NARGES: Prediction Model for Informed Routing in a Communications Network

Hooman Homayounfard[1], Paul J. Kennedy[1], and Robin Braun[2]

[1] Centre for Quantum Computation and Intelligent Systems, School of Software, Faculty of Engineering and Information Technology, University of Technology, Sydney. PO Box 123 Broadway NSW 2007, Australia

[2] Centre for Real-Time Information Networks, Faculty of Engineering and Information Technology, University of Technology, Sydney. PO Box 123 Broadway NSW 2007, Australia

{Hooman.Homayounfard,Paul.Kennedy,Robin.Braun}@uts.edu.au

Abstract. There is a dependency between packet-loss and the delay and jitter time-series derived from a telecommunication link. Multimedia applications such as Voice over IP (VoIP) are sensitive to loss and packet recovery is not a merely efficient solution with the increasing number of Internet users. Predicting packet-loss from network dynamics of past transmissions is crucial to inform the next generation of routers in making smart decisions. This paper proposes a hybrid data mining model for routing management in a communications network, called NARGES. The proposed model is designed and implemented for predicting packet-loss based on the forecasted delays and jitters. The model consists of two parts: a historical symbolic time-series approximation module, called HDAX, and a Multilayer Perceptron (MLP). It is validated with heterogeneous quality of service (QoS) datasets, namely delay, jitter and packet-loss time-series. The results show improved precision and quality of prediction compared to autoregressive moving average, ARMA.

Keywords: Time Series Data Mining, Communications Network, Packet-Loss Prediction, Time Series Approximation, Heterogeneous Data Sources.

1 Introduction

Rapid increases in the number of Internet users and services have prompted researchers within academia and industry to contemplate smart ways of supporting applications with the required Quality of Service (QoS). Service availability is a crucial part of QoS and the network infrastructure must be designed so as to provide high availability to meet QoS. The target of 99.999% availability permits five minutes of downtime per year [1].

There are certain QoS parameters including packet delay, jitter and loss, which may be used as decision factors for online routing management. Although considerable efforts have been placed on the Internet to assure QoS within autonomous

J. Pei et al. (Eds.): PAKDD 2013, Part I, LNAI 7818, pp. 327–338, 2013.

systems, the dominant best-effort communications policy does not provide sufficient guarantee without abrupt change in the protocols [2] for the Internet. Estimation and forecasting of the end-to-end delay, jitter and packet-loss are essential tasks in network routing management for detecting anomalies.

A considerable amount of research has been done to forecast time-series with "numerical" methods. However, while dealing with large online datasets, these methods are time consuming and may not be efficient for real-time application such as multimedia streaming. Zadeh [3] suggests a transition from "computing with numbers" to the manipulation of the "human perceptions." Consequently, research projects [4,5] have started focusing on approximation of time-series with non-numerical methods. In this way, the time cost for trivial forecasting accuracy in the numerical methods may be avoided.

By prioritising Internet traffic more efficiently, QoS functions can address performance issues related to emerging Internet applications such as real–time voice and video streaming. An effective routing mechanism and its management are crucial to satisfactorily support diverse services in such networks. Routing tables, as the maps in packet delivery throughout the network, are dynamic and get updated by network state–based events. Typical network events include node failure, link failure and congestion. However, a major issue with current routing mechanisms is that they generate routing tables that do not reflect the real–time state of the network and ignore factors like local congestion.

Packet-loss in the Internet mainly occurs due to congestion in the links [6]. Real–time multimedia applications are sensitive to packet-loss, and packet retransmission is not an acceptable solution with these sorts of application. Predicting packet-loss with a certainty from network dynamics of past transmissions is crucial knowledge to inform the next generation of smart routers with better decision factors. We propose a data mining model for classification of links that have a high probability of packet-loss. The model is originally intended to contribute to making informed decisions within smart edge routers where the quality of transmissions should be controlled and is primarily determined by the level of packet-loss.

