018/03/29	Kipawa	Dar Es Salaam	Pugu_Road	Transport	FACTIVA	Estimated
66	31027	Secondary Data	2017/12/05	Iringa	Central	Kawawa_Road	Financial	RASELLO	Estimated
68	31027	Secondary Data	2017/12/05	Iringa	Central	Kawawa_Road	Financial	FACTIVA	Estimated
70		Secondary Data	2018/03/29	Tanga	Coast	Tanganyika_District	Manufacturing	FACTIVA	Estimated
77	113	Secondary Data	2017/12/05	Upanga	Dar Es Salaam	Moski_Street	Health	FACTIVA	Estimated
80		Secondary Data	2017/12/20	Kurasini	Dar Es Salaam	Kurasini_Suburb	Manufacturing	FACTIVA	Estimated
82		Secondary Data	2017/12/20	Ubungo	Dar Es Salaam	Sam Nujoma_Road	Retail, Wholesale, Trade	FACTIVA	Estimated
96		Secondary Data	2018/03/29	Kipawa	Dar Es Salaam	Julius K Nyerere_Road	Construction	FACTIVA	Estimated
116		Secondary Data	2018/03/29	Arusha	North	Wapare_Street	Manufacturing	FACTIVA	Estimated
119		Secondary Data	2018/03/29	Mwanza	Lake District	Mwanza_Suburb	Retail, Wholesale, Trade	FACTIVA	Estimated
120		Secondary Data	2018/03/29	Kipawa	Dar Es Salaam	Julius K Nyerere_Road	Agriculture	FACTIVA	Estimated
121		Secondary Data	2018/03/29	Kinondoni	Dar Es Salaam	Ali Hassan Mwinyi_Road	Transport	FACTIVA	Estimated
129	212	Secondary Data	2018/03/29	Kigamboni	Dar Es Salaam	Kigamboni_Suburb	Tourism	FACTIVA	Estimated
131		Secondary Data	2017/12/06	Morogoro	Central	Selous Game_Reserve	Tourism	FACTIVA	Estimated
142		Secondary Data	2017/12/12	Kariakoo	Dar Es Salaam	Kariakoo Kipata_Street	Manufacturing	FACTIVA	FACTIVA
143		Secondary Data	2017/12/05	Kariakoo	Dar Es Salaam	Kipata_Street	Manufacturing	FACTIVA	FACTIVA
144		Secondary Data	2017/12/19	Upanga	Dar Es Salaam	Upanga_Suburb	Community & Personal Services	FACTIVA	Estimated
149		Secondary Data	2017/12/06	Kipawa	Dar Es Salaam	Kiwalani _Industrial Area	Retail, Wholesale, Trade	FACTIVA	Estimated
154		Secondary Data	2018/03/29	Sinza	Dar Es Salaam	Sinza Shekilango_Road	Transport	FACTIVA	Estimated
169		Secondary Data	2017/12/22	Arusha	North	Themi_Industrial Area	Manufacturing	FACTIVA	Estimated
178		Secondary Data	2018/03/29	Arusha	North	Arusha_Suburb	Business Services	FACTIVA	Estimated
188		Secondary Data	2018/03/29	Arusha	North	Arusha_Suburb	Tourism	FACTIVA	Estimated
200		Secondary Data	2017/12/22	Kisutu	Dar Es Salaam	Samora_Avenue	Tourism	FACTIVA	Estimated
211	120	Secondary Data	2018/03/29	Kipawa	Dar Es Salaam	Nyerere Road_Industrial Area	Manufacturing	FACTIVA	Estimated
213		Secondary Data	2017/12/14	Kawe	Dar Es Salaam	Mlalakuwa_Bridge	Tourism	FACTIVA	Estimated
218		Secondary Data	2018/03/29	Mabibo	Dar Es Salaam	Nelson Mandela_Road	Construction	FACTIVA	Estimated
221		Secondary Data	2018/03/29	Buguruni	Dar Es Salaam	Nelson Mandela/Alhamza_Road/Street	Retail, Wholesale, Trade	FACTIVA	Estimated
222		Secondary Data	2017/12/20	Mwanza	Lake District	Mwanza_Suburb	Retail, Wholesale, Trade	FACTIVA	Estimated
232	157	Secondary Data	2018/03/29	Lindi	Coast	Liwale_Town	Tourism	FACTIVA	Estimated
237	36767	Secondary Data	2017/12/05	Kariakoo	Dar Es Salaam	Kariakoo_Suburb	Business Services	Estimated	Estimated
238		Secondary Data	2017/12/13	Kipawa	Dar Es Salaam	Julius K Nyerere_Road	Retail, Wholesale, Trade	FACTIVA	Estimated
241		Secondary Data	2017/12/05	Arusha	North	Njiro_Road	Business Services	FACTIVA	Estimated
243		Secondary Data	2017/12/19	Mikocheni	Dar Es Salaam	Mikocheni Light_Industrial Area	Construction	FACTIVA	Estimated
262		Secondary Data	2017/12/05	Kivukoni	Dar Es Salaam	IT_Plaza	Retail, Wholesale, Trade	FACTIVA	Estimated
264		Secondary Data	2018/03/29	Keko	Dar Es Salaam	Keko Mwanga_Industrial Area	Transport	FACTIVA	Estimated
265		Secondary Data	2017/12/06	Keko	Dar Es Salaam	Keko Mwanga_Industrial Area	Transport	FACTIVA	Estimated
269	36099	Secondary Data	2017/12/18	Mikocheni	Dar Es Salaam	Mickocheni Light_Industrial Area	Manufacturing	FACTIVA	Estimated
288	128	Secondary Data	2018/03/29	Ukonga	Dar Es Salaam	Julius Nyerere International_Airport	Transport	FACTIVA	Estimated
295		Secondary Data	2018/03/29	Arusha	North	Sombetini_Suburb	Transport	FACTIVA	Estimated

A.3.3 Location priorities

One of the analysis variables, Location priority, was used in the analysis. Please find below the list of priorities per landmark.

Table A.12: Location priority per landmark

Landmark	Location Priority	Landmark	Location Priority
Tower	1	Railway Station	25
Building	2	Airport	26
House	3	Embassy	27
Business Park	4	Cemetery	28
Business Centre	5	Stadium	29
Office Complex	6	Hospital	30
Bank	7	School	31
Hotel	8	Worship Place	32
Apartment	9	Beach	33
Complex	10	Farm	34
Hall	11	Lane	35
Place	12	Avenue	36
Plaza	13	Drive	37
Mall	14	Road	38
Shopping Centre	15	Street	39
Centre	16	Road/Street	40
Roundabout	17	Map Area	41
Market	18	Bridge	42
Estate	19	Suburb	43
Industrial Area	20	Town	44
Petrol Station	21	Region	45
Bus Stop	22	District	46
Post Office	23	Reserve	47
Police Station	24	National Park	48

A.3.3 SIC economic divisions

The major divisions (one-digit level) and divisions (two-digit level) of the SIC are shown in Table A.13. The source is the South African Reserve Bank Institutional Sector Classification Guide (2011).

Table A.13: Standard Industrial Classification of Economic Activities

Category	Major division	Division
Agriculture, hunting, forestry and fishing	**1**	
Agriculture, hunting and related services		11
Forestry, logging and related services		12
Fishing, operation of fish hatcheries and fish farms		13
Mining and quarrying	**2**	
Mining of coal and lignite		21
Extraction of crude petroleum and natural gas; service activities incidental to oil and gas extraction, excluding surveying		22
oil and gas extraction, excluding surveying		22
Mining of gold and uranium ore		23
Mining of metal ores, except gold and uranium		24
Other mining and quarrying		25
Services activities incidental to mining of minerals		29
Manufacturing	**3**	
Manufacture of food products, beverages and tobacco products		30
Manufacture of textiles, clothing and leather goods		31
Manufacture of wood and of products of wood and cork, except furniture;		32
Manufacture of articles of straw and plaiting materials; manufacture of paper and paper products; publishing, printing and reproduction of recorded media		32
Manufacture of coke, refined petroleum products and nuclear fuel; manufacture of chemicals and chemical, rubber and plastic products		33
Manufacture of other non-metallic mineral products		34
Manufacture of basic metals, fabricated metal products, machinery and equipment and of office, accounting and computing machinery		35
Manufacture of electrical machinery and apparatus not elsewhere classified		36
Manufacture of radio, television and communications equipment and apparatus, and of medical, precision and optical instruments, watches and clocks		37
Manufacture of transport equipment		38
Manufacture of furniture; manufacturing not elsewhere classified; recycling		39
Electricity, gas and water supply	**4**	
Electricity, gas, steam and hot water supply		41
Collection, purification and distribution of water		42
Construction	**5**	
Wholesale and retail trade; repair of motor vehicles, motorcycles, and personal and household goods; catering and accommodation	**6**	
Wholesale and commission trade, except of motor vehicles and motor cycles		61
Retail trade, except of motor vehicles and motorcycles; repair of personal household goods		62
personal household goods		62
Sale, maintenance and repair of motor vehicles and motorcycles; retail trade in automotive fuel		63
retail trade in automotive fuel		63
Catering and accommodation		64
Transport, storage and communication	**7**	
Land transport; transport via pipelines		71
Water transport		72
Air transport		73
Supporting and auxiliary transport activities; activities of travel agencies		74
Post and telecommunications		75
Financial intermediation, insurance, real-estate and business services	**8**	
Financial intermediation, except insurance and pension funding		81
Insurance and pension funding, except compulsory social security		82
Activities auxiliary to financial intermediation		83
Real-estate activities		84
Renting of machinery and equipment, without operator, and of personal and household goods		85
Computer and related activities		86
Research and development		87
Other business activities		88
Community, social and personal services	**9**	
Public administration and defence activities		91
Education		92
Health and social work		93
Other community, social and personal service activities		94
Activities of membership organisations not elsewhere classified		95
Recreational, cultural and sporting activities		96
Other service activities		99
Private households, extra-territorial organisations, representatives of foreign governments and other activities not adequately defined	**0**	
Private households with employed persons		01
Extra-territorial organisations		02
Representatives of foreign governments		03
Other activities not adequately defined		04
Private households with employed persons		01
Extra-territorial organisations		02
Representatives of foreign governments		03
Other activities not adequately defined		04
Individuals and non-producing households		041
Employees		0411
Retired persons		0412
Other individuals or households, including unemployed persons, students		0413
Other		042

A.3.4 SIC economic groups

Note that the divisions may be further divided into major groups (three-digit level) and groups (four-digit level) and sub-groups (five-digit level). For the purpose of this research the SIC codes were used on the five-digit sub-group level and grouped together into sub industries, and industries. Where there were gaps in the data, research was done on the particular company to find the type of industry and relevant sub-group SIC code. The importance of populating all the SIC codes becomes apparent when data estimations need to be done as per section A.4.

The number of SIC codes at this level (±500) are too many to list here. Therefore the tables below displays only extracts of sub-group SIC codes from major groups and subgroups.

On the following two pages SIC groups and example sub-groups are listed for these major divisions:

Major Division 1: Agriculture, Hunting, Forestry and Fishing

Major Division 2: Mining and Quarrying

Major Division 3: Manufacturing

Major Division 4: Electricity, Gas and Water Supply

Major Division 5: Construction

Major Division 6: Wholesale and Retail Trade

Major Division 7: Transport, Storage and Communication

Major Division 8: Financial Intermediation, Insurance, Real Estate and Business

Major Division 9: Community, Social and Personal Services

Table A.14: SIC groups for trade related industries

Major Group	Group	Sub Group	Major Division	SIC CODE DESCRIPTION	Industry	Sub Industry
			1	MAJOR DIVISION 1: AGRICULTURE, HUNTING, FORESTRY & FISHING	Agriculture	Agriculture
111	1111	11110		Growing Of Cereals And Other Crops N.E.C.	Agriculture	Farming
115	1151	11510		Game Propagation	Agriculture	Game
116	1160	11600		Production Of Organic Fertilizer	Agriculture	Production
121	1210	12100		Forestry And Related Services	Agriculture	Forestry
131	1310	13100		Ocean And Coastal Fishing	Agriculture	Fishing
			2	MAJOR DIVISION 2: MINING & QUARRYING	Mining	Mining
210	2100	21000		Mining Of Coal And Lignite	Mining	Coal
221	2211	22110		Extraction Of Crude Petroleum And Natural Gas	Mining	Petroleum
230	2300	23000		Mining Of Gold And Uranium Ore	Mining	Gold
241	2410	24100		Mining Of Iron Ore	Mining	Iron
242	2420	24200		Mining Of Non-Ferrous Metal Ores, Except Gold And Uranium	Mining	Other Metals
242	2424	24240		Platinum Group Metals Mining And The Dressing, Beneficiating And Otherwise Preparing Of Such Ore	Mining	Platinum
251	2511	25110		Stone Quarrying: Dimension Stone (Granite, Marble, Slate, And Wonderstone)	Mining	Quarrying
252	2520	25200		Mining Of Diamonds (Including Alluvial Diamonds)	Mining	Diamonds
			3	MAJOR DIVISION 3: MANUFACTURING	Manufacturing	Manufacturing
301	3011	30111		Slaughtering, Dressing And Packing Of Livestock, Including Poultry And Small Game For Meat	Manufacturing	Foods
305	3052	30521		Breweries, Except Sorghum Beer Breweries	Manufacturing	Beverages
306	3060	30600		Manufacture Of Tobacco Products	Manufacturing	Tobacco
311	3111	31110		Spinning, Weaving And Finishing Of Yarns And Fabrics, Other	Manufacturing	Clothing & Fabrics
323	3239	32399		Manufacture Of Other Paper Products	Manufacturing	Sawmills & Timber
324	3241	32410		Publishing Of Books, Brochures, Musical Books And Other Publications	Media & Publishing	Books
324	3242	32420		Publishing Of Newspapers, Journals And Periodicals	Media & Publishing	Newspapers & Journals
324	3243	32430		Publishing Of Recorded Audio Media	Media & Publishing	Recorded Media
324	3249	32490		Other Publishing	Media & Publishing	Other Publishing
325	3251	32510		Printing	Media & Publishing	Printing
325	3252	32520		Service Activities Related To Printing	Media & Publishing	Printing Services
326	3260	32600		Reproduction Of Recorded Media	Media & Publishing	Recorded Media
331	3310	33100		Manufacture Of Coke Oven Products	Manufacturing	Cement, Ceramics & Stone
332	3322	33220		Manufacture Of Petrol, Fuel Oils, Lubricating Oils And Greases, Primarily From Coal	Manufacturing	Fuel & Oil Products
334	3342	33420		Manufacture Of Fertilizers And Nitrogen Compounds	Manufacturing	Chemicals & Fertilizers
335	3353	33530		Manufacture Of Pharmaceuticals, Medicinal Chemicals And Botanical Products	Health	Pharmaceutical
335	3354	33541		Manufacture Of Soap And Other Cleaning Compounds	Manufacturing	Cleaning Products
335	3354	33542		Manufacture Of Perfumes, Cosmetics And Other Toilet Preparations	Manufacturing	Cosmetics
335	3354	33549		Manufacture Of Other Preparations Such As Polishes, Waxes And Dressings	Manufacturing	Cleaning Products
335	3359	33591		Manufacture Of Edible Salt	Manufacturing	Foods
337	3371	33711		Manufacture Of Tyres And Tubes	Manufacturing	Motor Vehicles & Parts
337	3379	33790		Manufacture Of Other Rubber Products	Manufacturing	Rubber & Hosing
338	3380	33800		Manufacture Of Plastic Products	Manufacturing	Plastics & Glass
342	3421	34210		Manufacture Of Non-Structural Non-Refractory Ceramicware	Manufacturing	Cement, Ceramics & Stone
342	3429	34291		Manufacture Of Abrasives	Manufacturing	Plastics & Glass
342	3429	34299		Manufacture Of Other Non-Metallic Mineral Products N.E.C.	Manufacturing	Other Minerals
351	3510	35102		Steel Pipe And Tube Mills	Manufacturing	Metals
356	3562	35620		Manufacture Of Pumps, Compressors, Taps And Valves	Manufacturing	Pumps
356	3564	35640		Manufacture Of Ovens, Furnaces And Furnace Burners	Manufacturing	Cement, Ceramics & Stone
356	3565	35650		Manufacture Of Lifting And Handling Equipment	Manufacturing	Machinery
357	3577	35770		Manufacture Of Weapons And Ammunition	Manufacturing	Weapons & Ammunition
361	3610	36100		Manufacture Of Electric Motors, Generators And Transformers	Manufacturing	Electrical & Appliances
374	3741	37412		Manufacture Of Surgical, Medical And Dental Supplies	Health	Medical Supplies
375	3750	37500		Manufacture Of Optical Instruments And Photographic Equipment	Manufacturing	Photographic
381	3810	38100		Manufacture Of Motor Vehicles	Manufacturing	Transport Vehicles & Parts
391	3910	39103		Manufacture Of Furniture Made Predominantly Of Materials Other Than Metal, Plastic Or Concrete	Manufacturing	Furniture
392	3921	39212		Diamond Cutting And Polishing	Manufacturing	Jewellery
392	3922	39220		Manufacture Of Musical Instruments	Manufacturing	Musical Instruments
392	3923	39230		Manufacture Of Sports Goods	Manufacturing	Sports Goods
392	3924	39240		Manufacture Of Games And Toys	Manufacturing	Toys
392	3929	39291		Brushes And Brooms	Manufacturing	Household
392	3929	39292		Crayons, Chalk, Pens And Pencils	Manufacturing	Stationary
392	3929	39293		Buttons, Buckles, Slide Fasteners, Etc.	Manufacturing	Clothing & Fabrics
392	3929	39299		Other Industries Not Elsewhere Classified, Including Rubber Stamps, Taxidermists, Ostrich Feathers	Manufacturing	Household
395	3951	39510		Recycling Of Metal Waste And Scrap N.E.C.	Manufacturing	Waste
			4	MAJOR DIVISION 4: ELECTRICITY, GAS & WATER SUPPLY	Utilities	Utilities
411	4111	41111		Electricity Generation	Utilities	Electricity
412	4120	41200		Manufacture Of Gas; Distribution Of Gaseous Fuels Through Mains	Utilities	Gas
413	4130	41300		Steam And Hot Water Supply	Utilities	Water
			5	MAJOR DIVISION 5: CONSTRUCTION	Construction	Construction
501	5010	50100		Construction: Site Preparation	Construction	Building
503	5031	50310		Plumbing	Construction	Plumbing
503	5032	50320		Electrical Contracting	Construction	Electrical & Appliances
503	5033	50330		Shopfitting	Construction	Shopfitting
504	5041	50410		Painting And Decorating	Retail, Wholesale, Trade	Painting & Decorating
504	5049	50490		Other Building Completion N.E.C.	Construction	Other Building Completion
505	5050	50500		Renting Of Construction Or Demolition Equipment With Operators	Construction	Construction, Demolition Equipment

Table A.15: SIC groups for commerce related industries and services

Major Group	Group	Sub Group	Major Division	SIC CODE DESCRIPTION	Industry	Sub Industry
			6	MAJOR DIVISION 6: WHOLESALE & RETAIL TRADE	Retail, Wholesale, Trade	Retail, Wholesale, Trade
613	6131	61310		Wholesale Trade In Textiles, Clothing And Footwear	Retail, Wholesale, Trade	Clothing & Fabrics
613	6139	61394		Wholesale Trade In Pharmaceuticals, Toiletries And Medical Equipment	Health	Medical Supplies
613	6139	61399		Wholesale Trade In Other Household Goods N.E.C	Retail, Wholesale, Trade	Wholesale
614	6141	61410		Wholesale Trade In Solid, Liquid And Gaseous Fuels And Related Products	Retail, Wholesale, Trade	Gas
614	6149	61490		Wholesale Trade In Other Intermediate Products, Waste And Scrap	Retail, Wholesale, Trade	Scrap
621	6211	62110		Retail Trade In Non-Specialised Stores With Food, Beverages And Tobacco Predominating	Retail, Wholesale, Trade	Foods
621	6219	62190		Other Retail Trade In Non-Specialised Stores	Retail, Wholesale, Trade	Retail
634	6340	63400		Sale, Maintenance And Repair Of Motor Cycles And Related Parts And Accessories	Retail, Wholesale, Trade	Motor Vehicles & Parts
641	6410	64101		Hotels, Motels, Botels And Inns Registered With The Sa Tourism Board	Tourism	Accommodation
642	6420	64203		Take-Away Counters	Retail, Wholesale, Trade	Hospitality
			7	MAJOR DIVISION 7: TRANSPORT, STORAGE & COMMUNICATION	Transport	Transport
711	7111	71111		Inter-Urban Railway Transport	Transport	Transport
712	7122	71222		Safaris And Sightseeing Bus Tours	Tourism	Tours
741	7411	74110		Cargo Handling	Transport	Freight
741	7412	74120		Storage And Warehousing	Transport	Warehousing
751	7511	75110		National Postal Activities	Transport	Postal
751	7512	75120		Courier Activities Other Than National Postal Activities	Transport	Couriers
752	7520	75200		Telecommunications	ICT	Communications
			8	MAJOR DIVISION 8: FINANCIAL INTERMEDIATION, INSURANCE, REAL ESTATE & BUSINESS	Business	Business
811	8111	81110		Central Banking	Financial	Banking
819	8199	81990		Other Financial Intermediation N.E.C.	Financial	Credit
821	8212	82120		Pension Funding	Financial	Pension
831	8311	83110		Administration Of Financial Markets	Financial	Financial
832	8320	83200		Activities Auxiliary To Insurance And Pension Funding	Financial	Insurance
841	8411	84110		Property Owning And Letting	Financial	Real Estate
841	8412	84120		Developing Real Estate, Subdividing Real Estate Into Lots And Residential Development On Own Account	Construction	Building
852	8521	85210		Renting Of Agricultural Machinery And Equipment	Agriculture	Renting
861	8610	86100		Hardware Consultancy	ICT	Hardware
862	8620	86200		Software Consultancy And Supply	ICT	Software
864	8640	86400		Data Base Activities	ICT	Data
865	8650	86500		Maintenance And Repair Of Office, Accounting And Computing Machinery	ICT	IT
881	8812	88121		Activities Of Accountants And Auditors Registered In Terms Of The Public Accountants And Auditors Act	Business Services	Accounting & Auditing
881	8813	88130		Market Research And Public Opinion Polling	Business Services	Research
882	8821	88211		Consulting Engineering Activities	Engineering	Consulting
882	8821	88212		Architectural Activities	Construction	Building
882	8821	88215		Geological And Prospecting Activities On A Fee Or Contract Basis	Mining	Geological
882	8821	88216		Activities Of Non-Registered Architects, E.G. Tracers And Draughts-Men Of Plans For Dwellings	Construction	Building
883	8831	88311		Advertising: Activities Of Advertising Agents	Business Services	Advertising
889	8891	88911		Activities Of Employment Agencies And Recruiting Organisations	Business Services	Employment Agencies
889	8892	88920		Investigation And Security Activities	Business Services	Investigation Services
889	8893	88930		Building And Industrial Plant Cleaning Activities	Business Services	Industrial
889	8894	88940		Photographic Activities	Business Services	Photographic
889	8895	88950		Packaging Activities	Manufacturing	Packaging
889	8899	88991		Credit Rating Agency Activities	Financial	Credit
889	8899	88992		Debt Collecting Agency Activities	Financial	Debt Collecting
889	8899	88999		Other Business Activities N.E.C.	Business Services	Other Business Services
			9	MAJOR DIVISION 9: COMMUNITY, SOCIAL & PERSONAL SERVICES	Community and Personal Services	Community & Personal Services
920	9200	92002		Primary And Secondary Education	Education	Education
931	9311	93115		Detached Operation Theatres	Health	Health
931	9312	93121		Medical Practitioner And Specialist Activities	Health	Medical Practitioners
931	9319	93192		Clinics And Related Health Care Services	Health	Hospitals & Clinics
932	9320	93200		Veterinary Activities	Health	Veterinary Activities
933	9330	93300		Social Work Activities	Community & Personal Services	Community & Personal Services
940	9400	94000		Sewage And Refuse Disposal, Sanitation And Similar Activities	Government	Government
951	9511	95110		Activities Of Business And Employers' Organisations	Community & Personal Services	Employee Organisations
951	9512	95120		Activities Of Professional Organisations	Community & Personal Services	Professional Organisations
959	9591	95910		Activities Of Religious Organisations	Community & Personal Services	Religious Organisations
959	9592	95920		Activities Of Political Organisations	Community & Personal Services	Political Organisations
959	9599	95990		Activities Of Other Membership Organisations N.E.C.	Community & Personal Services	Other Membership Organisations
961	9611	96111		Motion Picture And Video Production And Distribution	Entertainment	Film
961	9613	96130		Radio And Television Activities	Entertainment	Radio & Television
961	9614	96140		Dramatic Arts, Music And Other Arts Activities	Entertainment	Dramatic Arts
961	9619	96190		Other Entertainment Activities N.E.C.	Entertainment	Other Entertainment
962	9620	96200		News Agency Activities	Entertainment	News
963	9631	96310		Library And Archives Activities	Entertainment	Libraries
963	9632	96320		Museum Activities And Preservation Of Historical Sites And Buildings	Entertainment	Museums
963	9633	96330		Botanical And Zoological Gardens And Nature Reserve Activities	Agriculture	Botanical & Zoological
964	9641	96410		Sporting Activities	Entertainment	Sporting Activities
964	9649	96490		Other Recreational Activities	Entertainment	Other Recreational Activities
990	9902	99023		Men's And Ladies' Hairdressing	Retail, Wholesale, Trade	Retail

A.3.5 ICT spend percentages

The ICT Expenditure, or ICT spend, relies on the proportion of the total industry expenditure. This translates to the ICT spend as a percentage of the turnover of a company as derived from key metrics data (Gartner, 2011). The IT enterprise key metrics report from 2011 was best suited to be adapted to the Tanzanian market.

There are two percentage types for ICT spend:

- ICT spend as percentage of turnover
- ICT spend as percentage of the country ICT spend

Below find a list of industries with the percentages used to calculate the ICT spend as described in sections A.3.4 and A.3.5.

