The Mining and Visualisation of Application Services Data

Ronald Knoetze, Janet Wesson, Charmain Cilliers
Department of Computer Science and Information Systems
PO Box 77000, Nelson Mandela Metropolitan University, Port Elizabeth, 6031
Tel: (041) 504 2323, Fax: (041) 504 2831
Email: {Ronald.Knoetze, Janet.Wesson, Charmain.Cilliers}@nmmu.ac.za
Topic: Network Engineering, Sub-Topic: Modelling and Simulation

ABSTRACT – Application services form an integral part of networks and allow organisations to operate efficiently. Many network monitoring tools exist but most of these do not provide in-depth reports on network usage. Techniques that can identify patterns of network usage can be useful to optimise network performance. This paper discusses the identification of data mining algorithms and visualisation techniques which can be used to assist network managers make predictions for network performance.

Index Terms –application services data, data mining, information visualisation

I. INTRODUCTION

The Nelson Mandela Metropolitan University (NMMU) has an extensive network that supports various application services. These application services allow users to work together in a collaborative environment while located remotely from each other. Problems can occur when many users access the application service simultaneously, which can slow the network down considerably.

One way to monitor network performance is with the use of network monitoring tools, such as PacketShaper. PacketShaper was installed on the NMMU network between the application service, Integrated Tertiary Software (ITS), and the client computers. Unfortunately this monitoring tool only provides reports that show basic network usage. There is no facility to make predictions about future trends with regard to the application service and the network.

This paper discusses application services and the type of data collected by network monitoring tools. An overview of different types of data mining and visualisation techniques is discussed together with the process involved in selecting the appropriate techniques. Based on these findings, the design of a prototype that presents these results is shown.

II. APPLICATION SERVICES

The NMMU has an extensive network infrastructure supporting various LAN-based application services that are distributed across the campus. The main application service on the NMMU network is ITS which contain student records, human resources and the finance system. Network monitoring tools are available that monitor networks, but many of these tools provide naïve reports regarding network performance and do not provide facilities to make predictions about network performance. To assist network administrators make predictions about network usage and prevent degradation of application service performance, data mining systems can be used.

III. RELATED WORK

Several data mining systems exist that perform data analysis, including Microsoft SQL Server 2005 called Yukon [14] and Clementine [18]. The architecture of each of these data mining systems is similar and each offer a selection of data mining algorithms for the user to choose from when performing analysis on the data. Complex visualisation techniques to assist the user in understanding the results are also provided. These systems will work on generic data, but are not specifically focused on application services data.

Yukon focuses on targeted mailing, forecasting, market analysis and sequence clustering. Clementine is used for mining large scale databases including text mining, fraud detection and market analysis. This paper, however, focuses on the design of a prototype for specifically mining and visualising application services data.

IV. DATA ANALYSIS

The selection of the data mining algorithm and corresponding visualisation technique depends heavily on the type of data that is to be analysed [1]. Data analysis is required to ensure correct data identification, data cleaning and data integration and transformation. This data can then be stored in a single subject-orientated, integrated, time-variant and non-volatile data warehouse.

Network attributes need to be identified to assist in removing irrelevant variables. Network attributes are properties of a network that relate to performance and reliability of the network [2]. Research by Lowecamp et al. [2], Leese [3] and Barnford [4] identified four main network attributes, including *Delay*, *Throughput*, *Response Time* and *Utilization*.

Network attributes	PacketShaper variables
Delay (ms)	Network delay, Server delay
Throughput (kbps)	Total pass-through bytes, Total received bytes, total sent bytes
Response Time (ms)	Average-round-trip-time
Utilization (bps)	Average bits per second, Total transferred bytes

Table 1 - Network attributes and associated PacketShaper variables

The network variables collected by PacketShaper are divided into three groups: *class*, *link* and *partition*. Table 1 indicates the network attributes mapped to the corresponding PacketShaper variables.

Delay is the amount of time it takes a packet of data to get from one point in the network to another, and is measured in milliseconds (ms). *Network delay* is the time data is spent in transit when a client and server exchange data. *Server delay* is the time taken for the server to process a client's request. *Throughput* is the total amount of data transferred in a specific period of time and is measured in kilobits per second (kbps). These variables indicate the number of bytes sent and received in one second. *Response time* is the time taken from when a request was made until receiving the application's response, and is measured in milliseconds (ms). *Utilization* measures the average use of a particular resource over time, measured in bits per second (bps).