The current work extends our previous work [7]. The delay and jitter provided by a historical symbolic delay approximation model, called HDAX, is used within the proposed data mining model to predict the average number of packets lost in a link. The experiments with HDAX show better accuracy in forecasting the delay and jitter time–series as well as a reduction in the time cost of the forecasting method. To make the forecasting faster, we changed the perception–based function to a deterministic mapping function to avoid the time-cost of fuzzification and defuzzification.

We propose an informed decision-making model for routing management in a communications network, called NARGES as shown in the Fig. 1. The basic idea of our proposed model is to predict packet-loss in a network node by approximating the trends and values of the delay according to observed patterns. As shown in Figure 1, the model estimates the current trend and value assigned to the node based on the most two recent trends and values. The approximated current values of delay and jitter are then fed into a multilayer perceptron for predicting the packet-loss.

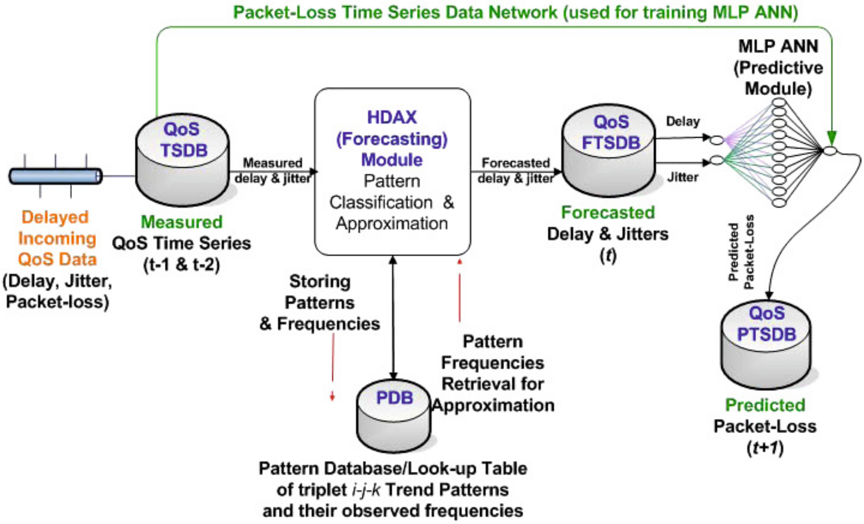

Fig. 1. Schema of NARGES Prediction Model

To evaluate the proposed loss prediction model, we used heterogeneous data. The data is categorised into ten categories in terms of the end-to-end network path and the network queuing policy. Accordingly, we ran a series of independent experiments with these different categories of datasets. The accuracy of the results, the speed of the algorithms and the cross-correlation of the forecasts of HDAX and predictions of the proposed data mining model are compared to the results from autoregressive moving average (ARMA) model.

The rest of this paper is organised as follows. Section 2 describes related work in predicting performance traces. In section 3 we describe the forecasting and predictive module of the proposed model. In section 4 the evaluation procedure is explained and later in section 5 we present results of our experiments on applying our model. Section 6 concludes the paper.

2 Related Work

This research addresses the issue of defining a prediction model based on a symbolic time-series forecasting model. The model uses historical frequencies of observed patterns in adjacent time-stamped windows within a time–series.

Originally, the research project defined to produce an "informed" [8] data mining model to be used in a smart network router for online routing management. We used a Hybrid model framework as suggested in [9]. In [10] and [11], they present ideas for the definition of a perception based function.

Packet delay and jitter show a temporal dependency with the packet-loss in the Internet [12,13]. Consequently, the research projects have studied the delay, jitter, packet-loss and other performance traces in the network so as to predict network anomalies. If packet n is lost, packet $n+1$ is likely to be lost. This can lead the network to a "bursty" packet-loss in a real–time communications network and may degrade the QoS and the effectiveness of recovery mechanisms.

A quantitative study of delay and loss correlation patterns from off-line analysis of measurement data from the Internet has been done by [14], although it did not consider real–time prediction of packet-loss from the jitter data of online communications [6].

