Journal of Universal Computer Science, vol. 13, no. 2 (2007), 287-316
submitted: 31/7/06, accepted: 15/1/07, appeared: 28/2/07 © J.UCS

Internet Path Behavior Prediction via Data Mining: Conceptual Framework and Case Study

Leszek Borzemski
(Wroclaw University of Technology Wroclaw, Poland
leszek.borzemski@pwr.wroc.pl)

Abstract: In this paper we propose an application of data mining methods in the prediction of the availability and performance of Internet paths. We deploy a general decision-making method for advising the users in further usage of Internet path at particular time and date. The method is based on the clustering and tree classification data mining techniques. The usefulness of our method for prediction the Internet path behavior has been confirmed in real-life experiment. The active Internet measurements were performed to gather the end-to-end latency and packet routing information. The knowledge gathered has been analyzed using a professional data mining package via neural clustering and decision tree algorithms. The results show that the data mining can be efficiently used for the purpose of the forecasting the network behavior. We propose to build a network performance monitoring and prediction service based on proposed data mining procedure. We address our approach especially to the non-networkers of such networking frameworks as Grid and overlay networks who want to schedule their network activity but who want to be left free from networking issues to concentrate on their work.

Keywords: Grids, Network Behavior Prediction, Knowledge Management, Data Mining, Internet Performance, End-to-end-performance
Categories: C.2.3, C.4, H.1.2, H.2.8

1 Introduction

Today's Internet users perceive good network operation by low latency, high throughput and high availability. Network performance is usually evaluated by the available bandwidth, end-to-end latency, and throughput of data transfers. But it has never been easy to determine whether slow responses are due to either network or end system on both sides. Moreover, we are not able exactly to diagnose and isolate key sources of Internet communication problems because they may be localized in various network appliances and at different communication layers. Even though the best effort networking is enough for Internet based applications that are usually employing stateless communication, many of new application with the end-to-end nature now require predictable network performance. With the advent of Grids [Avery and Foster 2001], overlay [Andersen et al. 2001] and peer-to-peer [Gnutella 2005] networks, the network behavior prediction issue becomes an essential task. The predictions can be used to schedule application/user activity and network communication, to select servers or network routes, as well as to organize parallel downloads.

Various network performance evaluation principles and practices are used in contemporary Internet. Several efforts in network performance measurement and monitoring concepts, tools and projects are reported at CAIDA's [CAIDA 2006] and

SLAC's [SLAC 2006] websites. Most tools, excluding some popular personal ones, can be used only by the network specialists and administrators because they need professional knowledge how to use specified network performance tools and how to read the results.

Here we concentrate on the needs of non-networkers who work within such class of network architectures like Grids or overly networks. These architectures allow end-to-end communication across the wide-area Internet on the top of the existing Internet organization (nodes, paths, routing and protocols) optimizing application-specific metrics that are connected with the properties of Grid-based distributed application such as inter-communication time/cost, message exchange, data/job migration or file operation/downloading time/cost. There are also such application services as DNS (*Domain Name Service*) and Web where it is necessary not only to measure network performance but also the response time of such servers [Borzemski 2006a]. These application specific metrics provide a deeper understanding of how the application is performing in a distributed system.

Grid computing is applying usually to a scientific or technical problem, in the academic and commercial world, that requires a great number of computers or access to large amounts of data in a network. Then the users can ran up against difficulties trying to get the computations completed in a proper time and at proper cost. They need methods and tools for end-to-end performance prediction to select a partner node (service). According to the Grid *utility computing* model the responsibility for network problems lies with system support stuff whereas the end users should be left free to concentrate on their applications. Therefore, network performance monitoring problem for the Grid must be treated in a special way, and both diagnostic tools and diagnostic methods must be developed taking into account the needs and capabilities of the non-networkers [Leese et al. 2005].

An overlay network is a network created dynamically between a group of cooperating Internet hosts. Basically overlay networks provide resource location services for different application services. In principle, if they want to provide the service they must monitor and evaluate the network. For example, resilient overlay networks determine whether the Internet path between two hosts is working on an end-to-end basis to find failure-disjoint paths between nodes, and forward traffic along a working path [Andersen 2005].

In this work, we propose to develop an intermediary Internet service for network performance measurement and prediction of network behavior via data mining (DM). Such service can describe performance conditions of network node-to-node paths and identify possible network behavior. Our data mining based performance prediction technique enables the end user to forecast the log-term network performance behavior and to advise the user in the decision when and which communication path should be used at a particular time and date [Borzemski 2004]. Such approach can be employed, e.g. for Grid links, for example to schedule of future usage of Internet connections or to select replicated resources. The experiment comprises *traceroute* transfers to measure the *Round-Trip Time* (RTT) to characterize the network behavior. We mine data sets using the IBM Intelligent Miner for Data [Baragoin et al. 2002, Cabena et al. 1999, IM4D 2002].

The rest of the paper is organized as follows. Section 2 gives the overview of the problems and solutions in the fields of network monitoring and network behavior

prediction. Section 3 discusses network monitoring in the context of Grids. Section 4 shows how data mining meets Internet diagnostics. Section 5 presents the outline of our proposal, the measurements and data mining methodology. The results of mining using sample data are showed in Section 6. Finally, the concluding remarks appear in Section 7.

2 Network monitoring and behavior prediction

Network behavior analysis can be performed from different points of view, including two key issues, i.e. performance and security. Here we deal with network performance issues. Internet performance is extremely difficult to study in an integrated way. It has never been easy to determine whether slow responses are due to either network or end-system problems, or both. Moreover, because most of the performance problems are transient and very complex in the relationships between different factors that may influence each other, therefore we cannot exactly diagnose and isolate their key sources. All these factors may affect ultimate Internet end-to-end performance. For example, in case of Web access, almost 60% latency, as perceived by users at their microscopic level, refers to the end-to-end path between user and Internet host [Cardellini et al. 2002].

2.1 Active vs. passive monitoring

We can monitor the Internet paths via passive or active experiments. Passive monitoring gathers data based on the analysis of sniffed network traffic. Active measurements produce data based on the response to the probe traffic transmitted between two (at least) end-points in the network. Active measurements allow studying host and service availability, routes of transmitted packets, packet delay, and packet loss and packet inter-arrival jitter [Prasad et al. 2003]. The end-to-end nature of active measurement system makes it suitable in the Grids, overlay networks and peer-to-peer networks.

2.2 Sound measurements

Measurements can merely report the network state at the time of the measurement [Paxson 2004]. They are effectively used by various protocols to test and monitor current performance and to take necessary action when changes are detected. If a network exhibits constant (stable) behavior over observed network life-time span, then the measurements can be used in forecasting the future of network behavior. The concept of constant network parameters is especially more useful for coarser time scale than for fine time scales [Zhang et al. 2001]. It was shown in [Wolski 1998] that in general the round-trip packet delay as measured by RTT appears to be well described as steady on time scale of 10-30 minutes. In our paper we also use RTT as performance characteristics of Internet path and assume the constancy property of measured network parameters in a similar time scale.

2.3 Performance forecasting

The prediction of Internet performance has been always a challenging and topical issue [Abusina et al. 2005, Arlitt et al. 2005, He et al. 2005]. The users may need short-term and long-term network performance forecasts. Short-term forecasting requires instantaneous measuring of network performance. In long-term forecasting we can apply less resource exhaustive monitoring and use historical information more extensively. Existing methods are able to predict the network behavior only at the current time (i.e. at the moment when the performance prediction is done), whereas we try to predict the network behavior at a future time.

The comprehensive lists of Internet measurement projects are presented at SLAC [SLAC 2006] and CAIDA [CAIDA 2006] websites. Mostly they are aimed to deal with the performance problem related to whole or a significant part of Internet where large amounts of measured data regarding, for instance, round trip delay among several node pairs over a few hours, days or months, and using specific measurements and data analysis infrastructures are obtained. These projects can build so called Internet weather at the IP level. Most of them only measure the traffic, present the results as some aggregated and temporary observations but do not provide any network performance forecasting.

Individual users cannot prepare long-term performance forecasting using the knowledge which is discovered in measurement projects that have been launched on the Internet [e.g., Brownlee and Loosley 2001, Luckie et al. 2001, Mogul 2002, Saroiu et al. 2002]. These projects present the network behavior reports that are mainly focused on the whole Internet or a significant part of it, as they use the measurements mainly performed in the core of Internet.

2.4 Web measurements

When browsing the Web, users are concerned with the performance of entire pages. Understanding and identifying the sources of the performance problems is a very important issue, especially for e-business. We can look at this problem from the perspective of the owner/administrator of a website. Then regular load tests originated from different Internet addresses are required. They must have a high degree of realism. They should model as closely as possible the user behavior that is expected in the real-life situations. Probably the service at Keynote's website [Brownlee and Loosley 2001, Keynote 2006] is the most advanced benchmarking service that measures website's performance and availability from a word-wide network of measurement agents. However Keynote does not provide performance predictions.

There is also the need to have a service for testing, measuring and diagnosing websites from the perspective of the end-users. Recent paper [Borzemski 2006a] presents user needs and surveys available services. We have also developed the WING system for Web probing, visualization and performance analysis [Borzemski and Nowak 2004a]. In [Borzemski and Nowak 2004b] we present how the system was used in an extensive performance evaluation study of Web access seen from the perspective of Wroclaw University of Technology (WUT) end users.

Current version of WING can be used in instant and periodical testing, measuring and diagnosing Web sites only from the perspective of the observer placed in WUT campus network. The results of instant performance evaluation can be accessed via

Web based visualization, whereas periodical measurements are stored in a local database collaborating in an off-line mode with any data mining package.

Now we are developing a new version of WING, MWING, which has a multiagent architecture spreading out the observers (agents) freely in the Internet, process and store measurements at WUT location and getting the final results of performance analysis at any other site [Borzemski et al. 2007]. In addition to Web access observation, the system is able to invoke any agents to measure needed network performance metrics. It also will provide data for network performance predictions.

3 Grids and network monitoring

Grid technologies provide seamless and scalable access to wide-area Internet based distributed resources. Grids are an approach for building dynamically constructed problem solving environments using geographically and organizationally dispersed high performance computing and data handling resources [Foster and Kesselman 2003]. Today's Grids are used mainly in the academic world for large-scale problem solving (e.g., high energy physics data analysis, aerospace systems design, cosmology studies) [Foster and Kesselman 2003, Johnston 2003, Luckie et al. 2001]. The interest in Grids is growing in the commercial sector, as well. There are also industry efforts to combine Grid services and Web services (IBM, HP, Microsoft, and Sun) [IBM 2005].