The network variables collected by PacketShaper are stored in multiple log files, one for each of the three groups. Data integration was used to merge these log files and store them in a data warehouse for application of data mining algorithms.

V. Data Mining Algorithms

Data mining is commonly defined as extracting hidden information that can be used for prediction from large amounts of data [5]. The main objective is to identify valid, novel and understandable patterns for existing data [6]. The choice of data mining algorithms depends largely on the type of data that is going to be used for analysis and what needs to be done to the data. In the problem domain of networks, the data that was identified is the application services data collected by PacketShaper. This data will be used to make predictions for network performance.

Data classification is required when selecting the data mining algorithm. The incorrect classification of the data could lead to the incorrect algorithm being chosen for analysis. Data mining is split into two kinds of learning: supervised and unsupervised. Supervised learning is used to predict values, while unsupervised learning tries to find relationships within the data [7]. Data mining algorithms are further categorised into three categories:

- Classification (supervised).
- Clustering (unsupervised), and
- Association (unsupervised).

A. Classification

Classification is a two-step process [1]. The first step creates a model to describe a pre-determined set of data classes, and the second step uses the model for classification. The accuracy of the model is estimated, and based on this accuracy the model can be considered acceptable to classify future data. Once the model has been created, any new data can be compared to the model and predictions can then be made.

There are two main types of classification algorithms: namely decision trees and Naïve Bayesian. Decision trees are able to generate understandable rules and work well with numeric data. They also provide a clear indication of which fields are most important for classification [8]. Naïve Bayesian also works well on numeric data but performs poorly when variables are highly correlated.

The data collected by the PacketShaper log files is numeric and this was used as a basis for initially selecting the decision tree for classification. Decision trees also provide an example of the training set and this assists in generating easy to understand rules.

The form of the output that is produced by the decision tree algorithm is *A(x)* → *B(y)*, where *A* is the parent node, *B* the child node, *x* is the user defined threshold that is used to split the log files, and *y* is the number of log file entries for the child node. For example, *normalised-network-delay(0-50)* → *server-delay-median(*145*)* can be interpreted as *normalised-network-delay* is the parent, with (0-50) the user defined threshold that splits the log file with s*erver-delay-median* as the child node with 145 log file entries satisfying the threshold. This form of output is hierarchical.

B. Clustering

Clustering is the process of grouping objects that are similar to one another based on certain criteria [9]. Clusters are collections of data objects that are similar to one another within clusters and dissimilar to objects in other clusters. Cluster analysis can also be used to identify sparse and dense regions.

Algorithms used in clustering include k-means, neural nets and DBSCAN. K-means algorithms are suitable for discovering clusters and utilises the entire data source rather than requiring training samples. Neural nets are suitable for non-linear data and can produce good results in complicated domains. The disadvantages of neural nets include that all non-numeric data is required to be converted to number values and must be normalised to be in the range 0-1. Neural nets are also time consuming and memory intensive.

Neural nets were disregarded as the data collected by PacketShaper is linear and numeric (Table 1). K-means was selected for clustering as it can use either the whole dataset or only portions of it. The output of the k-means algorithm is easy to understand and can handle outlier data.

Based on the grouping of the PacketShaper variables into their respective network attribute as shown in Table 1, cluster analysis can be performed within each attribute and can cluster averages, totals and means. For example, the k-means algorithm will be applied to *network delay* and s*erver delay* within the *Delay* network attribute.

The k-means algorithm identifies initial cluster medians per network variable and each is assigned a cluster number. A distance function is used to calculate the distance between

the initial cluster medians and all other log file entries. The log file entries are then assigned to the closest cluster. This process continues recursively until no more changes between the clusters occur.

The form of the output that is produced by the k-means algorithm is (*A, B, N*) where *A* is the cluster number, $B=\{x_1,x_2,\ldots,x_n\}$, $n\geq 2$ the number of network variables, x_i is the median of cluster *i*, and *N* is the number of log file entries that fall within that cluster. For example, when clustering within the *Delay* attribute, (2, (122.5,146.4),74) can be interpreted as cluster 2 with median (122.5,146.4) for the two *Delay* network variables with 74 data tuples within the cluster 2.