Table A.16: ICT spend as percentage of turnover and country ICT

Major Division	Division	% ICT of Turnover	% ICT of Country ICT	Sub Industry
1		1.4%	2.8%	Agriculture
	11	1.4%	0.03%	Farming
	11	1.4%	0.04%	Game
	11	1.4%	0.04%	Production
	12	1.4%	0.06%	Forestry
	13	1.4%	0.05%	Fishing
2		1.0%	3.9%	Mining
	21	1.0%	0.08%	Coal
	22	1.0%	0.07%	Petroleum
	23	1.0%	0.07%	Gold
	24	1.0%	0.07%	Iron
	24	1.0%	0.07%	Other Metals
	24	1.0%	0.07%	Platinum
	25	1.0%	0.07%	Quarrying
	25	1.0%	0.07%	Diamonds
	25	1.0%	0.07%	Other Minerals
	29	1.0%	0.07%	Other Minerals
3		2.1%	3.6%	Manufacturing
	30	1.3%	0.10%	Foods
	30	1.3%	0.09%	Beverages
	30	1.9%	0.07%	Tobacco
	31	1.9%	0.09%	Clothing & Fabrics
	32	1.8%	0.09%	Sawmills & Timber
	32	5.2%	0.08%	Books
	32	5.2%	0.08%	Newspapers & Journals
	32	5.2%	0.08%	Recorded Media
	32	5.2%	0.11%	Other Publishing
	32	5.2%	0.10%	Printing
	32	5.2%	0.08%	Printing Services
	33	1.8%	0.12%	Cement, Ceramics & Stone
	33	1.8%	0.10%	Fuel & Oil Products
	33	1.2%	0.09%	Chemicals & Fertilizers
	33	3.1%	0.10%	Pharmaceutical
	33	1.9%	0.10%	Cleaning Products
	33	1.9%	0.09%	Cosmetics
	33	1.3%	0.10%	Foods
	33	1.9%	0.10%	Clothing & Fabrics
	33	1.8%	0.11%	Motor Vehicles & Parts
	33	1.8%	0.12%	Rubber & Hosing
	33	1.8%	0.07%	Plastics & Glass
	34	1.8%	0.07%	Plastics & Glass
	34	1.8%	0.12%	Cement, Ceramics & Stone
	34	1.8%	0.07%	Other Minerals
	35	1.9%	0.07%	Metals
	35	4.3%	0.07%	Metals
	35	1.8%	0.07%	Pumps
	35	1.8%	0.07%	Cement, Ceramics & Stone
	35	1.8%	0.10%	Machinery
	35	1.8%	0.12%	Weapons & Ammunition
	35	2.4%	0.05%	Electrical & Appliances
	36	2.4%	0.05%	Electrical & Appliances
	37	2.4%	0.05%	Electrical & Appliances
	37	4.2%	0.10%	Medical Supplies
	37	2.4%	0.07%	Electrical & Appliances
	37	1.8%	0.08%	Photographic
	38	1.8%	0.09%	Transport Vehicles & Parts
	39	1.9%	0.11%	Furniture
	39	1.9%	0.10%	Jewellery
	39	1.8%	0.10%	Musical Instruments
	39	1.9%	0.10%	Sports Goods
	39	1.9%	0.10%	Toys
	39	1.9%	0.10%	Household
	39	1.9%	0.10%	Stationary
	39	1.9%	0.10%	Clothing & Fabrics
	39	1.8%	0.10%	Waste
4		2.5%	3.9%	Utilities
	41	2.0%	0.10%	Electricity
	41	2.8%	0.10%	Gas
	41	2.8%	0.07%	Water
	42	2.8%	0.07%	Water
5		1.6%	3.4%	Construction
	50	2.7%	0.10%	Building
	50	4.3%	0.10%	Building
	50	1.0%	0.05%	Plumbing
	50	2.4%	0.05%	Electrical & Appliances
	50	1.0%	0.10%	Shopfitting
	50	1.0%	0.10%	Other Building Completion
	50	1.9%	0.10%	Painting & Decorating
	50	1.0%	0.10%	Construction, Demolition Equipment
6		2.1%	3.9%	Retail, Wholesale, Trade
	61	1.5%	0.07%	Wholesale
	61	1.5%	0.04%	Foods
	61	1.9%	0.10%	Clothing & Fabrics
	61	3.1%	0.10%	Medical Supplies
	61	1.9%	0.10%	Gas
	61	1.8%	0.10%	Scrap
	62	1.5%	0.07%	Foods
	62	1.5%	0.10%	Retail
	62	3.1%	0.10%	Retail
	62	1.9%	0.09%	Clothing & Fabrics
	63	2.1%	0.11%	Motor Vehicles & Parts
	63	1.5%	0.07%	Retail
	64	5.0%	0.09%	Accommodation
	64	1.8%	0.08%	Hospitality
7		2.9%	3.5%	Transport
	71	2.7%	0.09%	Transport
	71	5.0%	0.10%	Tours
	71	2.7%	0.10%	Freight
	72	2.7%	0.09%	Transport
	73	2.7%	0.09%	Transport
	74	2.7%	0.09%	Freight
	74	2.7%	0.09%	Warehousing
	74	2.7%	0.09%	Transport
	75	2.7%	0.12%	Postal
	75	2.7%	0.09%	Couriers
	75	3.8%	0.12%	Communications
8		4.4%	3.8%	Business
	81	6.3%	0.10%	Banking
	81	6.3%	0.10%	Credit
	82	3.2%	0.10%	Insurance
	82	6.3%	0.09%	Pension
	83	6.3%	0.10%	Financial
	83	3.2%	0.10%	Insurance
	84	6.3%	0.10%	Real Estate
	84	1.0%	0.10%	Building
	84	6.3%	0.10%	Real Estate
	85	2.7%	0.09%	Renting
	85	1.4%	0.09%	Renting
	85	1.0%	0.09%	Renting
	85	1.8%	0.09%	Renting
	85	1.5%	0.09%	Retail
	86	5.4%	0.10%	Hardware
	86	7.0%	0.10%	Software
	86	6.7%	0.08%	Data
	86	5.4%	0.10%	IT
	87	4.3%	0.10%	Research
	87	4.2%	0.10%	Research
	88	4.3%	0.11%	Accounting & Auditing
	88	4.3%	0.11%	Research
	88	4.3%	0.10%	Consulting
	88	1.0%	0.11%	Building
	88	1.0%	0.11%	Geological
	88	4.8%	0.11%	Advertising
	88	4.8%	0.11%	Employment Agencies
	88	4.8%	0.10%	Investigation Services
	88	1.8%	0.07%	Industrial
	88	4.8%	0.12%	Photographic
	88	1.8%	0.11%	Packaging
	88	6.3%	0.10%	Credit
	88	6.3%	0.10%	Debt Collecting
	88	4.8%	0.11%	Business Services
	88	4.3%	0.11%	Other Business Services
9		3.5%	3.4%	Community & Personal Services
	91	2.0%	0.10%	Government
	92	4.7%	0.07%	Education
	93	4.2%	0.07%	Hospitals & Clinics
	93	4.2%	0.07%	Health
	93	4.2%	0.09%	Medical Practitioners
	93	4.2%	0.07%	Veterinary Activities
	93	1.4%	0.07%	Community & Personal Services
	94	2.0%	0.07%	Government
	95	4.3%	0.10%	Professional Organisations
	95	1.4%	0.10%	Employee Organisations
	95	1.4%	0.10%	Religious Organisations
	95	1.4%	0.10%	Political Organisations
	95	1.4%	0.10%	Other Membership Organisations
	96	5.2%	0.09%	Film
	96	5.2%	0.12%	Radio & Television
	96	5.2%	0.08%	Dramatic Arts
	96	5.2%	0.08%	Other Entertainment
	96	5.2%	0.10%	News
	96	5.2%	0.08%	Libraries
	96	5.2%	0.08%	Museums
	96	1.4%	0.08%	Botanical & Zoological
	96	5.2%	0.08%	Sporting Activities
	96	5.2%	0.07%	Other Recreational Activities
	99	1.5%	0.10%	Retail

A.4 Data estimations

For certain analysis methods there need to be no gaps in the data. Therefore, to fill in gaps for certain variables, estimations were done based on calculations of existing data. In other cases, the data need to be derived for every record. The calculations and estimations are shown below.

A.4.1 Number of Employees

Where the employee figure was not disclosed in the researched data, the number was derived, based on methods provided by the Who Owns Whom data source (Who Owns Whom contributors, 2016a).

- The average of all the employees per SIC code was taken per country.

$$E_{gap} = \frac{\sum_{i}^{n(SIC_C)} Ec_i}{n(SIC_C)}, \tag{A.1}$$

$for\ each\ SIC \in \{SIC\ code\ of\ E_{gap}\},$
$with\ E_{gap}\ the\ number\ of\ employees\ missing,$
$and\ Ec_i\ the\ number\ of\ employees\ for\ the\ SIC\ code\ per\ country,$
$and\ SIC_C\ the\ SIC\ code\ for\ the\ relevant\ country,$
$and\ n(SIC_C)\ the\ number\ of\ values\ per\ SIC\ code\ per\ country$

- If no country average of employees exist, the Tanzanian average for the SIC code was used multiplied by the ratio between the GDP and the GDP per Capita. This derived figure will then be multiplied by the ratio between the parent company (Group) employees and the group employees total for the SIC code and country.

$$E_{gap} = \left(\frac{E_p}{\sum_{i}^{n(SIC)} Ep_{SA_i}}\right) E_{Sconv} \tag{A.2}$$

$for\ each\ SIC \in \{SIC\ code\ of\ E_{gap}\},$
$with\ E_{gap}\ the\ number\ of\ employees\ missing,$

$and\ E_p\ the\ parent\ employee\ figure,$

$and\ Ep_{SA_i}\ a\ parent\ employee\ figure\ in\ South\ Africa\ and\ same\ SIC$

Here E_{Sconv} is calculated as (AverageTZN * GDP Ratio) with AverageTZN the average number of employees per company in Tanzania, in a specific SIC code.

$$E_{Sconv} = \left(\frac{\sum_i^{n(SIC)} E_{SA_i}}{n(SIC_{SA})}\right) \frac{GDP_{SA}}{(GDP_{SA}/P_{SA})} \quad (A.3)$$

$for\ each\ SIC \in \{SIC\ code\ of\ E_{gap}\},$

$E_{SA_i}\ the\ number\ of\ employees\ for\ a\ company\ per\ SIC\ code\ in\ South\ Africa,$

$and\ n(SIC_{SA})\ the\ number\ of\ companies\ per\ SIC\ code\ in\ South\ Africa,$

$and\ GDP_{SA}\ the\ total\ GDP\ of\ South\ africa,$

$and\ P_{SA}\ the\ total\ population\ of\ South\ Africa,$

$where\ \ GDP_{SA}/P_{SA}\ \ gives\ the\ GDP\ per\ capita\ of\ South\ Africa$

A.4.2 Turnover figures

Where turnover figures were undisclosed in the researched data, the Tanzanian average per SIC code was calculated against the number of employees. This method is based on the way Who Owns Whom calculated the missing values (Who Owns Whom contributors, 2016a).

The turnover value is therefore calculated in terms of (Average turnover / Actual number of employees), for the average turnover across all companies in Tanzania, in a specified SIC code.

$$T_{gap} = \left(\frac{\sum_i^{n(SIC_C)} Tc_i}{n(SIC_C)}\right) \Big/ E_{Tgap}, \quad (A.4)$$

$for\ each\ SIC \in \{SIC\ code\ of\ T_{gap}\},$

$with\ T_{gap}\ the\ missing\ turnover\ value,$

$and\ Tc_i\ the\ turnover\ for\ the\ SIC\ code\ per\ country,$

$and\ SIC_C\ the\ SIC\ code\ for\ relevant\ country,$

and $n(SIC_C)$ *the number of values with no gaps per SIC code for country, and* E_{Tgap} *the employees for the company with missing turnover values* (*note that* E_{Tgap} *is actual or derived number of employees*)

A.4.3. Company size classification

The classification variable in the target dataset, is derived from the SIC code, number of employees and turnover values.

A reference table using limits of company size to determine the classification, is provided by the Who Owns Whom data source (Who Owns Whom contributors, 2016b). The methodology using these limits originated from the National Small Business Act (Zulu, 2018), and is then revised to make provision for inflation and sub-sector disaggregation, where applicable.

Table A.17: Limits for the classification of companies by size

SECTOR / SUB-SECTOR	SIC Division	CATEGORY	EMPLOYEES (greater than or equal to)	TURNOVER ($'000) (greater than or equal to)
Agriculture	1	Large	101	303
		Medium	51	182
		Small	11	30
		Very Small	6	12
		Micro	0	
Mining & Quarrying	2	Large	201	2 364
		Medium	51	606
		Small	21	242
		Very Small	6	12
		Micro	0	
Manufacturing	3	Large	201	3 091
		Medium	51	788
		Small	21	303
		Very Small	6	12
		Micro	0	
Electricity, Gas, & Water	4	Large	201	3 091
		Medium	51	788
		Small	21	309
		Very Small	6	12
		Micro	0	
Construction	5	Large	201	1 576
		Medium	51	364
		Small	21	182
		Very Small	6	12
		Micro	0	
Wholesale Trade, Commercial Agents & Allied Services	61	Large	201	3 879
		Medium	51	1 939
		Small	21	364
		Very Small	6	12
		Micro	0	
Retail & Motor Trade & Repairs	62, 63	Large	201	2 364
		Medium	51	1 152
		Small	21	242
		Very Small	6	12
		Micro	0	
Catering, Accommodation & Other Trade	64	Large	201	788
		Medium	51	364
		Small	21	309
		Very Small	6	12
		Micro	0	
Transport, Storage, Communication	7	Large	201	1 576
		Medium	51	788
		Small	21	182
		Very Small	6	12
		Micro	0	
Insurance, Pension Funding, Financial Intermediation & Auxiliary Services	81, 82, 83	Large	201	48 485
		Medium	51	7 576
		Small	21	3 030
		Very Small	6	121
		Micro	0	
Real Estate & Other Business Services	84, 85, 86, 87, 88	Large	201	1 576
		Medium	51	788
		Small	21	182
		Very Small	6	12
		Micro	0	
Community, Social & Personal Services (excl. Government)	9	Large	201	788
		Medium	51	364
		Small	21	61
		Very Small	6	12
		Micro	0	

Sources: Who Owns Whom, derived from Schedule 1 of the National Small Business Act, 1996 (amended 2018)

A.4.4. ICT spend

Once the turnover value and number of employees are populated, the ICT spend is calculated in two ways (Who Owns Whom contributors, 2016c).

- ICT spend based on turnover is calculated as:
  (Turnover * % ICT of turnover)
  Here the turnover can be actual or derived, and %ICT of turnover is per SIC division in Table A.16.
- ICT spend based on number of employees is calculated as:
  (Average ICT spend per employee * Employees)
  Here Average ICT spend per employee is the total ICT spend per SIC division divided by the number of employees in the division, and number of employees is actual or derived.

A.4.5. % ICT country spend

The ICT spend of the country (Tanzania) is derived from the % of the GDP represented by ICT. The GDP was taken as $5.74 billion (Trading Economics, 2017). The ICT as percentage of this GDP for Tanzania is about 24.4%. This figure was derived from software and internet figures in a publication on ICT usage in sub-Saharan Africa (Cirera *et al.* 2016). Various factors could be taken into account for Tanzania ICT calculations, but were not analysed as part of the scope of this research. As further reading a study by Mwantimwa (2019) on the usage of ICT to enhance business processes is available in the literature.

The total ICT spend for the country as % of the GDP is then $14 billion approximately. Dividing the ICT spend as derived in section A.4.4, in this country ICT spend will give a percentage. For example, an ICT spend of $1 304 806 divided by total ICT spend for Tanzania, will give a percentage of 0.0093%. This percentage is the ICT as % of country ICT variable in the dataset.

A.4.6. ICT usage population

Taking the population of Tanzania as 56 318 348 (Trading Economics, 2017), the ICT usage can be calculated. The percentages given below are used per company size to calculate the ICT usage (Cirera *et al.*, 2016).

Table A.18: ICT usage and average % for selected African countries

Company Size	ICTs	ICT usage by selected countries (%)						
		Ghana	DRC	Tanzania	Uganda	Zambia	Kenya	Average
Small Very Small Micro	Computer	57.3	48.8	34.3	60	67.4	81.5	58.2
	Software	7	28.2	7.9	14	22.1	22.2	17.1
	Internet	43.8	20.9	18	21.3	44.2	64.4	35.5
Medium	Computer	89.7	74.4	59.6	35.3	67.5	96.8	70.5
	Software	17.9	37.8	16.6	14.2	24.2	39.7	25.4
	Internet	83	58.7	34.2	19.3	56.4	83.1	55.8
Large	Computer	98.6	94.1	69.9	89.6	100	100	91.9
	Software	25.6	61.1	22	21	58.5	46.1	38.4
	Internet	95.5	63.7	42.9	80.4	93.3	87.6	77.2
Company Size	**ICTs**	**Average ICT usage % by selected countries per Company Size**						
		Ghana	**DRC**	**Tanzania**	**Uganda**	**Zambia**	**Kenya**	**Average**
Small Very Small Micro	Computer	36%	33%	20%	32%	45%	56%	37%
	Software							
	Internet							
Medium	Computer	64%	57%	37%	23%	49%	73%	51%
	Software							
	Internet							
Large	Computer	73%	73%	45%	64%	84%	78%	69%
	Software							
	Internet							

For example, the ICT usage for medium sized companies is on average 37% of the total population, or 20 725 152 users. A specific medium sized company with 56 employees will then represent a potential of 56 out of 20 725 152 users of ICT, or a ICT as % of usage population value of 0.00027%.

Appendix B – DETAILED RESULTS OF TESTS

B.1 Descriptive Statistics

B.1.1 Numerical Variables

Below find a summary descriptive statistics of all numerical variables.

Table B.1: Numerical variables: descriptive statistics

Measure	n	Missing Values	Min	Max	Mean	Median	Mode	Range	Standard Deviation	Standard Error	Coefficient of Variation	Variance	Lower 90% Confidence Limit	Upper 90% Confidence Limit	Skewness	Kurtosis
Co_Employees	3362	0	1	12844	613	163	120	12843	1087	18.751	3%	1 182 031	583	644	3	11
Co_Turnover	3362	0	9899.99	6.504 × 109	1.661 × 108	7.895 × 106	2.189 × 106	6.504 × 109	5.739 × 108	9.898 × 106	6%	3.293 × 1017	1.498 × 108	1.824 × 108	5	31
ICT_Spend	3362	0	99	2.398 × 108	5.732 × 106	185 081.69	105 078.81	2.398 × 108	2.181 × 107	376 091.42	7%	4.755 × 1014	5.114× 106	6.351 × 106	5	31
Telecom_ARPU	827	2535	241.42	1.459 × 108	1.039 × 106	20 437.71	78 221.06	1.459 × 108	7.870 × 106	273 501.70	26%	6.186 × 1013	5.892 × 105	1.489 × 106	13	198
Telecom_Revenue	827	2535	4179.31	5.786 × 108	6.518 × 106	448 974.56	190 876.44	5.786 × 108	3.343 × 107	1.163 × 106	18%	1.116 × 1015	4.607 × 106	8.429 × 106	10	132
Telecom_Subscribers	827	2535	1	5711	115	25	1	5710	367	12.753	11%	134 497	94	136	8	89
Telecom_SolutionLines	592	2770	0	158630	538	14	3	158630	6847	281.390	52%	4.687 × 107	75	1 001	21	481
Telecom_ProductLines	592	2770	0	158629	532	17	3	158629	6847	281.411	53%	4.689 × 107	69	995	21	481
Telecom_DeviceLines	441	2921	0	13933	63	3	1	13933	704	33.547	53%	496 302	8	118	18	341

B.1.2 Ratio Variables

Below find a summary descriptive statistics of the two ratio variables.

Table B.2: Ratio variables: descriptive statistics

Measure	ICT as % of Country ICT	ICT as % of Usage Population
n	3362	3362
Missing Values	0	0
Min	0.0000007%	0.00001%
Max	1.713%	0.046%
Mean	0.04%	0.0023%
Median	0.0013%	0.0007%
Mode	0.00075%	0.00039%
Range	1.713%	0.0464%
Standard Deviation	0.156%	0.00391%
Standard Error	0.0027%	0.000067%
Coefficient of Variance	0.16%	0.15%
Variance	0.00024%	0.0000002%
Lower 90% Confidence Limit	1.4565%	0.0400%
Upper 90% Confidence Limit	1.9689%	0.0528%
Skewness	5.47	2.99
Kurtosis	30.78	11.33

The percentiles were calculated for the ratio variables and the data behind the analysis in section 5.1.4 is shown here.

Table B.3: Ratio variables: percentiles

Percentiles	ICT as % of country ICT	ICT as % of usage population
10%	0.000018%	0.000017%
20%	0.000047%	0.000028%
25%	0.000062%	0.000034%
30%	0.000072%	0.000039%
40%	0.000093%	0.000053%
50%	0.000132%	0.000066%
60%	0.000336%	0.000093%
70%	0.001075%	0.000138%
75%	0.001635%	0.000196%
80%	0.001984%	0.000289%
90%	0.005190%	0.000678%

B.1.3 Variable correlations

Correlations of variables with dependencies are shown below.

Table B.4: Variable correlations indicating dependencies

Measure	Co Employees	Co Turnover	ICT Spend	Telecom ARPU	Telecom Revenue	Telecom Subscribers	Telecom SolutionLines	Telecom ProductLines	Telecom DeviceLines
Co_Employees	1.00	-0.08	-0.08	-0.05	0.01	0.65	0.30	0.30	0.39
Co_Turnover	-0.08	1.00	0.94	0.20	0.37	-0.03	-0.01	-0.01	-0.01
ICT_Spend	-0.08	0.94	1.00	0.20	0.33	-0.04	-0.01	-0.01	-0.01
Telecom_ARPU	-0.05	0.20	0.20	1.00	0.44	-0.04	-0.01	-0.01	-0.02
Telecom_Revenue	0.01	0.37	0.33	0.44	1.00	0.00	0.00	0.00	0.00
Telecom_Subscribers	0.65	-0.03	-0.04	-0.04	0.00	1.00	0.62	0.62	0.77
Telecom_SolutionLines	0.30	-0.01	-0.01	-0.01	0.00	0.62	1.00	1.00	0.99
Telecom_ProductLines	0.30	-0.01	-0.01	-0.01	0.00	0.62	1.00	1.00	0.99
Telecom_DeviceLines	0.39	-0.01	-0.01	-0.02	0.00	0.77	0.99	0.99	1.00

B.2 K-means clustering

The variables used per test run, with their centres per cluster is shown below. For the first and last runs the initial iteration as well as the last iteration where the feasible solution was reached is shown.

Table B.5: Variable centres per cluster for test run KMC1

***Run* 1** $s(1) = 0.39$			*w(1)=0.34*			***Run* 11** $s(11) = 0.46$			*w(11)=0.46*	
Cluster	**1**	**2**	**3**	**4**	**5**	**Cluster**	**1**	**2**	**3**	**4**
Priority	0.364	-1.340	**0.705**	-0.667	0.102	**Priority**	**0.813**	0.126	0.263	**0.824**
SIC_Code	-1.263	0.311	0.373	0.836	-0.232	**SIC_Code**	**0.733**	**0.791**	0.338	0.271
Co_Employees	**0.678**	-0.050	-0.264	-0.339	-0.367	**Co_Employees**	0.046	0.052	0.043	0.050
Co_Turnover	-0.198	-0.165	-0.194	5.243	2.997	**Co_Turnover**	0.021	0.040	0.024	0.018
ICT_Spend	-0.215	-0.162	-0.179	5.786	1.799	**ICT_Spend**	0.021	0.044	0.014	0.009
Objects	**736**	**964**	**1526**	**86**	**50**	**Objects**	**1625**	**809**	**400**	**528**
Feasible clusters	**Y**	**Y**	**Y**			**Feasible clusters**	**Y**	**Y**	**Y**	**Y**

Table B.6: Test runs KMC2 – KMC4 cluster centres

KMC2 $s(7) = 0.66$		*w(7)=0.44*		
Cluster	**1**	**2**	**3**	**4**
ICT_Spend	0.011	**0.523**	0.005	0.002
Perc_ICTUsgPopulation	0.014	0.020	**0.310**	0.106
Telecom_ARPU	0.002	0.011	0.000	0.000
Telecom_Revenue	0.002	0.017	0.001	0.003
Telecom_Subscribers	0.002	0.002	0.030	0.008
Objects	**2578**	**97**	**248**	**439**
Feasible clusters	**Y**			**Y**
Number of runs	**7**			

KMC3 $s(5) = 0.54$			*w(5)=0.54*	
Cluster	**1**	**2**	**3**	**4**
Priority	**0.813**	**0.824**	0.127	0.263
SIC_Code	**0.733**	0.271	**0.791**	**0.337**
Telecom_ARPU	0.0019	0.0020	0.0019	0.0006
Telecom_Revenue	0.0028	0.0034	0.0023	0.0028
Telecom_Subscribers	0.0043	0.0074	0.0056	0.0031
Telecom_SolutionLines	0.0004	0.0022	0.0002	0.0002
Telecom_ProductLines	0.0004	0.0021	0.0002	0.0002
Telecom_DeviceLines	0.0005	0.0021	0.0001	0.0001
Objects	**1625**	**528**	**810**	**399**
Feasible clusters	**Y**	**Y**	**Y**	**Y**
Number of runs	**5**			

KMC4 $s(3) = 0.63$		*w(3)=0.42*		
Cluster	**1**	**2**	**3**	**4**
ICT_Spend	0.005	0.011	0.002	**0.526**
Perc_ICTCountry	0.005	0.011	0.002	**0.526**
Perc_ICTUsgPopulation	**0.310**	0.014	0.106	0.020
Telecom_ARPU	0.00004	0.0019	0.0001	0.0111
Telecom_Revenue	0.0010	0.0024	0.0035	0.0146
Telecom_Subscribers	0.0302	0.0021	0.0081	0.0018
Telecom_SolutionLines	0.0058	0.0002	0.0003	0.0001
Telecom_ProductLines	0.0058	0.0002	0.0003	0.0001
Telecom_DeviceLines	0.0062	0.0001	0.0001	0.0001
Objects	**248**	**2579**	**439**	**96**
Feasible clusters		**Y**	**Y**	
Number of runs	**3**			

Table B.7: Cluster centres for test run KMC5

***Run* 1** $s(1) = 0.57$			*w(1)=0.29*		***Run* 9** $s(9) = 0.43$			*w(9)=0.43*	
Cluster	**1**	**2**	**3**	**4**	**Cluster**	**1**	**2**	**3**	**4**
Priority	-0.533	**0.944**	-0.077	0.025	**Priority**	**0.813**	0.126	0.263	**0.824**
SIC_Code	**0.676**	-1.179	-0.028	-0.020	**SIC_Code**	**0.733**	**0.791**	0.338	0.271
Co_Employees	**-0.334**	4.846	2.836	-0.273	**Co_Employees**	0.046	0.052	0.043	0.050
Co_Turnover	5.158	-0.276	-0.197	-0.154	**Co_Turnover**	0.021	0.040	0.024	0.018
ICT_Spend	5.378	-0.259	-0.195	-0.162	**ICT_Spend**	0.021	0.044	0.014	0.009
Perc_ICTCountry	5.378	-0.259	-0.195	-0.162	**Perc_ICTCountry**	0.021	0.044	0.014	0.009
Perc_ICTUsgPopulation	-0.343	4.855	2.842	-0.274	**Perc_ICTUsgPopulation**	0.046	0.053	0.044	0.051
Telecom_ARPU	0.034	-0.132	-0.131	-0.101	**Telecom_ARPU**	0.002	0.002	0.001	0.002
Telecom_Revenue	0.264	-0.104	-0.156	-0.160	**Telecom_Revenue**	0.003	0.002	0.003	0.003
Telecom_Subscribers	-0.281	15.259	0.059	-0.270	**Telecom_Subscribers**	0.004	0.006	0.003	0.007
Telecom_SolutionLines	-0.076	23.091	-0.041	-0.074	**Telecom_SolutionLines**	0.0004	0.0002	0.0002	0.0022
Telecom_ProductLines	-0.075	23.090	-0.041	-0.074	**Telecom_ProductLines**	0.0004	0.0002	0.0002	0.0021
Telecom_DeviceLines	-0.088	19.688	-0.052	-0.086	**Telecom_DeviceLines**	0.0005	0.0001	0.0001	0.0021
Objects	**100**	**1**	**296**	**2965**	**Objects**	**1625**	**808**	**401**	**528**
Feasible clusters				**Y**	**Feasible clusters**	**Y**	**Y**	**Y**	**Y**

According to the last run in each table above, the centres for Priority and SIC code represent clusters the most distinctly.

A summary of the test run output in terms of feasible clusters, iterations, quality measurements and processing time (runtime) for all KMC test runs are shown below.

Table B.8: KMC test run metrics and run times (mm:ss)

Test (t)	Transformation	Run (r)	Clusters (k)	Feasible clusters (c)	Iterations (j)	Silhouette (*s*)	Weight (*w*)	RunTime
KMC1	None	1	5	3	500	0.94	0.71	00:45
	Z transform	2	5	4	500	0.39	0.35	00:38
	Scale 0 to 1	3	5	4	500	0.46	0.41	01:05
KMC2	None	1	5	1	500	0.65	0.22	00:39
	Z transform	2	5	2	500	0.74	0.42	00:35
	Scale 0 to 1	3	5	2	500	0.79	0.45	00:37
KMC3	None	1	5	2	500	0.91	0.52	00:43
	Z transform	2	5	3	500	0.40	0.30	00:45
	Scale 0 to 1	3	5	4	500	0.54	0.48	00:46
KMC4	None	1	5	1	500	0.87	0.29	00:51
	Z transform	2	5	1	500	0.75	0.25	00:53
	Scale 0 to 1	3	5	2	500	0.67	0.38	00:46
KMC5	None	1	5	1	500	0.88	0.29	01:11
	Z transform	2	5	3	500	0.35	0.26	01:13
	Scale 0 to 1	3	5	4	500	0.43	0.38	01:06

B.3 Particle Swarm Optimisation

Quality metrics for the two PSO run were generated for different number of iterations. The tabled results are shown below.