The Rocha-Mier et al. model [9] suggests the measurement and modelling of sequences of network variables based on data network statistics. They have created a useful network scenario using OPNET Modeller. Although real network data variables could be derived from the data logs by the use of intelligent agents or manually by system administrators, there may be violation in accessing data throughout the Internet. Therefore, they adopted the modeller to study various levels of the network traffic load as well as types of services and applications.

The motivation of our work is to take a perception–based approach inspired by [10] embedded in a machine learning framework to predict network variables similarly to [9]. We validate our work using a combination of historical network traffic data and simulated data. Specifically our experiments used traces from network traffic archives generated by Napoli University "Federico II" [15] to test the impact of various network congestion levels, from "quiet" to a network with "bursty" packet-loss, on the proposed model. Simulated offline traces generated by OPNET Modeller, were used to test the impact of different queuing policies on the proposed model in longer experiments. This is a standard approach in the networking domain.

3 Proposed Algorithm

The NARGES model, as presented in Figure 1, is a hybrid model that predicts the packet-loss at time $t+1$ based on the approximated values of delay and jitter at time t. The delay and jitter forecasts at time t, are approximations of the current values of the time-series forecasted by HDAX model according to the historical trends and values of the corresponding time-series in the previous "two" periods, i.e. $t-1$ and $t-2$. The following sections describe the two modules: HDAX and a multilayer perceptron, as the two constitutive parts of NARGES model.

3.1 Forecasting Module: HDAX

In this section, we briefly describe our approach to forecasting time-series values from previously observed patterns of delay and jitter. We use a mapping function for the definition of the patterns for time-series approximation, which is different from what we presented in our previous work [7].

The HDAX algorithm is defined based on the model definition suggested by Tresp [16].

$$y_t = f(y_{t-1}, y_{t-2}, \ldots, y_{t-N}) + \epsilon_t \tag{1}$$

where f is either known or approximated sufficiently well by a function approximator and ϵ_t is the zero-mean noise with probability density $P_\epsilon(\epsilon)$, which represents dynamics that are not modelled. Let y_t be the value of the discrete time-series at time t and Y_t the trend of two consecutive values at times $t-1$ and t. We make the underlying HDAX model of the time series with order $N = 3$ as

$$y_t = f(y_{t-1}, y_{t-2}, y_{t-3}) + \epsilon_t \tag{2}$$

We represent possible future trends of the QoS time series y_t of delay and jitter values at time intervals $t-1$ and $t-2$ with categorical terms from the

$$Alphabet = \langle SI, I, P, D, SD, OUT \rangle \tag{3}$$

where these symbols are defined in Table 1. The basic trends in Table 1 are defined with linguistic variables based on a deterministic "mapping function." Within the mapping function, each of the categorical terms maps an interval on the domain of real numbers to a linguistic representative.

Table 1. The scale of time-series trends used for mapping numerical values to the trends. Note that we use y_t to denote the time series value at time t and Y_t to denote the difference of the two consecutive values at time $t-1$ and t. For simplicity, the linear scale in our experiment also has six linguistic grades (defined in Eq 3) each of which is a categorical term (assigned to the case number of zero to five respectively).

Case	Id	Description	$Y_t = y_t - y_{t-1}$
0	P	Plain	$Y_t = 0$
1	I	Increase	$0 < Y_t \le \frac{max}{2}$
2	SI	Sharply Increase	$\frac{max}{2} < Y_t \le max$
3	D	Decrease	$-\frac{max}{2} \le Y_t < 0$
4	SD	Sharply Decrease	$-max \le Y_t < -\frac{max}{2}$
5	OUT	Outlier	$\lvert Y_t \rvert > max$

We define "*previous-current-next*" patterns (also called the $i-j-k$ patterns) with a combination of three consecutive trends, at times $t-2$, $t-1$ and t as shown in Figure 2. Our approach consists of two phases: training and simulation. The max in Table 1 is the maximum value of a time-series, shown later with the dashed line in Fig. 2.

