Data Mining in QoS-aware Media Grids

Xiuju Fu[1], Xiaorong Li[2], Lipo Wang[3], David Ong[4], and Stephen John Turner[5]

[1] Institute of High Performance Computing, Singapore 117528
fuxj@ihpc.a-star.edu.sg
[2] Institute of High Performance Computing, Singapore 117528
lixr@ihpc.a-star.edu.sg
[3] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798
elpwang@ntu.edu.sg
[4] School of Computing, Nanyang Technological University, Singapore 639798
david@ntu.edu.sg
[5] School of Computing, Nanyang Technological University, Singapore 639798
ASSJTurner@ntu.edu.sg

1 Introduction

With the advent of high-speed networking technology and multimedia compression, various network-based multimedia services have become available. One of most popular ones is media streaming, which delivers multimedia content as continuous data so that clients can browse this online without waiting for entire multimedia files to be downloaded. Such technologies have been widely explored to enable a variety of network-based multimedia applications like video-on-demand distance learning, video conferencing, and so on.

A QoS-aware media grid [17] aims at presenting a scalable, robust and secure media access over grid environments [21]. An ever-increasing demand in computational resources has prompted the growth of grid techniques, and has increased the heterogeneity of network services and connections. QoS-aware media streaming [18] is considered a critical part of a media grid, and involves the access of text, graphics, audio and video content. As the main components of multimedia services, audio and video streaming require qualified network resources, such as high bandwidth and low latency. Since media streaming servers operate in a shared network environment, an efficient traffic prediction mechanism on available bandwidth is important for choosing appropriate media streaming servers and resolutions of media content to provide Quality of Service (QoS) of media streaming in distributed heterogeneous grid environments. To support QoS-aware streaming services, multiple versions of media content with different resolutions are provided for adapting to variations in network conditions. If prediction of available bandwidth can be made beforehand, a suitable version of media content can be chosen based on this prediction, which could help decrease degradation, such as lost packets and jitter caused by insufficient in bandwidth.

Typical data mining (DM) tasks include prediction, classification, clustering, and discovery of association rules. With the objective of discovering unknown patterns from data, DM methodologies have been derived from the fields of machine learning (ML), artificial intelligence (AI), and statistics. Data mining techniques have begun to serve fields outside of computer science, scientific research and artificial intelligence, such as the financial area and factory assembly lines. DM has been shown to lead to improved efficiency in manufacturing, prompting marketing campaigns, detecting fraud, and predicting diseases based on medical records. With ever increasing demands on computing resources, DM applications have become desirable in grid computing environments.

Being an integrated environment, Grids are autonomous, share resources, heterogenous, and distributed in nature. A Media Grid shares these characteristics, but as well emphasizes media streaming functions. Many prediction techniques have been applied to media streaming for improving QoS Services through predicting the following:

1. *bandwidth* – in order to adapt to dynamic media content streaming;
2. *media streaming service request patterns* – in order to automatically duplicate media content;
3. *workload and resource availability of media servers* – to determine intelligent scheduling strategies;
4. *usage patterns of grid computing environments*;
5. *user request patterns in the media grid.*

Data are accumulated from heterogeneous resources in media grids, but there is currently a lack of an efficient data mining framework and appropriate techniques for analyzing data generated in such grids. The quality of media streaming is mainly affected by network capacity, bandwidth, and throughput. Capacity is the maximum possible bandwidth over a network path. There are two types of bandwidth: (i) *available bandwidth* is the maximum allowable bandwidth, and (ii) *consumption bandwidth* is the amount of bandwidth consumed. With bandwidth prediction, we could avoid congestion caused by heavy network loads and reduce overestimation (underestimation) of the bandwidth requirements from clients. In this chapter, we focus on applying neural networks for predicting bandwidth, which facilitates QoS in media streaming.

The chapter is organized as follows. Section 2 describes related work. Section 3 presents a media grid framework. Section 4 describes the data mining strategy for bandwidth prediction in media grids. Experiments of bandwidth prediction are presented in Section 5. Section 6 concludes the chapter.

2 Related Work

Grid computing is a relatively reecent technology that aggregates large amounts of geographically distributed resources – such as PC clusters, supercomputers, network storages/caches – to make applications which are impossible for a local machine/cluster to be possible over networks. During the last decade, grid technologies have been applied to computationally intensive applications in the field of high energy physics, and projects like GriPhyN[22], PPDG[23], BIRN[24], DataTAG[25] have achieved much success in data processing and distribution. While most current research focuses on providing high capacity computational services on grids, there is increasing interest in exploring the deployment of grid technologies for providing multimedia services over networks. Recent developments of media grids involve using industry standard streaming protocols [26] (for instance, RTP, RTSP, and RTCP) and integrating various grid technologies (for example, information, data management and resource management services) into media grids [17] to support a large population of internet streaming users. In this section, we review previous work on network bandwidth prediction over media grid environments, and briefly introduce neural networks.

2.1 Network Bandwidth Prediction

In [28] and [29], the bandwidth of out-going link of network node is modelled according to wavelet transformation coefficients. Neural networks are applied as well to predict these coefficients. However, it is difficult to determine the coefficients in the wavelet modelling method. The network traffic is dynamic and might vary dramatically under different conditions or during different time periods. It is therefore inappropriate to fix the coefficients for predicting the dynamic bandwidth in wavelet modelling.