B.3.1 PSO 1 test runs – Feature-based dimensions

The iterations and global fitness values are tabulated for the standard and two hybrid PSO algorithms used. Each run was done in three sets, namely, with 3 iterations, 15 iterations and 35 iterations.

Table B.9: Standard PSO1 iterations: no *k*-means centres

3 Iterations		
Iteration	Global Fitness	Limit
1	0.217	0.1
2	0.202	0.1
3	0.139	0.1

Last global fitness is 0.1393
No feasible solution

15 Iterations		
Iteration	Global Fitness	Limit
1	0.2683	0.1
2	0.2374	0.1
3	0.1905	0.1
4	0.1488	0.1
5	0.1405	0.1
6	0.1363	0.1
7	0.1249	0.1
8	0.1042	0.1
9	0.1022	0.1
10	0.0990	0.1
11	0.0986	0.1
12	0.0986	0.1
13	0.0986	0.1
14	0.0986	0.1
15	0.0985	0.1

Feasible global fitness is 0.0986
Reached after 10 iterations

35 Iterations		
Iteration	Global Fitness	Limit
1	0.201	0.1
2	0.1769	0.1
3	0.0998	0.1
4	0.0998	0.1
5	0.0998	0.1
6	0.0985	0.1
7	0.0985	0.1
8	0.0985	0.1
9	0.0985	0.1
10	0.0985	0.1
11	0.0985	0.1
12	0.0985	0.1
13	0.0985	0.1
14	0.0985	0.1
15	0.0985	0.1
16	0.0985	0.1
17	0.0985	0.1
18	0.0985	0.1
19	0.0985	0.1
20	0.0985	0.1
21	0.0985	0.1
22	0.0985	0.1
23	0.0985	0.1
24	0.0983	0.1
25	0.0982	0.1
26	0.0982	0.1
27	0.0982	0.1
28	0.0982	0.1
29	0.0982	0.1
30	0.0982	0.1
31	0.0982	0.1
32	0.0982	0.1
33	0.0982	0.1
34	0.0982	0.1
35	0.0982	0.1

Feasible global fitness is 0.0998
Reached after 3 iterations

Table B.10: Hybrid PSO1 iterations: MATLAB *k*-means centres

3 Iterations		
Iteration	Global Fitness	Limit
1	0.170	0.1
2	0.135	0.1
3	0.105	0.1

Last global fitness is 0.105
No feasible solution

15 Iterations		
Iteration	Global Fitness	Limit
1	0.2411	0.1
2	0.2248	0.1
3	0.2248	0.1
4	0.2248	0.1
5	0.1578	0.1
6	0.1578	0.1
7	0.1331	0.1
8	0.1331	0.1
9	0.1040	0.1
10	0.1040	0.1
11	0.1040	0.1
12	0.1040	0.1
13	0.1003	0.1
14	0.1003	0.1
15	0.1003	0.1

Feasible global fitness is 0.1003
Reached after 12 iterations

35 Iterations		
Iteration	Global Fitness	Limit
1	0.1628	0.1
2	0.1394	0.1
3	0.101	0.1
4	0.1010	0.1
5	0.1010	0.1
6	0.1010	0.1
7	0.1005	0.1
8	0.1005	0.1
9	0.1005	0.1
10	0.1005	0.1
11	0.0987	0.1
12	0.0983	0.1
13	0.0983	0.1
14	0.0983	0.1
15	0.0983	0.1
16	0.0983	0.1
17	0.0983	0.1
18	0.0983	0.1
19	0.0983	0.1
20	0.0983	0.1
21	0.0982	0.1
22	0.0982	0.1
23	0.0982	0.1
24	0.0982	0.1
25	0.0982	0.1
26	0.0982	0.1
27	0.0982	0.1
28	0.0982	0.1
29	0.0982	0.1
30	0.0982	0.1
31	0.0982	0.1
32	0.0982	0.1
33	0.0982	0.1
34	0.0982	0.1
35	0.0982	0.1

Last global fitness is 0.0987
Reached after 10 iterations

Table B.11: Hybrid PSO1 iterations: KMC *k*-means centres

3 Iterations		
Iteration	Global Fitness	Limit
1	0.196	0.1
2	0.196	0.1
3	0.196	0.1

Last global fitness is 0.196
No feasible solution

15 Iterations		
Iteration	Global Fitness	Limit
1	0.1964	0.1
2	0.1964	0.1
3	0.1964	0.1
4	0.1090	0.1
5	0.1089	0.1
6	0.1022	0.1
7	0.0997	0.1
8	0.0997	0.1
9	0.0989	0.1
10	0.0989	0.1
11	0.0989	0.1
12	0.0989	0.1
13	0.0987	0.1
14	0.0987	0.1
15	0.0987	0.1

Last global fitness is 0.0997
Reached after 6 iterations

35 Iterations		
Iteration	Global Fitness	Limit
1	0.1964	0.1
2	0.1964	0.1
3	0.1964	0.1
4	0.1090	0.1
5	0.1089	0.1
6	0.1022	0.1
7	0.0997	0.1
8	0.0997	0.1
9	0.0989	0.1
10	0.0989	0.1
11	0.0989	0.1
12	0.0989	0.1
13	0.0987	0.1
14	0.0987	0.1
15	0.0987	0.1
16	0.0987	0.1
17	0.0987	0.1
18	0.0987	0.1
19	0.0987	0.1
20	0.0987	0.1
21	0.0987	0.1
22	0.0987	0.1
23	0.0987	0.1
24	0.0987	0.1
25	0.0985	0.1
26	0.0985	0.1
27	0.0982	0.1
28	0.0982	0.1
29	0.0982	0.1
30	0.0982	0.1
31	0.0982	0.1
32	0.0982	0.1
33	0.0982	0.1
34	0.0982	0.1
35	0.0982	0.1

Last global fitness is 0.0997
Reached after 6 iterations

B.3.2 PSO 2 test runs – Value-based dimensions

The iterations and global fitness values are tabulated for standard PSO, hybrid PSO with initial MATLAB *k*-means centres and hybrid PSO with initial KMC centres. Each run was done with 3 iterations, 15 iterations and 35 iterations.

Table B.12: Standard PSO2 iterations: no *k*-means centres

3 Iterations		
Iteration	Global Fitness	Limit
1	4.366	0.5
2	4.168	0.5
3	1.799	0.5

Last global fitness is 1.799
No feasible solution

l5 Iterations		
Iteration	Global Fitness	Limit
1	3.955	0.5
2	2.735	0.5
3	1.677	0.5
4	1.535	0.5
5	1.364	0.5
6	1.338	0.5
7	1.020	0.5
8	0.765	0.5
9	0.738	0.5
10	0.686	0.5
11	0.656	0.5
12	0.656	0.5
13	0.656	0.5
14	0.644	0.5
15	0.644	0.5

Last global fitness is 0.644
No feasible solution

35 Iterations		
Iteration	Global Fitness	Limit
1	3.1382	0.5
2	2.5352	0.5
3	1.3586	0.5
4	0.9878	0.5
5	0.6995	0.5
6	0.6824	0.5
7	0.6824	0.5
8	0.6824	0.5
9	0.6754	0.5
10	0.6401	0.5
11	0.6401	0.5
12	0.5667	0.5
13	0.5667	0.5
14	0.5667	0.5
15	0.5483	0.5
16	0.5228	0.5
17	0.4904	0.5
18	0.4839	0.5
19	0.4484	0.5
20	0.4253	0.5
21	0.4125	0.5
22	0.3857	0.5
23	0.3857	0.5
24	0.3748	0.5
25	0.3748	0.5
26	0.3748	0.5
27	0.3748	0.5
28	0.3684	0.5
29	0.3578	0.5
30	0.3564	0.5
31	0.3542	0.5
32	0.3527	0.5
33	0.3527	0.5
34	0.3527	0.5
35	0.3525	0.5

Feasible global fitness is 0.4904
Reached after 16 iterations

Table B.13: Hybrid PSO2 iterations: MATLAB *k*-means

3 Iterations		
Iteration	Global Fitness	Limit
1	1.316	0.5
2	1.316	0.5
3	0.603	0.5

Last global fitness is 0.603
No feasible solution

15 Iterations		
Iteration	Global Fitness	Limit
1	4.7533	0.5
2	4.1253	0.5
3	1.7007	0.5
4	1.0633	0.5
5	0.7360	0.5
6	0.4884	0.5
7	0.4322	0.5
8	0.4318	0.5
9	0.4318	0.5
10	0.4318	0.5
11	0.4318	0.5
12	0.4318	0.5
13	0.4036	0.5
14	0.3820	0.5
15	0.3820	0.5

Last global fitness is 0.4884
Reached after 5 iterations

35 Iterations		
Iteration	Global Fitness	Limit
1	4.7533	0.5
2	4.1253	0.5
3	1.7007	0.5
4	1.0633	0.5
5	0.736	0.5
6	0.4884	0.5
7	0.4322	0.5
8	0.4318	0.5
9	0.4318	0.5
10	0.4318	0.5
11	0.4318	0.5
12	0.4318	0.5
13	0.4036	0.5
14	0.382	0.5
15	0.382	0.5
16	0.382	0.5
17	0.382	0.5
18	0.382	0.5
19	0.376	0.5
20	0.3709	0.5
21	0.3708	0.5
22	0.3641	0.5
23	0.3568	0.5
24	0.3551	0.5
25	0.3521	0.5
26	0.3485	0.5
27	0.3477	0.5
28	0.3476	0.5
29	0.3476	0.5
30	0.3476	0.5
31	0.3476	0.5
32	0.3469	0.5
33	0.3469	0.5
34	0.3469	0.5
35	0.3469	0.5

Last global fitness is 0.4884
Reached after 5 iterations

Table B.14: Hybrid PSO2 iterations: KMC *k*-means centres

3 Iterations		
Iteration	Global Fitness	Limit
1	1.316	0.5
2	1.316	0.5
3	0.603	0.5

Last global fitness is 0.603
No feasible solution

15 Iterations		
Iteration	Global Fitness	Limit
1	1.3159	0.5
2	0.7949	0.5
3	0.5057	0.5
4	0.5057	0.5
5	0.5057	0.5
6	0.5057	0.5
7	0.4784	0.5
8	0.4779	0.5
9	0.4779	0.5
10	0.4265	0.5
11	0.4265	0.5
12	0.4265	0.5
13	0.4265	0.5
14	0.4265	0.5
15	0.4265	0.5

Last global fitness is 0.4784
Reached after 6 iterations

35 Iterations		
Iteration	lobal Fitne	Limit
1	1.3159	0.5
2	1.3159	0.5
3	0.5276	0.5
4	0.4462	0.5
5	0.4462	0.5
6	0.4159	0.5
7	0.4159	0.5
8	0.4159	0.5
9	0.4159	0.5
10	0.4159	0.5
11	0.4159	0.5
12	0.3898	0.5
13	0.3898	0.5
14	0.3898	0.5
15	0.3898	0.5
16	0.3898	0.5
17	0.3898	0.5
18	0.3898	0.5
19	0.3898	0.5
20	0.3898	0.5
21	0.3898	0.5
22	0.3733	0.5
23	0.3711	0.5
24	0.3683	0.5
25	0.3628	0.5
26	0.3564	0.5
27	0.3563	0.5
28	0.3563	0.5
29	0.3563	0.5
30	0.355	0.5
31	0.3528	0.5
32	0.3497	0.5
33	0.3497	0.5
34	0.3497	0.5
35	0.3497	0.5

Last global fitness is 0.4462
Reached after 3 iterations

B.3.3 PSO test runs on raw data

The importance of transforming the data before running the clustering algorithms are highlighted by the very poor fitness values when runs were done on raw data, without transformations. From the tables below it can be seen, amongst others, that the POS2 run with initial MATLAB *k*-means centres did not move, each iteration stayed constant.

Table B.15: PSO1 test runs on raw data

PSO1 no Kmeans

15 Iterations		
Iteration	Global Fitness	Limit
1	227799664	0.1
2	212890650	0.1
3	141840695	0.1
4	140430604	0.1
5	140430604	0.1
6	140430604	0.1
7	139001207	0.1
8	135002209	0.1
9	131262520	0.1
10	129051806	0.1
11	129051806	0.1
12	128502200	0.1
13	127310814	0.1
14	126537878	0.1
15	126537878	0.1

Last global fitness is 126537878
No feasible solution

PSO1 MATLAB Kmeans

15 Iterations		
Iteration	Global Fitness	Limit
1	205671558	0.1
2	149711051	0.1
3	131675540	0.1
4	49651261	0.1
5	42084962	0.1
6	41953579	0.1
7	41953579	0.1
8	41953579	0.1
9	41412768	0.1
10	41042579	0.1
11	41042579	0.1
12	41042579	0.1
13	41042579	0.1
14	41023415	0.1
15	40879703	0.1

Last global fitness is 40879703
No feasible solution

PSO1 KMC Kmeans

15 Iterations		
Iteration	Global Fitness	Limit
1	210006822	0.1
2	210006822	0.1
3	210006822	0.1
4	208672047	0.1
5	141595674	0.1
6	141595674	0.1
7	139361520	0.1
8	136883127	0.1
9	133134659	0.1
10	131047437	0.1
11	127755285	0.1
12	127627231	0.1
13	125842295	0.1
14	125259010	0.1
15	125259010	0.1

Last global fitness is 125259010
No feasible solution

Table B.16: PSO2 test runs on raw data

PSO2 no Kmeans

5 Iterations		
Iteration	Global Fitness	Limit
1	30505317	0.5
2	30505317	0.5
3	29242510	0.5
4	10882932	0.5
5	958851	0.5
6	958851	0.5
7	958851	0.5
8	958851	0.5
9	958851	0.5
10	958851	0.5
11	947110	0.5
12	711825	0.5
13	711825	0.5
14	711825	0.5
15	711825	0.5

Last global fitness is 711825
No feasible solution

PSO2 MATLAB Kmeans

15 Iterations		
Iteration	Global Fitness	Limit
1	23434739	0.5
2	23434739	0.5
3	23434739	0.5
4	23434739	0.5
5	23434739	0.5
6	23434739	0.5
7	23434739	0.5
8	23434739	0.5
9	23434739	0.5
10	23434739	0.5
11	23434739	0.5
12	23434739	0.5
13	23434739	0.5
14	23434739	0.5
15	23434739	0.5

Last global fitness is 23434739
No feasible solution

PSO2KMC Kmeans

15 Iterations		
Iteration	Global Fitness	Limit
1	1675698	0.5
2	1675698	0.5
3	1675698	0.5
4	1675698	0.5
5	1675698	0.5
6	1675698	0.5
7	1292047	0.5
8	872786	0.5
9	872786	0.5
10	872786	0.5
11	850036	0.5
12	850036	0.5
13	802415	0.5
14	802415	0.5
15	802415	0.5

Last global fitness is 802415
No feasible solution

B.4 Chi-square AID

As mentioned in Chapter 6 for the analysis, some additional tests were run for CHAID. Additional output, i.e. gains tables, were also mentioned for the last test run. These results from the test runs are shown here. Note that processing completed in a very small fraction of the processing time of other methods, therefore CHAID processing time was approximated as it could not be measured.

B.4.1 Additional CHAID tests

The CHAID1 test run showed a combined decision tree with dependent variable showing values for customers and prospects. In further tests customers and prospects were evaluated separately.

a. Customer classification per industry

A test run was done on the predictor variables ICT spend and Industry to measure counts on the dependent variable classification. The node options used for this test run are shown below.

- Significance level for merging, $\alpha_{merge} = 0.08$
- Significance level for splitting, $\alpha_{split} = 0.05$
- Minimum node size, $m = 10$

The steps to grow the decision tree after the first split are shown here from the output generated by the Easy CHAID online application.

Table B.17: Customer classification per industry steps

Node	Pairs	Grouping of predictor values	Significance probabilities
Node 1	Pair: 0	Members: '$ 1M - $ 9M' x '>= $ 10M'	p-value: 0.42650172045050294
Node 1	Pair: 1	Members: '>= $ 10M' x '< $ 1M'	p-value: 0.5290262822202063
Node 2	Pair: 0	Members: '$ 1M - $ 9M' x '>= $ 10M'	p-value: 1
Node 2	Pair: 1	Members: '>= $ 10M' x '< $ 1M'	p-value: 0.5876145767179657
Node 4	Pair: 0	Members: '$ 1M - $ 9M' x '>= $ 10M'	p-value: 0.08030772655502627
Node 4	Pair: 1	Members: '>= $ 10M' x '< $ 1M'	p-value: 0.0032801554181027814
Node 5	Pair: 0	Members: '$ 1M - $ 9M' x '>= $ 10M'	p-value: 0.0000933330074857075
Node 5	Pair: 1	Members: '>= $ 10M' x '< $ 1M'	p-value: 0.00007983317267656886
Node 6	Pair: 0	Members: '$ 1M - $ 9M' x '>= $ 10M'	p-value: 0.5842247884209305
Node 6	Pair: 1	Members: '>= $ 10M' x '< $ 1M'	p-value: 0.0015014636882695331
Node 6	Pair: 0	Members: 'Health' x 'ICT'	p-value: 0.3968833972104727
Node 6	Pair: 1	Members: 'ICT' x 'Manufacturing'	p-value: 0.6494303394745295
Node 6	Pair: 2	Members: 'Manufacturing' x 'Media & Publishing'	p-value: 0.43237187188447057
Node 6	Pair: 3	Members: 'Media & Publishing' x 'Mining'	p-value: 0.6164294836495663
Node 6	Pair: 4	Members: 'Mining' x 'Retail, Wholesale, Trade'	p-value: 0.6679462080582619
Node 6	Pair: 0	Members: 'Health' x 'ICT'	p-value: 0.3968833972104727
Node 6	Pair: 1	Members: 'ICT' x 'Manufacturing'	p-value: 0.6494303394745295
Node 6	Pair: 2	Members: 'Manufacturing' x 'Media & Publishing'	p-value: 0.43237187188447057
Node 6	Pair: 3	Members: 'Media & Publishing' x 'Mining' 'Retail, Wholesale, Trade'	p-value: 0.2694959301975628
Node 6	Pair: 0	Members: 'Health' x 'ICT' 'Manufacturing'	p-value: 0.5669256048707656
Node 6	Pair: 1	Members: 'ICT' 'Manufacturing' x 'Media & Publishing'	p-value: 0.41879603704682933
Node 6	Pair: 2	Members: 'Media & Publishing' x 'Mining' 'Retail, Wholesale, Trade'	p-value: 0.2694959301975628
Node 6	Pair: 0	Members: 'Health' 'ICT' 'Manufacturing' x 'Media & Publishing'	p-value: 0.3829574030815167
Node 6	Pair: 1	Members: 'Media & Publishing' x 'Mining' 'Retail, Wholesale, Trade'	p-value: 0.2694959301975628
Node 7	Pair: 0	Members: '$ 1M - $ 9M' x '>= $ 10M'	p-value: 0.21190941713958533
Node 7	Pair: 1	Members: '>= $ 10M' x '< $ 1M'	p-value: 0.22269568203783174
Node 7	Pair: 0	Members: 'Tourism' x 'Transport'	p-value: 0.6148714601434004
Node 7	Pair: 1	Members: 'Transport' x 'Utilities'	p-value: 0.2596831968888884
Node 11	Pair: 0	Members: 'Health' x 'ICT'	p-value: 0.06207501135240279
Node 11	Pair: 1	Members: 'ICT' x 'Manufacturing'	p-value: 0.5997423679444169
Node 11	Pair: 2	Members: 'Manufacturing' x 'Media & Publishing'	p-value: 0.15207689149586057
Node 11	Pair: 3	Members: 'Media & Publishing' x 'Mining'	p-value: 0.1763022273922823
Node 11	Pair: 4	Members: 'Mining' x 'Retail, Wholesale, Trade'	p-value: 0.7943389712984114
Node 11	Pair: 0	Members: 'Health' x 'ICT'	p-value: 0.06207501135240279
Node 11	Pair: 1	Members: 'ICT' x 'Manufacturing'	p-value: 0.5997423679444169
Node 11	Pair: 2	Members: 'Manufacturing' x 'Media & Publishing'	p-value: 0.15207689149586057
Node 11	Pair: 3	Members: 'Media & Publishing' x 'Mining' 'Retail, Wholesale, Trade'	p-value: 0.11387742901517228
Node 11	Pair: 0	Members: 'Health' x 'ICT' 'Manufacturing'	p-value: 0.14743715853592687
Node 11	Pair: 1	Members: 'ICT' 'Manufacturing' x 'Media & Publishing'	p-value: 0.07248302883332669
Node 11	Pair: 2	Members: 'Media & Publishing' x 'Mining' 'Retail, Wholesale, Trade'	p-value: 0.11387742901517228
Node 11	Pair: 0	Members: 'Health' 'ICT' 'Manufacturing' x 'Media & Publishing'	p-value: 0.13553900861858947
Node 11	Pair: 1	Members: 'Media & Publishing' x 'Mining' 'Retail, Wholesale, Trade'	p-value: 0.11387742901517228
Node 12	Pair: 0	Members: 'Health' x 'ICT'	p-value: 0.415720846161313
Node 12	Pair: 1	Members: 'ICT' x 'Manufacturing'	p-value: 0.3765042942133766
Node 12	Pair: 2	Members: 'Manufacturing' x 'Media & Publishing'	p-value: 0.7216483501517859
Node 12	Pair: 3	Members: 'Media & Publishing' x 'Mining'	p-value: 0.6710490341766344
Node 12	Pair: 4	Members: 'Mining' x 'Retail, Wholesale, Trade'	p-value: 0.3370110780371157
Node 12	Pair: 0	Members: 'Health' x 'ICT'	p-value: 0.415720846161313
Node 12	Pair: 1	Members: 'ICT' x 'Manufacturing' 'Media & Publishing'	p-value: 0.3200702461864804
Node 12	Pair: 2	Members: 'Manufacturing' 'Media & Publishing' x 'Mining'	p-value: 0.09507051017513624
Node 12	Pair: 3	Members: 'Mining' x 'Retail, Wholesale, Trade'	p-value: 0.3370110780371157
Node 12	Pair: 0	Members: 'Health' 'ICT' x 'Manufacturing' 'Media & Publishing'	p-value: 0.4558239529323438
Node 12	Pair: 1	Members: 'Manufacturing' 'Media & Publishing' x 'Mining'	p-value: 0.09507051017513624
Node 12	Pair: 2	Members: 'Mining' x 'Retail, Wholesale, Trade'	p-value: 0.3370110780371157
Node 12	Pair: 0	Members: 'Health' 'ICT' 'Manufacturing' 'Media & Publishing' x 'Mining'	p-value: 0.18521752786810797
Node 12	Pair: 1	Members: 'Mining' x 'Retail, Wholesale, Trade'	p-value: 0.3370110780371157

The decision tree shows the split of the nodes on Industry and ICT spend.

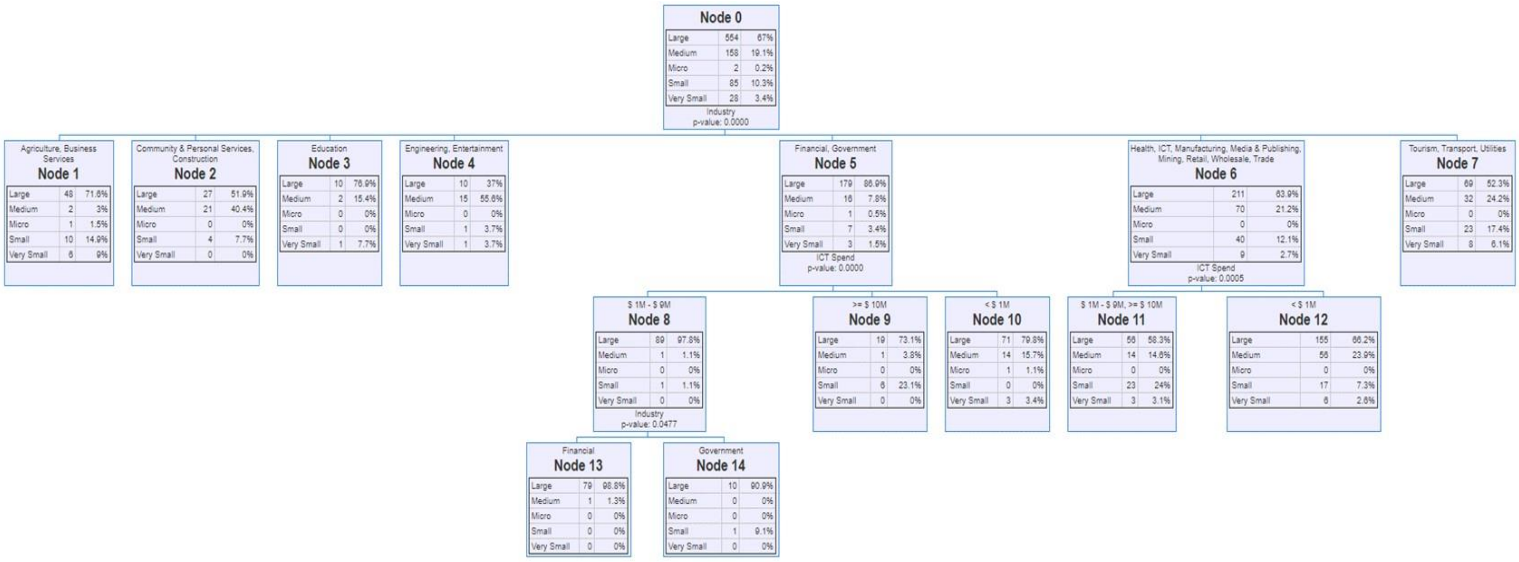


Figure B.1: Customer classifications per Industry tree diagram

b. Prospects classification per industry

The same predictor and dependent variables were used here, but only for prospects. The node options for this test run are shown below.

- Significance level for merging, $\alpha_{merge} = 0.05$
- Significance level for splitting, $\alpha_{split} = 1.8$
- Minimum node size, $m = 30$

The steps to create the terminating nodes of the decision tree are shown here in raw format, from the output generated by the Easy CHAID online application.

```
| Node 18
| ICT Spend
| Industry
| Node 18 cannot be split because all variables are the same or because splitting would generate groups smaller than the minimum node size.
| Node 19
| ICT Spend
| Industry
| Node 19 cannot be split because all variables are the same or because splitting would generate groups smaller than the minimum node size.
| Node 20
| ICT Spend
| Contingency table:
| Var: ICT Spend | test stat: 19.487179487179482 | p-value: 0.0006303215513641103
| p-value with Bonferroni adjustment: 0.0012606431027282206
| Industry
| Node 20 cannot be split because all variables are the same or because splitting would generate groups smaller than the minimum node size.
| Node 21
| ICT Spend
| Contingency table:
| Var: ICT Spend | test stat: 1.7106507193483547 | p-value: 0.6345684987119369
| p-value with Bonferroni adjustment: 1.2691369974238738
| Industry
| Since Industry in node 21 have more than 2 categories, attempting to merge similar categories...
| | Pair: 0 | Members: 'Education' x 'Health' | p-value: 0.09316283614266196
| | Pair: 1 | Members: 'Education' x 'Media & Publishing' | p-value: 0.5626712842861308
| | Pair: 2 | Members: 'Education' x 'Mining' | p-value: 0.41297523751008824
| | Pair: 3 | Members: 'Education' x 'Utilities' | p-value: 0.8055147514217518
| | Pair: 4 | Members: 'Health' x 'Media & Publishing' | p-value: 0.08585911386424039
| | Pair: 5 | Members: 'Health' x 'Mining' | p-value: 0.1838559527240401
| | Pair: 6 | Members: 'Health' x 'Utilities' | p-value: 0.1693779481403389
| | Pair: 7 | Members: 'Media & Publishing' x 'Mining' | p-value: 0.38623346348882204
| | Pair: 8 | Members: 'Media & Publishing' x 'Utilities' | p-value: 0.5953789108499024
| | Pair: 9 | Members: 'Mining' x 'Utilities' | p-value: 0.8441139453879911
| Highest p-value: 0.8441139453879911 in pair: 9
| | Pair: 0 | Members: 'Education' x 'Health' | p-value: 0.09316283614266196
| | Pair: 1 | Members: 'Education' x 'Media & Publishing' | p-value: 0.5626712842861308
| | Pair: 2 | Members: 'Education' x 'Mining' 'Utilities' | p-value: 0.3168331422674032
| | Pair: 3 | Members: 'Health' x 'Media & Publishing' | p-value: 0.08585911386424039
| | Pair: 4 | Members: 'Health' x 'Mining' 'Utilities' | p-value: 0.18594117672845534
| | Pair: 5 | Members: 'Media & Publishing' x 'Mining' 'Utilities' | p-value: 0.23595263090631136
| Highest p-value: 0.5626712842861308 in pair: 1
| | Pair: 0 | Members: 'Education' 'Media & Publishing' x 'Health' | p-value: 0.08801345992719023
| | Pair: 1 | Members: 'Education' 'Media & Publishing' x 'Mining' 'Utilities' | p-value: 0.31446595737637795
| | Pair: 2 | Members: 'Health' x 'Mining' 'Utilities' | p-value: 0.18594117672845534
| Highest p-value: 0.31446595737637795 in pair: 1
| Merging categories stopped because resulted in only 2 merged categories
| Contingency table:
| Var: Industry | test stat: 5.688102251709218 | p-value: 0.1278110169323189
| p-value with Bonferroni adjustment: 1.9171652539847832
| Node 21 cannot be split because all variables are the same or because splitting would generate groups smaller than the minimum node size.
| Node 22
| ICT Spend
| Industry
| Node 22 cannot be split because all variables are the same or because splitting would generate groups smaller than the minimum node size.
| Node 23
| ICT Spend
| Industry
| Node 23 cannot be split because all variables are the same or because splitting would generate groups smaller than the minimum node size.
```

Figure B.2: Creating terminating nodes on prospect classifications

The decision tree shows the terminating nodes (Node 18 – 23) and split of the nodes on Industry and ICT spend.