C. Association

Association rule mining is used to find interesting associations or correlation relationships within large sets of data results [7]. Association rule algorithms include Apriori and correlation analysis.

Apriori can be used for large databases. The disadvantage is that Apriori requires many scans of the database. Apriori also does not suit certain types of data. In the case of the data collected by PacketShaper, Apriori would not be a suitable choice. Apriori was initially considered as the algorithm for association, but the IF-THEN types of rules created by Apriori do not suit the data and Apriori did not provide a means to view comparisons between the network attributes as all the log file entries have the same variables.

Correlation was considered for association based on the network attributes identified, namely *Delay*, *Throughput*, *Response Time* and *Utilisation*. In the case of correlation, comparison could not be done within each attribute as it would provide no useful information. More interesting results could be provided when correlating data across the different network attributes. This could include correlating *Delay* with *Response Time*, or any combination of the four network attributes. The correlation algorithm provides a correlation coefficient that is used to calculate the gradient of the line showing positive or negative correlation between the network attributes.

The output of the correlation algorithm is of the form (*A, B, r*) where *A* and *B* is a subset of the network attributes and *r* is the correlation coefficient. For example, (*Delay, Response Time*, 1.34) can be interpreted as the combined dataset of *Delay* and *Response Time* with a correlation coefficient of 1.34.

VI. Visualisation of Data Mining Results

Data mining results are sometimes difficult to understand [1] and can be improved by use of information visualisation (IV). IV is the use of interactive, visual representations of data to gain knowledge [10]. Combining data mining with visualisation techniques can help the user by assisting the user's perception of data mining [15]. The use of data manipulation techniques such as pan, zoom, drill-down and scrolling assists the user in gaining insight into the results [16].

For each group of data mining algorithms, suitable visualisation techniques are needed. The visualisation techniques depend on the type of output produced by the data mining algorithms. For each of the algorithms identified in the previous sections the output of the algorithm was identified. This output was used to select an appropriate visualisation technique for each of the three algorithms.

A. Visualising Classification

In the previous section, the data output of the decision tree algorithm was identified as hierarchical data. Research conducted by Barlow and Neville [11] regarding hierarchical data compared four different decision tree layouts to decide which visualisation technique would best display decision trees. The four layouts were:

- Organisation Charts,
- Tree Rings,
- Icicle plots, and
- Treemaps.

Organisation charts are the most common visualisation technique used to display decision trees. Tree rings and treemaps display the topology of the tree and node size. Icicle plots are similar to tree rings except the icicle plot contains empty spaces. The results obtained by [11] indicated that organisation charts and icicle plots were the preferred visualisation techniques with the icicle plot being better for viewing large trees.

Due to the hierarchical nature of the output produced by the decision tree algorithm, the organisation chart was selected as the preferred visualisation technique for decision trees. The icicle map can be more compact, but based on the grouping of the variables of the application services data; the size of the organisation chart is limited to a maximum number of levels. The network attribute with the largest amount of variables is *Delay*, which contains eight variables. As a result, the maximum height of the organisation chart is eight.

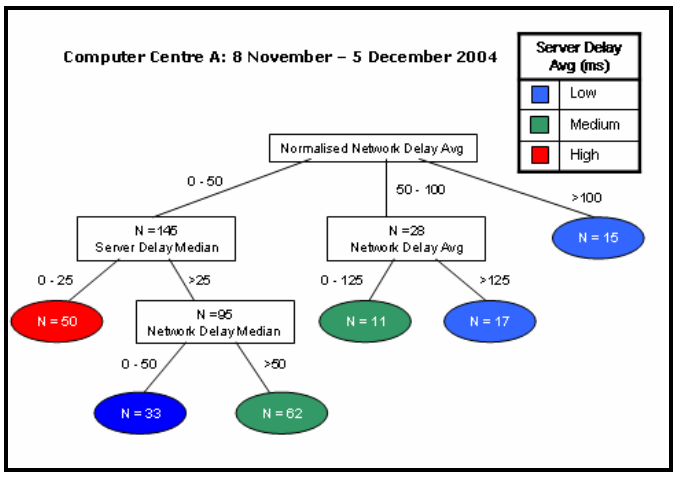

Figure 1: Organisation Chart for Visualising Decision Tree

Figure 1 shows an example of a decision tree using an organisation chart. The data used was from the *Delay* network attribute. The visualisation shows a legend identifying which PacketShaper variable was being predicted, with colour indicating different levels of server delay, ranging from *low* to *high*. The user only needs to follow the path from root to leaf to see which variables affect server delay. Each branch identifies the testing condition with the resultant frequency count of log file entries that satisfy the condition.