Besides the Grids there is the world of overlay networks [Aberer et al. 2005, Andersen 2005] including peer-to-peer (P2P) applications (such as Gnutella [Gnutella 2005]). P2P networks are also built among scientific communities (e.g., GriPhyN project [Avery and Foster 2001]) that need predictable network behavior, as well [Despotovic and Aberer 2004].

3.1 Utility computing model

Grids can be compute-intensive and data-intensive, i.e. Grids developed for speeding up the computations and efficient usage of computing power accessible in local systems in the network, and Grids for very large data handling problem solving. Grids use Internet, preferably high-speed Internet, to communicate between distributed computational resources, transfer data and jobs, and to synchronize computations. Grids share widely distributed computing resources providing a uniform look and service of distributed systems.

The *utility computing* model is a service provisioning model in which a service provider makes computing resources available to the customer as needed, and charges them for specific usage rather than a flat rate [Leese 2005]. Grids are built around this concept. Grid systems seek to maximize the efficient use of resources and/or minimize associated costs. Grid end-users need effective services and tools that allow easy sharing of costly and rare computing resources without knowing the specifics of each local environment. They want to match their requirements using the publicly available wide area networking developed around the Internet.

Grids are based on the basic network services – among them there are services for resource discovery, resource co-scheduling, and resource brokering as well as

communication services. Many Grids currently use Globus [Foster and Kesselman 1997] to provide the basic services that characterize and locate resources, initiate, transfer and monitor jobs, provide secure authentication of users, provide uniform access to data, etc. In the Globus Toolkit there is a communications component for providing communication mechanisms for a wide range of communication methods and networks, taking into account network quality of service parameters such as jitter, reliability, latency, and bandwidth.

3.2 Performance monitoring and prediction

Performance network monitoring in Grids is a fundamental problem for them. There are several approaches and systems available for taking measurements and predicting of future performance [e.g. Foster and Kesselman 1997, Foster and Kesselman 2003, Wolski 1998, Yousaf and Welzl 2005]. Most of them are based on the last observations and estimate the network characteristics in a short-term.

One of the most challenging Grid problems is that of scheduling scarce resources such as supercomputers, clusters and large instruments. Such resources have to be co-scheduled for several users and for limited periods of time, or at a particular time and date. CPU advance reservation scheduling and network bandwidth advance reservation are critical components to the co-scheduling services. But if we cannot make network bandwidth reservation we should know the behavior of the network connections to decide when and which remote resources are to be used. Therefore we need both short-term and long-term prediction of network performance.

The successful deployment of Grid-based applications in a great extent depends on the availability of good network behavior over the requested time range, even on an agreement-based service [Borzemski 2005, Leese et al. 2005, Zhang et al. 2004]. However existing methods are able to predict the network behavior only at the current time (i.e. at the moment when the performance prediction is done), whereas we need to predict the network behavior at a future time or at a future time period.

Another motivation standing behind the usage of the predictive performance models for Grids derives from the resource replication habits (e.g. databases in different locations). Huge data transfers are often constrained by network bandwidth. Therefore we need the prediction of network latency and network throughput when selecting the best replica.

The Network Weather Service (NWS) (its functionality is being analogous to weather forecasting) [Wolski 1998] can be used for forecasting the performance of various resource components, including the network itself by sending out and monitoring lightweight probes through the network to the sink destinations at regular intervals. It is intended to be a lightweight, non-invasive monitoring system. This service operates over a distributed set of performance sensors network monitors from which it gathers readings of the instantaneous network conditions. It can also monitor and forecast the performance of computational resources. It uses numerical models to generate short-term forecasts of what the conditions will be for a given time frame starting at the moment of the prediction.

The NWS uses simple prediction methods (mean-based, median-based, autoregressive) that cannot satisfactorily capture temporal network dependencies, thus new methods for network performance prediction need to be studied, especially approaches based on some learning techniques. In [Eswaradass et al. 2006] an

Artificial Neural Network (ANN) based predictive mechanism for available bandwidth is proposed. Experimental results based on passive measurements made on 1-day and 6 ½ week traffic traces and short-term active measurements showed that ANN approach provided an improved prediction over that of NWS. Both mechanisms were more accurate for short-term predictions (60 - 300 seconds). The ANN mechanism can be used with the NWS as alternative prediction functionality.

A limitation of NWS lies in that it runs only in UNIX operating system environments and requires much of installation and administration work in a distributed framework. NWS basic prediction techniques are not representative of the transfer bandwidth obtainable for large files (10 MB to 1 GB) and do not support long-term forecasts. New NWS developments partially address these problems, e.g. [Swany and Wolski 2002] shows the correlation based forecasting method that combines data from two measurements streams which may be correlated in same way. This technique was developed for forecasting long HTTP transfers using a combination of short TCP/IP bandwidth probes and historical observations of HTTP transfers.

Another interesting alternative prediction method is proposed in [Schultz et al. 2001]. The major feature of this approach is the consideration of the human rhythm of life. Prediction values are calculated using selected number of arguments, depending on daytime and weekday. This algorithm does not consider public holidays and possible network load disturbances that may occur following the special events like sport games or socio-political events. We also take advantage of the human rhythm of life but our approach supports all days in a week and is more general because it takes into account end results in network performance.

A different approach, which is similar to our proposal, is to develop a network service providing communication performance prediction for the non-networkers. The UK e-Science Programme community [Williams 2005] has been highly active in network performance measurement with projects such as GridMon developed for the end-to-end measurements for Grid users. An adaptive Grid has been proposed with the concept of data transfer strategy that can be dynamically adapted to network behavior, for example by scheduling bulk data transfers for quiet times on the network. Such monitoring tool as the NSW is expected to be used in predictions.

4 Data mining meets Internet diagnostics

Nowadays the classical statistical data analysis is usually performed to derive network performance characteristics [Andersen et al. 2001, Avery and Foster 2001, Ballintijn et al. 2000, Barford et al. 2001, Brownlee and Loosley 2001, Foster and Kesselman 1997, Luckie et al. 2001, Gummadi et al. 2002, Wolski 1998, Zhang et al. 2001]. Network monitoring systems produce huge datasets that we are not able to fully explore using conventional data analysis methods and tools. From the decision point of view, we can expect that the useful information might be hidden in these datasets. Such hidden information usually is not easy to discern using conventional data queries and statistical calculations. In such situations the data mining can be helpful.

4.1 Predictive data mining

Data mining algorithms analyze of the data in large databases to identify trends, similarities, and patterns to support decision making. Data mining is a promising area of current research, which can provide important advantages to the users. It is an interdisciplinary area focusing upon methodologies for extracting useful knowledge from data which can yield substantial knowledge from data gathered [Grossman et al. 2001, Han and Kamber 2000].

The process of data mining consists of three main stages: (1) the initial preprocessing involving cleaning raw data and raw data transformations to prepare data for further processing, (2) model deployment, and (3) model application. The data mining can be descriptive or predictive. The predictive data mining is the most common type of data mining and one that has many applications.

4.2 Performance data mining

During monitoring of Internet we collect a large amount of measurement data concerning different parameters characterizing Internet performance [Barford et al. 2001, Brownlee and Loosley 2001]. We can use different metrics such as latency of transmissions, available bandwidth, router hops, and packet loss or packet routes for particular end-to-end paths. All of them can be used in construction of a performance prediction model of network behavior. Better understanding the performance and availability of the underlying network may help in better planning and utilization of the network, and nodal resources.

Existing network performance prediction methods use complicated stochastic approaches that extrapolate historical data to provide a rough estimate of future network performance. The ongoing rapid growth of measured data on the Internet has created an immense need for data mining and knowledge discovery methodologies applied to the Internet performance data. In this paper, we address network performance prediction, in the context of transmission latency, as a data mining problem. Our proposal has also practical implications different than other solutions. Namely, instead of predicting one single value of predicted performance measure we predict a range within the network performance may lie.

The knowledge discovery in data bases filled with performance data related to technical systems, such as Internet, is of growing interest [Faloutsos et al. 2002, Wang et al. 2002]. However the research is still at early development stage and there is a room for new data mining research.

Predictive data mining may characterize network performance at two levels, namely at the level of the IP-based Internet communication subsystem what we are presenting here, and at the application level when we deal with data transfers made by a file transfer protocol FTP and similar solutions (e.g. GridFTP - a protocol extensions to FTP for the Grids). The specific application level context is referred to the Web data transfers where the HTTP (HyperText Transfer Protocol) protocol and Web client to Web server interaction add its specificity [Borzemski 2006b].

This contribution shows how the data mining methods can be used in the analysis of long-term network behavior. Our overall strategy involves discovering knowledge that may characterize performance behavior of Internet path, and then making use of

this knowledge to advise the user in future path utilization. Other problems, including end-to-end routing behavior and topology discovery are also addressed in our project.

4.3 Seasonal behavior of Internet

Internet seasonal behavior is observed. It is evidenced by Web and network event logs. [Hellerstein et al. 2001] show that the number of HTTP operations per second is non-stationary and its five-minute mean changes with time of day and day of week issues. It was also shown that time of day explains 53% of the variability in the raw data of HTTP transactions collected over eight months from a production Web server. For our exploratory analysis we assume that weekly and daily period phenomenon exists.

4.4 Web mining

As for today Internet data mining is mainly devoted to *Web mining* [Chakrabarti 2003, Fürnkranz 2005, Srikant and Yang 2001] i.e. the application of data mining methods and techniques to discover useful knowledge from the World Wide Web data. Web Mining deals with the HTTP application layer data. Web mining focuses now on four main research directions related to the categories of Web data they are dealing with, namely *Web content mining*, *Web usage mining*, *Web structure mining* and *Web user profile mining*.

In a recent paper [Borzemski 2006b] we show how data mining may offer a promising strategy for discovering and building knowledge usable in prediction of Web performance. We introduce a new Web mining dimension – a *Web performance mining* that discovers the knowledge about Web performance issues using data mining.

5 Network monitoring and prediction framework

We propose to deploy a new Internet service called Network Monitoring Broker (NMB) which can measure network performance, describe network characteristic and publish the forecasts of network behavior, especially for automatic network resource selection. We focus on end-to-end performance seen by end users of some virtual organization (like Grids or overlay networks) deployed over the set of network nodes linked by a mesh of Internet communication paths [Borzemski 2005].