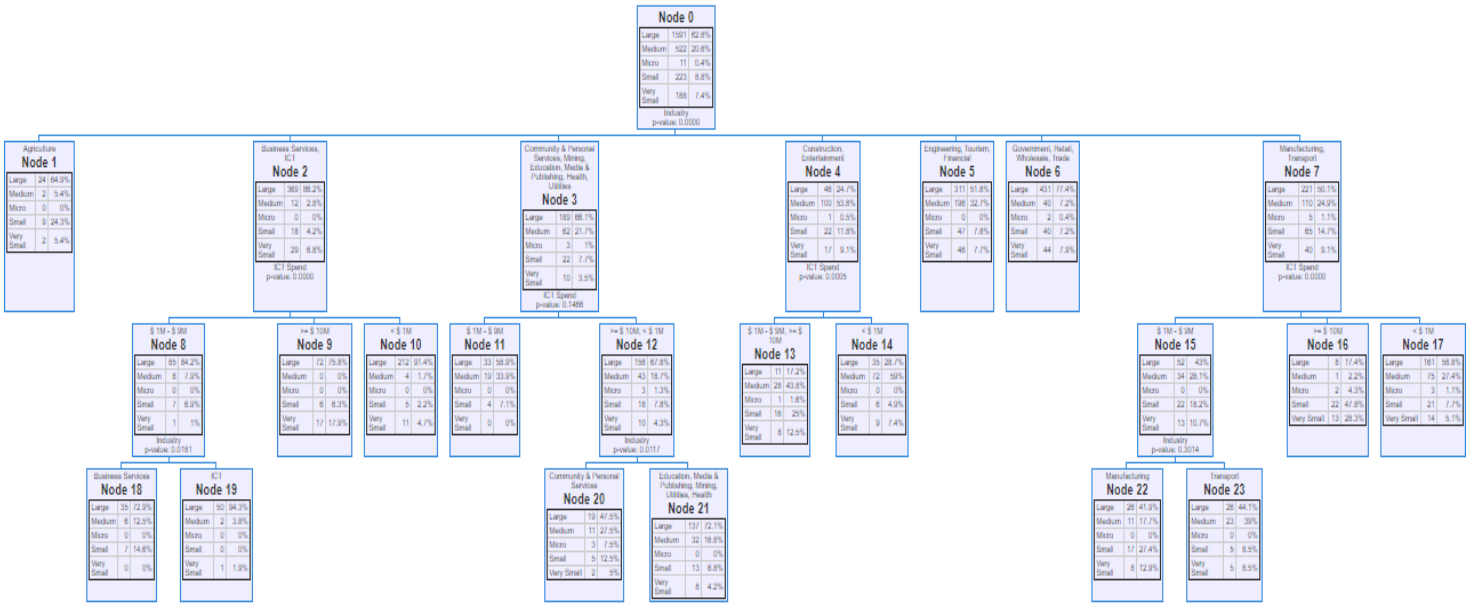


Figure B.3: Customer classifications per Industry tree diagram

c. Customer ICT spend per industry and device type

The predictor variables Telecom device type, Industry and Telecom subscribers were used here, with measurements on the dependent variable ICT spend. The node options for this test run are shown below.

- Significance level for merging, $\alpha_{merge} = 0.5$
- Significance level for splitting, $\alpha_{split} = 0.8$
- Minimum node size, $m = 30$

The steps to create the last split of the decision tree are shown here from the output generated by the Easy CHAID online application.

Table B.18: Last split for Customer ICT spend per industry

Node	Pairs	Grouping of predictor values	Significance probabilities
Node 13	Pair: 0	Members: 'Mobile Handset' x 'Mobile Hotspot'	p-value: 0.5595442219311644
Node 13	Pair: 1	Members: 'Mobile Hotspot' x 'Router'	p-value: 0.3613104285261789
Node 13	Pair: 2	Members: 'Router' x 'Tablet'	p-value: 0.6925693242051977
Node 13	Pair: 3	Members: 'Tablet' x 'USB Modem'	p-value: 0.8265654376242381
Node 13	Pair: 0	Members: 'Mobile Handset' x 'Mobile Hotspot'	p-value: 0.5595442219311644
Node 13	Pair: 1	Members: 'Mobile Hotspot' x 'Router'	p-value: 0.3613104285261789
Node 13	Pair: 2	Members: 'Router' x 'Tablet' 'USB Modem'	p-value: 0.7316156289466418
Node 13	Pair: 0	Members: 'Mobile Handset' x 'Mobile Hotspot'	p-value: 0.5595442219311644
Node 13	Pair: 1	Members: 'Mobile Hotspot' x 'Router' 'Tablet' 'USB Modem'	p-value: 0.5571058618121738
Node 13	Pair: 0	Members: '>=50' x '10-24'	p-value: 0.04964296696826698
Node 13	Pair: 1	Members: '10-24' x '25-49'	p-value: 0.18301841584547507
Node 13	Pair: 2	Members: '25-49' x '5-9'	p-value: 0.6768496439102307
Node 13	Pair: 0	Members: '>=50' x '10-24'	p-value: 0.04964296696826698
Node 13	Pair: 1	Members: '10-24' x '25-49' '5-9'	p-value: 0.10843887514694506
Node 14	Pair: 0	Members: 'Mobile Handset' x 'Mobile Hotspot'	p-value: 0.5181235251706815
Node 14	Pair: 1	Members: 'Mobile Hotspot' x 'Modem'	p-value: 0.34993774911115527
Node 14	Pair: 2	Members: 'Modem' x 'Module'	p-value: 0.5761501220305789
Node 14	Pair: 3	Members: 'Module' x 'Router'	p-value: 0.3864762307712327
Node 14	Pair: 4	Members: 'Router' x 'Tablet'	p-value: 0.7093881150142266
Node 14	Pair: 5	Members: 'Tablet' x 'USB Modem'	p-value: 0.0050017993807101035
Node 14	Pair: 0	Members: 'Mobile Handset' x 'Mobile Hotspot'	p-value: 0.5181235251706815
Node 14	Pair: 1	Members: 'Mobile Hotspot' x 'Modem'	p-value: 0.34993774911115527
Node 14	Pair: 2	Members: 'Modem' x 'Module'	p-value: 0.5761501220305789
Node 14	Pair: 3	Members: 'Module' x 'Router' 'Tablet'	p-value: 0.27332167829229836
Node 14	Pair: 4	Members: 'Router' 'Tablet' x 'USB Modem'	p-value: 0.0061698993205441255
Node 14	Pair: 0	Members: 'Mobile Handset' x 'Mobile Hotspot'	p-value: 0.5181235251706815
Node 14	Pair: 1	Members: 'Mobile Hotspot' x 'Modem' 'Module'	p-value: 0.3148825535494548
Node 14	Pair: 2	Members: 'Modem' 'Module' x 'Router' 'Tablet'	p-value: 0.19670560245894708
Node 14	Pair: 3	Members: 'Router' 'Tablet' x 'USB Modem'	p-value: 0.0061698993205441255
Node 14	Pair: 0	Members: 'Mobile Handset' 'Mobile Hotspot' x 'Modem' 'Module'	p-value: 0.4764153364235376
Node 14	Pair: 1	Members: 'Modem' 'Module' x 'Router' 'Tablet'	p-value: 0.19670560245894708
Node 14	Pair: 2	Members: 'Router' 'Tablet' x 'USB Modem'	p-value: 0.0061698993205441255
Node 14	Pair: 0	Members: 'Mobile Handset' 'Mobile Hotspot' 'Modem' 'Module' x 'Router' 'Tablet'	p-value: 0.2316256322868422
Node 14	Pair: 1	Members: 'Router' 'Tablet' x 'USB Modem'	p-value: 0.0061698993205441255
Node 14	Pair: 0	Members: '>=50' x '1-4'	p-value: 0.7479950291067547
Node 14	Pair: 1	Members: '1-4' x '10-24'	p-value: 0.8703247258333906
Node 14	Pair: 2	Members: '10-24' x '25-49'	p-value: 0.4276524506600088
Node 14	Pair: 0	Members: '>=50' x '1-4' '10-24'	p-value: 0.6087328291682352
Node 14	Pair: 1	Members: '1-4' '10-24' x '25-49'	p-value: 0.4071014708613435

The decision tree shows the last split on node 12 (node 13 and 14) and split of the nodes on Industry and Telecom device type and Telecom subscribers.

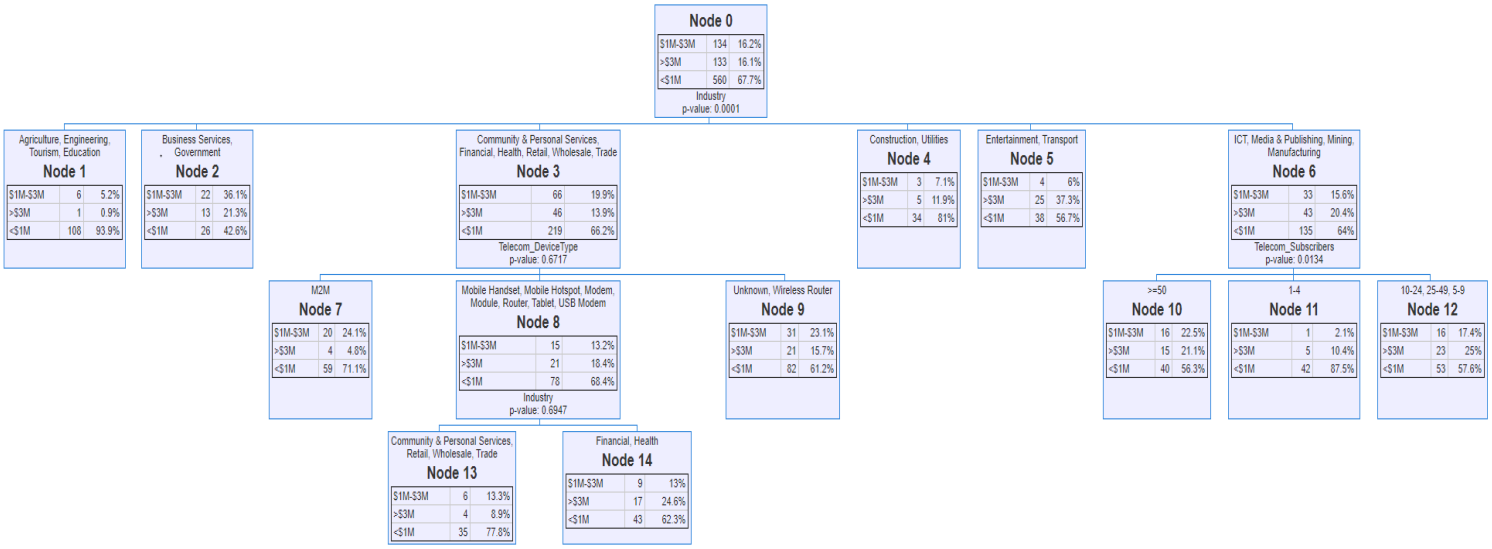


Figure B.4: Customer ICT spend per Industry and Device type

B.4.2 Additional CHAID output

For the final test run, CHAID4, steps to create the decision tree and additional gains tables are shown here.

a. Steps for CHAID4 decision tree

The results of this run shows a decision tree with a dependent variable that flags Customer or Prospect, and predictor variables Company employees, ICT spend and Region.

The steps to grow the decision tree on the predictor Region are shown here.

Table B.19: Last split of ICT spend on region in CHAID4

Node	Pairs	Grouping of predictor values	Significance probabilities
Node 12	Pair: 0	Members: 'Central' x 'Coast'	p-value: 0.5581846494226577
Node 12	Pair: 1	Members: 'Central' x 'Dar Es Salaam'	p-value: 0.6254596668156873
Node 12	Pair: 2	Members: 'Central' x 'Lake District'	p-value: 0.07103122859545696
Node 12	Pair: 3	Members: 'Central' x 'North'	p-value: 0.5491849631348769
Node 12	Pair: 4	Members: 'Central' x 'South West'	p-value: 0.05777957112359722
Node 12	Pair: 5	Members: 'Coast' x 'Dar Es Salaam'	p-value: 0.7647942500847046
Node 12	Pair: 6	Members: 'Coast' x 'Lake District'	p-value: 0.18470755633593727
Node 12	Pair: 7	Members: 'Coast' x 'North'	p-value: 0.14277243461411881
Node 12	Pair: 8	Members: 'Coast' x 'South West'	p-value: 0.12471041265268423
Node 12	Pair: 9	Members: 'Dar Es Salaam' x 'Lake District'	p-value: 0.049610247677007724
Node 12	Pair: 10	Members: 'Dar Es Salaam' x 'North'	p-value: 0.050121383809357045
Node 12	Pair: 11	Members: 'Dar Es Salaam' x 'South West'	p-value: 0.05640693427320442
Node 12	Pair: 12	Members: 'Lake District' x 'North'	p-value: 0.00348343554388586
Node 12	Pair: 13	Members: 'Lake District' x 'South West'	p-value: 0.4601809354471206
Node 12	Pair: 14	Members: 'North' x 'South West'	p-value: 0.006060707369366147
Node 12	Pair: 0	Members: 'Central' x 'Coast'	p-value: 0.5581846494226577
Node 12	Pair: 1	Members: 'Central' x 'Dar Es Salaam'	p-value: 0.6254596668156873
Node 12	Pair: 2	Members: 'Central' x 'Lake District' 'South West'	p-value: 0.022121314679261483
Node 12	Pair: 3	Members: 'Central' x 'North'	p-value: 0.5491849631348769
Node 12	Pair: 4	Members: 'Coast' x 'Dar Es Salaam'	p-value: 0.7647942500847046
Node 12	Pair: 5	Members: 'Coast' x 'Lake District' 'South West'	p-value: 0.07498240375843501
Node 12	Pair: 6	Members: 'Coast' x 'North'	p-value: 0.14277243461411881
Node 12	Pair: 7	Members: 'Dar Es Salaam' x 'Lake District' 'South West'	p-value: 0.00844720668374177
Node 12	Pair: 8	Members: 'Dar Es Salaam' x 'North'	p-value: 0.050121383809357045
Node 12	Pair: 9	Members: 'Lake District' 'South West' x 'North'	p-value: 0.0003904560502286935 6
Node 13	Pair: 0	Members: 'Central' x 'Dar Es Salaam'	p-value: 0.3884622611010973
Node 13	Pair: 1	Members: 'Central' x 'North'	p-value: 0.15729920705028533
Node 13	Pair: 2	Members: 'Dar Es Salaam' x 'North'	p-value: 0.27426057534756987
Node 14	Pair: 0	Members: 'Central' x 'Coast'	p-value: 0.49015296041582535
Node 14	Pair: 1	Members: 'Central' x 'Dar Es Salaam'	p-value: 0.8940098361017719
Node 14	Pair: 2	Members: 'Central' x 'Lake District'	p-value: 0.1868575026630407
Node 14	Pair: 3	Members: 'Central' x 'North'	p-value: 0.6357884468154802
Node 14	Pair: 4	Members: 'Central' x 'South West'	p-value: 0.5049850750938459
Node 14	Pair: 5	Members: 'Coast' x 'Dar Es Salaam'	p-value: 0.2184196323283769
Node 14	Pair: 6	Members: 'Coast' x 'Lake District'	p-value: 0.022821503671488053
Node 14	Pair: 7	Members: 'Coast' x 'North'	p-value: 0.11272987360132947
Node 14	Pair: 8	Members: 'Coast' x 'South West'	p-value: 0.6861678498552393
Node 14	Pair: 9	Members: 'Dar Es Salaam' x 'Lake District'	p-value: 0.051459677157791095
Node 14	Pair: 10	Members: 'Dar Es Salaam' x 'North'	p-value: 0.31840188878558484
Node 14	Pair: 11	Members: 'Dar Es Salaam' x 'South West'	p-value: 0.4434448784226176
Node 14	Pair: 12	Members: 'Lake District' x 'North'	p-value: 0.1910707321856756
Node 14	Pair: 13	Members: 'Lake District' x 'South West'	p-value: 0.1213352503584827
Node 14	Pair: 14	Members: 'North' x 'South West'	p-value: 0.34739684126343684
Node 14	Pair: 0	Members: 'Central' 'Dar Es Salaam' x 'Coast'	p-value: 0.2194048731549374
Node 14	Pair: 1	Members: 'Central' 'Dar Es Salaam' x 'Lake District'	p-value: 0.05097883652977375
Node 14	Pair: 2	Members: 'Central' 'Dar Es Salaam' x 'North'	p-value: 0.3149795787419146
Node 14	Pair: 3	Members: 'Central' 'Dar Es Salaam' x 'South West'	p-value: 0.44403713854025206
Node 14	Pair: 4	Members: 'Coast' x 'Lake District'	p-value: 0.022821503671488053
Node 14	Pair: 5	Members: 'Coast' x 'North'	p-value: 0.11272987360132947
Node 14	Pair: 6	Members: 'Coast' x 'South West'	p-value: 0.6861678498552393
Node 14	Pair: 7	Members: 'Lake District' x 'North'	p-value: 0.1910707321856756
Node 14	Pair: 8	Members: 'Lake District' x 'South West'	p-value: 0.1213352503584827
Node 14	Pair: 9	Members: 'North' x 'South West'	p-value: 0.34739684126343684
Node 14	Pair: 0	Members: 'Central' 'Dar Es Salaam' x 'Coast' 'South West'	p-value: 0.15738252961529198
Node 14	Pair: 1	Members: 'Central' 'Dar Es Salaam' x 'Lake District'	p-value: 0.05097883652977375
Node 14	Pair: 2	Members: 'Central' 'Dar Es Salaam' x 'North'	p-value: 0.3149795787419146
Node 14	Pair: 3	Members: 'Coast' 'South West' x 'Lake District'	p-value: 0.014943431380168182
Node 14	Pair: 4	Members: 'Coast' 'South West' x 'North'	p-value: 0.07724095789929186
Node 14	Pair: 5	Members: 'Lake District' x 'North'	p-value: 0.1910707321856756

b. Additional gains tables from CHAID4 test run

The analysis of the CHAID4 test run output resulted in a total gains table as described in section 4.4.3f. Two additional gains tables were created to evaluate customers and prospects separately. These are shown here.

Table B.20: CHAID4 gains table for the customer base

Segment ID	Segment Count	Percent of Total	Average ICT Spend ($ '000)	Segment Index	Cum. Count	Cum. Percent	Cum. $ ICT Spend	Cum. Index
5	12	1.50%	66 772.10	1 439	12	1.50%	66772.08	1 439
2	12	1.50%	48 677.49	1 049	24	2.90%	57724.79	1 244
14	10	1.20%	32 106.41	692	34	4.10%	50189.97	1 082
8	96	11.60%	7 482.22	161	130	15.70%	18651.94	402
11	41	5.00%	3 571.34	77	171	20.70%	15036.12	324
12	11	1.30%	3 482.75	75	182	22.00%	14337.84	309
4	23	2.80%	3 424.54	74	205	24.80%	13113.42	283
1	33	4.00%	3 357.17	72	238	28.80%	11760.67	253
7	339	41.00%	2 822.92	61	577	69.80%	6509.55	140
9	2	0.20%	2 567.30	55	579	70.00%	6495.93	140
10	4	0.50%	2 463.67	53	583	70.50%	6468.27	139
13	5	0.60%	2 023.59	44	588	71.10%	6430.47	139
3	119	14.40%	258.51	6	707	85.50%	5391.63	116
6	37	4.50%	248.47	5	744	90.00%	5135.85	111
15	83	10.00%	198.19	4	827	100.00%	4640.29	100

Table B.21: CHAID4 gains table for the prospect base

Segment ID	Segment Count	Percent of Total	Average ICT Spend ($ '000)	Segment Index	Cumulative Count	Cumulative Percent	Cumulative $ ICT Spend	Cumulative Index
2	5	0.2%	51 331.51	843	5	0.2%	51331.6	843
5	62	2.4%	37 355.59	614	67	2.6%	38398.57	631
14	8	0.3%	22 566.28	371	75	3.0%	36709.79	603
7	1 143	45.1%	8 946.09	147	1 218	48.0%	10655.68	175
8	211	8.3%	6 370.72	105	1 429	56.4%	10022.98	165
4	72	2.8%	4 307.81	71	1 501	59.2%	9748.84	160
1	120	4.7%	4 030.42	66	1 621	63.9%	9325.51	153
11	39	1.5%	3 550.95	58	1 660	65.5%	9189.84	151
13	14	0.6%	3 283.50	54	1 674	66.0%	9140.45	150
12	1	0.0%	2 219.20	36	1 675	66.1%	9136.32	150
9	3	0.1%	1 663.81	27	1 678	66.2%	9122.96	150
10	3	0.1%	1 417.38	23	1 681	66.3%	9109.2	150
15	134	5.3%	213.82	4	1 815	71.6%	8452.47	139
6	125	4.9%	188.32	3	1 940	76.5%	7919.98	130
3	595	23.5%	117.37	2	2 535	100.0%	6088.6	100

B.5 Artificial Neural Networks

The network diagrams, and learning curves are shown here for the ANN1 and ANN2 training batches. As part of the network design, the most significant features (principle components, or highest priority variables) are key, therefore the input importance of the variables are also shown. The figures below were combined from the output of JustNN from Neural Planner Software, Copyright © 2002-2016 (Wolstenholme, 2002).

B.5.1 ANN1 training batches

For purposes of classification (not pattern recognition) sufficient examples for training of the total analysis dataset of 3362 records were used (15% to 25% sample rate).

The normalised batch training data was tested on 889 examples with one hidden layer and learning rate of 0.6 and momentum of 0.8. Training was done for one output (batch 1.1a) and for 4 outputs (batch 1.1b). To compare, standard data (without transformation) was used for training in similar way for batches 1.1c and 1.1d. For the same settings as above the results proved the same for batches 1.1c and 1.1d. The number of hidden layers were therefore adjusted to more than one, and learning rate and momentum settings were tested until a learning rate of 0.4 and momentum of 0.2 were used, as this resulted in the lowest training error. An additional iteration was run in batch 1.1d, after reseeding random numbers to initialise weights again randomly between -0.5 and +0.5.

Batches 1.2 and 1.3 were run as validation by using the same hyper-parameters on a network with one hidden layer, one output layer and no data transformations. Up to this point 889 examples were used. To finally assess the output, two iterations were done on different random sample batches of 500 examples each. The first iteration used transformed data (batch 1.4a) and the last iteration had no transformation (batch 1.4b).