B. Visualising Clustering

There are two main visualisation techniques for visualising clustering: scatterplots and bubble charts. Scatterplots are created by plotting values of the log file entries onto a graph. One data dimension is represented on each axis, with each log file entry representing a point in the scatterplot. Scatterplots have some disadvantages. They start to lose their effectiveness when the number of data points becomes too large. To counter this, the use of voxelisation is proposed [12]. Voxelisation makes use of binning and volume rendering, where each voxel represents one bin.

Bubble charts are also an effective way of visualising the results of correlation. A bubble chart is a form of scatterplot with variable size symbols. Bubble charts can be used to visualise larger data sets better. Each bubble represents a data point and the number of log file entries that fall in that group. Bubble charts have two features that differentiate it from normal scatterplots: they allow users to view any outliers easier and they show gaps in the occurrence of the bubble [13]. Bubble charts make use of three values: the median of the cluster and the count of the log file entries in the cluster.

In the previous section, the results of the k-means algorithm were identified as the cluster number, the median for a cluster and the frequency count of the number of log file entries that fall within the cluster boundary. Based on the description of the scatterplot and bubble chart and the output of the k-means algorithm, the bubble chart was selected as the appropriate visualisation technique for clustering.

Figure 2 shows the results of the k-means algorithms being performed on *network delay* and *server delay*. The use of colour helps the user distinguish between the different clusters and the size of the bubble indicates the number of log file entries that fall within each cluster. This technique also makes it easy to identify outliers.

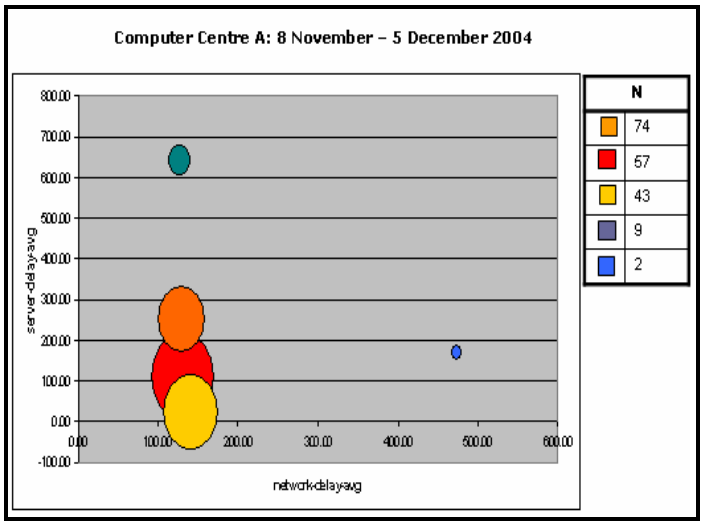

Figure 2 - Bubble Chart for Visualising Clustering

C. Visualising Correlation

The most popular approach to viewing correlation is the use of scatterplots as it visualises the relationship between two or more variables [17]. The log file entries are plotted on the graph to indicate all the transactions that took place. A line is drawn through these points, with the gradient of the line determined by the correlation coefficient. This line indicates if there is positive or negative correlation between the variables being compared. Bubble charts can also be used for correlation [13], but can result in desired information being hidden from the user.

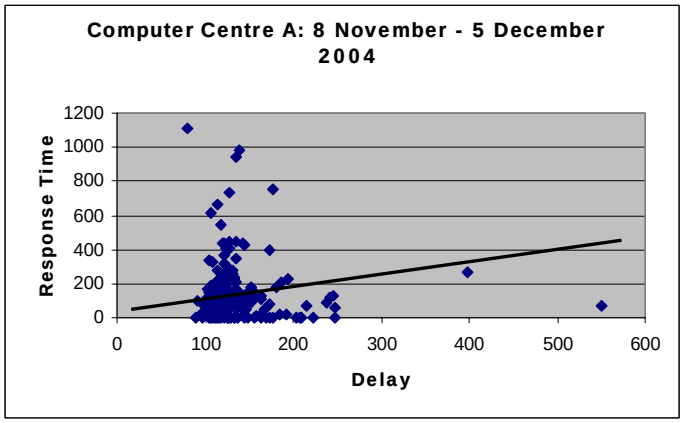

Figure 3 - Scatterplot for Visualising Correlation

The choice of visualisation technique depends on the output of the correlation algorithm. The output of correlation is a subset of the data warehouse and the correlation coefficient. The dataset is used to plot the log file entries in the scatterplot. The