HDAX Training. Figure 2 shows a "sliding window" that moves over the training data and makes patterns consisting of three consecutive trends, i, j and k, at times $t-2$, $t-1$ and t (previous, current and next) together with

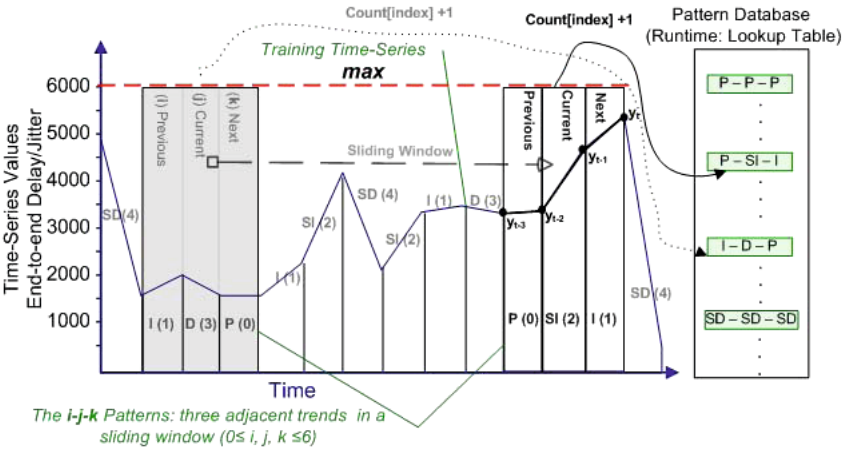

Fig. 2. The training phase uses a time series dataset values to recognise $i-j-k$ patterns and train the look–up table that maps each of these patterns to a respective frequency. The table is then used for forecasting the k trend at time $t+1$ in the simulation phase.

their frequency in the look–up table. The look–up table is then used in the simulation phase to approximate the next trend, at time t, and the associated delay/jitter value from the current and previously observed trend patterns, where i, j and k are the respective indices for previous, current and next trends ($0 \leq i, j, k \leq 5$) and $F(i, j, k)$ is their respective frequency. From the look–up table, the probability of $i - j - k$ patterns is estimated as

$$\bar{P}(i, j, k) = \frac{F(i, j, k)}{N_k}, \quad N_k = \sum_{k=0}^{5} F(i, j, k). \tag{4}$$

where i, j and k are the indices for the respective patterns at times $t-1$, $t-2$ and t, and N_k is the number of total observations for all $i-j-k$ patterns.

HDAX Simulation. Based on the most frequently observed patterns in the last two consecutive trends at time $t-1$ and $t-2$, the HDAX algorithm uses the estimated conditional probability to approximate the trend at time t. The Y_t is a trend value at time t in the time-series of trends, defined based on the last two trends seen at times $t-1$ and $t-2$. Formally speaking, we estimate the $P_k(i, j)$ as follows

$$\bar{P}_k(i, j) = \bar{P}(Y_t = k | Y_{t-1} = i, Y_{t-2} = j) + P_\epsilon(\epsilon) \tag{5}$$

where i and j are the indices of the observed trends at times $t-1$ and $t-2$, respectively. The trend at time t, with the index k, may take six possible values from the alphabets in Eq. 3. Based on this, the look-up table is used to forecast the next trend and value of the time-series based on the trend with highest frequency in this table:

$$i = t - 1, j = t - 2 : \hat{Y}_t = \arg\max_k (\bar{P}_k(i, j)) \quad (6)$$

With the trend k is estimated, the $\widehat{y_t}$ can be approximated based on time step-size between y_{t-1} and y_t and the slope of the line at y_{t-1}.

3.2 Predictive Module: Multi-layer Perceptron

The predictive module calculates the average packet-loss. As described above, the outputs of HDAX are approximations of the values of the network traces at time t. They are used within a multilayer perceptron (MLP) to get better precision for real-time packet-loss prediction at time $t + 1$.