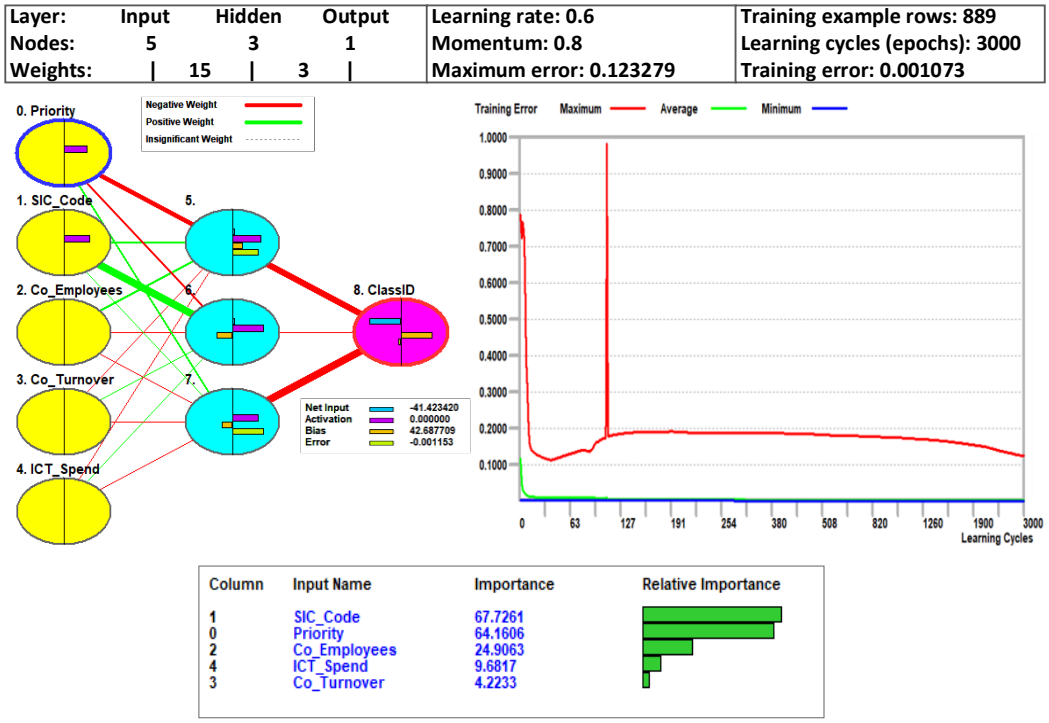


Figure B.5: Learning batch 1.1a (normalised), 1 output neuron

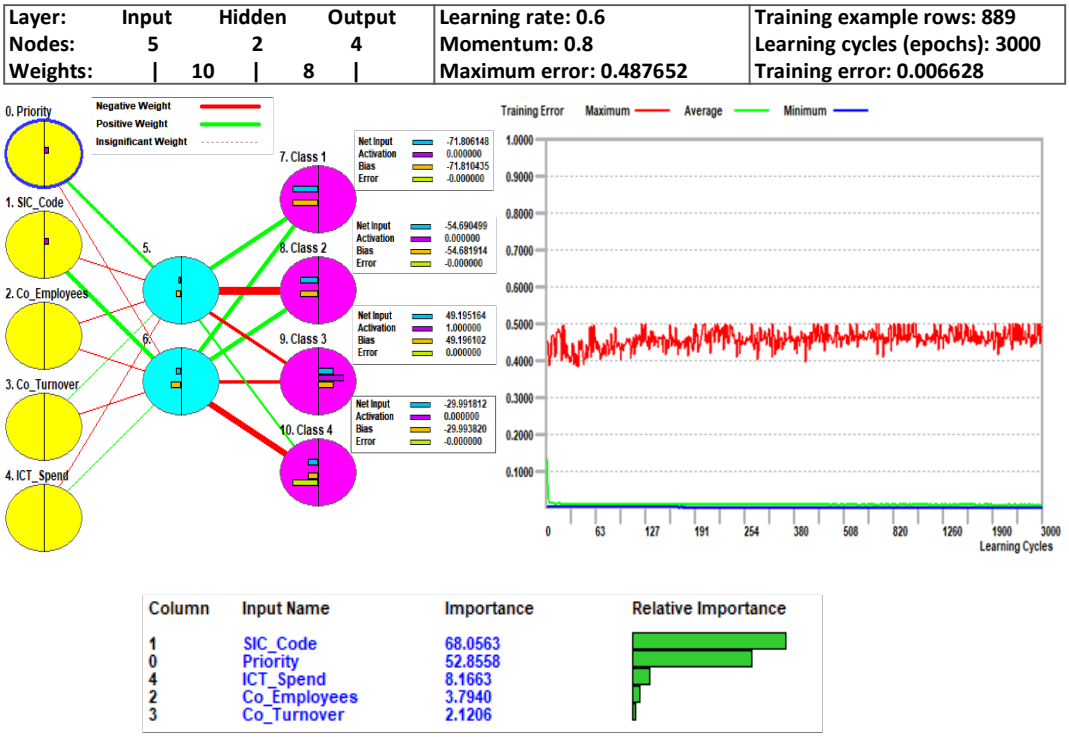


Figure B.6: Learning batch 1.1b (normalised), 4 output neurons

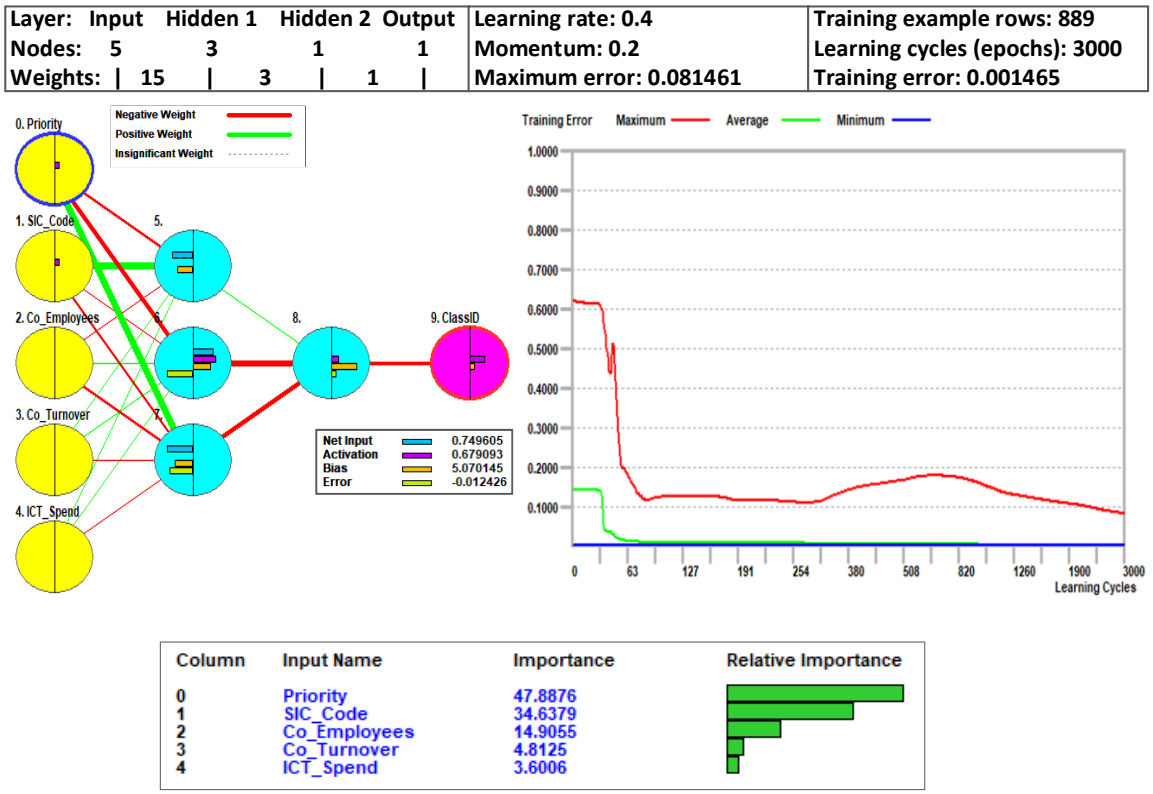
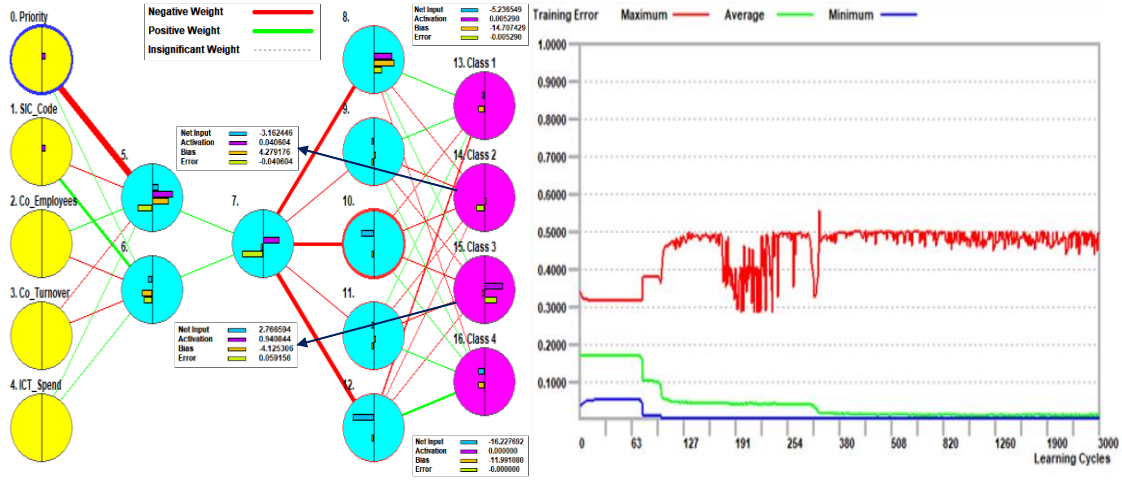


Figure B.7: Learning batch 1.1c (standard), 1 output neuron

Layer:	Input	Hidden 1	Hidden 2	Hidden 3	Output
Nodes:	5	2	1	5	4
Weights:	10	2	5	20	

Learning rate: 0.4
Momentum: 0.2
Maximum error: 0.494795

Training example rows: 889
Learning cycles (epochs): 3000
Training error: 0.009395



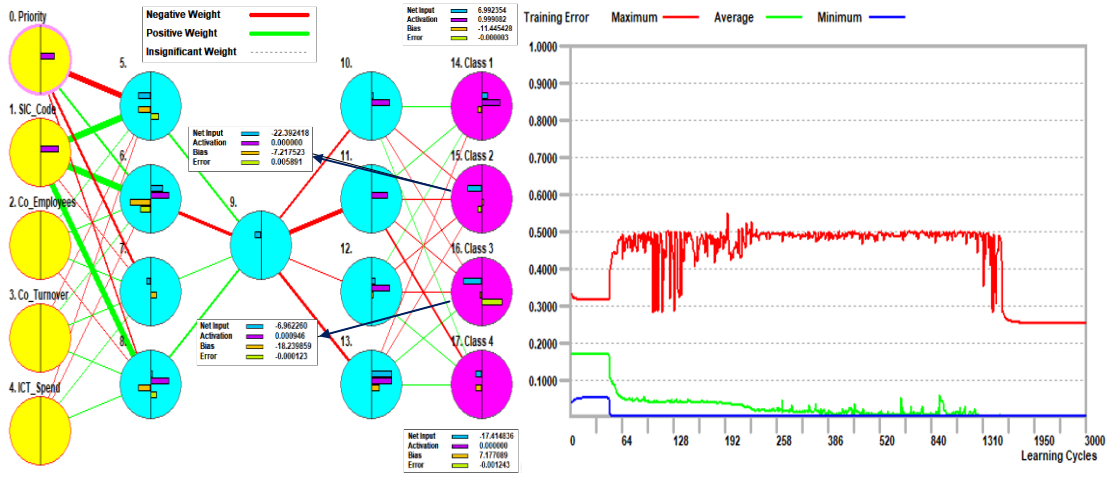
Column	Input Name	Importance	Relative Importance
0	Priority	62.7253	
1	SIC_Code	34.5578	
2	Co_Employees	16.8036	
3	Co_Turnover	13.9211	
4	ICT_Spend	3.3155	

Figure B.8a: Learning batch 1.1d (standard) 1st run, 4 output neurons

Layer:	Input	Hidden 1	Hidden 2	Hidden 3	Output
Nodes:	5	4	1	4	4
Weights:	20	4	4	16	

Learning rate: 0.4
Momentum: 0.2
Maximum error: 0.250454

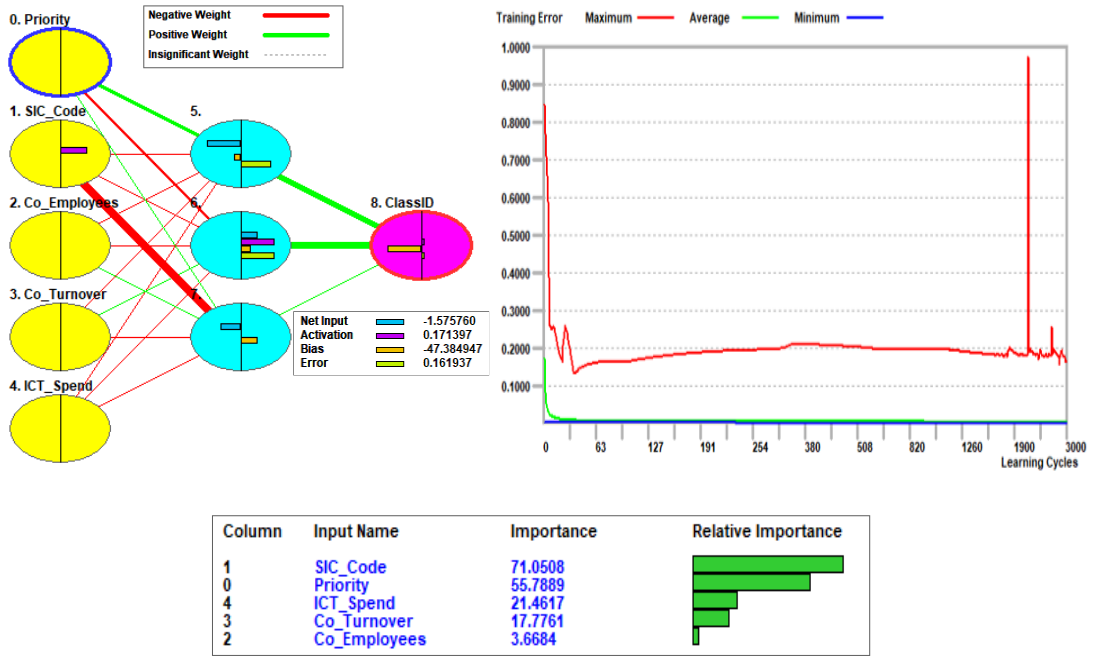
Training example rows: 889
Learning cycles (epochs): 3000
Training error: 0.000849



Column	Input Name	Importance	Relative Importance
1	SIC_Code	167.0312	
0	Priority	115.6027	
2	Co_Employees	12.3972	
4	ICT_Spend	9.0770	
3	Co_Turnover	3.2179	

Figure B.8b: Learning batch 1.1d (standard) 2nd run, 4 output neurons

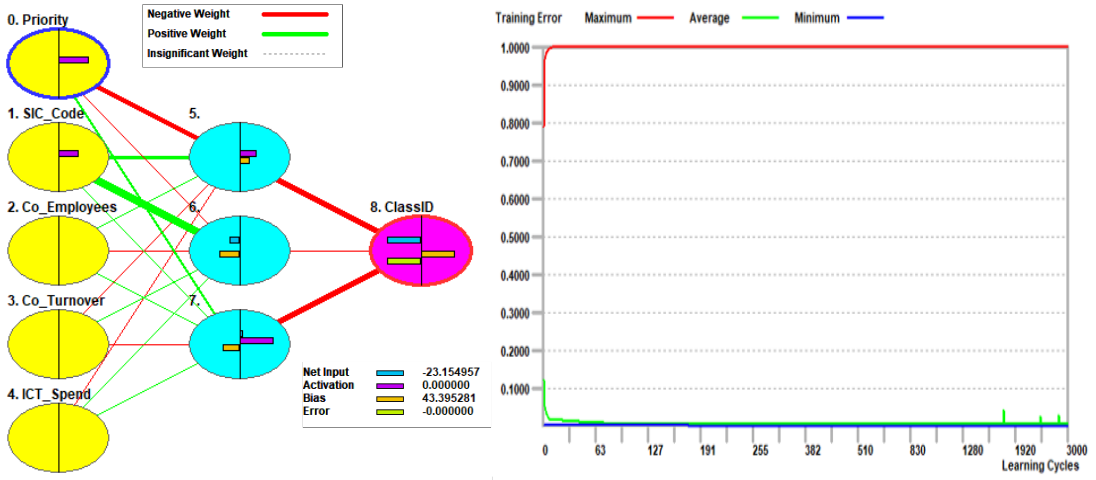
Layer:	Input	Hidden	Output	Learning rate: 0.6	Training example rows: 889
Nodes:	5	3	1	Momentum: 0.8	Learning cycles (epochs): 3000
Weights:	| 15	| 3	|	Maximum error: 0.160471	Training error: 0.002793



Column	Input Name	Importance	Relative Importance
1	SIC_Code	71.0508	
0	Priority	55.7889	
4	ICT_Spend	21.4617	
3	Co_Turnover	17.7761	
2	Co_Employees	3.6684	

Figure B.9: Learning batch 1.2 (standard), 1 output neuron

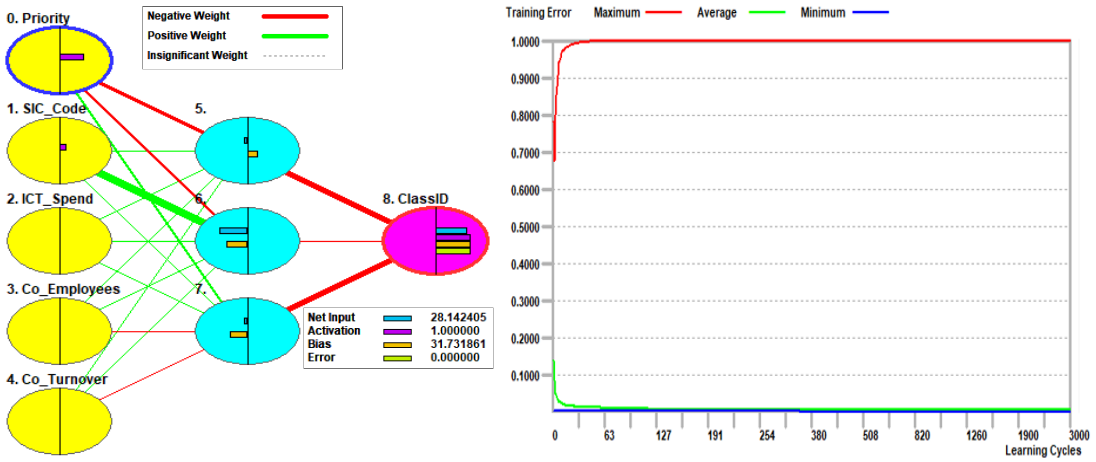
Layer:	Input	Hidden	Output	Learning rate: 0.6	Training example rows: 889
Nodes:	5	3	1	Momentum: 0.8	Learning cycles (epochs): 3000
Weights:	| 15	| 3	|	Maximum error: 1.000000	Training error: 0.004131



Column	Input Name	Importance	Relative Importance
1	SIC_Code	92.9943	
0	Priority	70.3049	
3	Co_Turnover	16.7052	
4	ICT_Spend	12.0035	
2	Co_Employees	3.3986	

Figure B.10: Learning batch 1.3 (standard), 1 output neuron

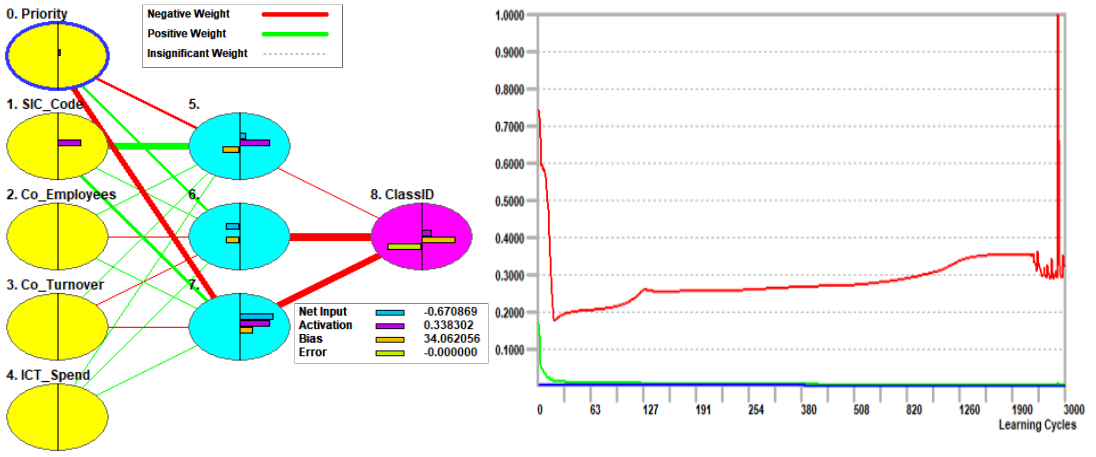
Layer:	Input	Hidden	Output	Learning rate: 0.6	Training example rows: 500
Nodes:	5	3	1	Momentum: 0.8	Learning cycles (epochs): 3000
Weights:	| 15	| 3	|	Maximum error: 0.100000	Training error: 0.003944



Column	Input Name	Importance	Relative Importance
1	SIC_Code	59.7876	
0	Priority	53.6159	
3	Co_Employees	9.2207	
2	ICT_Spend	7.3363	
4	Co_Turnover	5.9764	

Figure B.11: Learning batch 1.4a (normalised), 1 output neuron

Layer:	Input	Hidden	Output	Learning rate: 0.6	Training example rows: 500
Nodes:	5	3	1	Momentum: 0.8	Learning cycles (epochs): 3000
Weights:	| 15	| 3	|	Maximum error: 0.322076	Training error: 0.002492



Column	Input Name	Importance	Relative Importance
1	SIC_Code	60.2311	
0	Priority	59.9321	
3	Co_Turnover	10.4827	
4	ICT_Spend	7.7867	
2	Co_Employees	3.9860	

Figure B.12: Learning batch 1.4b (standard), 1 output neuron

B.5.2 ANN1 validation batches

As mentioned before, two training batches in test group ANN1 were run as validation after different combinations of certain hyper-parameters were tested. Together with batches 1.2 and 1.3, additional number of validation tests were run with one hidden layer and no data transformations. The validation tests were performed with different sample sizes, ranging between five and 400 examples for validation.

Additional iterations were run for validation, but as part of training batch runs with the same control settings as in the training runs. As control measure, iterations using validation examples with random, incorrect output values were also run. The learning curves for the batch runs with their different validation examples, for actual output and random output validation, are shown here.

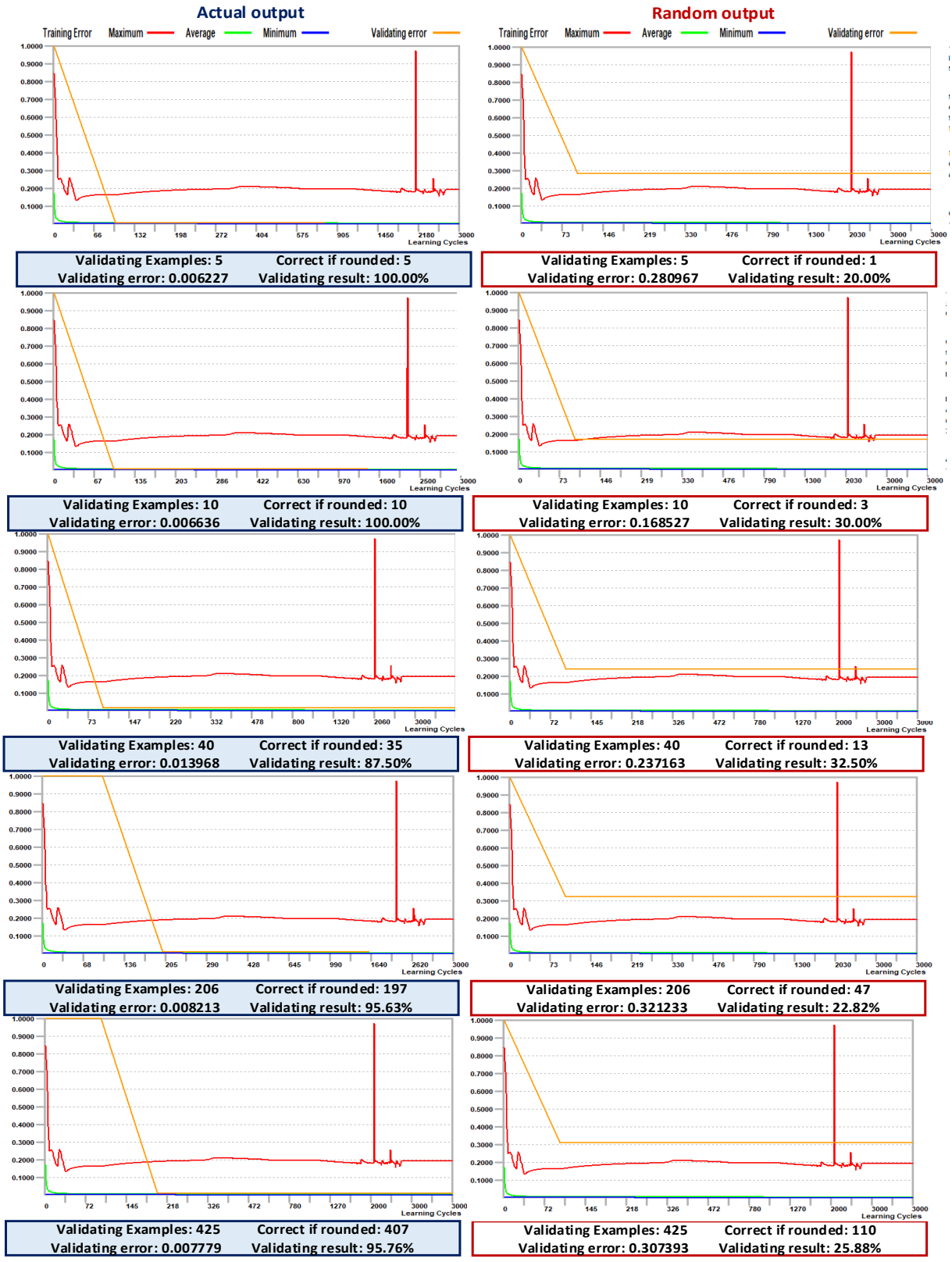


Figure B.13: Validation tests for batch 1.2 (standard), 1 output neuron

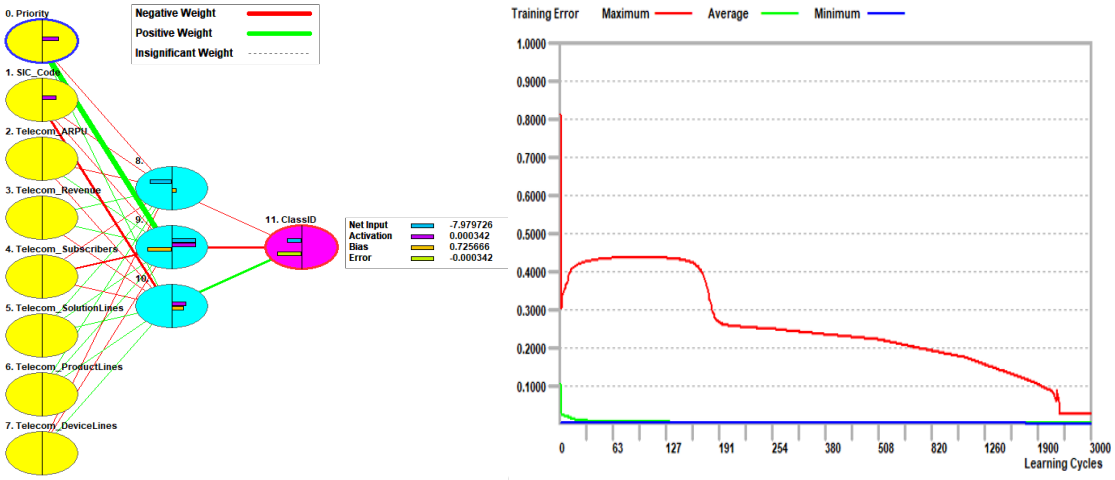
Figure B.14: Validating tests for batch 1.3 (standard), 1 output neuron

B.5.3 ANN2 training batches

Having done ANN batch learning on feature-based metrics of the target market, learning could be done on value-based metrics, for customers, while retaining two target market metrics in the datasets, Priority and SIC code. As a result, only 300 examples (9% sample rate) were sufficient for training to reach a realistic error level.

One normalised batch dataset were used for training first, followed by two subsequent batches with standard data for training. For each training iteration, one hidden layer, and one output neuron were used, with numerical values as output representing the four classes. A learning rate of 0.6 and momentum of 0.8 were used each time. Additional iterations were run for each batch training set (normalised and standard training data), after randomly initialising the weights again (without seeding this time) to between -0.5 and +0.5. The number of nodes (neurons) in the hidden layer were increased in the additional iteration for each training run to test any improvement in training error.