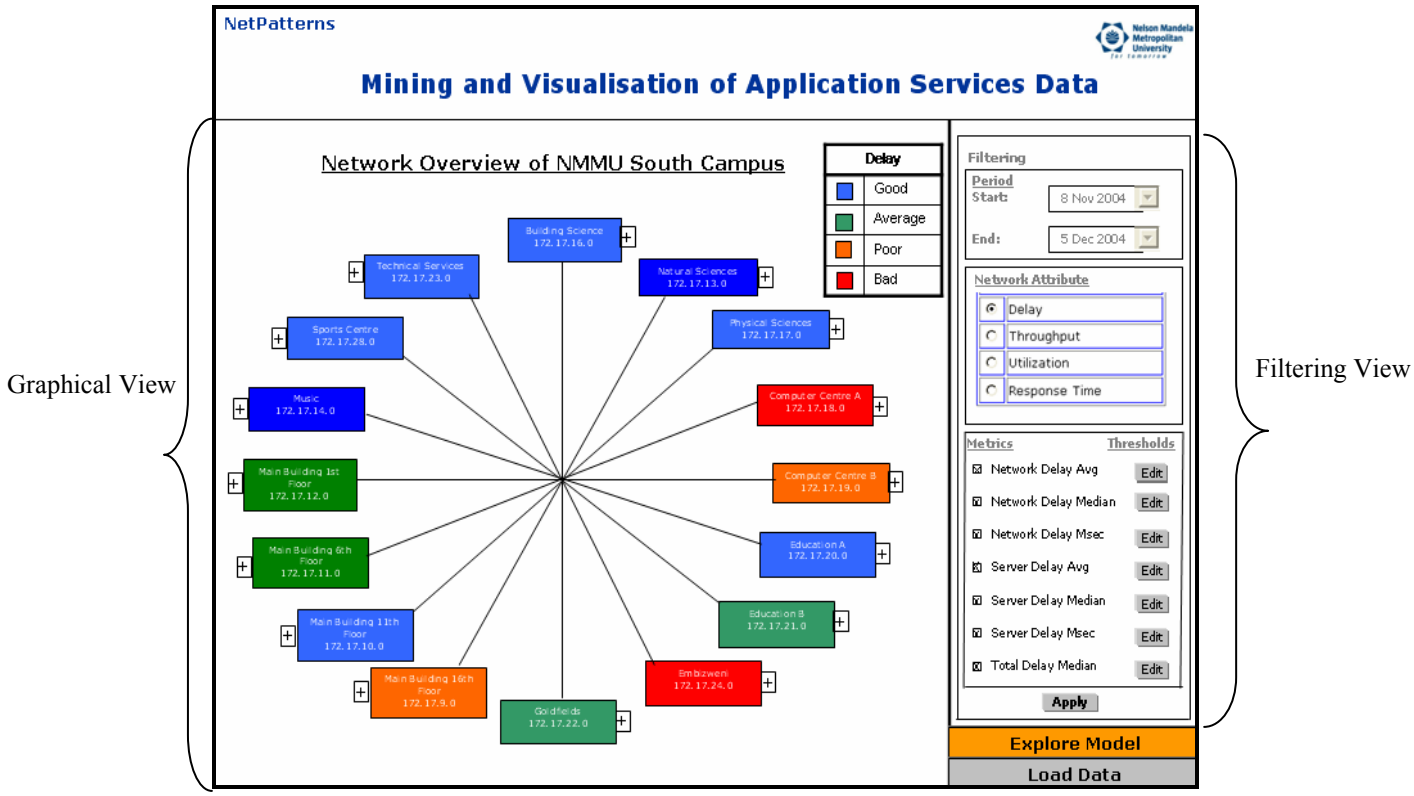

Figure 4 – UI Design showing Network Overview

correlation coefficient is used to calculate the gradient of the line which indicates if the correlation between the network attributes is positive or negative. Figure 3 indicates the relationship between *Delay* and *Response Time* and shows a positive correlation between these variables.