The MLP is a feed-forward network with back-propagation learning rule and one hidden layer. As shown in Fig. 1, the MLP has two input layer nodes and forecasts delay and jitter. It is designed and implemented using the MATLAB neural network toolbox.

Optimum parameter values for number of hidden layer neurons, learning rate and momentum was defined empirically by using a training data-set and choosing the parameter values with the highest test accuracy. The network was tested with between 1 and 100 neurons in the hidden layer. Based on these experiments, ten hidden layer neuron were chosen.

4 Experimental Evaluation

The section describes experiments for evaluating the accuracy, speed and quality of the output of our model and ARMA. We implemented ARMA algorithms based on [17]. The results of HDAX and NARGES are compared with ARMA results. In the experiments, the accuracy of algorithms was compared using normalised root mean square error (NRMSE). Performance was compared using the run time. Prediction quality was evaluated with the normalised correlation coefficient with MATLAB's cross-correlation function.

4.1 Datasets

As mentioned above, the data consists of three QoS traces of delay, jitter and packet-loss. For each experiment three sets of data are considered: the original data generated by D-ITG or OPNET, the output of HDAX (or NARGES) and the output of ARMA. Each dataset is divided into training and test datasets (25% and 75% respectively).

The results were categorised according to datasets used for the experiments in two ways: (i) according to the end-to-end path and (ii) according to the queuing policy used. Forty-six datasets were used for computing results. Each dataset includes time-stamped traces of delay, jitter and loss values. There are

Table 2. Characteristics of the end-to-end paths for the data obtained from D-ITG (Dataset Categories: 1-7) and queuing policies for the data obtained from OPNET Modeler (Dataset Categories: 8-10)

Dataset Category	Access Networks (e2eP)	Operating Systems	User Device
1	ADSL-to-Ethernet	Linux-to-Linux	PC-to-Workstation
2	GPRS-to-Ethernet	Windows XP-to-Linux	Laptop-to-Workstation
3	UMTS-to-Ethernet	Windows XP-to-Linux	Laptop-to-Workstation
4	Ethernet-to-ADSL	Linux-to-Linux	Workstation-to-PC
5	Ethernet-to-GPRS	Linux-to-Windows XP	Workstation-to-Laptop
6	Ethernet-to-UMTS	Windows XP-to-Windows XP	Workstation-to-Laptop
7	Ethernet-to-WLAN	Linux-to-Windows XP	Workstation-to-Laptop
	Queuing Policy	Description	QoS Enabled
8	WFQ	Weighted Fair Queuing	Yes
9	FIFO	First in First out	No
10	PQ	Priority Queuing	Yes

36 datasets from D-ITG with an average 3000 values in each of delay, jitter and packet-loss time-series. There are also 10 datasets generated by OPNET, 9 with 48000 values and one with 1,000,000 values.

The data generated by D-ITG are gathered in two ways: an archive from the University of Napoli [15] and data probed over a University network test-bed. Samples were obtained by sending probe packets with a packet rate of 100 packets per seconds. The measurement unit of the delay and jitter is either milliseconds or nanoseconds while for packet-loss the value represents the average number of lost packets in the window of the times. The network test-bed was formed by two nodes on the University LAN: a laptop with 1.70 GHz processor, IntelPro 2200BG network connection and a node on a HPC Linux Cluster running Red Hat Enterprise Linux 6 (64bit) with processor rating between 3.06-3.46GHz and a gigabit inter-node connection. An end-to-end path (e2eP) definition is considered for the datasets obtained from D-ITG as shown in Table 2.

We used OPNET datasets to study the effect of different packet transmission policies, longer experiments and network congestion states on the performance of the implemented data mining model. The selection of service discipline in the routers (FIFO, WFQ and PQ) can affect VoIP applications and link congestion. Thus, the performance of the prediction model is evaluated using these datasets. Simulations ran on a laptop with 1.70 GHz processor with IntelPro 2200BG network connection and OPNET Modeler 15.0.A.