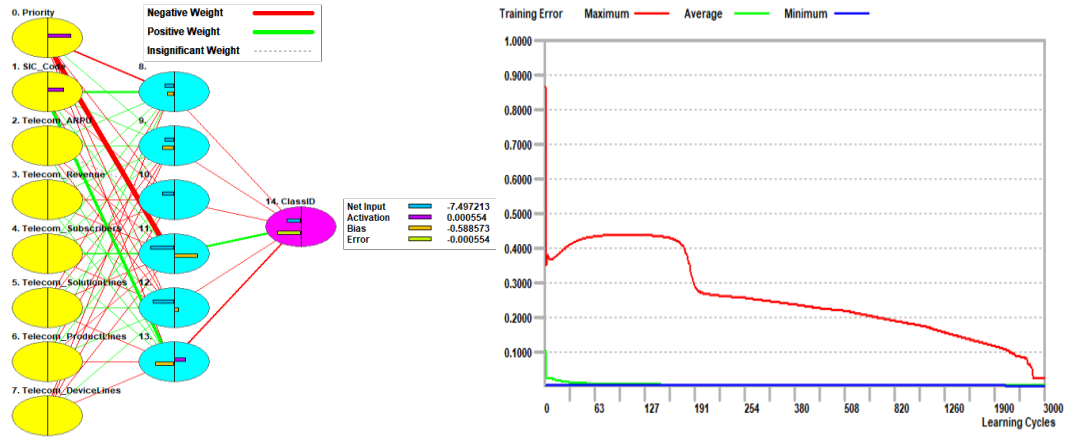
Layer:	Input	Hidden	Output	Learning rate: 0.6	Training example rows: 300
Nodes:	8	3	1	Momentum: 0.8	Learning cycles (epochs): 3000
Weights:	| 24	| 3	|	Maximum error: 0.024727	Training error: 0.000403



Column	Input Name	Importance	Relative Importance
0	Priority	63.9353	
1	SIC_Code	46.0867	
4	Telecom_Subscribers	29.2449	
3	Telecom_Revenue	7.6400	
5	Telecom_SolutionLines	6.8734	
6	Telecom_ProductLines	4.8871	
2	Telecom_ARPU	4.3004	
7	Telecom_DeviceLines	3.7083	

Figure B.15a: Learning batch 2.1 (normalised) 1st run

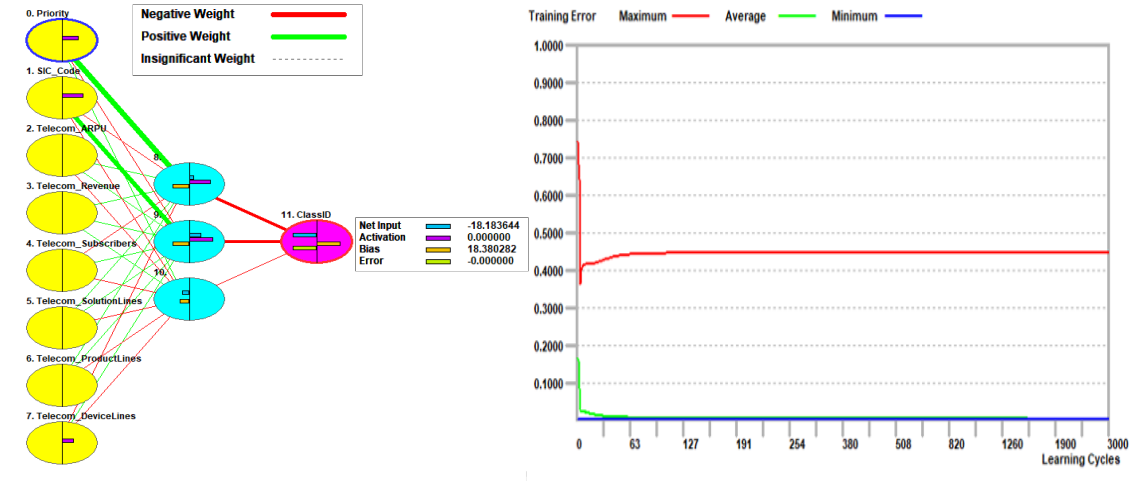
Layer:	Input	Hidden	Output	Learning rate: 0.6	Training example rows: 300
Nodes:	8	6	1	Momentum: 0.8	Learning cycles (epochs): 3000
Weights:	| 48	| 6	|	Maximum error: 0.020110	Training error: 0.000462



Column	Input Name	Importance	Relative Importance
0	Priority	81.0040	
1	SIC_Code	68.4235	
4	Telecom_Subscribers	35.4534	
5	Telecom_SolutionLines	7.0765	
6	Telecom_ProductLines	6.2861	
2	Telecom_ARPU	6.0493	
3	Telecom_Revenue	5.1671	
7	Telecom_DeviceLines	4.3882	

Figure B.15b: Learning batch 2.1 (normalised) 2nd run

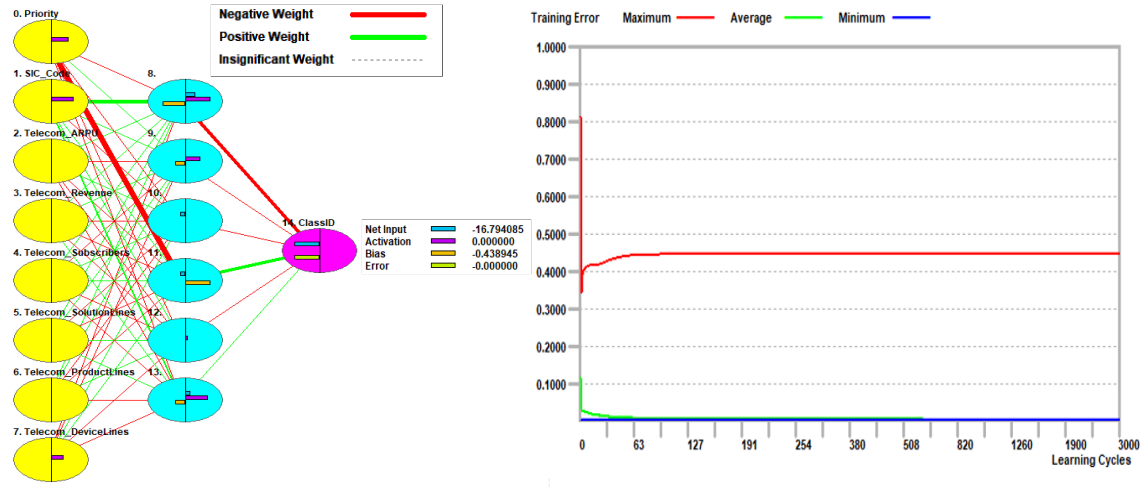
Layer:	Input	Hidden	Output	Learning rate: 0.6	Training example rows: 300
Nodes:	8	3	1	Momentum: 0.8	Learning cycles (epochs): 3000
Weights:	| 24	| 3 |		Maximum error: 0.44444	Training error: 0.003237



Column	Input Name	Importance	Relative Importance
0	Priority	40.1351	
1	SIC_Code	35.8377	
3	Telecom_Revenue	9.6378	
4	Telecom_Subscribers	5.9233	
2	Telecom_ARPU	2.4382	
7	Telecom_DeviceLines	2.3695	
5	Telecom_SolutionLines	1.6133	
6	Telecom_ProductLines	1.6022	

Figure B.16a: Learning batch 2.2 (standard) 1st run, 1 output neuron

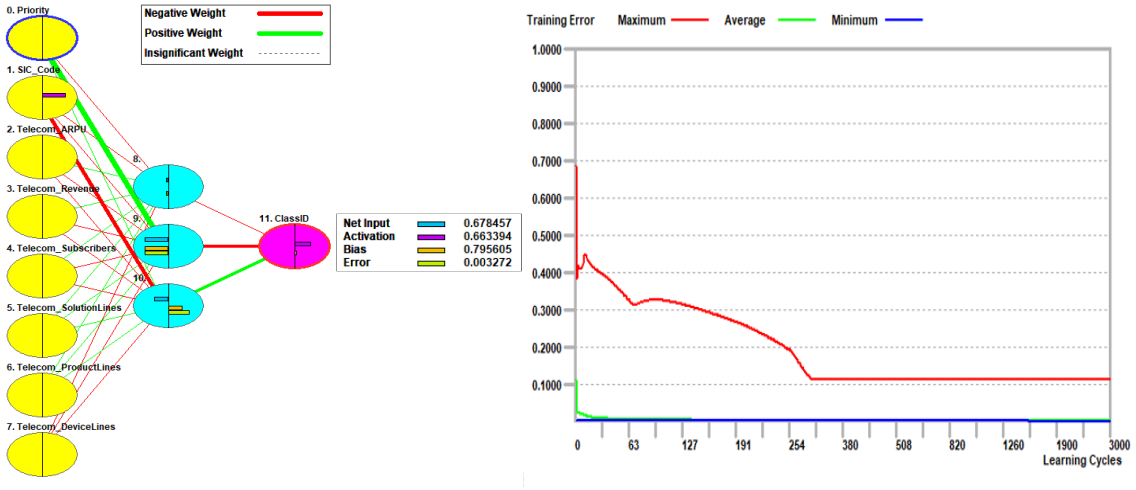
Layer:	Input	Hidden	Output	Learning rate: 0.6	Training example rows: 300
Nodes:	8	6	1	Momentum: 0.8	Learning cycles (epochs): 3000
Weights:	| 48	| 6 |		Maximum error: 0.44444	Training error: 0.003083



Column	Input Name	Importance	Relative Importance
1	SIC_Code	45.7836	
0	Priority	42.7650	
3	Telecom_Revenue	8.9446	
4	Telecom_Subscribers	8.4092	
5	Telecom_SolutionLines	2.9441	
2	Telecom_ARPU	2.6088	
7	Telecom_DeviceLines	2.0877	
6	Telecom_ProductLines	2.0876	

Figure B.16b: Learning batch 2.2 (standard) 2nd run, 1 output neuron

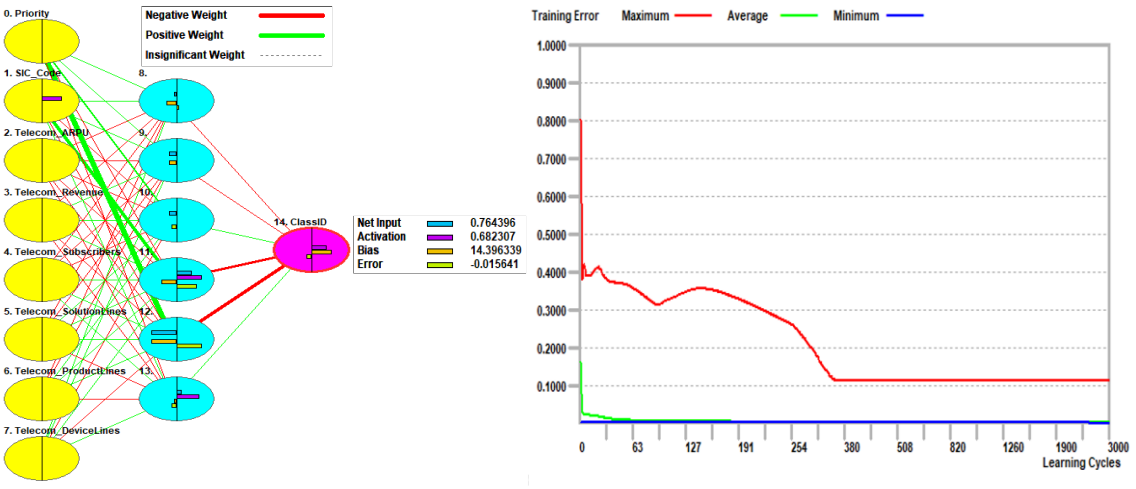
Layer:	Input	Hidden	Output	Learning rate: 0.6	Training example rows: 300
Nodes:	8	2	1	Momentum: 0.8	Learning cycles (epochs): 3000
Weights:	| 24	| 3	|	Maximum error: 0.111111	Training error: 0.000719



Column	Input Name	Importance	Relative Importance
0	Priority	47.5504	
1	SIC_Code	35.0770	
4	Telecom_Subscribers	4.8819	
3	Telecom_Revenue	4.4809	
7	Telecom_DeviceLines	3.6548	
2	Telecom_ARPU	3.4660	
5	Telecom_SolutionLines	1.7364	
6	Telecom_ProductLines	1.6023	

Figure B.17a: Learning batch 2.3 (standard) 1st run, 1 output neuron

Layer:	Input	Hidden	Output	Learning rate: 0.6	Training example rows: 300
Nodes:	8	2	1	Momentum: 0.8	Learning cycles (epochs): 3000
Weights:	| 16	| 2	|	Maximum error: 0.111111	Training error: 0.000544



Column	Input Name	Importance	Relative Importance
0	Priority	62.1832	
1	SIC_Code	46.1681	
3	Telecom_Revenue	6.6062	
7	Telecom_DeviceLines	6.0918	
2	Telecom_ARPU	5.7850	
4	Telecom_Subscribers	4.5301	
5	Telecom_SolutionLines	1.5781	
6	Telecom_ProductLines	1.5497	

Figure B.17b: Learning batch 2.3 (standard) 2nd run, 1 output neuron

B.6 Evaluation Criteria results

Below find all hypotheses, quality and processing scores as well as indices for all the test runs and methods.

Table B.22: All scores and indices for test runs and methods

Hypotheses, Quality and Processing Scores for Test Runs															Analysis Methods Scores			
Analysis	KMC1	KMC2	KMC3	KMC4	KMC5	PSO1	PSO2	CHAID1	CHAID2	CHAID3	CHAID4	ANN1	ANN2	Total Average Score	KMC	PSO	CHAID	ANN
Segment H1	0.500	0.900	0.400	0.950	0.600	0.900	0.500	1.000	0.500	0.650	0.990	0.950	0.400	0.711	0.67	0.70	0.79	0.68
Segment H2	0.900	0.500	0.900	0.300	0.950	0.750	0.700	1.000	0.500	0.850	0.650	0.990	0.900	0.761	0.71	0.73	0.75	0.95
Segment H3	0.500	0.400	0.550	0.350	0.300	0.500	0.600	0.500	0.500	0.600	0.600	0.450	0.750	0.508	0.42	0.55	0.55	0.60
Segment H4	0.750	0.750	0.550	0.850	0.650	0.500	0.600	0.500	0.600	0.650	0.550	0.450	0.600	0.615	0.71	0.55	0.58	0.53
Segment Hypotheses Score	0.663	0.638	0.600	0.613	0.625	0.663	0.600	0.750	0.525	0.688	0.698	0.710	0.663	0.649	0.63	0.63	0.67	0.69
Research H1	0.900	0.900	0.900	0.750	0.800	0.850	0.800	0.700	0.900	0.850	0.950	0.990	0.750	0.849	0.85	0.83	0.85	0.87
Research H2	0.950	0.750	0.800	0.750	0.700	0.600	0.600	0.880	0.900	0.800	0.950	0.950	0.750	0.798	0.79	0.60	0.88	0.85
Research H3	0.800	0.700	0.750	0.800	0.700	0.650	0.600	0.850	0.900	0.850	0.950	0.970	0.800	0.794	0.75	0.63	0.89	0.89
Research Hypotheses Score	0.883	0.783	0.817	0.767	0.733	0.700	0.667	0.810	0.900	0.833	0.950	0.970	0.767	0.814	0.80	0.68	0.87	0.87
Research Q1	0.900	0.850	0.850	0.600	0.700	0.750	0.650	0.900	0.800	0.750	0.950	0.990	0.700	0.799	0.78	0.70	0.85	0.85
Research Q2	0.900	0.800	0.700	0.800	0.600	0.450	0.500	0.950	0.900	0.800	0.950	0.800	0.800	0.765	0.76	0.48	0.90	0.80
Research Q3	0.800	0.650	0.850	0.600	0.900	0.550	0.650	0.800	0.700	0.700	0.900	0.700	0.800	0.738	0.76	0.60	0.78	0.75
Research Q4	0.850	0.950	0.550	0.900	0.650	0.800	0.850	0.850	0.750	0.750	0.950	0.850	0.850	0.812	0.78	0.83	0.83	0.85
Research Q5	0.400	0.650	0.500	0.900	0.800	0.650	0.700	0.800	0.700	0.700	0.900	0.850	0.850	0.723	0.65	0.68	0.78	0.85
Research Q6	0.400	0.650	0.700	0.800	0.900	0.600	0.650	0.850	0.900	0.650	0.950	0.800	0.700	0.735	0.69	0.63	0.84	0.75
Research Q7	0.350	0.650	0.750	0.800	0.900	0.700	0.750	0.800	0.850	0.600	0.900	0.750	0.650	0.727	0.69	0.73	0.79	0.70
Research Questions Score	0.657	0.743	0.700	0.771	0.779	0.643	0.679	0.850	0.800	0.707	0.929	0.820	0.764	0.757	0.73	0.66	0.82	0.79
Hypotheses Score (H)	0.734	0.721	0.706	0.717	0.712	0.668	0.648	0.803	0.742	0.743	0.859	0.833	0.731	0.740	0.72	0.66	0.79	0.78
Hypotheses & Time Average Score	0.862	0.857	0.848	0.853	0.848	0.620	0.324	0.902	0.871	0.871	0.929	0.916	0.865	0.813	0.85	0.470	0.89	0.890
Hypotheses & Iterations Average Score	0.471	0.401	0.453	0.372	0.498	0.834	0.823	0.890	0.858	0.862	0.915	0.800	0.833	0.693	0.44	0.830	0.88	0.820
Hypotheses & Processing Average Score	0.666	0.629	0.651	0.612	0.673	0.727	0.573	0.896	0.864	0.867	0.922	0.858	0.849	0.753	0.65	0.650	0.89	0.850
Quality Score (Q)	0.554	0.640	0.558	0.676	0.503	0.817	0.613	0.740	0.640	0.790	0.680	0.955	0.882	0.696	0.59	0.71	0.71	0.92
Quality & Time Average Score	0.772	0.816	0.774	0.833	0.744	0.736	0.307	0.870	0.820	0.895	0.840	0.867	0.647	0.763	0.79	0.52	0.86	0.76
Quality & Iterations Average Score	0.381	0.360	0.380	0.351	0.393	0.951	0.805	0.858	0.807	0.886	0.825	0.751	0.614	0.643	0.37	0.88	0.84	0.68
Quality & Processing Average Score	0.576	0.588	0.577	0.592	0.569	0.844	0.556	0.864	0.813	0.891	0.833	0.809	0.631	0.703	0.58	0.70	0.85	0.72
Combined Quality Hypotheses Score	0.644	0.681	0.632	0.696	0.608	0.743	0.631	0.772	0.691	0.766	0.769	0.784	0.513	0.687	0.65	0.69	0.75	0.65
Process Time	00:00:49.33	00:00:37.00	00:00:44.67	00:00:50.00	00:01:10.00	00:32:23.33	01:15:32.00	00:00:00.50	00:00:00.58	00:00:00.52	00:00:00.46	00:00:03.93	00:00:04.00	00:08:38.18	00:00:50.20	00:53:57.41	00:00:00.52	00:00:04.32
Process Time (T) Score	0.98922	0.99194	0.99025	0.98907	0.98466	0.57130	0.00010	0.99999	0.99997	1.000	1.000	0.999	0.999	0.886	0.99	0.29	0.99999	1.00
Iterations (I) or Nodes (N)	277	320	279	338	251	9	10	17	18	15	19	88	31	129	293	10	17	59
Iterations (I) Score	0.207	0.080	0.201	0.027	0.284	1.000	0.997	0.976	0.973	0.982	0.970	0.768	0.935	0.646	0.16	1.00	0.98	0.85
Process Score (P)	0.598	0.536	0.596	0.508	0.634	0.786	0.499	0.988	0.987	0.991	0.985	0.883	0.967	0.766	0.57	0.64	0.99	0.93
I & N Reverse Count	70	27	68	9	96	338	337	330	329	332	328	259	316	218	54	338	330	288
Percent of Total Iterations	2.5%	1.0%	2.4%	0.3%	3.4%	11.9%	11.9%	11.6%	11.6%	11.7%	11.6%	9.1%	11.1%	7.7%	1.9%	11.9%	11.6%	10.1%
Q x H x P Combined Scores	0.629	0.632	0.620	0.634	0.616	0.757	0.587	0.844	0.789	0.841	0.841	0.890	0.860	0.7339	0.63	0.67	0.83	0.88
Analysis Sequence	1	2	3	4	5	6	7	8	9	10	11	12	13	7	3	2	1	13
Analysis Method Index	1 014	1 020	1 000	1 022	994	1 221	946	1 361	1 273	1 357	1 357	1 436	1 387	1 184	1 010	1 084	1 337	1 412
Cumulative Reverse Count	70	97	165	174	270	608	945	1 275	1 604	1 936	2 264	2 523	2 839	1 136	155	777	1 770	2 681
Cumulative Percentage	2.5%	3.4%	5.8%	6.1%	9.5%	21.4%	33.3%	44.9%	56.5%	68.2%	79.7%	88.9%	100.0%	40.0%	5.5%	27.4%	62.3%	94.5%
Total Score Count	44	17	42	6	59	256	198	278	260	279	276	231	272	171	34	227	273	251
Cumulative Average Score	0.63	0.63	0.63	0.63	0.62	0.70	0.66	0.71	0.72	0.74	0.76	0.77	0.78	0.69	0.63	0.68	0.73	0.78
Cumulative Index	1 016	1 016	1 016	1 016	1 000	1 129	1 065	1 145	1 161	1 194	1 226	1 242	1 258	1 114	1 013	1 097	1 182	1 250

The hypotheses, quality and processing scores for the segmentation methods are depicted graphically below.

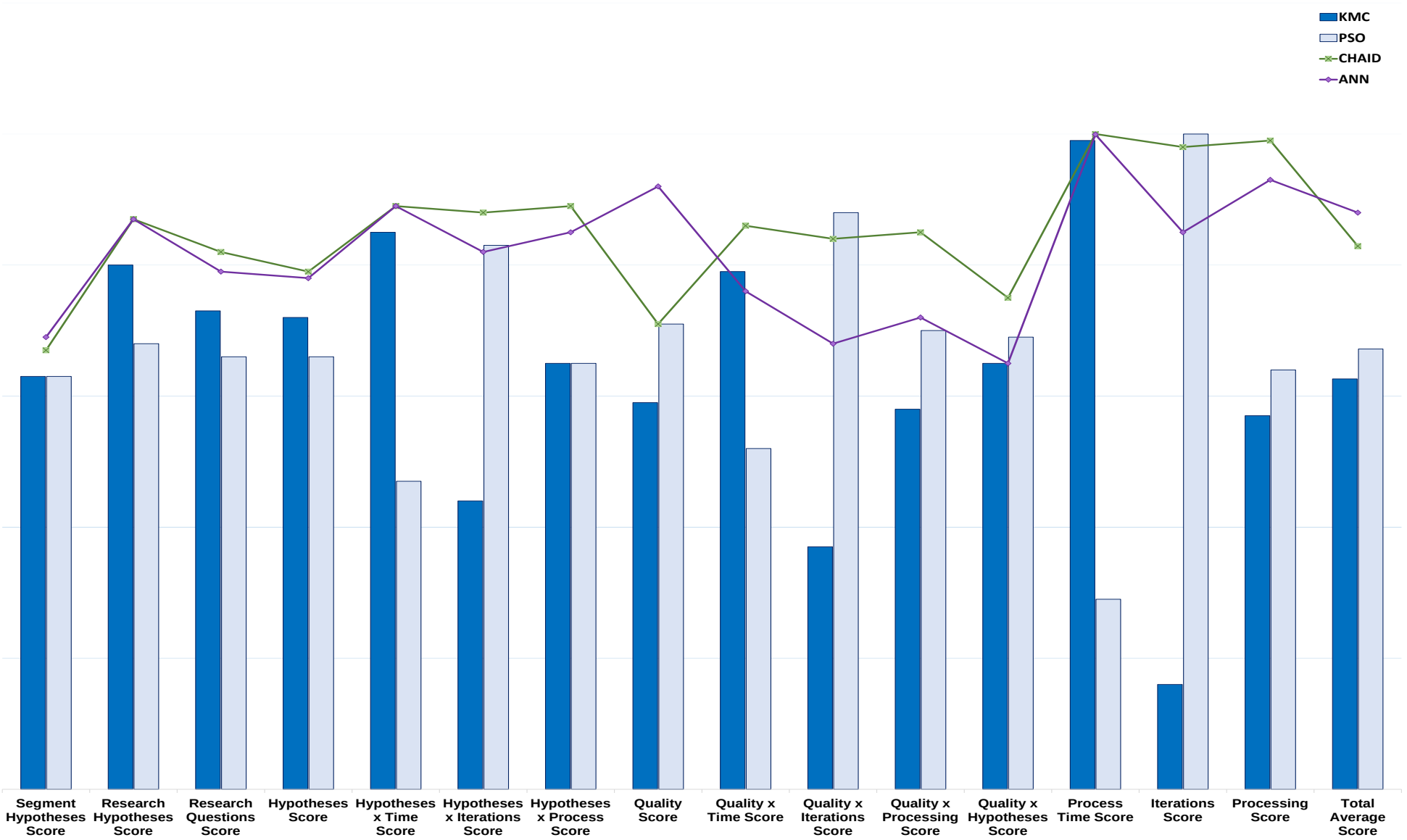


Figure B.18: All processing scores for segmentation methods

Segment and research hypotheses and segment questions scores for the all the methods are depicted graphically below.

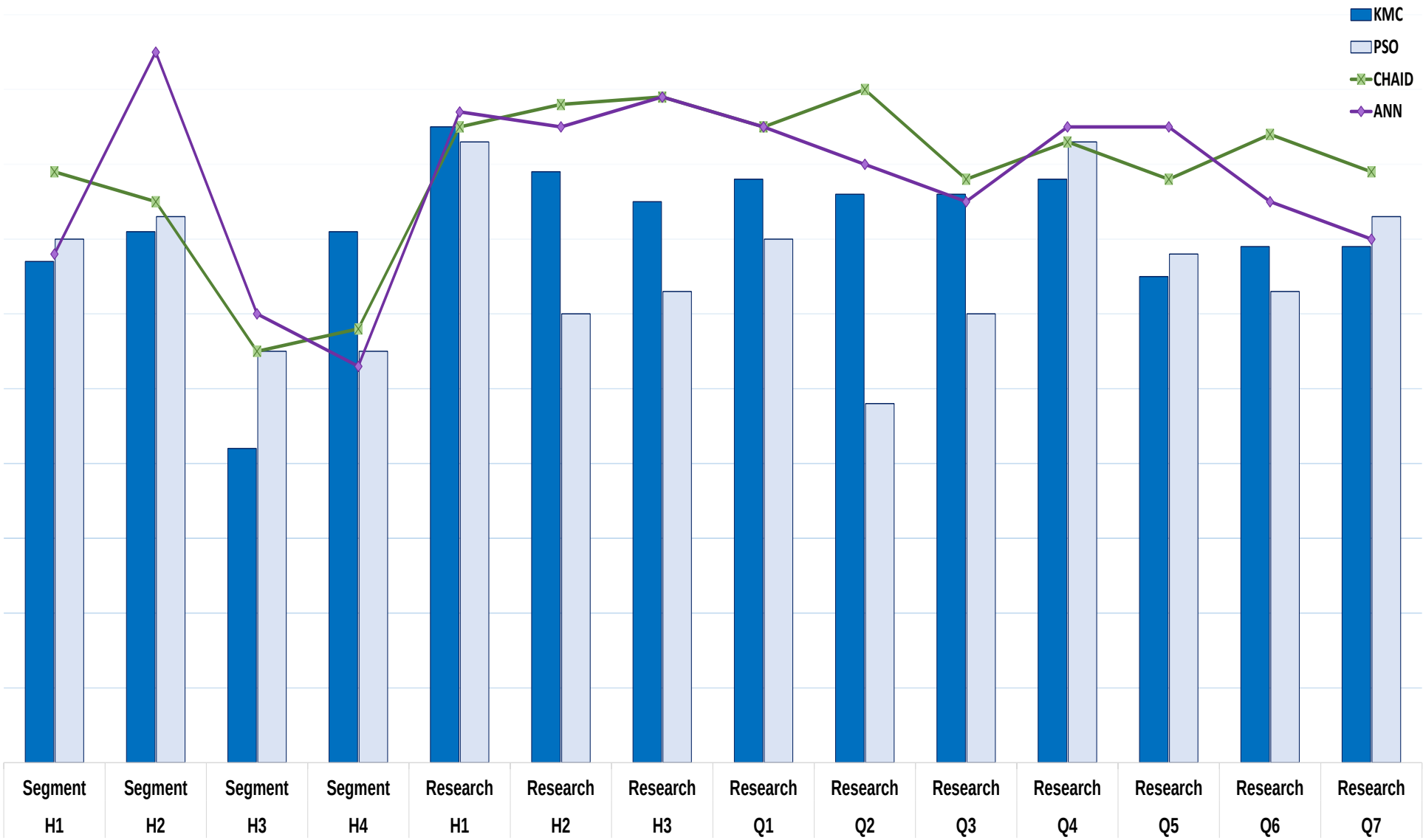


Figure B.19: Fit for purpose scores of segmentation methods

Appendix C – SOURCE CODE AND APPLICATIONS

C.1 Exploratory data tool

For the initial exploratory data analysis, an online tool from Statistics Solutions were used (Intellectus Statistics, 2020). The application is online available at https://www.intellectusstatistics.com.

Statistics Solutions is a dissertation editing service with expertise in both a quantitative and qualitative approach (Statistics Solutions, 2020). Intellectus Statistics is software that allows analyses without requiring statistical expertise. This made the creation of descriptive statistics for this research more effective, leaving time for actual analysis of the quantitative methods. The following descriptive statistics are provided after analysis of the target dataset through the Intellectus Statistics online application: mean, standard deviation, standard error, minimum, maximum, skewness, kurtosis, correlation coefficients, and percentiles. Cross tabulations between values can also be done. The variable types on which statistics can be calculated are scale (numerical), nominal and ordinal. Frequency of values are also shown initially to decide which variables to include or to define the variable type. Other descriptive statistics like median, mode and correlations are provided by the KMC Excel based tools, which is described in the next section.

C.2 KMC Excel based VBA code

Please find below VBA code for each of the process steps. The code is from the KMC application developed by Prof Dr Rauh, Hochschule Heilbron (Rauh, 2013a). It is given without any alterations, except for the translation of comments from German in the solution validation code. The application is available from https://www.orauh.de/software/kmc-clustering-tool.