VII. DESIGN

An iterative approach was used in the user interface (UI) design of the interactive visualisation system, NetPatterns. The design process focused on producing low-fidelity prototype designs of the user interface for evaluation by the users. The evaluation technique made use of conceptual model extraction. Conceptual model extraction involves asking the user to explain elements on the screen and provide feedback regarding these screen elements. This iterative process refined the design of the UI based on the feedback received from the users.

The UI was designed to enable the user to view the results of the data mining algorithms and to interact with them. Figure 4 shows the network overview after the default network attribute *Delay* is selected for the initial visualisation. The interface has been divided into two main views: a graphical view (left pane) and a filtering view (right pane) as shown in Figure 4. The placement of these views remains fixed during data analysis and exploration. The graphical view occupies most of the screen space as this is the focal point for the user. The graphical view is used to assist the user in understanding the results while the filtering view allows the user to modify the data that is being viewed.

Figure 4 shows the results once an initial algorithm has been performed on the *Delay* component. A star visualisation technique is used to provide an overview of the NMMU network. Each node represents a VLAN on the network and colour denotes the various levels of delay that exists within that VLAN. In Figure 4, *Delay* is represented as ranging from low to high. To drill-down into a VLAN, the user can click on the node for that VLAN. Once the user selects a VLAN, the user is presented with the results of one of the three data mining algorithms (Figure 5).

Filtering can be performed on the data to allow users to select different network attributes and corresponding variables. The user can choose the time period to view as well as which network attribute to be used. Depending on which data mining algorithm the user selects, the graphical and filtering views are updated, as can be seen in Figures 4 and 5. For decision trees, all the variables within each network attribute are available for the user to select. For clustering, the user can select to cluster averages, medians or totals. For correlation, the user is allowed to select two or more network attributes to correlate. No textual display is provided as the visualisations are intended to assist the user in understanding the results of the data mining algorithms. Once the user has made the relevant selections for the specific algorithm, the user will be able to save and print the visualisations generated in the graphical view.

VIII. CONCLUSION & FUTURE WORK

Techniques that can identify patterns of network usage can be useful in optimising network performance. This paper has identified four network attributes, namely delay, throughput, response time and utilization and the associated PacketShaper variables (Table 1). Three data mining algorithms were selected based on the characteristics of the data, namely decision trees, k-means and correlation. The visualisation techniques selected were organisation charts, bubble charts and scatterplots.