5 Results

Three methods were used analysis: accuracy, speed and correlation analysis for the proposed models versus ARMA. Friedman's test [18] is used to rank algorithms and to test the hypothesis of the similarity of the algorithms based on Holm's test. The $p-value$ is used to reject or accept the above "null hypothesis."

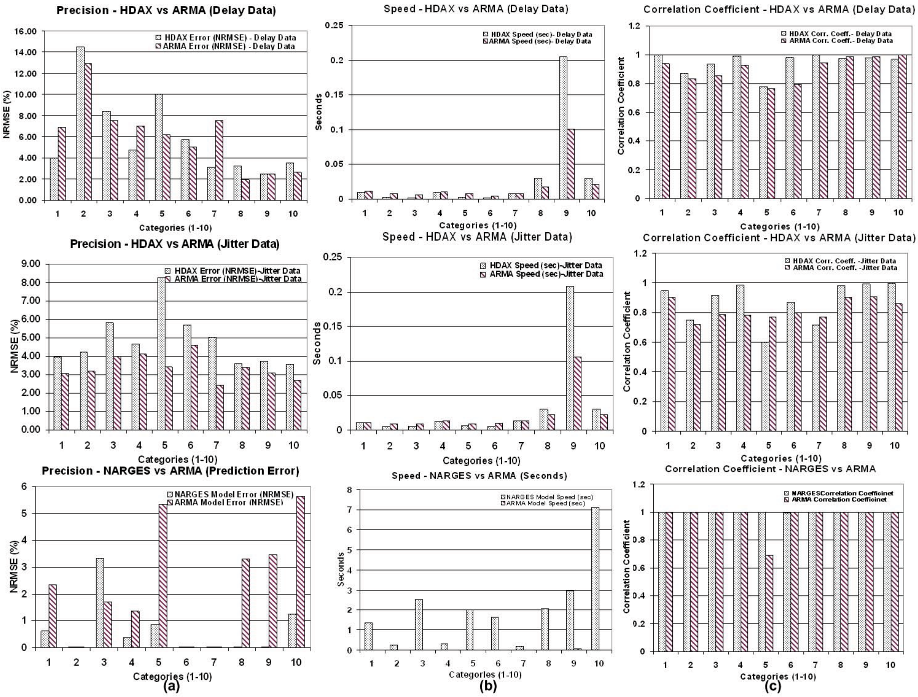

Fig. 3. Error (NRMSE) of HDAX and NARGES vs ARMA together with speed of algorithm and cross-correlation coefficients are shown in the column (a) to (c), respectively. The first and second rows are the HDAX results and the last row shows the whole model (NARGES) results. In the twin bar charts, the left "gray" bars shows HDAX (in the first two rows) and NARGES (in the last row) while the right bar filled with wide downward diagonal pattern denotes ARMA outcomes.

5.1 Model Performance

The accuracy and speed of NARGES model and its HDAX subcomponent was examined by comparing the NRMSE between the model outputs and the original traces. Figure 3, shows the results of HDAX for delay and jitter traces as well as the NARGES results over packet-loss traces in a top down order.

The average precision of the forecasted delay and jitter time-series of HDAX and ARMA as well as the NARGES precision for predicted packet-loss are shown within column (a) in Figure 3. Column (b) shows a comparison between the speed of the models with ARMA. In terms of similarity measurement between the original datasets and the output by HDAX, NARGES and ARMA, the maximum similarity of normalised cross-correlations was used for correlation analysis between the time-series. Column (c) in Figure 3 shows the respective correlation coefficient between the output of the models and the original data.