C.2.1. Data input

The company dataset is selected in Excel, linked to the KMC algorithm and preliminary statistics are output by using the VBA code below.

```
'linking to the data sheet in Excel
Private Sub ConnectBtn_Click( )
   If Not formalValidationOK Then Exit Sub
   Set PopRng = Range(Me.PopRngREd.Text)
   If Not logicValidationOK Then Exit Sub
   pop.setUp PopRng
   popname = Me.PopNameTBx.Text
   If Not sheetExists(popname and "Copy") Then
      Set copyWS = Application.ActiveWorkbook.Worksheets.Add
      copyWS.Name = popname and "Copy"
   Else
      Set copyWS = Worksheets(popname and "copy")
   End If
   OutputCharacteristics
   Me.MsgTBx.Text = "population stats have been printed to worksheet " and _
                popname and "Char"
   updateSolDelCBx                    'solutions available for deletion or scatter plot
   connected
End Sub

'writes stats of the (still unclustered) population to worksheet
Private Sub OutputCharacteristics( )
   Application.DisplayAlerts = False
   Dim prr As Long, prc As Long      'point of reference
   Dim i As Long, j As Long, n As Long
   n = pop.numAttr
   On Error Resume Next
   Worksheets(popname and "Char").Delete
   Set charWS = Application.ActiveWorkbook.Worksheets.Add
   charWS.Name = popname and "Char"
   charWS.Cells(1, 1) = "Characteristics of population " and popname
   charWS.Cells(3, 1) = "number of attributes"
   charWS.Cells(3, 3) = n
   charWS.Cells(4, 1) = "number of objects"
   charWS.Cells(4, 3) = pop.numObj

   prr = 6
   prc = 1
   charWS.Cells(prr, prc + 1) = "attribute characteristics"
   charWS.Cells(prr + 2, prc + 1) = "no of missing values"
   charWS.Cells(prr + 3, prc + 1) = "min"
   charWS.Cells(prr + 4, prc + 1) = "max"
   charWS.Cells(prr + 5, prc + 1) = "mean"
```

```
    charWS.Cells(prr + 6, prc + 1) = "median"
    charWS.Cells(prr + 7, prc + 1) = "mode"
    charWS.Cells(prr + 8, prc + 1) = "range"
    charWS.Cells(prr + 9, prc + 1) = "standard deviation"
    charWS.Cells(prr + 10, prc + 1) = "variance"

    For i = 1 To n
       charWS.Cells(prr + 1, prc + 2 + i) = pop.attrName(i)
       charWS.Cells(prr + 2, prc + 2 + i) = pop.numMiss(i)
       charWS.Cells(prr + 3, prc + 2 + i) = pop.minVal(i)
       charWS.Cells(prr + 4, prc + 2 + i) = pop.maxVal(i)
       charWS.Cells(prr + 5, prc + 2 + i) = pop.mean(i)
       charWS.Cells(prr + 6, prc + 2 + i) = pop.medianVal(i)
       charWS.Cells(prr + 7, prc + 2 + i) = pop.modeVal(i)
       charWS.Cells(prr + 8, prc + 2 + i) = pop.rangeVal(i)
       charWS.Cells(prr + 9, prc + 2 + i) = pop.standDev(i)
       charWS.Cells(prr + 10, prc + 2 + i) = pop.variance(i)
    Next i
    GR.formatTable Range(charWS.Cells(prr + 1, prc + 1), charWS.Cells(prr + 10,
prc + 2 + n)).Cells

    prr = 20
    prc = 1
    charWS.Cells(prr, prc + 2) = "Correlations"
    Dim r( ) As Variant
    r = pop.corrMatrixOut( )
    charWS.Range(Cells(prr + 1, prc + 2), _
              Cells(prr + UBound(r, 1), prc + 1 + UBound(r, 2))) = r
    GR.formatTable BharWS.Range(Cells(prr + 1, prc + 2), _
               Cells(prr + UBound(r, 1), prc + 1 + UBound(r, 2)))

    prr = prr + UBound(r, 1) + 2
    prc = 1
    charWS.Cells(prr, prc + 2) = "Covariances"
    r = pop.covMatrixOut( )
    charWS.Range(Cells(prr + 1, prc + 2), _
              Cells(prr + UBound(r, 1), prc + 1 + UBound(r, 2))) = r
    GR.formatTable BharWS.Range(Cells(prr + 1, prc + 2), _
               Cells(prr + UBound(r, 1), prc + 1 + UBound(r, 2)))
    Application.DisplayAlerts = True
End Sub
```

C.2.2. Exploring the data

In performing test runs certain values are initialised, and checks are done to explore the data. The VBA code below shows functions for these actions.

```
'insert random centers
   For i = 1 To nSeg
      cenOld(i, 0) = i                                  'insert cids
      cenNew(i, 0) = i                                   'insert cids
      For j = 1 To nAtt
         cenOld(i, j) = Rnd * (maxi(j) - mini(j)) + mini(j)
      Next j
   Next i

   'do iterations
   Dim it As Long, nextC As Integer, changed As Boolean
   it = 0                                               'num of iterations
   Do
      cenNewToZero
      changed = False
      For i = 1 To nObj
         nextC = nextCenter(i)                              'determine nearest centre
         If nextC <> CInt(pop(i, nAtt + 1)) Then changed = True
         pop(i, nAtt + 1) = nextC                           'assign cid to object

         'increase num of objects in segment
         cenNew(nextC, nAtt + 1) = cenNew(nextC, nAtt + 1) + 1
         For j = 1 To nAtt                                 'add values of object
            cenNew(nextC, j) = cenNew(nextC, j) + pop(i, j)
         Next j
      Next i

     'calculate new centers
     'divide accumulated values by number of objects in segment
      For i = 1 To nSeg                For j = 1 To nAtt
            If cenNew(i, nAtt + 1) > 0 Then
               cenNew(i, j) = cenNew(i, j) / cenNew(i, nAtt + 1)
            End If
        Next j
      Next i

      cenNewToCenOld
      it = it + 1

      Loop Until Not changed Or it >= nIt
         segments = cenNew
      End Function
```

```
'calculates euclidian distance between p1 and p2,
‘where p1 and p2 are tuples without id
Private Function EuDist(ByRef p1( ) As Double, ByRef p2( ) As Double) As Double
    Dim i As Integer
       For i = 1 To UBound(p1)
          EuDist = EuDist + (p1(i) - p2(i)) ^ 2
       Next
       EuDist = EuDist ^ 0.5
    End Function

'forms a tuple of the attribute values of a row in pop, without id
    Private Function ptuple(ByVal rowno As Long) As Double( )
      Dim t( ) As Double
         ReDim t(1 To nAtt)
         Dim i As Long
         For i = 1 To nAtt
            t(i) = pop(rowno, i)  'col 0 is id
         Next
         ptuple = t
    End Function

'forms a tuple of the attribute values of a row in cenOld
    Private Function ctuple(ByVal rowno As Integer) As Double( )
       Dim t( ) As Double
       ReDim t(1 To nAtt)
       Dim i As Long
       For i = 1 To nAtt
          t(i) = cenOld(rowno, i)   'col  0 is cid
       Next
       ctuple = t
    End Function

'finds cid of the center next to pop(rowno)
    Private Function nextCenter(ByVal rowno As Long) As Integer
       Dim n As Integer, i As Integer, ndist As Double
       n = 1
       ndist = EuDist(ptuple(rowno), ctuple(1))
       For i = 2 To UBound(cenOld, 1)
          If EuDist(ptuple(rowno), ctuple(i)) < ndist Then
             ndist = EuDist(ptuple(rowno), ctuple(i))
             n = i
          End If
       Next
       nextCenter = n
    End Function
```

C.2.3. Transforming the population

After initial tests, the blank values have to be filled, and transformation of data may be needed, before clustering can start. The functions in VBA for replacing blanks in three ways and for transforming the data in two ways are shown below.

```
'replaces missing values by 1 mean, 2 median, 3 mode
Public Sub replaceMissing(ByVal subs As Byte)
Dim r( ) As Double
Dim i As Long, j As Long, n As Long, m As Long
   n = Me.numAttr
   m = Me.numObj
   ReDim r(1 To n)
   If subs = 1 Then
      For i = 1 To n
         r(i) = Me.mean(i)
      Next
   Else
      If subs = 2 Then
         r(i) = Me.medianVal(i)
      Else
         r(i) = Me.modeVal(i)
      End If
   End If
   For i = 1 To n
      For j = 1 To m
         If IsEmpty(PopRng.Cells(j + 1, i + 1)) Then PopRng.Cells(j + 1, i + 1) = r(i)
      Next j
   Next i
End Sub

'standardizes the values using 1 min-max or 2 z-transformation
Public Sub standardize(ByVal method As Byte)
   Dim i As Long, j As Long, n As Long, m As Long
   n = Me.numAttr
   m = Me.numObj
      If method = 1 Then
   Dim maxv As Double, minv As Double
   For i = 1 To n
    minv = Me.minVal(i)
    maxv = Me.maxVal(i)
    For j = 1 To m
      PopRng.Cells(j + 1, i + 1) = (PopRng.Cells(j + 1, i + 1) - minv) / (maxv - minv)
         Next j
     Next i
```

```
    Else
      Dim meanv As Double, sdev As Double
      For i = 1 To n
          meanv = Me.mean(i)
          sdev = Me.standDev(i)
          For j = 1 To m
             PopRng.Cells(j + 1, i + 1) = (PopRng.Cells(j + 1, i + 1) - meanv) / sdev
          Next j
      Next i
    End If
End Sub
```

C.2.4. Clustering process

Extracting (transformed) data and updating the cluster id's (CIDs) form part of the clustering process in the code below.

```
Option Explicit

Private segs( ) As segment    'segments
Private atts( ) As String     'attributes
Dim noOfAttr As Integer

'extract data from range;
'range is with headers and cluster assignments, but without object ids
Public Sub extractSegments(ByVal pop As Range)
    Dim noOfObj As Long
    Dim i As Long, j As Long
    Dim sInd As Long
    ReDim segs(1 To 1)
    With pop

       noOfObj = CLng(.Rows.Count - 1)
       noOfAttr = CInt(.Columns.Count - 1)


       'get ids and number of objects of clusters
       segs(1).segId = CInt(.Cells(2, .Columns.Count).Value)
       segs(1).noOfObj = 1
       For i = 3 To .Rows.Count
          sInd = getIndex(CInt(.Cells(i, .Columns.Count)))
          If sInd < 1 Then
             ReDim Preserve segs(1 To UBound(segs) + 1)
             segs(UBound(segs)).segId = CInt(.Cells(i, .Columns.Count))
             segs(UBound(segs)).noOfObj = 1
          Else
             segs(sInd).noOfObj = segs(sInd).noOfObj + 1
          End If
```

```
        Next i

        'set dimensions for the matrices holding object values
        For i = 1 To UBound(segs)
           ReDim segs(i).x(1 To segs(i).noOfObj, 1 To noOfAttr)
        Next i

        'receive attribute values of objects
        Dim objcount( ) As Long          'array of object counters
        ReDim objcount(1 To UBound(segs))
        For i = 2 To .Rows.Count
           sInd = getIndex(CInt(.Cells(i, .Columns.Count)))
           objcount(sInd) = objcount(sInd) + 1
           For j = 1 To noOfAttr
              segs(sInd).x(objcount(sInd), j) = CDbl(.Cells(i, j))
           Next j
        Next i

        'write attribute names into array atts
        ReDim atts(1 To noOfAttr)
        For i = 1 To noOfAttr
           atts(i) = Trim(CStr(.Cells(1, i).Value))
        Next i
     End With 'pop
End Sub

'provides values for attribute attrName of cluster segId
Public Function attrValues(ByVal attrName As String, ByVal segId As Integer)
As Double( )
     attrValues = extractColumn(segs(segId).x, attrNameToId(attrName))
End Function

'provides number of segments
Public Function numSegments( ) As Integer
     numSegments = UBound(segs, 1)
End Function

'extracts values of column colNo from matrix of values
Private Function extractColumn(ByRef attrMatrix, ByVal colNo As Integer) As
Double( )
     Dim c( ) As Double
     ReDim c(1 To UBound(attrMatrix, 1))
     Dim i As Long
     For i = 1 To UBound(attrMatrix, 1)
        c(i) = attrMatrix(i, colNo)
     Next
     extractColumn = c
End Function
```

```
'gives index for segId, but -1 if cluster is not contained in segs
Private Function getIndex(ByVal sId As Integer) As Integer
   Dim found As Boolean
   Dim i As Long
   i = 1
   Do While i <= UBound(segs) And Not found
      If segs(i).segId = sId Then found = True
      i = i + 1
   Loop
   getIndex = IIf(found, i - 1, -1)
End Function

'provides an array with names of all attributes
Public Function getAttributeNames( ) As String( )
   getAttributeNames = atts
End Function


'gives id of attribute named attrName, -1 if not found
Private Function attrNameToId(ByVal attrName As String) As Integer
   Dim i As Integer
   For i = 1 To noOfAttr
      If atts(i) = attrName Then
         attrNameToId = i
         Exit Function
      End If
   Next i
   attrNameToId = -1
End Function
```

C.2.5. Solution validation

Using the silhouette index, the quality of the solution can be measured, and weights can be calculated on the output. The code below calculates the silhouette for each cluster centre and averages the values.

```
'calculates the total square error of the cluster distribution contained in PopRng
'Public Function FQuadSuNum(ByVal PopRng As Range) As Double
   Dim NoOfClust As Integer
   Dim noOfAttr As Integer
   Dim noOfObj As Long
   Dim Cent( ) As Double      'Attribute values of the centers; Column 0: ClustId
   Dim Obj( ) As Double      'Matrix; Column 0: ObjId; last column: ClustId
   Dim Clust( ) As String     'Cluster names; Index acts as a ClustId
   Dim NoOfLmts( ) As Long  'Number of Obj in clusters; Index match to ClustId
   Dim Errors( ) As Double    'Centre sum of squares on ClustElem, index: ClustId
   Dim i As Long, j As Long, k As Long
```

```
ReDim Clust(1 To 1)
ReDim NoOfLmts(1 To 1)
'Accept values from the range in arrays Obj and Clust
With PopRng
   noOfObj  .Rows.Count
   noOfAttr  .Columns.Count  1
   ReDim Obj(1 To noOfObj, 0 To noOfAttr + 1)
   'Register the clusters in Clust
   Clust(1)  CStr(.Cells(1, noOfAttr + 1))   'set the first cluster
   Dim aktClu As String
   Dim aktCid As Long
   For i  1 To noOfObj
      aktClu  Trim(CStr(.Cells(i, noOfAttr + 1)))
      k  1
      Dim found As Boolean
      found  False
      Do While k < UBound(Clust) And Not found
         If aktClu  Clust(k) Then
            found  True
         Else
            k  k + 1
         End If
      Loop
      If Not found Then
         ReDim Preserve Clust(1 To UBound(Clust) + 1)
         Clust(UBound(Clust))  aktClu
         ReDim Preserve NoOfLmts(1 To UBound(Clust) + 1)
      End If
   Next i
   'Insert objects in Obj, set ObjId and ClustId and update NoOfLmts
   For i  1 To noOfObj
      Obj(i, 0)  i         'ObjId
      Obj(i, noOfAttr + 1)  sucheClustId(Trim(CStr(.Cells(i, noOfAttr + 1))), Clust)
      NoOfLmts(Obj(i, noOfAttr + 1))  NoOfLmts(Obj(i, noOfAttr + 1)) + 1
      For j  1 To noOfAttr                  'assign attribute values
         Obj(i, j)  CDbl(.Cells(i, j))
      Next j
   Next i
End With
'Calculate cluster centres and store them in the array Cent
NoOfClust  UBound(Clust)
ReDim Cent(1 To NoOfClust, 0 To noOfAttr)
For i  1 To NoOfClust
   Cent(i, 0)  i                      'ClusterIds in Spalte 0
Next i
Dim cid As Long                          'for ClusterId
'Add up attribute values separately according to clusters
For i  1 To noOfObj
   cid  Obj(i, noOfAttr + 1)                 'Get ClusterId from last column
   For j  1 To noOfAttr
      Cent(cid, j)  Cent(cid, j) + Obj(i, j)    'Include values of Obj i
   Next j
```

```
    Next i
    'Divide attribute sums by number of elements
    For i  1 To NoOfClust
       For j  1 To noOfAttr
          Cent(i, j)  Cent(i, j) / NoOfLmts(i)
       Next j
    Next i
    'Sum of squares and average.
    'Find the cluster squares and store them in the Errors array
    ReDim Errors(1 To NoOfClust, 1 To 2) 'Column 1: e; Column 2: average e

    Dim errSum As Double

    'Sums of squares
    For i  1 To noOfObj
       cid  Obj(i, noOfAttr + 1)
       Errors(cid, 1)  Errors(cid, 1) + _
          quadEuDist(CentAttr(Zeilenvektor(Cent, cid), noOfAttr), _
                ObjAttr(Zeilenvektor(Obj, i), noOfAttr))
    Next i
    ' Total error sum of squares
    Dim sum As Double
    sum  0
    For i  1 To NoOfClust
       sum  sum + Errors(i, 1)
    Next i
    FQuadSuNum  sum
End Function
'calculates the overall silhouette of the cluster distribution described in PopRng
Public Function GSilhouette(ByVal PopRng As Range) As Double
    Dim NoOfClust As Integer
    Dim noOfAttr As Integer
    Dim noOfObj As Long
    Dim Obj( ) As Double       'Matrix; Spalte 0: ObjId; letzte Spalte: ClustId
    Dim Clust( ) As String     'Clusternamen; Index fungiert als ClustId
    Dim NoOfLmts( ) As Long    'Anzahl Obj in den Clustern; Index entspricht ClustId
    Dim i As Long, j As Long, k As Long
    ReDim Clust(1 To 1)
    ReDim NoOfLmts(1 To 1)




    'Accept values from the range in arrays Obj and Clust
       With PopRng
       noOfObj  .Rows.Count
       noOfAttr  .Columns.Count  1
       ReDim Obj(1 To noOfObj, 0 To noOfAttr + 1)

       'Register the clusters in Clust
       Clust(1)  CStr(.Cells(1, noOfAttr + 1))   'erstes Cluster setzen
       Dim aktClu As String
```

```
    Dim aktCid As Long
    For i  1 To noOfObj
       aktClu  Trim(CStr(.Cells(i, noOfAttr + 1)))
       k  1
       Dim found As Boolean
       found  False
       Do While k < UBound(Clust) And Not found
          If aktClu  Clust(k) Then
             found  True
          Else
             k  k + 1
          End If
       Loop
       If Not found Then
          ReDim Preserve Clust(1 To UBound(Clust) + 1)
          Clust(UBound(Clust))  aktClu
          ReDim Preserve NoOfLmts(1 To UBound(Clust) + 1)
       End If
    Next i
    'Insert objects in Obj, set ObjId and ClustId and update NoOfLmts
    For i  1 To noOfObj
       Obj(i, 0)  i          'ObjId
       Obj(i, noOfAttr + 1)  sucheClustId(Trim(CStr(.Cells(i, noOfAttr + 1))), Clust)
       NoOfLmts(Obj(i, noOfAttr + 1))  NoOfLmts(Obj(i, noOfAttr + 1)) + 1
       For j  1 To noOfAttr                  'Attributwerte zuweisen
          Obj(i, j)  CDbl(.Cells(i, j))
       Next j
    Next i
End With
NoOfClust  UBound(Clust)
  Dim dd( ) As Double                'takes cumulative distances from active Obj
ReDim dd(1 To NoOfClust)
ReDim siClu(1 To NoOfClust + 1)      'for silhouettes of Cluster u
Dim cid As Integer, cid2 As Integer   'for ClustId compared objects
Dim ooDist As Double          'Auxiliary variable for distance from current object
Dim a As Double, b As Double         'Members of the silhouette formula
Dim siObj As Double                  'Silhouette of the current object
'determine distances to all other objects u. sum them up in dd by clusters
   For i  1 To noOfObj
    cid  CInt(Obj(i, noOfAttr + 1))         'ClustId of the current object
    For k  1 To NoOfClust
       dd(k)  0
    Next k

    'calculate average distances
    For k  1 To NoOfClust
       dd(k)  dd(k) / NoOfLmts(k)
    Next k
    'calculate the terms of the silhouette formula for the object i
    a  dd(cid)
    If a  0 Then
       siObj  0
```

```
        Else
            b  1.79769313486232E+200
            For k  1 To NoOfClust
                If k <> cid Then
                    If dd(k) < b Then b  dd(k)
                End If
            Next k
            siObj  (b  a) / IIf(a > b, a, b)
        End If
        'Update the totals for the cluster silhouette of cluster cid
        siClu(cid)  siClu(cid) + siObj
        'continue the sum of the silhouette of the cluster division
        siClu(NoOfClust + 1)  siClu(NoOfClust + 1) + siObj
    Next i
    'find the average silhouettes
    For i  1 To NoOfClust
        siClu(i)  siClu(i) / NoOfLmts(i)
    Next i
    siClu(NoOfClust + 1)  siClu(NoOfClust + 1) / noOfObj
    GSilhouette  siClu(NoOfClust + 1)
End Function

'provides the silhouettes of all clusters and overall silhouette of those in PopRng
'describes the cluster distribution, together with the names of the clusters, in the form of a two-dimensional array.
Public Function SilhouetteM(ByVal PopRng As Range) As Variant
    Dim c As Variant   'für die ClusterNamen
    Dim s As Variant   'für die Silhouettenwerte
    Dim m( ) As Variant 'für Clusternamen und Silhouettenwerte
    Dim i As Integer
    c  Clusters(PopRng)
    s  Silhouette(PopRng)
    ReDim m(1 To UBound(c) + 1, 1 To 2)
    For i  1 To UBound(m, 1)  1
        m(i, 1)  c(i)
        m(i, 2)  s(i)
    Next i
    m(UBound(m, 1), 1)  "Together"
    m(UBound(m, 1), 2)  s(UBound(s))
    SilhouetteM  m
End Function

'provides silhouettes of all clusters and overall silhouette of those in PopRng
'describes cluster distribution as one-dimensional array with indices: ClustId.
Public Function Silhouette(ByVal PopRng As Range) As Variant
    Dim NoOfClust As Integer
    Dim noOfAttr As Integer
    Dim noOfObj As Long
    Dim Obj( ) As Double       'Matrix; Spalte 0: ObjId; letzte Spalte: ClustId
    Dim Clust( ) As String     'Clusternamen; Index fungiert als ClustId
    Dim NoOfLmts( ) As Long    'Anzahl Obj in den Clustern; Index entspricht ClustId
    Dim i As Long, j As Long, k As Long
```

```
    ReDim Clust(1 To 1)
    ReDim NoOfLmts(1 To 1)
    'Accept values from the range in arrays Obj and Clust
    With PopRng
       noOfObj  .Rows.Count
       noOfAttr  .Columns.Count  1
       ReDim Obj(1 To noOfObj, 0 To noOfAttr + 1)
      'Register the clusters in Clust
       Clust(1)  CStr(.Cells(1, noOfAttr + 1))   'erstes Cluster setzen
       Dim aktClu As String
       Dim aktCid As Long
       For i  1 To noOfObj
          aktClu  Trim(CStr(.Cells(i, noOfAttr + 1)))
          k  1
          Dim found As Boolean
          found  False
          Do While k < UBound(Clust) And Not found
             If aktClu  Clust(k) Then
                found  True
             Else
                k  k + 1
             End If
          Loop
          If Not found Then
             ReDim Preserve Clust(1 To UBound(Clust) + 1)
             Clust(UBound(Clust))  aktClu
             ReDim Preserve NoOfLmts(1 To UBound(Clust) + 1)
          End If
       Next i
       'Insert objects in Obj, set ObjId and ClustId and update NoOfLmts
       For i  1 To noOfObj
          Obj(i, 0)  i         'ObjId
          Obj(i, noOfAttr + 1)  sucheClustId(Trim(CStr(.Cells(i, noOfAttr + 1))), Clust)
          NoOfLmts(Obj(i, noOfAttr + 1))  NoOfLmts(Obj(i, noOfAttr + 1)) + 1
          For j  1 To noOfAttr                 'Attributwerte zuweisen
             Obj(i, j)  CDbl(.Cells(i, j))
          Next j
       Next i
    End With
NoOfClust  UBound(Clust)
Dim dd( ) As Double                    'takes cumulative distances from active objects
ReDim dd(1 To NoOfClust)
ReDim siClu(1 To NoOfClust + 1)       'for silhouettes of clusters/cluster division
Dim cid As Integer, cid2 As Integer        'for ClustId compared objects
Dim ooDist As Double                   'Auxiliary variable for current object distance
Dim a As Double, b As Double       'Members of the silhouette formula
Dim siObj As Double                     'Silhouette of the current object
'determine the silhouettes of all objects and use them to continue the cluster
silhouettes
          For i  1 To noOfObj
             cid  CInt(Obj(i, noOfAttr + 1))         'ClustId des aktuellen Objekts
             For k  1 To NoOfClust
```

```
            dd(k)  0
        Next k
'determine distances to all other objects u. sum them up in dd by clusters
        For j  1 To noOfObj
            If i <> j Then
                cid2  CInt(Obj(j, noOfAttr + 1)) 'ClustId des Vergleichsobjekts
                ooDist  EuDist(ObjAttr(Zeilenvektor(Obj, i), noOfAttr), _
                        ObjAttr(Zeilenvektor(Obj, j), noOfAttr))
                dd(cid2)  dd(cid2) + ooDist
            End If
        Next j
        'calculate average distances
        For k  1 To NoOfClust
            dd(k)  dd(k) / NoOfLmts(k)
        Next k
        'calculate the terms of the silhouette formula for the object i
        a  dd(cid)
        If a  0 Then
            siObj  0
        Else
            b  1.79769313486232E+200
            For k  1 To NoOfClust
                If k <> cid Then
                    If dd(k) < b Then b  dd(k)
                End If
            Next k
            siObj  (b  a) / IIf(a > b, a, b)
        End If
        'Update the totals for the cluster silhouette of cluster cid
        siClu(cid)  siClu(cid) + siObj
        'continue the sum of the silhouette of the cluster division
        siClu(NoOfClust + 1)  siClu(NoOfClust + 1) + siObj
    Next i
    'find the average silhouettes
    For i  1 To NoOfClust
        siClu(i)  siClu(i) / NoOfLmts(i)
    Next i
    siClu(NoOfClust + 1)  siClu(NoOfClust + 1) / noOfObj
    Silhouette  WorksheetFunction.Transpose(siClu)
    Silhouette  siClu
End Function
```

C.3 PSO-clustering MATLAB code

Please find below the code for all processing of the PSO-clustering algorithm in MATLAB (MathWorks, 2019). The same process as KMC were used, adjusted for PSO. The author of the code is Augusto Ballardini from IraLab. Some alterations were done to input test data, and to use the KMC cluster centroids as input for the hybrid PSO runs (Ballardini, 2018a).