The Network Weather Service (NWS) [34] is a well-known network performance measurement and has been used to predict network traffic in Grid computing. Besides probes which are used to record network performance, NWS uses prediction methods such as mean-based, median-based and autoregression to forecast the network traffic. Despite its strengths, this approach does have disadvantages similar to the wavelet modelling method – in other words, it is difficult to adapt to the dynamic network conditions of grid environments.

As a popular data mining tool, neural networks have been employed to predict network bandwidth. For example, [5] used neural networks to predict available network bandwidth. In their work, the recorded network traffic is divided into non-overlapped and continuous bins. The time stamp, minimum packet rate, maximum packet rate, average packet rate, minimum bit rate, maximum bit rate, and average bit rate are derived from each bin data and used as the inputs to neural networks. The outputs of the neural network predictor are the bandwidth of later K-step bins, K being defined by the user. This method holds promise for predicting network bandwidth. Our network bandwidth prediction method is similar to this method. However, we propose a new performance metric to better evaluate the performance of neural network predictors.

2.2 Brief Overviews on Neural Networks

It is estimated that there are 50?? different types of artificial neural networks (ANNs) in use today [ref??]. ANNs can be categorized according to different aspects, such as learning algorithm, the number of network layers, the direction of signal flow, the activation function, and so on.

Based on the learning algorithm, neural networks can be classified into three major categories:

- In *supervised learning*, pairs of input and target vectors are required to train networks, so that appropriate outputs corresponding to input signals are generated accordingly. With an input vector applied, the error between the output of the neural network and its target output is calculated, which is then used to tune weights in order to minimize this error. The least mean square (LMS) method is a well-known method in minimizing errors.???? Supervised learning includes error-correction learning, reinforcement learning and stochastic learning.;
- *Unsupervised learning* does not require target vectors for the outputs. Without input-output training pairs as external teachers, unsupervised learning is self-organized to produce consistent output vectors by modifying weights. Paradigms of unsupervised learning include Hebbian learning and competitive learning. Kohonen's self-organizing map (SOM)is a typical neural network whose learning is unsupervised.;
- Some neural networks employ *hybrid learning* . For example, counterpropagation networks and radial basis function (RBF) networks use both supervised (at the output layer) and unsupervised (at the hidden layer) learning. Counterpropagation neural networks combine network paradigms of SOM and Grossberg's outstar [33]. The counterpropagation neural network can be used to produce corresponding outputs when an input vector is incomplete, noisy or partially in error.

According to the direction of signal flow, neural networks can be categorized as *feedforward* – in which weight connections are fed forward from inputs through hidden neurons to output neurons – and *recurrent* – in which feedback connections are present. Compared with feed-forward neural networks, recurrent neural networks can be unstable and dynamic. Hopfield neural networks [33] are well-known recurrent neural networks. Recurrent neural networks have been studied as examples of chaotic systems [30][31]. The bidirectional associative memory (BAM) network is another type of neural network which employs feedback. The BAM is heteroassociative. Both BAM and Hopfield neural networks are able to produce correct outputs when inputs are partially missing or incorrect.

Adaptive resonance theory (ART) networks [33] deal well with the stability-plasticity dilemma.???????? ART classifies an input vector into its class according to the stored pattern with which it is most similar. The stored pattern is tuned to make it closer with the input vector. Without finding its matching pattern – it is within a predefined tolerance for matching purposes – a new category is generated by storing a pattern which is similar to the input vector.

Another important category of neural networks is based on how the activation of a hidden unit is determined. In multi-layer perceptron (MLP) neural networks, the activation of a hidden unit is determined by the scalar product of the input vector and its corresponding weight vector. In Radial basis function (RBF) neural networks the activation is determined by the distance between the input vector and a prototype vector.

In this paper, we employ MLPs exclusively.

3 System Model of Data Analysis over Media Grid

In this section, we describe our data analysis system model in which data mining techniques are employed.

3.1 Architecture

Our media grid multi-agent data analysis system adopts distributed analysis agents to provide on-line monitoring services for heterogeneous Internet users. Figure 1 shows the layout of the agent-based data analysis system where media servers, clients, and analysis agents are geographically distributed over the network. Each analysis agent can work independently or cooperatively to monitor and predict the resource usage of media servers over a certain area so as to improve the global resource utilization. Distributed analysis agents reduce the processing burden of streaming servers and are sufficiently flexible to deal with various streaming applications. Such a system is highly scalable and can ameliorate the effects of failure and overloading in a centralized monitoring system.

A Media Grid is a distributed infrastructure which brings together various distributed resources into a virtual environment and provide customized streaming services in a cost-effective way. It accumulates various types of abundant resources on the Internet (such as network storage, Personal Computers, or PC clusters) to improve the system computation and storage capacity. Figure **??** shows the hierarchical layered architecture of a media grid.