The predicted average packet-loss for NARGES was compared to ARMA. As the Fig. 3 shows, NARGES predicts more precisely than ARMA with OPNET

Table 3. Holm / Hochberg Table for $\alpha = 0.05$ and $z = (R_0 - R_i)/SE$. Note that in testing the algorithms for accuracy (ACC), speed (SPD) and cross-correlation (CCF) over Delay (D), Jitter (J) and Packet-Loss (L) models printed in **bold** are statistically significantly better.

	D-ITG Datasets			OPNET Datasets		
Data/Test	algorithm	z	p	algorithm	z	p
D/ACC	**HDAX**	3.0000	0.0027	**HDAX**	3.1628	0.0016
D/SPD	**HDAX**	1.9999	0.0455	**ARMA**	3.1623	0.0016
D/CCF	**ARMA**	5.3333	9.6×10^{-8}	**HDAX**	2.5298	0.0114
J/ACC	HDAX	1.0000	0.3173	HDAX	1.4×10^{-15}	0.9999
J/SPD	HDAX	1.3333	0.1824	**HDAX**	3.1623	0.0016
J/CCF	ARMA	0.9999	0.3173	**ARMA**	3.1623	0.0016
L/ACC	NARGES	0.4999	0.6171	**NARGES**	3.1623	0.0015
L/SPD	**ARMA**	6.0000	2.0×10^{-9}	**ARMA**	3.1623	0.0016
L/CCF	NARGES	2.7×10^{-15}	0.9999	**NARGES**	3.1623	0.0016

datasets but is slower. This is because NARGES has more modules and processes more data than ARMA to predict the final packet-loss values. The training time of the MLP module accounts for the longer time taken to run our model.

5.2 Model Ranking

Friedman test was run and the stored statistic used to calculate the Holm's test $p-value$, which is a decision factor for rejecting or accepting the null hypothesis. It also calculates the average ranking of the algorithms used in each test.

Friedman's test is a non-parametric equivalent to the parametric repeated measures ANOVA test. It computes the ranking of the measured outputs for an algorithm with other algorithms, assigning the best of them the ranking 1 and the worst the ranking k. According to the null hypothesis, it is supposed that the results of the algorithms are equivalent and the rankings are also similar.

A similarity test between the precision, performance (speed) and the correlation of the output of the models with the original test data is performed via nonparametric Holm tests. The tests were conducted over the results collected from the runs of HDAX, ARMA and NARGES models with the three traces of delay, jitter and loss within each datasets. We have done this to rank the algorithms and test the similarity hypothesis between HDAX and NARGES models and ARMA. The Holm tests are testing the similarity of algorithms accuracy, speed and correlation coefficient for delay, jitter and packet-loss time-series.

In Table 3, the algorithm shown in each row works significantly better than ARMA if their corresponding average ranks differ by at least the critical difference, which are the ones with $p-values \leq 0.05$. In Table 3, the algorithm name showed in each row are taken as the better when the null hypothesis is rejected.

The Holm's tests shows that HDAX ranking is better than ARMA for the precision of the results and the speed of the algorithms whereas ARMA ranking is better than HDAX for cross-correlation between the forecasted and original time-series. Moreover, according to the $p-values$, HDAX forecasts significantly better and faster than ARMA for delay traces while ARMA has more correlated outputs in comparison to the original delay data in the 36 runs for the D-ITG datasets in section 4.1. Two algorithms, HDAX and ARMA, perform the same in forecasting jitter values because Holm's tests accept all null hypothesis. For the whole model outputs accuracy and quality of the predictions, NARGES predicts the packet-loss the same as ARMA does, although NARGES ranks higher. In terms of the speed of the models, the $p-value$ of the ARMA speed is less than the significance level ($\alpha = 0.05$).

The Holm's test for OPNET datasets, explained in section 4.1, shows that HDAX forecasts significantly more accurate than ARMA over the delay datasets and has a better quality in terms of the cross-correlation between its forecasts and the original data. HDAX is faster over jitter datasets while ARMA forecasts faster over delay datasets. Unlike the tests with D-ITG datasets, the NARGES outputs with OPNET datasets show significantly better precision and quality of predictions. It means that our model shows better precision and prediction compared to ARMA in longer experiments. In terms of the speed of the models, the $p-value$ of the ARMA speed is less than the significance level ($\alpha = 0.05$).