The original code is available from https://github.com/iralabdisco/pso-clustering.

```
% Author:  Augusto Luis Ballardini
% Email:   augusto.ballardini@disco.unimib.it
% Website: http://www.ira.disco.unimib.it/people/ballardini-augusto-luis/

% This library is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
% Permission is granted to copy, distribute and/or modify this document
% under the terms of the GNU Free Documentation License, Version 1.3
% or any later version published by the Free Software Foundation;
% with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
% A copy of the license is included in the section entitled "GNU
% Free Documentation License".

% The following code is inspired by the following paper:
% Van Der Merwe, D. W.; Engelbrecht, AP., "Data clustering using particle swarm optimization,"
% Evolutionary Computation, 2003. CEC '03. The 2003 Congress on , vol.1, no., pp.215,220 Vol.1, 8-12 Dec. 2003
% doi: 10.1109/CEC.2003.1299577
% URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1299577&isnumber=28874

clear;
close all;

%rng('default') % For reproducibility

% INIT PARTICLE SWARM
centroids = 4;        % == clusters here (aka centroids)
dimensions = 5;       % how many dimensions in each centroid
plot3D = 1;           % three dimensional plot (1 = Yes; 0 = No)
particles = 3362;     % how many particles in the swarm, aka how many solutions
iterations = 20;      % iterations of the optimization alg.
simtime=0.01;         % simulation delay btw each iteration
dataset_subset = 1;   % for the IRIS dataset, change this value from 0 to 2
write_video = true;   % enable to grab the output picture and save a video
hybrid_pso = true;    % enable/disable hybrid_pso
```

```
manual_init = false;    % enable/disable manual initialization (only for dimensions={2,3})

% VIDEO GRUB STUFF...
if write_video
   writerObj = VideoWriter('PSO.avi');
   writerObj.Quality=100;
%    writerObj.FrameRate=30;
   open(writerObj);
end

% LOAD B2B ANALYSIS DATA (IRIS DATASET); USE WITH CARE!
%load fisheriris.mat
%meas = meas(:,1+dataset_subset:dimensions+dataset_subset); %RESIZE THE DATASET WITH
CURRENT DIMENSIONS; USE WITH CARE!
PSO1_Data = readtable("PSO1_Analysis_Data.xlsx");

%Target dataset replacing Iris data
Swarm = PSO1_Data{:,1+dataset_subset:dimensions+dataset_subset};
dataset_size = size (Swarm);

% EXECUTE K-MEANS
% only for hybrid PSO, else random centres are used initially
if hybrid_pso
   fprintf('Extract KMC K-Means centroids\n');
 %Using MATLAB kmeans process to obtain intial cluster centres
   [idx,KMEANS_CENTROIDS] = kmeans(Swarm,centroids, 'dist','sqEuclidean',
'display','iter','start','uniform','onlinephase','off');
% Using KMC output of centroids
  KMC1_Data = readtable("PSO1_KMC1_Centres.xlsm");
  KMEANS_CENTROIDS = KMC1_Data{:,1+dataset_subset:dimensions+dataset_subset};

   fprintf('\n');
end

% GLOBAL PARAMETERS (the paper reports this values 0.72;1.49;1.49)
w  = 0.72; %INERTIA
c1 = 1.49; %COGNITIVE
c2 = 1.49; %SOCIAL

% PLOT STUFF... HANDLERS AND COLORS
pc = []; txt = [];
cluster_colors_vector = rand(particles, 3);

% PLOT DATASET
fh=figure(1);
%hold on;  %will cause original plot to be overwritten with following plots?
if dimensions >= 3 && plot3D == 1
   plot3(Swarm(:,1),Swarm(:,2),Swarm(:,3),'k*');
   view(3);
else
   plot(Swarm(:,1),Swarm(:,2),'k*');
end
```

```
 % PLOT STUFF .. SETTING UP AXIS IN THE FIGURE
axis equal;
%axis(reshape([min(Swarm)-2; max(Swarm)+2],1,[]));
%axis(reshape([min(Swarm)-2; max(Swarm)+2],1,[]));
hold off;

% SETTING UP PSO DATA STRUCTURES
% Here the variables needed in the pso clustering are pre-initialized.
% Please note that swarm vel, swarm pos and swarm best maintains the values
% for all the swarms (aka particles)
% 'c' =
% 'ranges' is used to scale the initial randomized values to something
% inside the range of the input data (just to not have useless values
% outside the valid range, i.e. the range of the data).
% 'swarm fitness' is initially set as infinite. this is the "value" that
% will become smaller and smaller (i.e. minimizating the fitness function)

swarm_vel = rand(centroids,dimensions,particles)*0.1;
swarm_pos = rand(centroids,dimensions,particles);
swarm_best = zeros(centroids,dimensions);
c = zeros(dataset_size(1),particles);
ranges = max(Swarm)-min(Swarm); %%scale
swarm_pos = swarm_pos .* repmat(ranges,centroids,1,particles) +
repmat(min(Swarm),centroids,1,particles);
swarm_fitness(1:particles)=Inf;

% KMEANS_INIT
if hybrid_pso
   swarm_pos(:,:,1) = KMEANS_CENTROIDS;
end

% MANUAL INITIALIZATION (only for dimension 2 and 3)
if manual_init
   if dimensions >= 3 && plot3D == 1
      % MANUAL INIT ONLY FOR THE FIRST PARTICLE
         swarm_pos(:,:,1) = [6 3 4; 5 3 1];
   else
      % KEYBOARD INIT ONLY FOR THE FIRST PARTICLE
         swarm_pos(:,:,1) = ginput(2);
   end
end

for iteration=1:iterations

   %CALCULATE EUCLIDEAN DISTANCES TO ALL CENTROIDS
   distances=zeros(dataset_size(1),centroids,particles);
   for particle=1:particles
      for centroid=1:centroids
         distance=zeros(dataset_size(1),1);
         for data_vector=1:dataset_size(1)
            %Swarm(data_vector,:)
            distance(data_vector,1)=norm(swarm_pos(centroid,:,particle)-Swarm(data_vector,:));
```

```
        end
        distances(:,centroid,particle)=distance;
      end
    end

    %ASSIGN MEASURES with CLUSTERS
    for particle=1:particles
      [value, index] = min(distances(:,:,particle),[],2);
      c(:,particle) = index;
    end

    % PLOT STUFF... CLEAR HANDLERS
    delete(pc); delete(txt);
    pc = []; txt = [];

    % PLOT STUFF...
    hold on;
    for particle=1:particles
      for centroid=1:centroids
        if any(c(:,particle) == centroid)
          if dimensions >= 3 && plot3D == 1
            pc                                                    =                                                    [pc
plot3(swarm_pos(centroid,1,particle),swarm_pos(centroid,2,particle),swarm_pos(centroid,3,particle)
,'*','color',cluster_colors_vector(particle,:))];
          else
            pc                                                    =                                                    [pc
plot(swarm_pos(centroid,1,particle),swarm_pos(centroid,2,particle),'*','color',cluster_colors_vector(
particle,:))];
          end
        end
      end
    end
    set(pc,{'MarkerSize'},{12})
    hold off;
    %CALCULATE GLOBAL FITNESS and LOCAL FITNESS:=swarm_fitness
    average_fitness = zeros(particles,1);
    for particle=1:particles
      for centroid = 1 : centroids
        if any(c(:,particle) == centroid)
          local_fitness=mean(distances(c(:,particle)==centroid,centroid,particle));
          average_fitness(particle,1) = average_fitness(particle,1) + local_fitness;
        end
      end
      average_fitness(particle,1) = average_fitness(particle,1) / centroids;
      if (average_fitness(particle,1) < swarm_fitness(particle))
        swarm_fitness(particle) = average_fitness(particle,1);
        swarm_best(:,:,particle) = swarm_pos(:,:,particle);    %LOCAL BEST FITNESS
      end
    end
    [global_fitness, index] = min(swarm_fitness);      %GLOBAL BEST FITNESS
    swarm_overall_pose = swarm_pos(:,:,index);        %GLOBAL BEST POSITION
```

```
   % SOME INFO ON THE COMMAND WINDOW
   fprintf('%3d. global fitness is %5.4f\n',iteration,global_fitness);
   %uicontrol('Style','text','Position',[40 20 180 20],'String',sprintf('Actual fitness is: %5.4f',
global_fitness),'BackgroundColor',get(gcf,'Color'));
   pause(simtime);

   % VIDEO GRUB STUFF...
   if write_video
      frame = getframe(fh);
      writeVideo(writerObj,frame);
   end

   % SAMPLE r1 AND r2 FROM UNIFORM DISTRIBUTION [0..1]
   r1 = rand;
   r2 = rand;
   % UPDATE CLUSTER CENTROIDS
   for particle=1:particles
      inertia = w * swarm_vel(:,:,particle);
      cognitive = c1 * r1 * (swarm_best(:,:,particle)-swarm_pos(:,:,particle));
      social = c2 * r2 * (swarm_overall_pose-swarm_pos(:,:,particle));
      vel = inertia+cognitive+social;

      swarm_pos(:,:,particle) = swarm_pos(:,:,particle) + vel ;   % UPDATE PARTICLE POSE
      swarm_vel(:,:,particle) = vel;                    % UPDATE PARTICLE VEL
   end
end
% PLOT THE ASSOCIATIONS WITH RESPECT TO THE CLUSTER
hold on;
particle=index; %select the best particle (with best fitness)
cluster_colors = ['m','g','y','b','r','c','g'];
for centroid=1:centroids
   if any(c(:,particle) == centroid)
      if dimensions >= 3 && plot3D == 1

plot3(Swarm(c(:,particle)==centroid,1),Swarm(c(:,particle)==centroid,2),Swarm(c(:,particle)==centroi
d,3),'o','color',cluster_colors(centroid));
      else
plot(Swarm(c(:,particle)==centroid,1),Swarm(c(:,particle)==centroid,2),'o','color',cluster_colors(centr
oid));
      end
   end
end
hold off;
 % VIDEO GRUB STUFF...
if write_video
   frame = getframe(fh);
   writeVideo(writerObj,frame);
   close(writerObj);
end

% SAY GOODBYE
fprintf('\nEnd, global fitness is %5.4f\n',global_fitness);
```

C.4 Easy CHAID SPSS functions used

The Easy Chaid opensource application from Rafael Troiani (2016) was tested against SPSS functions (IBM, 2020). The JavaScript[14] code for Easy Chaid (Troiani, 2016) is available from:

https://github.com/rafatro/EasyCHAID

The SPSS functions used are shown here as described in syn_tree_chaid.html and tree_credit_howto_01.html in the online IBM knowledge center (IBM, 2014). The application itself is available in the link http://www.easychaid.com and runs online.

C.4.1 CHAID Subcommand (TREE command)

The CHAID subcommand sets parameters for a CHAID tree.

Each keyword in the subcommand is followed by an equals sign (=) and the value for that keyword (IBM, 2020).

Example

```
TREE risk [o] BY income age creditscore
 /METHOD TYPE=CHAID
 /CHAID ALPHASPLIT=.01 INTERVALS=age income (10) creditscore (5).
```

ALPHASPLIT Keyword

The ALPHASPLIT keyword specifies the significance level for splitting of nodes. An independent variable will not be used in the tree if significance level for the split statistic (χ^2or F) is less than or equal to specified value.

- Specify a value greater than zero and less than 1.
- The default value is 0.05.

ALPHAMERGE Keyword

The ALPHAMERGE keyword specifies the significance level for merging of predictor categories. Small values tend to result in a greater degree of merging.

- Specify a value greater than zero and less than or equal to 1.
- The default value is 0.05.
- If you specify a value of 1, predictor categories are not merged.

[14] Often abbreviated as JS, JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions (Wikipedia contributors, 2020b)

- ALPHAMERGE is available only for the CHAID method. For Exhaustive CHAID, the keyword is ignored, and a warning is issued.

SPLITMERGED Keyword

The SPLITMERGED keyword specifies whether predictor categories that are merged in a CHAID analysis are allowed to be resplit.

NO. *Merged predictor categories cannot be resplit.* This is the default.
YES. *Merged predictor categories can be resplit.*

CHISQUARE Keyword

For nominal dependent variables, the CHISQUARE keyword specifies the χ^2 measure used in CHAID analysis. For ordinal and scale dependent variables, the keyword is ignored and a warning is issued.

PEARSON. *Pearson* χ^2. This is the default.

LR. Likelihood-ratio χ^2.

CONVERGE Keyword

For nominal and ordinal dependent variables, the CONVERGE keyword specifies the convergence value for estimation of the CHAID model.

- Specify a value greater than zero and less than 1.
- The default value is 0.05.
- If the dependent variable is nominal or scale, this keyword is ignored, and a warning is issued.

MAXITERATIONS Keyword

For nominal and ordinal dependent variables, the MAXITERATIONS keyword specifies the maximum number of iterations for estimation of the CHAID model.

- Specify a positive integer value.
- The default value is 100.
- If the dependent variable is nominal or scale, this keyword is ignored, and a warning is issued.

ADJUST Keyword

The ADJUST keyword specifies how to adjust significance values for multiple comparisons.

BONFERRONI. *Significance values are adjusted using the Bonferroni method.* This is the default.
NONE. *Significance values are not adjusted.*

INTERVALS Keyword

In CHAID analysis, scale independent (predictor) variables are always banded into discrete groups (for example, 0-10, 11-20, 21-30, and so on) prior to analysis. You can use the INTERVALS keyword to control the number of discrete intervals for scale predictors.

- By default, each scale predictor is divided into 10 intervals that have approximately equal numbers of cases.
- The INTERVALS keyword is ignored if the model contains no scale independent variables.

  Example: INTERVALS=5. The value must be a positive integer less than or equal to 64.

  Multiple lists of variables can be specified. Example: INTERVALS=age income (10) creditscore (5). The value must be a positive integer less than or equal to 64.

C.4.2 About Easy CHAID

Every node is split according to the variable that better discriminates the observations on that node. The new nodes are split again and again until reaching the minimum node size (user-defined) or the remaining variables don't differentiate enough for a new split. Independent variables must be categorical like gender male and female or like marital status married, single, divorced and widow(er). Numerical continuous variables like age, height, weight, or income must be transformed into categories before using them in CHAID. Whereas original CHAID algorithm accepts numerical continuous variable as the dependent variable, this implementation of CHAID accepts only categorical variables (Troiani, 2016).

C.5 JustNN Network setup guide

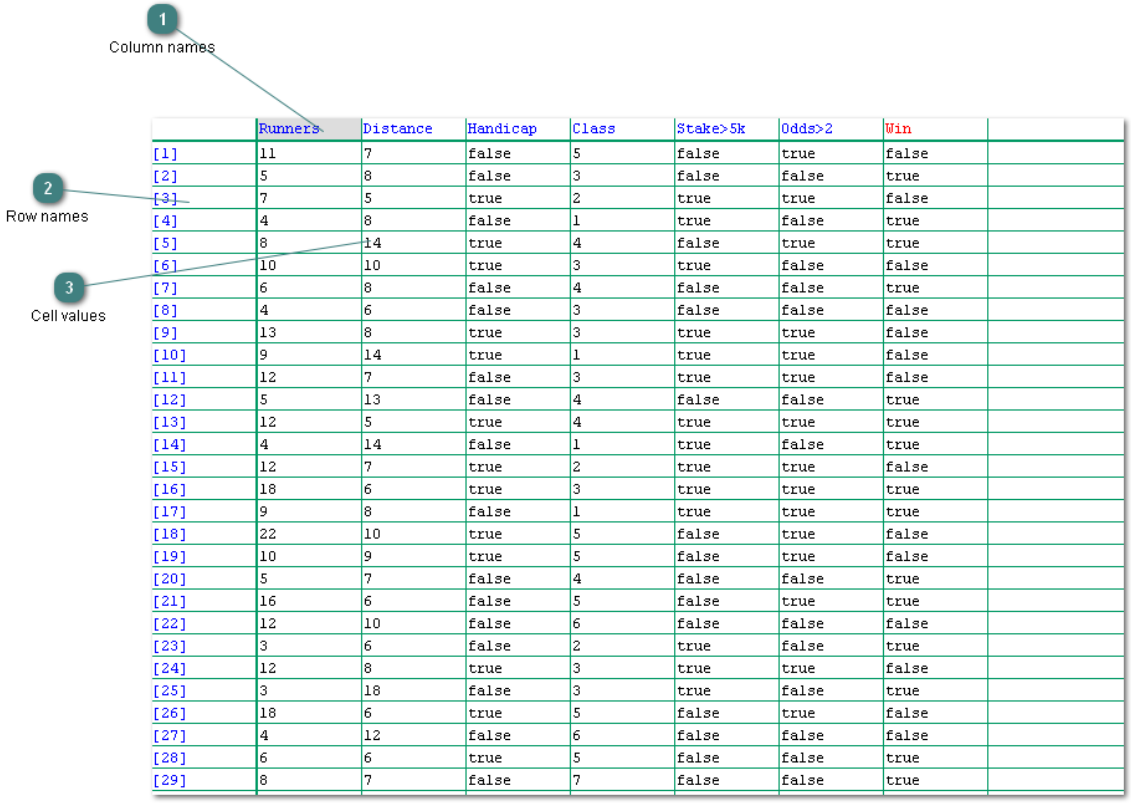
The code for JustNN is not available. Instead, extracts from the JustNN user guide (Wolstenholme, 2015) is given below. References to the commercial version, EasyNN (Wolstenholme, 2002), can be found on Github at some links from developers, to list a few:

https://github.com/tristan099/EasyNN

out/production/TextAnalysis/InputData/262_05169341.txt

https://github.com/BeTechLabs/Neural-Networks-Tutorials

C.5.1 The Grid

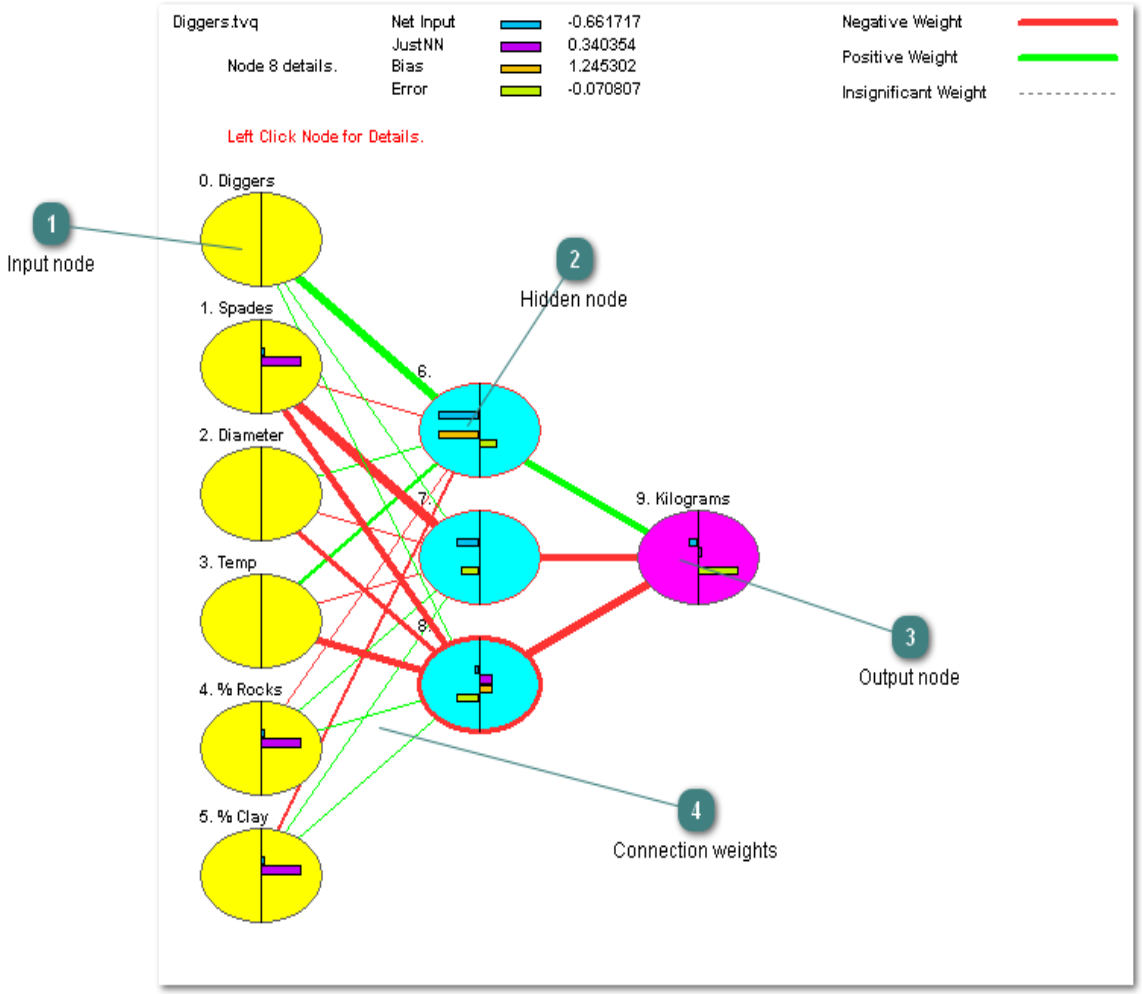


	Runners	Distance	Handicap	Class	Stake>5k	Odds>2	Win
[1]	11	7	false	5	false	true	false
[2]	5	8	false	3	false	false	true
[3]	7	5	true	2	true	true	false
[4]	4	8	false	1	true	false	true
[5]	8	14	true	4	false	true	true
[6]	10	10	true	3	true	false	false
[7]	6	8	false	4	false	false	true
[8]	4	6	false	3	false	false	false
[9]	13	8	true	3	true	true	false
[10]	9	14	true	1	true	true	false
[11]	12	7	false	3	true	true	false
[12]	5	13	false	4	false	false	true
[13]	12	5	true	4	true	true	true
[14]	4	14	false	1	true	false	true
[15]	12	7	true	2	true	true	false
[16]	18	6	true	3	true	true	true
[17]	9	8	false	1	true	true	true
[18]	22	10	true	5	false	true	false
[19]	10	9	true	5	false	true	false
[20]	5	7	false	4	false	false	true
[21]	16	6	false	5	false	true	true
[22]	12	10	false	6	false	false	false
[23]	3	6	false	2	true	false	true
[24]	12	8	true	3	true	true	false
[25]	3	18	false	3	true	false	true
[26]	18	6	true	5	false	true	false
[27]	4	12	false	6	false	false	false
[28]	6	6	true	5	false	false	true
[29]	8	7	false	7	false	false	true

The Grid view shows all the Examples arranged in rows and all the Input/Outputs arranged in columns. The first column contains the Example types and names. The first row contains the Input/Output types and names. Everything on the Grid can be edited by moving to the cell containing the

value and then pressing the enter key to start the Edit Grid dialog. The cell can be selected either using the arrow keys or the mouse. A single click will select the cell and a double click will start the Edit Grid dialog. A double click on the Example name cell will select the whole row and a double click on the Input/Output name cell will select the whole column. The row or the column can be deselected by pressing the Esc key.

C.5.2 The Network



The **Network** view shows how the nodes in a **JustNN** neural network are interconnected.

a. How to create a new neural network

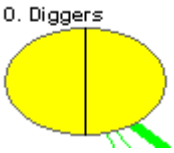
A new neural network can be created from the **Grid** by pressing the **New Network** toolbar button or selecting **Action > New Network**. This will produce the New Network dialog. This dialog allows the neural network configuration to be specified. The dialog will already contain the necessary information to generate a neural network that will be capable of learning the information in the Grid. However, the generated network may take a long time to learn and it may give poor results when tested. A better neural network can be generated by checking Grow hidden layer 1 and allowing **JustNN** to determine the optimum number of nodes and connections.

It is rarely necessary to have more than one layer of hidden nodes but **JustNN** will generate two or three hidden layers if Grow hidden layer 2 and Grow hidden layer 3 are checked.

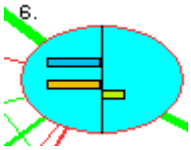
The time that JustNN will spend looking for the optimum network can be controlled by setting the Growth rate variables. Every time that the period expires JustNN will generate a new neural network slightly different from the previous one. The best network is saved.

b. Components of the neural network

1 **Input node**

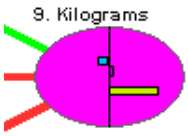


2 **Hidden node**



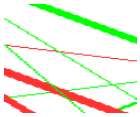
Hidden nodes are fully connected to input nodes, output nodes or other layers of hidden nodes.

3 Output node



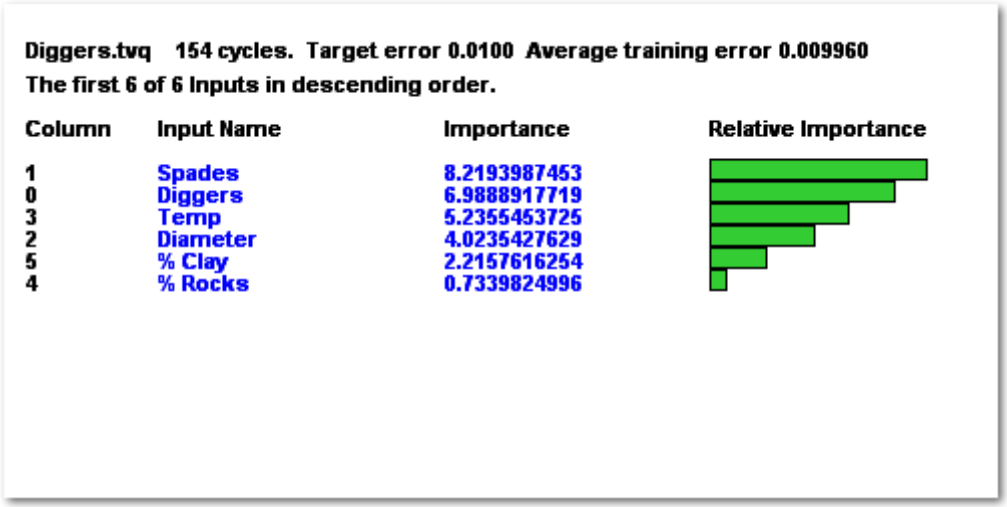
Output nodes are connected to the output columns in the grid.

4 Connection weights



The input layer is fully connected to the first hidden layer. Each connection has a weight that is updated while the network is learning. Hidden layers are fully connected the next hidden layer or the output layer.

C.5.3 Input Importance



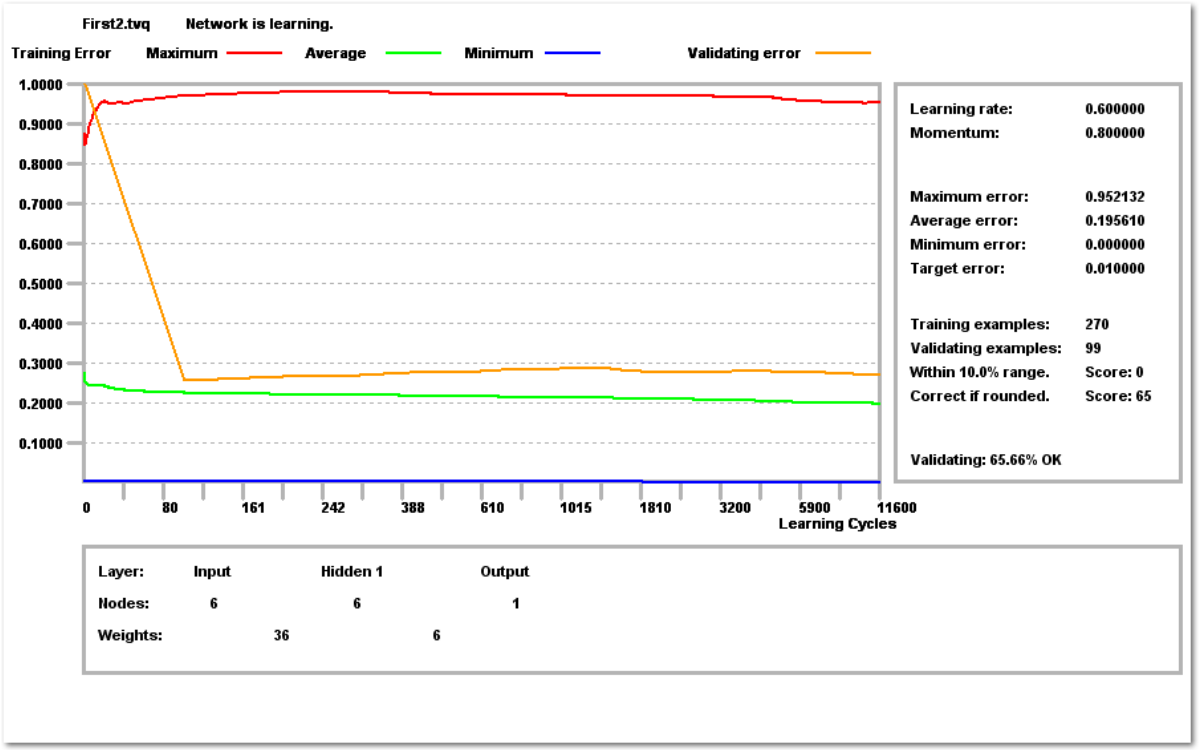
Diggers.tvq 154 cycles. Target error 0.0100 Average training error 0.009960

The first 6 of 6 Inputs in descending order.

Column	Input Name	Importance	Relative Importance
1	Spades	8.2193987453	
0	Diggers	6.9888917719	
3	Temp	5.2355453725	
2	Diameter	4.0235427629	
5	% Clay	2.2157616254	
4	% Rocks	0.7339824996	

The **Input Importance** view shows the importance and the relative importance of each Input column. The Importance is the sum of the absolute weights of the connections from the input node to all the nodes in the first hidden layer. The inputs are shown in the descending order of importance from the most important input.

C.5.4 Learning Progress



The **Learning Progress** view shows how learning is progressing. Up to 5000 graph points are recorded. This is sufficient for over 200,000,000 learning cycles. The graph is produced by sampling these points. The horizontal axis is nonlinear to allow the whole learning progress to be displayed. As more cycles are executed the graph is squashed to the left. The scaled errors for all example rows are used. The red line is the maximum example error, the blue line is the minimum example error and the green line is the average example error. The orange line is the average validating error.