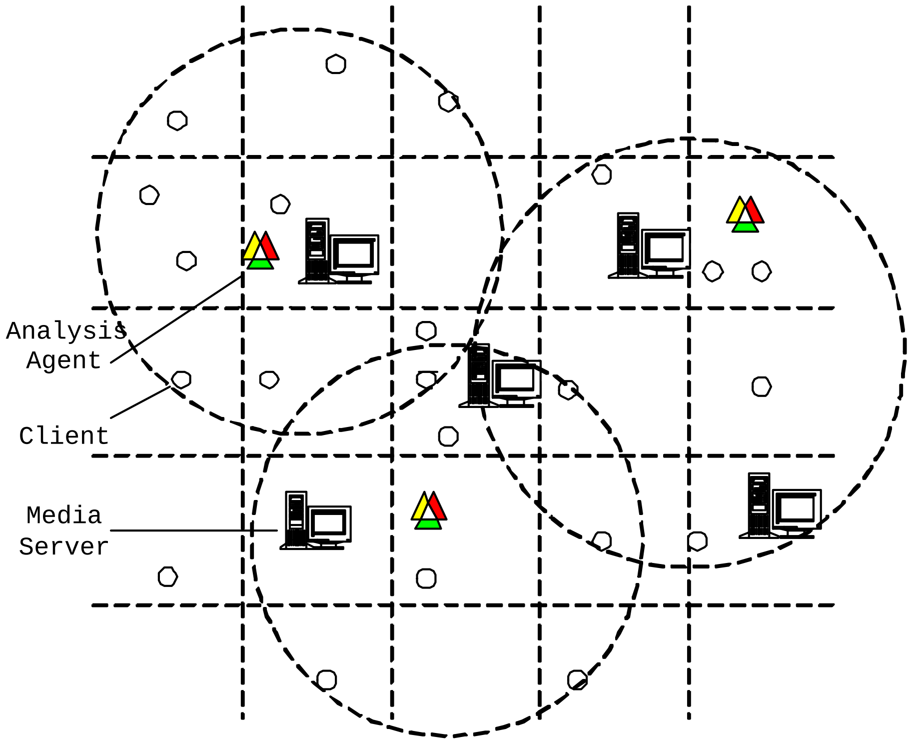

Fig. 1. Topology layout of a multi-agent based data analysis system

The lowest layer is the resource layer which manipulates distributed physical resources – media servers, storage, camera, and the like – to support higher level grid services. Above the resource layer is the grid middleware layer which includes the grid component services, e.g., information service [19], data management service [20], and so on. The grid middleware provides a firm foundation to support large-scale streaming services. It inherits such features of grids generally as the integration of loosely-coupled network resources, a service-oriented structure, and self-organization and decentralized methods of resource management. Media grid components are built above the grid component layer to provide multimedia-related services for Internet users. The media grid portal serves as an abstraction layer for the various media grid components. Through this portal, clients can subscribe to access media content with their QoS requirements such as playback time, bit rate, resolution, and so forth.

3.2 System Components

A media grid multi-agent data analysis system monitors and predicts network resource usage over distributed media grid environments. It consists of four main components: an analysis agent, a web portal, data storage, and a grid information service.

- *Analysis Agent*: The analysis Agent collects QoS information from a streaming server/client or (both), analyze it, provide feedback to the user, and store the data into a database.
- *Web Portal*: The web portal is a web service based interface which allows users to customize the quality monitoring by specifying their quality metrics or measurement conditions.
- *Data Storage*: QoS information and results are stored in a distributed database for archiving or long-term analysis.
- *Grid Information Service* [19]: When the user submits requests to web portal, the system needs to find a suitable analysis agent. The Grid Information Service will help to provide information on the available resources.

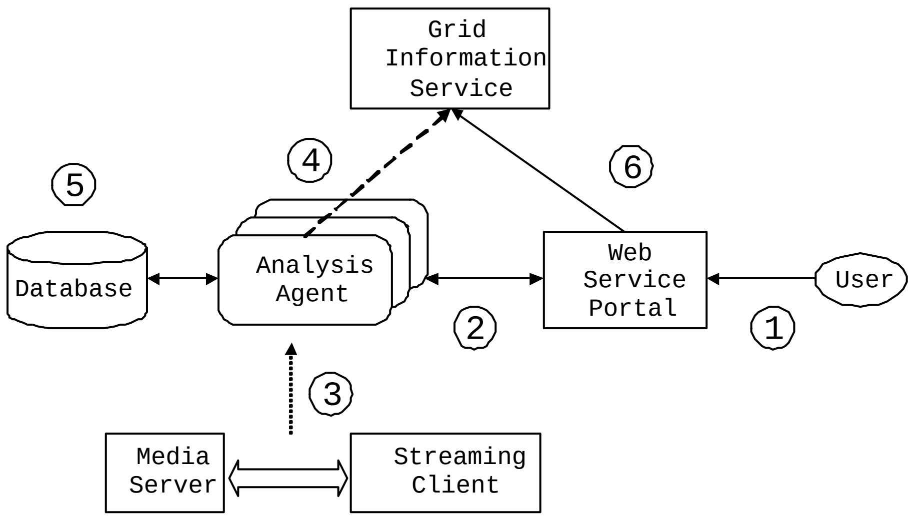

Fig. 2. Architecture and component interaction

Figure 4 demonstrates a working scenario of the multi-agent data analysis service, including and the interaction between each component. Each analysis agent will register itself to the grid information service as a computational resource over the network. The data analysis time sequence includes:

1. clients submit requests via the web service portal;

2. data is collected from the streaming servers and passed to the analysis agent for analysis;
3. the analysis agent performs data processing and predicts bandwidth consumption;
4. results are stored into the database for archiving or future analysis;
5. the results of the analysis or quality reports are sent back to the service providers to assist with resource management.