5.1 An intermediary NMB service

A general idea of our approach is shown in Figure 1. The ultimate goal of our service is to build an advanced Internet access providing a means whereby users (whether human or machine) improve likelihood that they use network at good performance. NMB infrastructure is established at each of the nodes of the virtual organization and can be organized in the likeness of the intermediary servers that mediate the interaction between clients and servers of the World Wide Web [Borzemski and Nowak 2005].

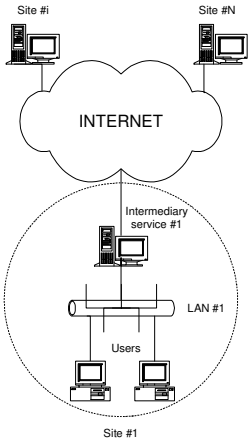

Figure 1: Network performance monitoring and prediction architecture

The general purpose of intermediaries is to extend the functionality and performance without violating the principles employed in the design of the Internet. The intermediaries are developed based on their functionality and focus. A typical operation involves the modification of origin server responses and the creation of final content in response to client requests. Intermediaries can shape the server response in different ways, e.g. by adding, omitting or changing the response prepared by the server in order to prepare the final response with the value added information. Examples include message relays, caches, proxies, content mirrors, load balancers, protocol gateways, and virus scanners.

5.2 Our data mining approach

There are three basic data mining functions: classification, clustering, and associations and sequencing. Classification assigns data to predefined categories (classes) based on predefined rules. Clustering is similar to classification in that different concept categories called clusters are identified through analysis of characteristics of the data using some proximity measures, but there is no predefined clusters. The clusters are generated through patterns identified in the data. Associations and sequencing generate descriptive models that identify rules (such as *if-then*) to allow for prediction of future trends.

Data mining models can be predictive and are often used for forecasting of future behavior of the system under consideration. Here we propose a two-step mining approach to create a prediction model using clustering and classification mining functions. The clustering function is followed by the classification function in such a way that the results of clustering are the inputs to classification. Using clustering function we discover the rules of grouping performance data (records) with similar properties. Next, the decision tree is build based on clusters that are identified. The resulting tree can be used as the decision-making model in advising the user how to use Internet at a particular time and day. Our network performance forecasting method is explained in the real-life study presented in Section 6.

5.3 Measurement tools

The monitoring framework NMB is concerned with the understanding and characterization of Internet performance as seen from the end user side. For this purpose, we can use various performance metrics like Round-Trip Time (RTT), packet loss, available network bandwidth, TCP and UDP throughput, as well as the application specific metrics. Existing active measurement tools support different performance metrics and can provide diverse active measurement schemes [CAIDA 2006, SLAC 2006].

The ultimate aim of the experiment described here is to study two-way network connectivity between WUT campus network (represented by a source host) and the Internet (represented by a set of destination hosts). We consider two quantitative characteristics of a source-destination pair: the IP path hop count and the RTT. RTT is a common measure of the current delay on a network. We developed the TRACE tool to continuously measure the forward IP path and RTT from a source to a destination, and in the backward direction. TRACE is based on the *traceroute* and *reverse traceroute* network monitoring tools.

Traceroute is a tool commonly available on Internet hosts. It causes low network traffic overhead while testing the route over the network between source and target hosts. It lists all the intermediate network nodes (routers/hosts) a testing packet (UDP packet) is passing through to reach the destination host, as well as the latency between the source host and intermediate nodes. This latency is the Round-Trip Time that tells us how long a packet goes from a source host to a destination system and back again.

Usually RTT is measured by the ping tool which monitors network performance by a means of two measures, namely the RTT, and packet loss. The latter shows a percentage of packets discarded due to overloads and congestions. We also considered a ping program but traceroute is more valuable for our experiments because it deliver route information. Moreover, traceroute's RTT can be used for performance monitoring along the whole route from the sender to the target node.

Traceroute gives the responding nodes and the RTTs of the three probes for all nodes on the route from the sender to the destination node. Using traceroute we can detect failing nodes, monitor routes and monitor performance.

But traceroute operates only in one direction, i.e. from the source host to the destination host. However any connection over the Internet actually depends on two routes: the route from source host to the destination host, and the route from destination host back to source host. These routes may be (and often are) different (asymmetric) [Paxson 1997, Tsuru and Oie 2001].

To discover reverse routing, we used *reverse traceroute* that is simply the traceroute but diagnosing the network in an opposite direction, i.e. from the target host to the source host. This service is not commonly installed on Internet hosts but it can be easily installed. In our experiment we used the reverse traceroute hosts (servers) published at SLAC website [SLAC 2006].

5.4 Testbed and experiment setup

The following destination hosts were used in the measurements: icfamon.dl.ac.uk, www.sdsc.edu, umaxp1.physics.isa.umich.edu, v2.ph.gla.ac.uk, cgi.cs.wisc.edu, netdb3.es.net, alf.nbi.dk, katherine.physics.umd.edu. All destination servers were

probed every half an hour over 46 weeks in 2003/2004. The source host named szafir.ists.pwr.wroc.pl with IP address 156.17.130.49 run in WUT computer network.

The measurements were done and collected by the IBM RISC/6000 F80 computer running AIX 4.3.3 and DB2 8.1 database. Data mining was done using the IBM Intelligent Miner for Data 8.1 (IM4D) [IM4D 2002] running on that server and on PC computer equipped with P4 2.4 GHz, MS Windows XP Professional PL and DB2 8.1 database. The traces were stored in the relational table (mining database). A row (record) of this table contains either the IP address, the IP path hop count and the RTT to the ultimate destination, or the IP address, the IP path hop count and RTT of intermediate router that responded on the path traversed by the probe packets from the source to the destination host. Each record is stamped by the time of the day, day of the week, day of the month and month of the year of the probe.

5.5 Challenges and opportunities

Our intermediary service can help in adapting data transport strategy to changing network behavior and thereby advance the productivity of the users. Network behavior prediction is based on data mining performed on the measurement datasets. The intermediary service plays two roles. It is a network measurement engine which, from the local perspective, by means of active measurements monitors end-to-end performance to other sites. It analyses the measurements via data mining and publishes current prediction of network behavior for all paths that originate from the local place to all other places in the virtual organization. It must be placed at, or near, a place where there are local resources.

NMB service must continually monitor Internet links and periodically re-evaluate network behavior predictions to obtain accepted prediction accuracy within certain tolerances and to adapt to drifting network changes. Based on Internet periodic behavior we consider time of day and day of week factors. But of course we are conscious that many network events may have sudden and time-limited effect on path performance. Such events can be treated as the outliers in data mining analysis or must be included in the analysis with all consequences.

The problem is that in general question how many historical data are to be considered in the analysis, and how long the model is valid. Based on our previous research we prefer to use of a moving window data mining instead of possible incremental data mining. The re-evaluation period and the optimum size of the moving observation window must be determined. We can take advantage of a seasonal behavior of Internet. All the time we must remember that the service should not impose too heavy additional overhead on the computing resources.

6 Creating a prediction model: a real-life study

Two-phase data mining method to analyze and predict network performance has been used in a real-life example. This case study is an explanation how the method works, what results it achieves, and is a proof of concept of our idea. In this section the prediction model is shown using sample data obtained by the TRACE system for the forward path from the host working at the WUT computer network named szafir.ists.pwr.wroc.pl with IP address 156.17.130.49 and the ultimate destination host

at the Glasgow University in Scotland named v2.ph.gla.ac.uk with IP address 194.36.1.69. From the whole trace we selected here the records concerning the ultimate destination host.

The prediction method uses two data mining techniques, namely clustering and classification. Clustering segments performance data into groups which are classes of network behavior whereas the classification builds a decision-making structure (a decision tree) from examples of past decisions that can be used to make decisions for unseen cases.

In the presentation of the results we begin with the analysis made on a sample dataset collected in the duration of almost 15 weeks. Next, we present some results of the mining analysis performed on the whole dataset obtained in the duration of 46 weeks.

6.1 Data preprocessing

The initial preprocessing involved cleansing raw data and raw data transformations to prepare data for further processing. Only non-error transactions were used in data mining. Data was checked to avoid missing and invalid field values. Missing RTT and HOP values were estimated as averages. After the cleaning phase we obtained the database with 5004 records. Each record has the following fields: the time stamp of the measurement, the hour of day (HOUR € {0, 1, 2, 3,… ,23}), the day of week (DAY € {1, 2, 3, 4, 5, 6, 7}, where 1 denotes Monday and so on), the length of the path between source host (156.17.130.49) and destination host (194.36.1.69), i.e. the number of hops (HOP € {1, 2,…,30}) as well as three samples of the average round-trip time. The value of RTT used in further analysis is the average value calculated from these three samples.

In preprocessing phase we introduced the discretization in case of continuous variable RTT. Discretization facilitates interpretation of the results, for both the clustering and classification functions, and takes care of outliers. The quantiles (buckets) for the discretization of RTT data were automatically generated in an automated fashion by IM4D based on the standard deviation width technique. This method works well and the result is satisfactory. IM4D builds an *equal-width* data-driven even partitioning for a range of continuous variable between two cut-points *Low* and *High* that stand out two end quantiles: (-∞ - *Low*] and [*High* - +∞). Both ends intervals may contain outliers. The values of the RTT at the intermediate cut-points are determined and data is divided into equal-width *N* quantiles. Therefore the whole variable range is always divided into $N + 2$ quantiles. Cut-points are well-established values, such as 25, 50, 75 or 100 because a range [50 - 75] looks more natural than [51.80 – 75.92]. In this paper the quantile breaks are independently generated for every new sample dataset under consideration.

In the sample of 5004 records the range of RTT values was discretized into 18 quantiles using the following breakpoints: 25, 50, 75, 100, 125, 150,…, 375, 400, and 425 ms. RTT values were limited by the value 10,000 ms which was used to mark of situation that the host/router did not respond before the time limit specified by the traceroute tool (default wait time is 3 seconds).

6.2 Clustering mining function

First what we need is to obtain the classes (profiles) of characteristic behavior of the network. Here we perform network behavior profiling through clustering. The purpose of clustering is to partition a database into groups of records that have similar characteristics. A cluster profile represents the typical values of the fields for records in their assigned cluster.