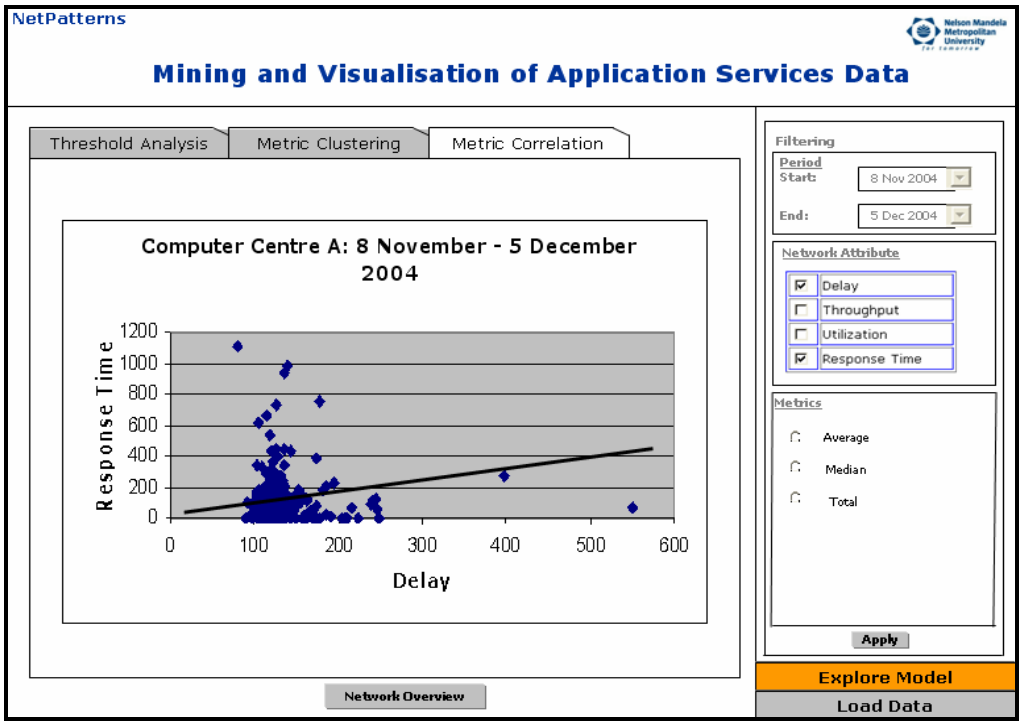

Figure 5 – UI Design of NetPatterns

An interactive prototype was designed to support dynamic analysis and exploration of the data mining results. The effectiveness of this interface still needs to be evaluated.

ACKNOWLEDGMENT

We would like to thank the Telkom Centre of Excellence programme and the Department of Computer Science and Information Systems at the Nelson Mandela Metropolitan University for making this research possible.

REFERENCES

[1] HAN, J. and KAMBER, M (2001): *Data Mining: Concept and Techniques*. Morgan Kaufmann.

[2] LOWECAMP, B., TIERNEY, B., COTTRELL, L., HUGHES-JONES, R., KIELMAN, T. and SWANY, M. (2004): A Hierarchy of Network Performance Characteristics for Grid Applications and Services. http://www-didc.lbl.gov/NMWG/docs/draft-ggf-nmwg-hierarchy-02.pdf

[3] LEESE, M. (2003): GridMon – Grid Network Performance Monitoring for UK e-science. http://www.gridmon.dl.ac.uk

[4] BARNFORD, P. (2003): Network Performance Measurement and Analysis. www.cs.wisc.edu/~pb/640/perform.ppt

[5] THEARLING, K. (2000): An Introduction to Data Mining. www.thearling.com/text/dmwhite/dmwhite.htm

[6] CHUNG, H.M, and GRAY, P.: Special Section: Data Mining. *Journal of Management Information Systems* **16**(1):pp11-16. http://jmis.bentley.edu/articles/v16_n1_p11/

[7] ORACLE (2002): Oracle 9i Data Mining – Concepts

[8] BROOKS, P. (1997): Data Mining Today, DBMS Magazine. www.dbmsmag.com/9702d16.html

[9] THEARLING, K. (2002): An Overview of Data Mining Techniques. www.thearling.com/text/dmtechniques/dmtechniques.htm

[10] CARD, S.K., MACKINLAY, J.D. and SHNEIDERMAN, B. (1999): *Readings in Information Visualisation: Using Visions to Think*. Morgan Kaufmann.

[11] BARLOW, T. and NEVILLE, P. (2001): A comparison of 2-D Visualisation Hierarchies, *Proc IEEE Symposium on Information Visualisation*, San Diego, California, Oct 22-23, 2001

[12] SAHLING, G.N. (2002): *Interactive 3D Scatterplots – From High Dimensional Data to Insight*, Masters Dissertation, Institute of Computer Graphics and Algorithms.

[13] BINKLEY, D. and HARMAN, M. (2004): Analysis and Visualization of Predicate Dependence on Formal Paremeters and Global Variables. IEEE Transactions of Software Engineering **30**(11), November 2004.

[14] PAUL, S., MACLENNAN, J., TANG, Z. and OVERSON, S. (2004): Microsoft SQL Server 2005: Data Mining Tutorial.

[15] ANKERST, M. (2003): Visual Data Mining with Pixel-Orientated Visualization Techniques. http://www.dbs.informatik.uni-muenchen.de/~ankerst/poDM.kdd2001.pdf

[16] SHNEIDERMAN, B. (1996): The Eyes Have It: A Task by Data Type Taxonomy for Information Visualisation. *Proc IEEE Symposium on Visual Languages*, pp 336-343.

[17] STATSOFT, INC. (2004). *Electronic Statistics Textbook*. http://www.statsoft.com/textbook/stathome.html

[18] SPSS, INC. (2004). Clementine. http://www.spss.com/clementine/

Ronald Knoetze received his BSc Hons degree in 2003 from the University of Port Elizabeth. He is presently doing his MSc in Computer Science at the NMMU. His current field of research involves using data mining algorithms to obtain patterns from application services data.