For a large scale network with multiple media servers distributed over different domains, it is not practical to use only one centralized server to monitor and analyze the system information due to the limited CPU and bandwidth resources. Such a system connects multiple agents and improves capacity by utilizing distributed computation power such as CPU and workstations over the Internet.

4 Data Mining Strategy for Bandwidth Prediction

In this section, we briefly introduce MLPs as neural network predictors for predicting network traffic in media grids. A data mining strategy for training these neural networks are is presented later (Sect.??).

4.1 Multi-Layer Perceptron Neural Network

A typical multi-layer perceptron (MLP) neural network classifier is shown in Figure **??**.

A hidden layer is required for MLPs to classify linearly inseparable data sets and make prediction. The input nodes do not carry out any processing. A hidden neuron in the hidden layer is shown in Figure **??**.

The jth output of a feed-forward MLP neural network is:

$$y_j = f(\sum_{i=1}^{K} W_{ij}^{(2)} * \phi_i(\mathbf{x}) + b_j^{(2)}) \tag{1}$$

where $W_{ij}^{(2)}$ is the weight connecting hidden neuron i with output neuron j, K is the number of hidden neurons, $b_j^{(2)}$ is the bias of output neuron j, $\phi_i(\mathbf{x})$ is the output of hidden neuron i, and $\mathbf{x}$ is the input vector.

$$\phi_i(\mathbf{x}) = f(\mathbf{W}_i^{(1)} \cdot \mathbf{x} + b_i^{(1)}) \tag{2}$$

where $\mathbf{W}_i^{(1)}$ is the weight vector connecting the input vector with hidden neuron i, and $b_i^{(1)}$ is the bias of hidden neuron i.

A common activation function f is the sigmoid or logistic function:

$$f(z) = \frac{1}{1 + e^{-\beta z}} \tag{3}$$

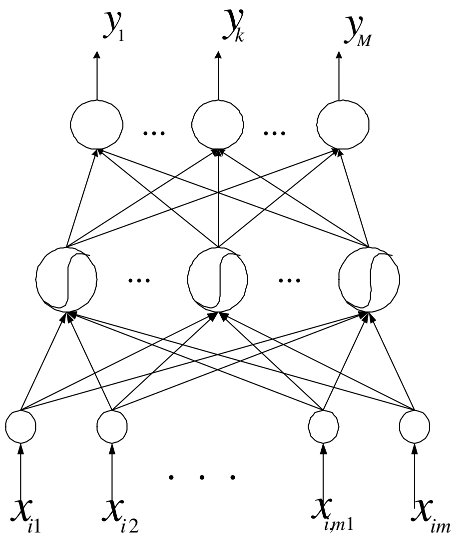

Fig. 3. A two-layer MLP neural network with one hidden layer

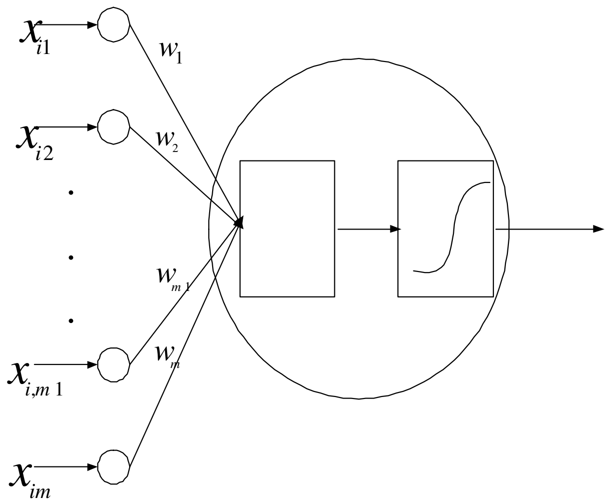

Fig. 4. A hidden neuron of the MLP.

where β is the gain.

Another activation function often used in MLP neural networks is the hyperbolic tangent, which takes on values between -1 and +1 (instead of 0..1, as with the sigmoid):

$$f(z) = \frac{e^{\beta z} - e^{-\beta z}}{e^{\beta z} + e^{-\beta z}} \tag{4}$$

There are many training algorithms for MLP neural networks reported in the literature, for example [11]: gradient decent error back-propagation (BP), back-propagation with adaptive learning rate, back-propagation with momentum, Quasi-Newton back-propagation, Bayesian regularization back-propagation, conjugate gradient back-propagation, and the Levenberg-Marquardt algorithm.

The *back-propagation* technique [15] is commonly used to train MLPs. This technique could learn more than two layers of network. The key idea of the back-propagation technique is that the error obtained from the output layer is propagated backward to the hidden layer and is used to guide training of the weights between the hidden layer and the input layer.