IM4D shows the number of detected clusters and the characteristics that make up each cluster. Additionally, the results present how these characteristics are distributed within the clusters. The presentation of the results is provided via the Visualizer which is a Java-based patented GUI [IBM 2000] displaying the results produced by a mining or statistical function [IM4D 2002]. The results can be saved or exported in different file formats, including PMML [PMML 2006] which is XML-based format allowing to interchange results between different DM systems, in particular the predictive models.

In the textual description of the clustering results the discrete numerical fields DAY, HOUR and HOP are described by their modal values counted among all records belonging to the cluster. IM4D uses the term "predominantly" for showing the highest frequency value.

For continuous numerical fields such as RTT, the fields can be described by the intervals having the highest frequency, or by an interpretation (small, average, large) if the statistical information necessary for this interpretation is available. It is not written in the manual what is a general rule for that. The Visualizer builds this interpretation; let's say for cluster C, using the following scheme:

- The mean value of RTT M for the total population of records in the database is calculated as well as the standard deviation SD.
- The mean value of RTT $M(C)$ for records belonging to the cluster C is calculated.
- By applying these calculated numbers we get three intervals to interpret the RTT behavior for cluster C, namely:
  - *Low*, if $M(C) < LB$,
  - *Medium*, if $LB \le M(C) \le UB$,
  - *High*, if $M(C) > UB$,

where $LB = M - SD/2$ is the lower RTT bound, and $UB = M + SD/2$ is the upper RTT bound. Such representation is justified. Of course, it can be calculated for the whole population, as well. Then $M(C) = M$. Clusters interpreted as having *High* RTT are called in this paper as "*High* clusters". The same name rule is for the rest of RTT categories. In 5004 database the Vizualizer showed *Low* clusters if $M(C) < 50$ ms, *Medium* if 50 ms $\le M(C) \le 150$ ms and *High* if $M(C) > 150$ ms.

6.2.1 Dataset characteristics

The basic statistical description of our dataset with 5004 records is shown in Figure 2. From the point of view of the aim of our analysis the most important is the characteristics of RTT variable. The most frequent RTT quintile is [50 - 75] owning 1,966 records. The maximum RTT value is 2,749.66, whereas the minimum RTT is 49.66. The mean M of RTT data is 128.35, the standard deviation SD is 140.21, and the median is 79.00. Thus, the lower limit is 58.25 and upper limit is 198.46. Hence the total population of 5004 measurements generally features *Medium* RTT.

Field Name	Type	Modal Value	Modal Freq.	Min	Max	Mean	Standard Deviation
HOP	DN	20	1,929	19	25	19.94	N/A
RTT	CO	[50-75]	1,966	49.66	2,749.66	128.35	140.21

Figure 2: Characteristics of HOP and RTT (Field Types: DN=Discrete Numeric, CO=Continuous Numeric)

This description shows that on average, the path under consideration features quite good connectivity conditions (i.e. as characterized by *Medium* RTT). However, the RTT data distribution is far from normality what can be showed in terms of *skewness* and *kurtosis*. The skewness is a measure of the asymmetry of the data distribution. If a distribution has positive skew it is right-skewed. Kurtosis is a quality measure of the "peakedness" (broad or narrow) of a data distribution. For a normal distribution a skew is 0 and kurtosis is 3. A kurtosis coefficient greater than 3 indicates a high peak, thin midrange on either side of the mean and fat tails. Figure 3 shows the histogram of RTT. RTT has an extreme distribution as its skew is 8.17 and kurtosis is 115.90. The distribution of RTT has a long tail on the right, in other words the majority of values are low (i.e. *Medium* in our categorization) with the smaller number of extreme high values. Because of that we may expect often longer than medium delays for some of the time. Our analysis is to discover these durations of high values of round-trip time.

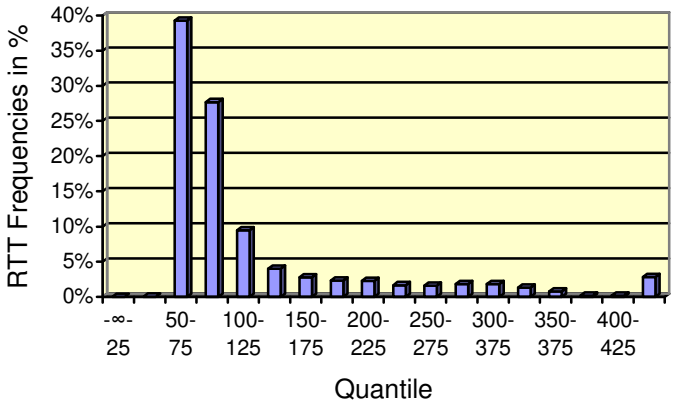

Figure 3: Histogram of RTT from 5004 records

Dataset size is a vital issue in data mining and should not be ignored in the data analysis. The in-depth discussion of the "right" duration of experiments for DM considerations is given later in the Section 6.4. Based on this analysis we can say that we have enough examples of all classes and all values of all fields.

6.2.2 Neural clustering

Clustering addresses segmentation problems and assigns data records described by a set of attributes into a set of groups of records called "clusters". This process is performed automatically by clustering algorithms that identify the distinguishing characteristics of the dataset and then partition the space defined by the dataset attributes along natural cleaving boundaries.

Cluster	Size (%)	Cluster Description
14	7.93%	RTT is high, DAY is predominantly 3, HOP is predominantly 19 and HOUR is predominantly 16.
3	7.79%	DAY is predominantly 1, HOUR is predominantly 2, HOP is predominantly 20 and RTT is medium.
7	7.69%	DAY is predominantly 1, HOUR is predominantly 9, HOP is predominantly 20 and RTT is medium.
12	7.05%	DAY is predominantly 7, HOUR is predominantly 23, RTT is medium and HOP is predominantly 20.
8	6.97%	DAY is predominantly 7, HOUR is predominantly 15, RTT is medium and HOP is predominantly 20.
4	6.91%	DAY is predominantly 7, HOUR is predominantly 11, RTT is medium and HOP is predominantly 20.
0	6.89%	DAY is predominantly 7, HOUR is predominantly 1, RTT is medium and HOP is predominantly 20.
6	6.29%	DAY is predominantly 3, HOUR is predominantly 7, RTT is medium and HOP is predominantly 19.
1	6.20%	DAY is predominantly 5, HOUR is predominantly 5, HOP is predominantly 20 and RTT is medium.
5	6.18%	DAY is predominantly 4, HOUR is predominantly 7, RTT is medium and HOP is predominantly 19.
11	6.14%	DAY is predominantly 2, HOUR is predominantly 23, HOP is predominantly 21 and RTT is medium.
15	5.68%	DAY is predominantly 2, RTT is high, HOUR is predominantly 21 and HOP is predominantly 19.
9	5.26%	DAY is predominantly 5, HOUR is predominantly 12, RTT is medium and HOP is predominantly 20.
13	5.26%	DAY is predominantly 5, HOUR is predominantly 23, RTT is medium and HOP is predominantly 20.
10	4.34%	DAY is predominantly 3, HOP is predominantly 21, HOUR is predominantly 17 and RTT is medium.
2	3.42%	HOUR is predominantly 0, RTT is high, DAY is predominantly 3 and HOP is predominantly 19.

Figure 4: Textual overview of clusters

The clustering analysis can help to find the differences in network behavior. For finding the groups of records describing similar network behavior in the input mining database we used the neural clustering algorithm, which employs a Kohonen Feature Map neural network [IM4D 2002]. In order to find accurate and homogenous groups of objects and simultaneously avoiding the production of too many small clusters, we assumed that the algorithm may produce maximally 16 clusters [IM4D 2002]. We also assumed 20 maximum passes of the algorithm one can made. Specifying multiple passes through the input data improves the quality of the generated clusters but also increases the processing time required to perform clustering.

As the active fields (attributes), i.e. record fields that participate in the creation of clusters, we chose: the day of week (DAY), the hour of measurement (HOUR), the length of the path between source and destination host i.e. the number of hops (HOP) and the average round-trip time (RTT). The average RTT was calculated using three RTT values published by TRACE. We obtained 16 clusters. The size of clusters ranges from 3.42% to 7.93% of total number of records. Figure 4 presents the textual overview for obtained clusters. Each row shows cluster id, cluster size and textual description of the cluster. The textual description is the statement of the conditions met by all fields in that cluster. The most representative fields are analyzed first. Figures 5, 6, 7 and 8 show frequencies in % for RTT, DAY, HOUR and HOP, respectively. Each column in the table describes one cluster by showing how the field values are distributed for the records in this cluster.

Cluster / Quantile	14	3	7	12	8	4	0	6	1	5	11	15	9	13	10	2
-∞ - 25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25 - 50	0	0	0	0	0	0	0	0	0	0.32	0	0	0	0	0	0
50 - 75	0	52.82	48.05	29.18	49.86	88.73	61.16	59.05	48.06	73.14	16.61	0	18.63	23.95	25.81	0
75 - 100	0	35.91	34.29	42.78	40.69	4.34	27.54	27.94	36.13	17.15	42.67	0	38.78	57.79	33.18	0
100 - 125	2.02	7.95	11.95	11.33	7.45	6.93	6.67	11.75	8.06	8.08	23.45	0.71	11.79	10.65	20.74	7.02
125 - 150	9.07	2.05	3.12	8.78	1.15	0	2.91	0.63	1.61	0	4.56	4.57	7.22	4.18	5.99	12.87
150 - 175	7.31	0.26	1.04	3.68	0	0	0.58	0	0.65	0	2.93	9.86	4.56	1.52	2.76	16.96
175 - 200	12.59	0	0.24	1.42	0	0	0.29	0	0.32	0	0	14.08	2.28	0	0.46	7.02
200 - 225	13.85	0	0	0.85	0	0	0.58	0	0	0	0	11.97	0	0	0	11.11
225 - 250	8.56	0	0	0.85	0	0	0	0	0	0	0	12.32	0	0	0	6.43
250 - 275	7.56	0	0	0.57	0	0	0.27	0	0	0	0	11.62	0	0	0	7.61
275 - 300	12.59	0	0	0.28	0	0	0	0	0	0	0	9.15	0	0	0	8.19
300 - 325	12.59	0	0	0	0	0	0	0	0	0	0	9.51	0	0	0	8.19
325 - 350	7.05	0	0	0	0	0	0	0	0	0	0	8.1	0	0	0	8.19
350 - 375	4.28	0	0	0	0	0	0	0	0	0	0	4.93	0	0	0	4.68
375 - 400	1.01	0	0	0	0	0	0	0	0	0	0	1.78	0	0	0	0.58
400 - 425	1.52	0	0	0	0	0	0	0	0	0	0	0.69	0	0	0	0.58
425 - +∞	0	1.01	1.31	0.28	0.85	0	0	0.63	5.17	1.31	9.78	0.71	16.74	1.91	11.06	0.57