[6] showed that the BP algorithm is very sensitive to initial weight selection. Prototype patterns [4] and the orthogonal least square algorithm [13] can be used to initialize the weights. The initialization of weights and biases has a great impact on both the network training (convergence) time and generalization performance. Usually, the weights and biases are initialized to small random values. If the random initial weights happen to be far from a good solution or they are near a poor local minimum, training may take a long time or become trapped there [8]. Proper weight initialization will place the weights close to a good solution, which reduces training time and increases the possibility of reaching a good solution.

[8] proposed initialization of weights using a clustering algorithm based on mean local density (MLD). This method easily leads to good performance, whereas random weight initialization leads to a wide variety of different results, many of which are poor. However, it is noted that the *best* result from random weight initialization was much better than the result obtained from the MLD initialization method.

4.2 Data Mining Strategy

Rather than streaming high resolution media content which suffers jitter or long delays due to insufficient bandwidth, smooth streaming of content using a relatively lower resolution might be preferable. Media content is available in multiple formats with different resolutions. By having knowledge of bandwidth in advance, our strategy could be to automatically switch to an appropriate resolution of media content in order to adapt to changing network traffic conditions. Another reason for predicting network traffic is to better allocate the job load to different media servers distributed in different physical locations.

The Data mining model is as follows:

1. *Raw data collection* – data is collected for training neural networks in order to predict future network traffic;
2. *Data cleaning and transformation* – The collected raw data is in the form of a time-series, and therefore needs to be transformed into multi-dimensional format prior to inputting to the neural network. Moreover, when noise is present in the data, de-noising (filtering) is usually required;
3. *Neural networks are trained according to historical data* – The performance of neural network predictors is evaluated to meet the needs of media grid QoS;
4. The trained neural network predictors are used for predicting network traffic, and for determining an adaptive streaming media strategy.

Data Collection for Network Bandwidth

In order to predict network traffic, it is important to collect historical data. Future trends can then be discovered (predicted) using various techniques, based on this historical data. In general, if the network varies dramatically day-by-day, at least two-weeks of data is preferred. In order to collect historical network traffic data for training and testing purposes, a C-based program – Info Daemon – is used to capture the server incoming and outgoing bandwidth every 2 seconds. The captured data is then passed to the analysis agent for processing and analysis. In this chapter, we focus on the analysis and prediction of the outgoing bandwidth consumption, since most traffic is caused by streams transmitted from media servers.

Data Preprocessing

Data preprocessing is critical due to its significant influence on the performance of data mining models. Data preprocessing usually includes noise elimination, feature selection, data partition, data transformation, data integration, and missing data processing.

The time-series data representing the network bandwidth can be written as:

$$Y = \{y(t)|t = 1, 2, ..., N\} \tag{5}$$

where N is the number of data records. The data are transformed into the format which are used as the inputs to neural network predictors. The one-dimensional (1D) data are transformed into multi-dimensional data, as follows:

$$Y_1 = \begin{pmatrix} y(1) & y(m+1) & y(2m+1) & \cdots \\ y(2) & y(m+2) & y(2m+2) & \cdots \\ \multicolumn{4}{c}{\dotfill} \\ y(m) & y(2m) & y(3m) & \cdots \end{pmatrix} \tag{6}$$

where m is the size of the window which divides the 1D time-series data into m-dimensional, m being determined by the user according to media streaming strategy. If a longer period of bandwidth prediction is needed, m is set as a larger number.

Including the m bandwidth rates in one vector, we need to add additional variables which are significant for prediction, and these are transformed from the m variables. They include time stamp, minimum bandwidth rate, maximum bandwidth rate, and average bandwidth of the m bandwidth rates. Thus a new datum X with $m + 4$ dimensions is generated accordingly:

$$X = \begin{pmatrix} y(1) & y(m+1) & y(2m+1) & \cdots \\ y(2) & y(m+2) & y(2m+2) & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ y(m) & y(2m) & y(3m) & \cdots \\ t & t+1 & t+3 & \cdots \\ a(1) & a(2) & a(3) & \cdots \\ m_i(1) & m_i(2) & m_i(3) & \cdots \\ m_a(1) & m_a(2) & m_a(3) & \cdots \end{pmatrix} \tag{7}$$

where $a(i)$, $m_i(i)$ and $m_a(i)$ are the average, minimum and maximum bandwidth of the vector $Y_1(i) = \{y(i*m+1), y(i*m+2), \cdots, y((i+1)*m)\}$, respectively. t is the initial time stamp, and which can be set as $t = 1$.

4.3 Performance Metrics

Several performance metrics for measuring the accuracy of a predictor exist in the literature. For instance, the coefficient of determination which is the mean squared error (MSE) normalized by the variance of the actual data is used as the performance metric in [9]. The disadvantage of the coefficient of determination lies in that the performance evaluation is inappropriate when the actual data only vary around the mean value. In [14], the maximum and mean error are used as the measurement of performance. In [5], the relative prediction error is used as the metric to evaluate predictor performance:

$$err = \frac{PredictedValue - ActualValue}{ActualValue} \tag{8}$$

The mean error and relative mean error suffer the same problem in which the performance values produced are affected significantly if there are isolated errors with a large magnitude in value.

In order to overcome the aforementioned disadvantages, we propose a new multi-level performance metric represented by the vector, $P = \{p_1, p_2, ..., p_l\}$, where l is determined empirically by the user, and reflects the level of performance metric needed. The relative prediction errors from Eq. 8 are sorted in ascending order. Assume $l = 6$. p_1, p_2,...,p_6 are the mean relative errors of the first 20%, 40%, 60%, 80%, 90%, 100% of sorted relative errors, respectively. This multi-level performance metric is represented by a multi-level relative mean error vector. l can be set empirically; in this chapter, we set $l = 6$.

The network traffic data is recorded every day. The neural network is trained according to one day's data, and then used to predict the the next day's network traffic. The data are transformed into a $(m+4)$-dimensional data set according to Eq. 6 and Eq. 7. For example, if $m = 5$, the neural network is trained to predict the average bandwidth of the next 10 seconds according to the preceding 10-second traffic (recall that the original data is recorded every two seconds).

5 Experimental System and Performance Evaluation

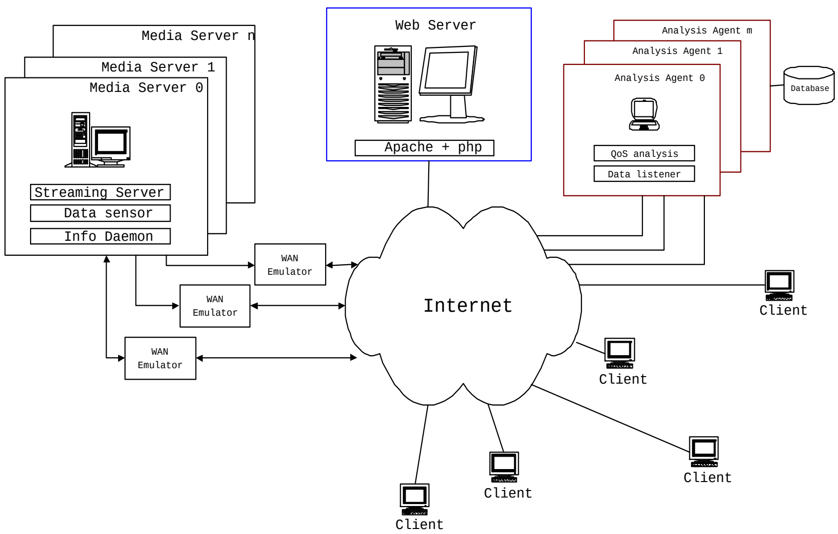

Fig. 5. Experimental system setup

5.1 System Hardware and Software

Figure 5 illustrates the prototype system which consists of analysis agents, a web service portal, media servers, and clients. The media servers are Linux PCs with Fedora Core 4, running Darwin Streaming Server version 5.5.1. Darwin Server – an open source server developed by Apple Computer Inc. – streams media to clients over industry standard protocols (such as RTP, RTSP, RTCP). To monitor certain streaming sessions, users (either the service provider or streaming clients) can query the web server and customize their measurement metrics for quality measurement and assessment. The web portal runs on the web server, which is in charge of accepting/rejecting requests for quality measurement and assessment. Once a request has been accepted by the web portal, it will be allocated to an analysis agent. In our experimental system, each analysis agent is a Java program located on a Linux machine; however, it is suitable for both MS-Windows and Linux platforms. Table 5.1 summarizes the hardware and software used in the prototype system.

Table 1. Hardware and Software

Function	OS	Mem	Number	Software
Media server	Linux	1GB	3	Darwin server, Data Sensor, Info Daemon
WAN emulator	Linux	1GB	3	Iproute2
Analysis agent	Linux	512	2	Data Collector, QoS Analyzer
Clients	WinXP	256	n	Quicktime/Realplayer/IBMToolkitForMPEG4

5.2 Request Arrival Pattern

We consider a request rate reported in [35], where a client submits requests to VWHs with various probability at different time in a 24-hour period. Requests arrive in *Poisson* distribution according to the probability described in Figure 6. There are three media servers, and 1500 users are randomly generated to stream videos from any server according to the request arrival pattern.

5.3 Results and Analysis

The data files are generated by Info Daemon located at each media server (Figure 5), which records the bandwidth incoming and outgoing traffic every two seconds. Simple noise detection is carried out by detecting those samples with the extreme values and isolating them.

The neural networks are used to predict the network bandwidth of media streaming in a grid environment. In this work, 20 neurons are used in the first layer, and 10 neurons are used in the second layer of the neural network predictor. The activation function is a sigmoid. The weights of the neural network predictors are determined automatically during the training process.