Figure 5: Frequencies in % for RTT

Cluster / Category	14	3	7	12	8	4	0	6	1	5	11	15	9	13	10	2
1	0	51.28	58.18	0	0	0	0	0	0	0	48.86	43.66	0	0	0	4.68
2	0	43.33	41.82	0	0	0	0	16.51	0	0	51.14	50.71	0	0	0	16.37
3	41.81	5.39	0	0	0	0	0	83.49	1.61	0	0	5.63	0.38	0	81.57	39.77
4	39.04	0	0	0	0	0	0	0	48.39	51.46	0	0	38.78	39.16	18.43	22.22
5	17.63	0	0	0	0	0	0	0	50	48.54	0	0	60.46	60.84	0	15.21
6	1.52	0	0	49.58	47.85	47.98	48.41	0	0	0	0	0	0.38	0	0	1.75
7	0	0	0	50.42	52.15	52.02	51.59	0	0	0	0	0	0	0	0	0

Figure 6: Frequencies in % for DAY

Cluster / Category	14	3	7	12	8	4	0	6	1	5	11	15	9	13	10	2
0	0	13.08	0	0	0	0	16.81	0	14.52	0	0	0	0	0	0	36.84
1	0	13.59	0	0	0	0	17.11	1.59	13.23	0	0	0	0	0	0	29.82
2	0	16.15	0	0	0	0	16.81	3.81	16.77	0	0	0	0	0	0	13.45
3	0	14.62	0	0	0	0	16.23	6.67	17.42	0	0	0	0	0	0	10.53
4	0	15.13	0	0	0	0	16.23	7.62	18.39	0	0	0	0	0	0	5.26
5	0	12.82	0	0	0	0	16.81	12.06	18.71	0	0	0	0	0	0	2.34
6	0	12.82	0	0	0	16.76	0	11.75	0.96	18.12	0	0	0	0	0	1.76
7	0	1.79	8.05	0	0	16.76	0	15.87	0	19.42	0	0	0	0	0	0
8	0	0	12.73	0	0	16.76	0	12.06	0	19.09	0	0	0.76	0	0	0
9	0	0	15.32	0	0	16.76	0	8.89	0	17.48	0	0	2.28	0	0	0
10	0.51	0	14.55	0	0	16.19	0	7.94	0	15.53	0	0.35	4.56	0	0	0
11	1.76	0	14.81	0	0	16.77	0	7.62	0	10.03	0	0.35	9.89	0	0	0
12	5.04	0	14.03	0	16.05	0	0	3.49	0	0.33	0	1.76	20.53	0	2.31	0
13	7.56	0	11.17	0	16.33	0	0	0.63	0	0	0	4.23	18.25	0	5.52	0
14	9.82	0	8.57	0	16.62	0	0	0	0	0	3.26	5.63	16.35	0	5.53	0
15	10.33	0	0.77	0	16.62	0	0	0	0	0	8.79	9.86	15.59	0	5.53	0
16	11.59	0	0	0.85	16.05	0	0	0	0	0	11.07	8.10	9.89	0	10.14	0
17	9.07	0	0	2.83	14.32	0	0	0	0	0	10.42	9.15	1.90	7.6	15.21	0
18	9.07	0	0	12.76	4.01	0	0	0	0	0	11.07	9.15	0	12.55	11.52	0
19	9.57	0	0	16.72	0	0	0	0	0	0	12.05	8.10	0	14.07	9.22	0
20	8.56	0	0	16.71	0	0	0	0	0	0	9.45	11.27	0	15.21	7.83	0
21	6.55	0	0	16.71	0	0	0	0	0	0	10.10	12.32	0	16.35	8.29	0
22	5.54	0	0	16.71	0	0	0	0	0	0	11.40	10.21	0	16.35	10.14	0
23	5.03	0	0	16.71	0	0	0	0	0	0	12.39	9.52	0	17.87	8.76	0

Figure 7: Frequencies in % for HOUR

Cluster / Category	14	3	7	12	8	4	0	6	1	5	11	15	9	13	10	2
19	64.48	23.08	28.83	36.54	24.93	31.21	29.86	45.40	31.29	38.51	14.66	53.17	30.80	22.05	12.44	63.16
20	30.73	41.79	42.34	40.23	43.55	41.04	41.74	32.38	39.68	37.86	39.09	37.68	38.40	43.73	33.64	25.15
21	4.79	30.77	26.23	23.23	31.52	27.75	27.54	22.22	26.45	22.98	44.63	8.80	30.42	34.22	53.00	10.53
22	0	3.08	2.34	0	0	0	0	0	0.32	0	1.62	0.35	0	0	0.92	1.16
23	0	1.28	0	0	0	0	0.86	0	2.26	0.65	0	0	0.38	0	0	0
25	0	0	0.26	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8: Frequencies in % for HOP

6.2.3 Details for clusters

The description of clusters shows how the clusters differ from each other and from the total population. Each cluster defines a class of network behavior.

For 61% of all records the RTT is medium and HOP is predominantly 20. However, the analysis shows that 17% of records exhibit high RTT. They are grouped into Clusters 14 (the biggest one), Cluster 15 and Cluster 2. For these clusters HOP is predominantly 19, i.e. one hop less than on average. 12.5% of records is characterizing by HOP=19 and medium RTT. For 9.5% of records RTT is medium and HOP is predominantly 21 (Clusters 10 and 11), i.e. one hop more than on average.

Several empirical works showed that the distributions of RTT and hops are markedly different. Whereas the distribution of hops is fairly symmetric about the mean, the round-time latency has long-tailed distribution. Therefore it is not surprising that RTT and hop count are not strongly correlated. This problem can be observed in Clusters 11 and 15. Namely, they have almost identical DAY and HOUR

distributions whereas strongly different RTT distributions. Using clustering we can find such "anomalies".

Cluster 14 which is the biggest one (7.93% of total population) defines the set of records where RTT is high (modal value is 200-225 ms), DAY is predominantly 3, HOP is predominantly 19, and HOUR is predominantly 16, whereas Cluster 3 (7.79%) defines the set of records where DAY is predominantly 1, HOUR is predominantly 2, HOP is predominantly 20, and RTT is medium (modal value is 50-75 ms).

More details about clusters can be derived. For instance, in case of Cluster 14 about 42% of measurements belonging to that cluster is performed on Wednesday (DAY=3), and 39% on Thursday (DAY=4), while the number of hops is predominantly 19 for 65% records. The measurements are performed mostly between 12:00 and 23:00 (98%) with the predominant hour 16:00 (12%). The other 2% of records include remaining twelve values that might appear in the HOUR field. Cluster 14 characterizes bad network conditions. RTT is high, and has evident long-tailed distribution. RTT>150 ms is for 89% of records in this cluster. The interval 200 - 225 ms has the highest frequency (14%). Each of the RTT ranges 175-200, 275-300, 300-325 ms includes 12% of records. There are mostly 19 hops (65%). 20 hops are identified in 31% records whereas only 4% of paths are with 21 hops. Clusters 2 and 15 have similar RTT characteristics.

6.3 Classification mining function

One of the disadvantages of cluster models is that there are no explicit rules to define each cluster and there is no clear understanding of how the model assigns data to clusters. The clustering result cannot be easily used in advising to use particular Internet path. Therefore, we employ the classification mining function in the next step of building the model.

IBM's IM4D provides two classification mining functions, namely Tree-Induction algorithm and Neural Classification function (employing back-propagation neural network) [IM4D 2002]. In our case human understanding of the learned rules is crucial. Therefore we used the decision tree approach. A tree can be easily converted to a rule set by traversing each path from the root to each leaf node. Unlike neural networks, trees are easy to understand and modify.

In the data mining community most research is devoted to seeking detailed descriptions and rules within datasets. However busy people [Menzies and Hu 2003] need ease-to-read and immediately useful data mining rules. One of the challenges for data mining is to make data mining accessible to a wider range of users that are not experts in the field of data mining. Then the automation of data mining applications to hide the complexity of the underlying data mining process can help them to solve their problems. Such solution is available in IBM's DB2 Intelligent Miner which provides Easy Mining Procedures for an easy-to-use SQL interface for the main steps of the data mining process [IBM 2005, IM4D 2003]. This set of procedures is complemented by some important mining operations such as the computation of a data model for customer segmentation.

Another challenge is the integration of data mining procedures and decision-support systems to build automated customized data mining solutions [Pisharath et al. 2006] or knowledge-based services, e.g. in Grids [Talia 2006].

6.3.1 Tree-induction algorithm

The Tree-Induction Algorithm implemented in IM4D is a modified CART regression algorithm [Mehta et al. 1996]. CART (Classification and Regression Trees) is a method of creating rules to distinguish between clusters of observations and determine the class of new observations invented in [Breiman et al. 1984]. A particular feature of CART is that decision rules are represented via easy for interpretation and implementation binary trees. CART is non-parametric and can use continuous and/or categorical data. It can handle datasets with complex structures and extreme distributions, i.e. which are far from normality as measured by their skew (>1) and kurtosis (>7), and/or characterized by a high percentage of binary/categorical attributes (>38%) [King et al. 1995]. CART is robust to the effect of outliers. Although CART analyses all possible solutions it is efficient in implementation.

The trees develop arbitrary accuracy and use validation data sets to avoid spurious detail. In many instances the decision tree produces a very accurate representation of the cluster model (>90% accuracy). If the tree representation is accurate, it is preferable to implement a tree, because it provides explicit, easy-to-understand rules for each cluster. Decision trees represent the knowledge in the form of *if-then* rules. For each path from the root to a leaf such rule can be created. The leaf node holds the class prediction. That is why we used a decision tree to classify the cluster IDs using the output data that was obtained as a result of applying the clustering algorithm. Decision trees are also attractive because they show clearly how to reach a decision.

However decision trees can become too large and be no longer particularly understandable. They are also prone to significant changes due to the small perturbations in data. They can be overfitted over a training dataset. Such overfitted trees poorly generalize to unseen cases. Smaller consistent decision trees typically have higher generalization accuracy than larger trees. Therefore in most real life cases they must be corrected via pruning to obtain more reliable the "right-sized" trees [Esposito et al. 1997].