Figures 7 through 12 show the target and predicted values with m=5, 10, 15, 20, 25, 30, which represent the size of bandwidth records with 10s, 20s, 30s, 40s, 50s and 60s, respectively. For one-day data, there are originally $24 * 60 * 60/2 = 43200$ samples. The horizontal axis shows the number of data samples

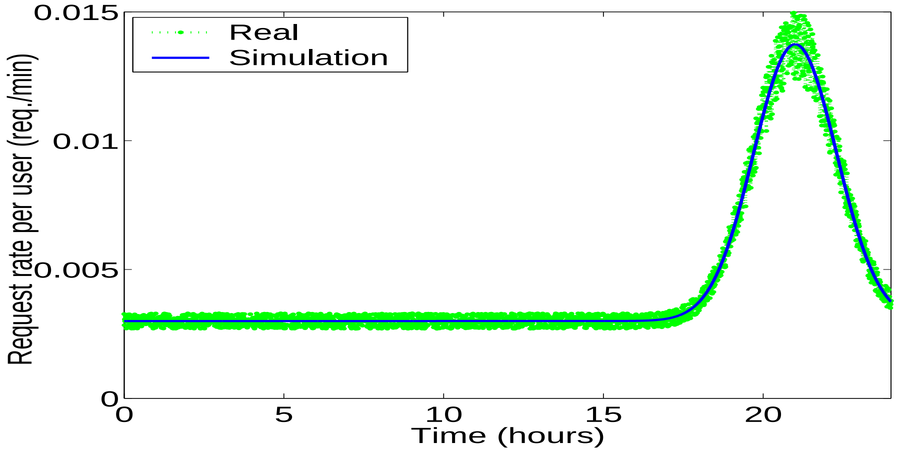

Fig. 6. Request rate per user over 24 hours.

following transformation. A different number of training samples is generated with m. For example, when $m = 5$, there are $43200/5 = 8640$ samples with $m + 4 = 9$ dimensions.

Fig. 7. Targets and predicted values with $m = 5$

Fig. 8. Same as Fig.7 with $m = 10$

Fig. 9. Same as Fig.7 with $m = 15$

Fig. 10. Same as Fig.7 with $m = 20$

Fig. 11. Same as Fig.7 with $m = 25$

Fig. 12. Same as Fig.7 with $m = 30$

From Figures 7 through 12, it is clear that a bigger window size of bandwidth records lead to a lower prediction accuracy to the mean bandwidth value. The multi-level relative mean error vectors for different values of m are shown in Table 2 and Figure 13. In the relative mean error vector, there is a sharp increase from 90% sorted relative errors to the whole relative mean error (100%). This means there are isolated relative errors of very large magnitude. This also shows that bigger window size corresponds to lower prediction accuracy.

Table 2. The mean relative prediction errors

The multi-level mean error						
m	20%	40%	60%	80%	90%	100%
5	1.540%	3.510%	5.690%	8.726%	11.302%	16.231%
10	2.302%	5.055%	8.021%	11.928%	14.786%	20.936%
15	2.622%	5.852%	9.484%	14.027%	17.143%	24.059%
20	3.464%	7.407%	11.567%	16.654%	19.848%	27.282%
25	3.771%	8.383%	13.017%	18.577%	22.198%	31.932%
30	4.632%	9.950%	15.336%	21.471%	26.068%	36.931%

Fig. 13. The multi-level prediction errors with different m

6 Conclusions

In this chapter, we presented a practical multi-agent based data analysis system which employs MLP neural networks to predict network traffic over media grids. Instead of using the mean square error and relative mean error as the performance metric for evaluating neural network predictor performance, we proposed a multi-level performance metric which presents the relative mean errors of predictors with a vector. Each item of the vector represents a certain percentage of the relative mean error of the relative prediction errors, sorted in ascending order. The metric can reflect overall performance of the predictors in a multi-level way, at the same time revealing isolated large errors. There is a lot of scope for applying data mining techniques in media grid environments. In our future work, DM techniques will be applied in real time to other media streaming tasks.

References

1. Agrawal R, Imielinski T, Swami A (1993) Database mining: a performance perspective. *IEEE Transactions Knowledge and Data Engineering* 5:914–925
2. Berry MJA, Gordon SL (2000) *Mastering Data Mining: The Art and Science of Customer Relationship Management.* Wiley, New York, NY
3. Deutsch JM (2003) Evolutionary algorithms for finding optimal gene sets in microarray prediction. *Bioinformatics* 19:45–52
4. Denoeux T, Lengelle R (1993) Initializing back propagation network with prototypes. *Neural Computation* 6:351-363
5. Eswaradass A, Sun X-H, Wu M (2005) A neural network based predictive mechanism for available bandwidth. In ??(eds) *Title* Proc. 9th IEEE Intl. Parallel and Distributed Processing Symposium, date??, place??, IEEE Press??, Piscataway, NJ??, 33a–33a
6. Kolen JF, Pollack JB (1990) Back Propagation is sensitive to initial conditions. Technical Report TR 90-JK-BPSIC, Laboratory for Artifical Intelligence Research, Computer and Information Science Department, UNiversity of ??
7. Fu XJ, Wang LP (2001) Linguistic rule extraction from a simplified RBF neural network. *Computational Statistics* (special issue on Data Mining and Statistics) 16(3):361–372
8. Geva S, Wong MT, Orlowski M (1997) Rule extraction from trained artificial neural network with functional dependency preprocessing. In ??(eds) *Title* Proc. 1st Intl. Conf. Knowledge-Based Intelligent Electronic Systems, date??, place??, publisher??, place??, 2:559–564
9. Hall J, Mars P, (1998) The limitations of artificial neural networks for traffic prediction. In ??(eds) *Title* Proc. Third IEEE Symposium on Computers and Communications, date??, place??, IEEE Press??, place??, 8–12
10. Halgamuge S, Wang LP (eds.) (2005) *Computational Intelligence for Modeling and Prediction.* Springer-Verlag, Berlin
11. Haykin S (1999) *Neural Networks: A Comprehensive Foundation (2nd ed).* Prentice Hall, Englewood Cliffs, NJ
12. Wang LP, Fu, XJ (2005) *Data Mining with Computational Intelligence.* Springer-Verlag Berlin.
13. Lehtokangas M, Saarinen J, Kaski K, Huuhtanen P (1995) Initialization weights of a multiplayer perceptron by using the orthogonal least squares algorithm. *Neural Computation* 7:982–999
14. Ramaswamy S, Gburzynski P, (1998) A neural network approach to effective bandwidth characterization in ATM networks. In *Modelling and Evaluation of ATM Networks* (ed??) publisher??, place?? 431-450

15. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. *Nature* 323:533536
16. Sun J, Li HB (2004) 3-D physical motion-based bandwidth prediction for video conferencing *IEEE Transactions Circuits and Systems for Video Technology* 14(5):pp??
17. Walsh AE (2005) The media grid: a public utility for digital media. *Dr. Dobb's J. Distributed Computing* Vol??(no??) (November):16-23
18. Li X, Hung T, Veeravalli B (2006) Design and implementation of a multimedia personalized service over large scale networks. In ??(eds) *Title* Proc. ICME = ???, date??, place??, publisher??, place??, pp??
19. Wei J, Hung T, Wentong C (2005) An information service for grid virtual organization: architecture, implementation and evaluation. *J. Supercomputing* 34(3):273–290
20. Globus Toolkit, available online at http://www.globus.org/toolkit/docs/4.0/data/ (last accessed date??).
21. Foster I, Kesselman C, (1998) The Grid: Blueprint for a New Computing Infrastructure. publisher??, place??
22. Grid Physics Network (GriPhyn) available online at http://www.griphyn.org (last accessed date??).
23. Partical Physics Data Grid (PPDG) available online at http://www.ppdg.net (last accessed date??).
24. Biomedical Informatics Research Network (BIRN) available online at http://www.birn.net (last accessed date??).
25. DataTAG available online at http://www.datatag.org (last accessed date??).
26. Apostolopoulos JG, Tan W-T, Wee SJ (2003) Video Streaming: Concepts, Algorithms, and Systems. In ??(eds.) *Handbook of Video Databases* CRC Press, place?? pp?
27. Teo KK, Wang LP, Lin ZP (2000) Wavelet multi-layer perceptron neural network for time-series prediction. In ??(eds) *Title* Proc. Proc. 10th Intl. Conf. Computing and Information – ICCI2000, Kuwait, 18-21 November, publisher??, place??, pp??
28. Ting R (1998) A multiscale Analysis and Analysis Technique for Management of Resources in ATM Networks. PhD Thesis, Department??, City University of New York, NY.
29. Turner CF, Jeremiah RM (2002) The independent wavelet bandwidth allocation algorithm. In ??(eds) *Title* Proc. Proc. Global Telecommunications Conf., place??, date??, publisher??, place??, 1:107–111
30. Wang LP, Li S, Tian FY, Fu XJ (2004) A noisy chaotic neural network for solving combinatorial optimization problems: Stochastic chaotic simulated annealing. *IEEE Transactions System, Man, Cybernetics, Part B - Cybernetics* 34(5):2119–2125
31. Wang LP, Smith K (1998) On chaotic simulated annealing. *IEEE Transactions Neural Networks* 9:716–718
32. Wang LP, Teo KK, Lin Z (2001) Predicting time series using wavelet packet neural networks. In ??(eds) *Title* Proc. Intl. Joint Conf. Neural Networks, date??, place??, publisher??, place??, 1593–1597
33. Wasserman PD (1989) *Neural Computing: Theory and Practice.* Van Nostrand Reinhold, New York, NY

34. Wolski R, Spring NT, Hayes J (1999) The network weather service: a distributed resource performance forecasting service for metacomputing. *J. Future Generation Computing Systems* 15(5-6):757-768
35. de Haar PG, et al.=????? (1997) DIAMOND project: Video-on-demand system and trials. *European Transactions Telecommunication??* 8(4):337-244

1

Resources

Your text comes here. Separate text sections

7 Key Books

Haykin S? (1999?) *Neural Networks: a Comprehensive Foundation (2nd ed).* Prentice Hall, Upper Saddle River, NJ

8 Key Survey/Review Articles

e.g. *Proc. IEEE*, *ACM Computer Surveys*, or general interest magazines like *CACM*, *IEEE Computer*, or SIG ones...

9 Organisations, Societies, Special Interest Groups

IEEE Neural Network Society (publisher of *IEEE Trans. Neural Networks*) International Neural Network Society (publisher of *Neural Network* Elsevier??) ??? *Neural Computation* MIT Press ???

10 Research Groups

11 Discussion Groups, Forums

12 Key International Conferences/Workshops

Neural Information Processing Symposium - NIPS (published as *Advances in Neural Information Processing Systems* Morgan Kaufmann, San Franscisco, CA?? Intl. Joint Conf. Neural Networks

13 (Open Source) Software

SNNS (stuttgart)??????????

14 Data Bases

UCI ML?????????????

Index