Tree model is developed in two phases: training and testing (validation). The most common measure of the quality of classification algorithms is *accuracy* or *error rate*. Prediction accuracy is normally measured on a separate dataset. In training phase we develop a model using historical data, whereas in testing phase we try out the model on new input data. The cross validation can be used, especially in case of small datasets. Temporal and expanding datasets challenge new problems of tree-induced classification and prediction that are solved via on-line, incremental or re-execution based solutions [Walid et al. 2004]. Very often the classification algorithms can work only on a small dataset limited by the computer memory size. These problems do not apply to CART implementation in IM4D because it runs using a DB2 database.

6.3.2 Decision tree

When building the tree, we did not limit the number of node levels for the tree to be created. The default parameter settings were used. For these settings the maximum tree depth was unlimited. The maximum purity per internal node was 100%. The minimum number of records per internal node was 5. For theses settings the final

decision tree (Figure 9) with a total of 46 nodes and the maximum tree depth of 8 was reached. The purity in leaf node indicates the percentage of correctly predicted records in that node. Scores shows classes or in our case, simply clusters' IDs.

We preceded the following to evaluate the model. The dataset of 5004 records was divided into two subsets, namely the *training set* and *test set*. Test set is usually takes of 30% of the total data although there is no strong reason for that. We used this setting and others, as well. Usually this partitioning is made in a random fashion, but in our case, taking into account the character of our application, we split a dataset of 5004 record up into two separate groups, each containing records collected in consecutive hours and days. The training set started each time from the beginning of the measurements. The training set was used to derive the decision tree and the test set was used to validate obtained tree. IM4D has the feature to provide such calculations by using options "Application Clustering" while clustering and "Test Mode" while classification. Using the Application Clustering we can apply the clustering model to the new data. In this way we obtained scores for classes for testing records. In Test Mode we can employ the decision tree reached in the Training Mode in classification of unknown records.

#	**Dataset division**	**Classification accuracy**	
	Training set/Test set	**Training mode**	**Test mode**
1	70%/30%	88.12%	74.14%
2	75%/25%	85.79%	67.33%
3	80%/20%	81.90%	69.12%
4	90%/10%	77.56%	71.43%

Figure 10: Classification accuracy for set of 5004 records

Figure 10 shows the results of classification accuracy for four divisions defined by the number of records in the training set/test set: 3500/1504 (70%/30%), 3750/1254 (75%/25%), 4000/1004 (80%/20%), and 4500/5004 (90%/10%). In the context of the classification accuracy measure our classifier can be evaluated positively. However, the question is, is the classification accuracy on old data likely to be a good indicator of the classification accuracy on new data? This is a general question which remains fully unanswered. The question is much deeper, is the classification accuracy or error rate appropriate quality measure in our application? We will touch this problem in Section 6.4.

6.3.3 Decision rules

Once a decision tree has been constructed it is simpler to interpret the results that we have obtained using clustering, and determine how we can map them to study network performance problem. The decision tree consists of well-defined rules, which can be easily applied to the decision making how to use the network. Terminal nodes represent classes of network behavior, i.e. clusters obtained in the previous mining step. Classes can be reached by traversing down paths through the tree, starting from the root node and recording the test outcomes in internal nodes and the terminal node

as the result of forecasting. Each such path defines the *if-then* rule that is easy for people to understand. For example, if the user wants to transfer data on Saturdays or Sundays morning between 6 and 11 o'clock then he or she may expect fine network behaviour because such network behavior forecast can be read from the following rule: *If* (DAY ≥5.5) *and* (HOUR < 11.5) *and* (HOUR ≥5.5) *then* Cluster = 4. In this forecast the RTT is *Medium* as described by Cluster 4.

The following general rule for describing RTT behavior can be derived from the decision tree:

If [(DAY = 3) *and* (11.5 ≤ HOUR < 16.5)]
or [(DAY = 4) *and* (15.5 ≤ HOUR < 19.5)]
then RTT = *High*
else
RTT = *Medium*.

The decision tree from Figure 9 defines a total of 24 rules among them we have four rules which concern *High* RTT conditions. The maximum length of such rule is 8. As we mentioned earlier, the Internet path under consideration has generally a good network "weather", however the user have to use it carefully within time ranges which are shown in the rules that produce *High* RTT, for example on Wednesdays between 12 and 16 o'clock. Then the rule warns: *If* (DAY < 5.5) *and* (DAY ≥2.5) *and* (HOUR ≥11.5) *and* (HOUR < 16.5) *and* (DAY < 3.5) *then* Cluster = 14. Cluster 14 exploits high RTT.

6.4 Data mining on an expanding database

Our database naturally is continuously expanding which can invalidate discovered patterns and introduce new ones. This problem is addressed in a recent work [Cavalcanti and Belo 2005] where an approach, called incremental or on-line mining, was proposed to handle the problem of mining user patterns when new transactions are added to the Web log file by only considering user patterns obtained by an earlier mining.

Here we use a conventional data mining approach with techniques and tools available in IM4D [IM4D 2002]. Thus, our two-step data mining procedure needs to be re-executed periodically to update the mining results.

Two history-based re-execution schemes, namely an *incremental window* data mining and *moving window* data mining are proposed and evaluated. The former assumes that at each re-execution point a dataset includes whole historical data available. The latter consists in the analysis of historical data available within a time window ending at re-execution point. Both techniques were applied to the total dataset of transactions collected during 46 weeks. We assumed that in both schemes re-executions are made every weekend.

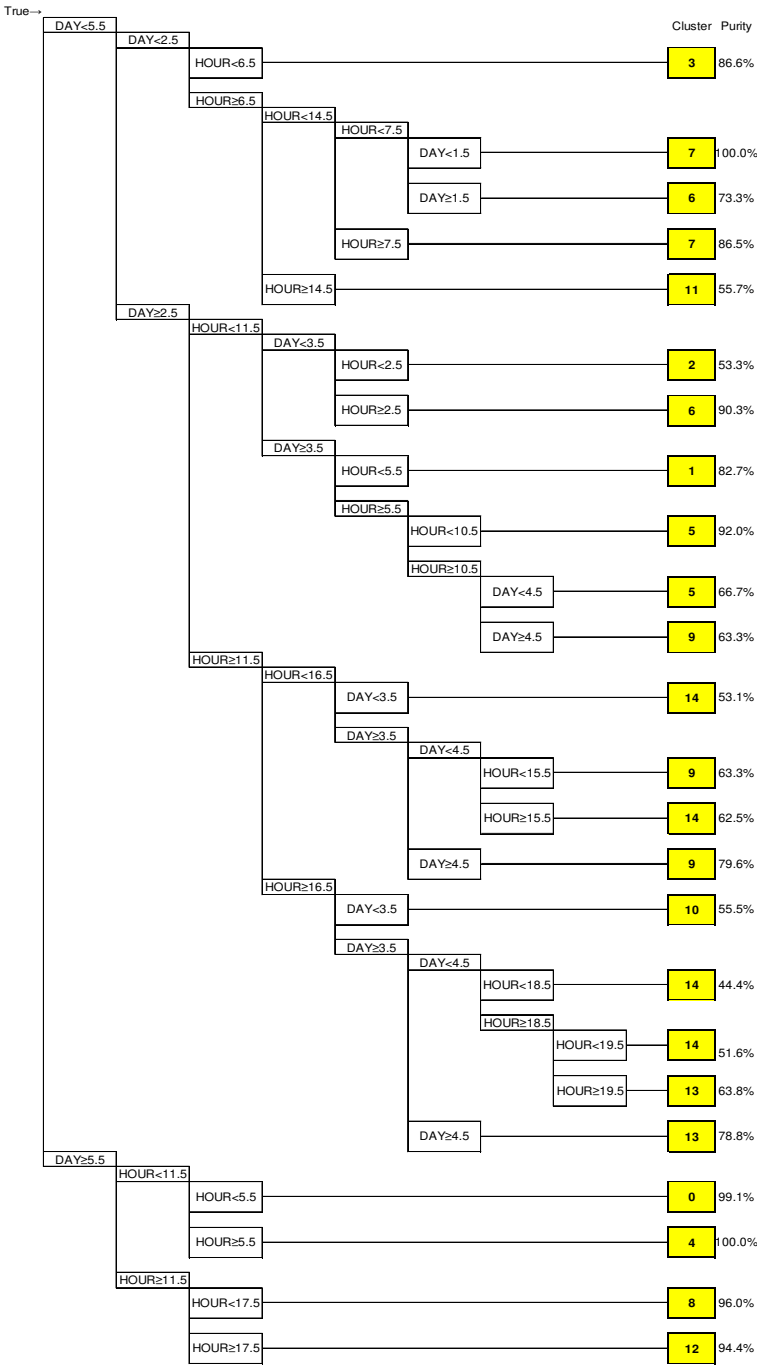

Figure 9: A decision tree model

6.4.1 Incremental window scheme

The re-executions are made at the weekend. Considering periodic characteristics of network behavior we use one week increment. Thus the dataset size varies from 1 to 46 weeks. Figure 11a presents the classification accuracy vs. time (week). Generally, the result shows that more data gives better classification, achieving around 92% for ten last re-executions. For small datasets the prediction accuracy fluctuates what is can be connected with the changes in the network configuration in consecutive weeks, whereas these changes are not so influential and we observe the "smoothing" effect at the reasonable level of the prediction accuracy.

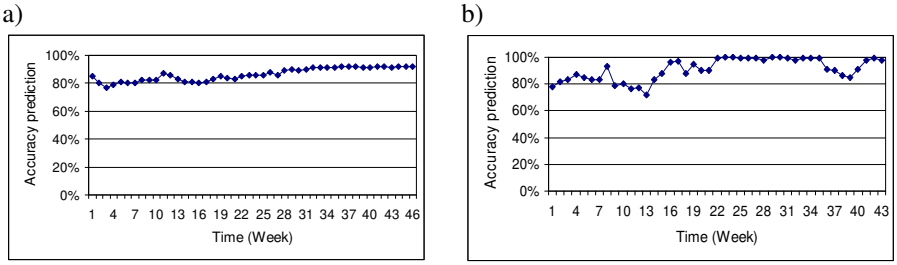

Figure 11: Prediction accuracy: (a) incremental window scheme, (b) four-week moving window scheme

The accuracy is not a key issue here as it is commonly considered in classification problems. In our application domain the most important is the ability to predict *High* RTT states. Figure 12 shows *High* RTT scores achieved in incremental window mining scheme. The vertical axis represents the hours-of-the-week from Monday to Thursday. Fridays, Saturdays and Sundays were omitted in the graph because none *High* RTT behavior was observed during these days. The horizontal axis corresponds to the successive weeks. The graph from Figure 12 may guide the user when to use the network.

Because a process of incremental mining we mine more and more data while consecutive re-executions and a model gets lost sensitivity to the changes in RTT, and what is more important, it loses *High* RTT rules. It means that after some weeks of incremental mining the decision trees which are re-mined have no rules that may guide about such network state. It is an open problem which is still under study.

6.4.2 Moving window scheme

Further analysis includes the moving window data mining scheme. Moving window data mining is the approach which addresses timeliness matters in our history-based predictions. The mining over the entire history may be impractical in this application as Internet is dynamically changing.

The *n*-week window is defined on the basis of the timestamp and include all samples from the dataset stream in the last *n* units of time (*n*=1, 2, 3, 4 and 6 weeks) before that timestamp. The prediction accuracy of hops and RTT based on the moving four-week window scheme is shown in Figure 11b. The accuracy varies very much

from one re-execution point to another. After analyzing all window sizes we have found that four week window gives pretty high accuracies, on average, 91% (RTT) and 96% (HOP). Bigger windows do not significantly improve the result and may be too long for justified network monitoring. Very small window (one week) is not enough for prediction. But even one-month windows cannot help in the prediction when the network changes too quickly. Both schemes showed that from 6th to 12th week something alarming happened in the network. Then even the prediction of hops was unsuccessful.

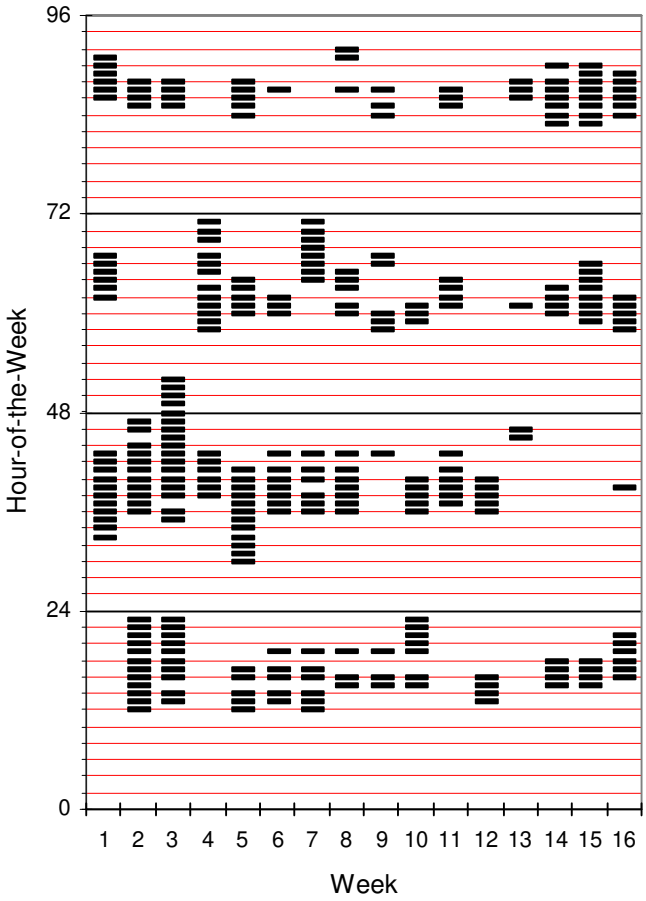

Figure 12: High RTT scores in incremental window data mining scheme

7 Conclusions

This work showed how data mining techniques can be applied to the analysis of Internet paths performance. Using data mining we can add new points of view and better understanding of diagnostic challenges. We demonstrated how two data mining

functions, namely clustering and classifying, can be used to discover the rules that may be used in long-range forecasting of performance of end-to-end Internet path.

The data mining based modeling is not supported by a unique and solid mathematical background. Thus, it is essential to discuss parameter settings in detail. Furthermore, in this case we assumed that most parameter settings and preprocessing actions (e.g. RTT discretization) are done in an automatic fashion by IM4D itself.

This paper provides a proof-of-principle that the proposed data mining driven modeling approach is applicable to computer networks research. Real-life case study showed that the resulting decision tree may help the user in deciding how to use the particular Internet link. The sample model gave pretty high accuracies.

A decision tree based classification commonly requires a relatively large training dataset, which may not be feasible to obtain. Probably this is not our case. Presented results show that in our application too much data may be impractical because using this data the decision maker is not able to distinguish bad network behavior cases, in spite of the fact that the classifier is very accurate.

We believe that our approach can be used in the development of network monitoring brokers allowing link selection and data transfer scheduling in response to changing network conditions. This approach can be profitable especially for the non-networkers who are working e.g. in Grids environments.

To generalize the result we need further deep study, including the analysis of other RTT observations performed in various network conditions. Our future research include: (1) Developing a Network Monitoring Broker with automated data mining and path selection mechanism; (2) Deployment of on-line data mining algorithms supported by the expert-based knowledge about Internet behavior; (3) Advancing an approach by dealing and integration with higher network layers transmission problems.

Acknowledgements

This work was supported by the Polish Ministry of Science and Higher Education under Grant No. N516 032 31/3359 (2006-2009).

References

[Aberer et al. 2005] Aberer, K., Alima, O., Ghodsi, A., Girdzijauskas, S., Haridi, S., Hauswirth, M.: „The Essence of P2P: A Reference Architecture for Overlay Networks"; P2P2005, The 5th IEEE International Conference on Peer-to-Peer Computing, Konstanz, Germany, (August 31-September 2, 2005).

[Abusina et al. 2005] Abusina, Z.U.M., Zabir, S.M.S., Asir, A., Chakraborty, D., Suganuma, T., Shiratori, N.: "An Engineering Approach to Dynamic Prediction of Network Performance from Application Logs"; Int. Network Mgmt 15(3) (2005), 151-162.

[Andersen 2005] Andersen, D.G.: "Improving End-to-End Availability Using Overlay Networks"; Ph. D Dissertation, Massachusetts Institute of Technology, (February 2005).

[Andersen et al. 2001] Andersen, D., Balakrishnan, H., Kaashoek, F., Morris, R.: "Resilient Overlay Networks"; Proc. of 18th ACM Symp. on Operating Systems Principles, Banff, Canada (2001), 131-145.

[Arlit et al. 2005] Arlit, M., Krishnamurthy, B., Mogul, J.C.: "Predicting Short-Transfer Latency from TCP Arcane: A Trace-Based Validation"; Proc. of International Measurement Conference IMC'05. USENIX Association, Berkeley (2005), 119-124.

[Avery and Foster 2001] Avery, P., Foster, I.: "The GriPhyN Project: Towards Petascale Virtual-Data Grids"; GriPhyN TR2001-14, http://www.griphyn.org (2001).

[Ballintijn et al. 2000] Ballintijn, G., Van Steen, M., Tanenbaumn, A. S.: "Characterizing Internet Performance to Support Wide-Area Application Development"; Operating Systems Review, 34 (4) (2000), 41-47.

[Baragoin et al. 2002] Baragoin, C., Andersen, C.M., Bayerl, S., Bent, G., Lee, J., Schommer, C.: "Mining Your Own Business in Telecoms Using DB2 Intelligent Miner for Data"; IBM Redbooks, SG24-6273-00 (2002).

[Barford et al. 2001] Barford, P., Bestavros, A., Byers, J., Crovella, M.: "On the Marginal Utility of Network Topology Measurements"; ACM SIGCOMM Internet Measurement Workshop, San Francisco, ACM Press, New York (2001), 5-17.

[Borzemski 2004] Borzemski, L.: "Data Mining in Evaluation of Internet Path Performance"; Lecture Notes in Artificial Intelligence, Vol. 3029, Springer-Verlag Berlin (2004), 643-652.

[Borzemski 2005] Borzemski, L.: "Mining Internet Data Sets for Computational Grids"; Lecture Notes in Artificial Intelligence, Vol. 3683, Springer-Verlag, Berlin (2005), 268-274.

[Borzemski 2006a] Borzemski, L.: "Testing, Measuring and Diagnosing Web Sites from the User's Perspective"; Intl J. of Enterprise Information Systems, 2(1) (2006), 54-66.

[Borzemski 2006b] Borzemski, L.: "The Use of Data Mining to Predict Web Performance"; Cybernetics & Systems: An International Journal, 37(6), (September 2006), 587-608.

[Borzemski et al. 2007] Borzemski, L., Cichocki, L., Fras, M., Kliber, M., Nowak, Z.: "MWING: A Multiagent System for Web Site Measurements"; Proc. of 1st KES Symposium on Agent and Multi-Agent Systems – Technologies and Applications, Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin (2007), to appear.

[Borzemski and Nowak 2004a] Borzemski, L., Nowak, Z.: „WING: A Web Probing, Visualization and Performance Analysis Service"; Lecture Notes in Computer Science. Vol. 3140. Springer-Verlag Berlin (2004), 601-602.

[Borzemski and Nowak 2004b] Borzemski, L., Nowak, Z.: "An Empirical Study of Web Quality: Measuring the Web from the Wroclaw University of Technology Campus"; Matera, M., Comai, S. eds. Engineering Advanced Web Applications. Rinton Publishers, Princeton (2004), 307-320.

[Borzemski and Nowak 2005] Borzemski, L., Nowak, Z.: "Using the Geographic Distance for Selecting the Nearest Agent in Intermediary-Based Access to Internet Resources"; Lecture Notes in Artificial Intelligence, Vol. 3683, Springer-Verlag, Berlin (2005), 261-267.

[Breiman et al. 1984] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: "Classification and Regression Trees"; Wadsworth International Group (1984).

[Brownlee and Loosley 2001] Brownlee, N., Loosley, C.: "Fundamentals of Internet Measurement: A Tutorial"; Keynote Systems, USA (2001).

[Cabena et al. 1999] Cabena, P., Choi, H. H., Kim, I. S., Otsuka, S., Reinschmidt, J., Saarenvirta, G.: "Intelligent Miner for Data Application Guide"; IBM Redbooks, SG24-5252-00 (1999).

[CAIDA 2006] The Cooperative Association for Internet Data Analysis; http://www.caida.org/tools/taxonomy/performance.xml (2006).

[Cardellini et al. 2002] Cardellini, V., Casalicchio, E., Colajanni, M., Yu, P.S.: "The State of the Art in Locally Distributed Web-Server Systems" ACM Computing Surveys, Vol. 34, No. 2, June (2002), 263-311.

[Cavalcanti and Belo 2005] Cavalcanti F., Belo, O.: "Improving Effectiveness of Web Sites Using Incremental Data Mining over Clickstreams"; Data Mining VI: Data Mining, Text Mining and their Business Applications, Vol. 35. Zanasi, A., Brebbia, C.A.,Ebecken, N. (eds.), WIT Press, Southampton (2005), 533-542.

[Chakrabarti 2003] Chakrabarti, S.: "Mining the Web: Analysis of Hypertext and Semi Structured Data". Morgan Kaufmann, San Francisco (2003).

[Despotovic and Aberer 2004] Despotovic, Z., Aberer, K.: "Probabilistic Prediction of Peers' Performances in P2P Networks"; Lecture Notes in Computer Science. Vol. 3191. Springer-Verlag Berlin (2004), 62-76.

[Esposito et al. 1997] Eposito, F., Malerba, D., Semeraro, G.:"A Comparative Analysis of Methods for Pruning Decision Trees"; IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no 5, May (1997), 476-491.

[Eswaradass et al. 2006] Eswaradass, A., Sun, X-H., Wu, M.: "Network Bandwidth Predictor (NBP):A System for Online Network Performance Forecasting"; Proc. of Sixth IEEE Symposium on Cluster Computing and the Grid (CCGRID'06), IEEE CS Press, Los Alamitos (2006), 265-268.

[Faloutsos et al. 2002] Faloutsos, M., Faloutsos, Ch.: "Data-Mining the Internet: What We Know, What We Don't, and How We Can Learn More" Full day Tutorial ACM SIGCOMM 2002 Conference, Pittsburgh (2002).

[Foster and Kesselman 1997] Foster, I., Kesselman, C.: "Globus: A Metacomputing Infrastructure Toolkit"; Intl J. Super-computer Applications, 11(2), (1997), 115-128.

[Foster and Kesselman 2003] Foster I., Kesselman, C. (Eds.): "The Grid: Blueprint for a New Computing Infrastructure"; Second Edition, Morgan Kaufmann, Elsevier, San Francisco (2003).

[Fürnkranz 2005] Fürnkranz, J.: "Web Mining"; Maimon, O., Lior, R., eds. Data Mining and Knowledge Discovery Handbook. Springer-Verlag, Berlin (2005), 899-920.

[Gnutella 2005] http://www.gnutellahosts.com (2005).

[Grossman et al. 2001] Grossman, R. L., Kamath, Ch., Kegelmeyer, P., Kumar, V., Namburu, R. R. (Eds.): "Data Mining for Scientific and Engineering Applications"; Kluwer Academic Publishers, Boston, Dordrecht, London (2001).

[Gummadi et al. 2002] Gummadi, K. P., Saroiu, S., Gribble, S. D.: "King: Estimating Latency between Arbitrary Internet End Hosts"; SIGCOMM Internet Measurement Workshop, Marseille, France, ACM Press, New York (2002), 5-18.

[Han and Kamber 2000] Han, J., Kamber, M.: "Data Mining: Concepts and Techniques"; The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor, Morgan Kaufmann Publishers, San Francisco (2000).

[He et al. 2005] He, Q., Dovrolis, C., Ammar, M.: "On the Predictability of Large Transfer TCP Throughput"; Proc. SIGCOMM'05. New York: ACM Press, New York (2005), 145-156.

[Hellerstein et al. 2001] Hellerstein, J., Zhang, F., Shahabuddin, P.: "A Statistical Approach to Predictive Detection"; Computer Networks 35 (2001), 77-95.

[IBM 2000] "GUI Guide for Data Mining"; United States Patent 6108004, http://www.freepatentsonline.com/6108004.html (2000).

[IBM 2005] IBM developerWorks: "Developing Grid Computing Applications and Web Services"; http://www-106.ibm.com/developerworks/ (2005).

[IM4D 2002] "Using the Intelligent Miner for Data V8 Rel. 1"; IBM Redbooks, SH12-6394-00 (2002).

[IM4D 2003] "Data Mining with Easy Mining procedures Version 8.2"; IBM Redbooks, SH12-6771-00 (2004).

[Johnston 2003] Johnston, W. E.: "Computational and Data Grids in Large-Scale Science and Engineering"; LBNL and NASA Ames Research Center, Meeting of the Japanese National Research Grid Initiative project, Tokyo, Japan, http://www-itg.lbl.gov/~wej/ (2003).

[Keynote 2006] Keynote Systems; http://www.keynote.com (2006).

[King et al. 1995] King, R.D., Feng, C., Sutherland, A.:"Statlog: Comparison of Classification Algorithms on Large Real-World Problems"; Applied Artificial Intelligence, 9 (1995), 289-333.

[Leese et al. 2005] Leese, M., Tyer, R., Tasker, R.: "Network Performance Monitoring for the Grid"; UK e-Science, 2005 All Hands Meeting (2005).

[Luckie et al. 2001] Luckie, M. J., McGregor, A. J., Braun, H.-W.: "Towards Improving Packet Probing Techniques" ACM SIGCOMM Internet Measurement Workshop, San Francisco, ACM Press, New York (2001), 145-150.

[Mehta et al. 1996] Mehta, M., Agrawal, R., Rissanen, J.: "SLIQ: A Fast Scalable Classifier for Data Mining"; Lecture Notes in Computer Science. Vol. 1057. Springer-Verlag Berlin (1996), 18-32.

[Menzies and Hu 2003] Menzies, T,. Hu, Y.: "Data Mining for Very Busy People"; Computer October (2003), 18-25.

[Mogul 2002] Mogul, J., "Clarifying the Fundamentals of HTTP"; Proc. of WWW11 Conference, Honolulu, ACM Press, New York (2002), 25-36.

[Paxson 1997] Paxson, V.: "End-to-End Routing Behavior in the Internet"; IEEE/ACM Transactions on Networking, 5(5), (1997), 601 - 615.

[Paxson 2004] Paxson, V.: "Strategies for Sound Internet Measurement"; Proc. ACM SIGCOMM Internet Measurement Conference, Taormina, Italy, ACM Press, New York (October 2004), 263 - 271.

[Pisharath et al. 2006] Pisharath, J., Zambreno, J., Ozisikyilmaz, B., Choudhary, A.: "Accelerating Data Mining Workloads: Current Approaches and Future Challenges in System Architecture Design"; 9th International Workshop on High Performance and Distributed Mining, Bethesda April 22 (2006).

[PMML 2006] "Predictive Model Markup Language (PMML)"; Data Mining Group, http://www.dmg.org/ (2006).

[Prasad et al. 2003] Prasad,R. S., Murray, M., Dovrolis, C., Claffy, K.: "Bandwidth Estimation: Metrics, Measurement Techniques, and Tools", IEEE Network, November/December 17(6) (2003), 27- 35.

[Saroiu et al. 2002] Saroiu, S., Gummadi, K. P., Dunn, R. J., Gribble, S. D., Levy, H. M.: "An Analysis of Internet Content Delivery System" Proc. of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002), Boston (2002), 315 - 327.

[Schulz et al. 2001] Schulz, J., Hochberger, C., Tavangarian, D.: "Prediction of Communication Performance for Wide Area Computing Systems"; Proc. of Ninth Euromicro Workshop on Parallel and Distributed Processing (PDP'01), IEEE CS Press, Las Alamitos (2001), 480-486.

[SLAC 2006] Stanford Linear Accelerator Center; http://www.slac.stanford.edu (2006).

[Srikant and Yang 2001] Srikant, R., Yang, Y.: "Mining Web Logs to Improve Website Organization"; Proc. of WWW10 Conference, Hong Kong, ACM Press, New York (2001), 430 - 437.

[Swany and Wolski 2002] Swany, M., Wolski, R.: "Multivariate Resource Performance Forecasting in the Network Weather Service"; Proceedings of the IEEE/ACM SC2002 Conference, Baltimore, ACM Press, New York (2002), 1-10.

[Talia 2006] Talia, D.:"Grid-Based Distributed Data Mining Systems, Algorithms and Services"; 9th International Workshop on High Performance and Distributed Mining, Bethesda April 22 (2006).

[Tsuru and Oie 2001] Tsuru, M., Oie, Y.: "Introduction to the Network Tomography"; GENESIS Technical Report IEICE Tech. Rep., IN2001-106 (2001).

[Walid et al. 2004] Walid, G.A., Elfeky, M.G., Elmagarmid A.K.: " Incremental, Online, and Merge Mining of Partial Periodic Patterns in Time-Series Databases"; IEEE Trans. On Knowledge and Data Engineering, 16(2) (2004), 1-11.

[Wang et al. 2002] Wang, M., Madhyastha, T., Chan, N.H., Papadimitriou, S., Faloutsos, C.: "Data Mining Meets Performance Evaluation: Fast Algorithms for Modeling Bursty Traffic"; Proc. of 18th International Conference on Data Engineering, San Jose, ACM Press, New York (2002), 507-516.

[Williams 2005] Williams, S.: "JANET Development Programme 2004-2005. Performance Measurement Overview"; www.ja.net/development/mm/ (2005).

[Wolski 1998] Wolski, R.: "Dynamically Forecasting Network Performance Using the Network Weather Service"; Cluster Computing, 1(1) (1998), 119-132.

[Yousaf and Welzl 2005] Yousaf, M. M., Welzl, M. "A Reliable Network Measurement and Prediction Architecture for Grid Scheduling"; 1st IEEE/IFIP International Workshop on Autonomic Grid Networking and Management AGNM'05, Barcelona (2005).

[Zhang et al. 2004] Zhang, H., Keahey, K., Allock W., "Providing Data Transfer with QoS as Agreement-Based Service"; Proc. IEEE International Conference on Services Computing SCC'04, (2004), 344-353.

[Zhang et al. 2001] Zhang, Y., Duffield, N., Paxson, V., Shenker, S.: "On the Constancy of Internet Path Properties"; ACM SIGCOMM Internet Measurement Workshop, San Francisco, ACM Press, New York (2001), 197-211.