Currently network routers must send information between routers to inform about the peer status. The results in this paper demonstrate, in a simulated setting at least, that a data mining agent can predict the peer status to reduce or perhaps eliminate the unnecessary network data transmission overhead and the time required for sending and receiving repeated packets.

6 Conclusions

This paper presented a packet-loss prediction model based on our earlier work [7]. We designed and implemented a hybrid data mining model, NARGES, which is using the forecasted current values of delay and jitter, to predict the future packet-loss rate assigned to a network node. We used a non-numerical approach to predict packet-loss in multimedia streams by observing the delay and jitter time-series. The Model is validated with heterogeneous QoS traces and the results show that the quality and the precision of the proposed model is significantly better than AMRA. However, NARGES was slower than ARMA because it has to process more inputs. As can be seen from the competing speed of HDAX module, it is the training time of the MLP module that degrades the speed of the model. In Table 3, the significant difference between the $p-value$ of the Holm's tests for the D–ITG and the longer OPNET datasets implies that our model can work faster in a real–time network experiment.

References

1. Torell, W.: Network-critical physical infrastructure: Optimizing business value. In: Twenty-Seventh International Telecommunications Conference, INTELEC 2005, pp. 119–124 (September 2005)

2. Floyd, S., Allman, M.: Comments on the usefulness of simple best-effort traffic. Request for Comments 5290, IETF (July 2008)
3. Zadeh, L.A.: From computing with numbers to computing with words from manipulation of measurements to manipulation of perceptions. Annals of the New York Academy of Sciences 929(1), 221–252 (2001)
4. Keogh, E., Lin, J., Fu, A.: Hot SAX: Efficiently finding the most unusual time series subsequence. In: ICDM 2005, Houston, pp. 27–30. IEEE (2005)
5. Batyrshin, I.Z., Sheremetov, L.B.: Perception–based approach to time series data mining. Applied Soft Computing Journal 8(3), 1211–1221 (2008)
6. Roychoudhuri, L., Al-Shaer, E.: Real–time packet loss prediction based on end–to–end delay variation. IEEE Trans. Network Service Manager 2(1) (2005)
7. Homayounfard, H., Kennedy, P.J.: HDAX: Historical symbolic modelling of delay time series in a communications network. In: Kennedy, P.J., Ong, K., Christen, P. (eds.) AusDM 2009. CRPIT, vol. 101, pp. 129–138. ACS, Melbourne (2009)
8. Debenham, J., Simoff, S., Leaney, J., Mirchandani, V.: Smart communications network management through a synthesis of distributed intelligence and information. In: Artificial Intelligence in Theory and Practice II, pp. 415–419 (2008)
9. Rocha-Mier, L.E., Sheremetov, L., Batyrshin, I.: Intelligent agents for real time data mining in telecommunications networks. In: Gorodetsky, V., Zhang, C., Skormin, V.A., Cao, L. (eds.) AIS-ADM 2007. LNCS (LNAI), vol. 4476, pp. 138–152. Springer, Heidelberg (2007)
10. Miloucheva, I., Hofmann, U., Gutiérrez, P.A.A.: Spatio-temporal QoS pattern analysis in large scale internet environment. In: Ventre, G., Canonico, R. (eds.) MIPS 2003. LNCS, vol. 2899, pp. 282–293. Springer, Heidelberg (2003)
11. Batyrshin, I., Panova, A.: On granular description of dependencies. In: Proc. 9th Zittau Fuzzy Colloquium, Zittau, Germany, pp. 1–8 (2001)
12. Jiang, W., Schulzrinne, H.: Modeling of packet loss and delay and their effects on real-time multimedia service quality. ACM Network and Operating Systems Support for Digital Audio and Video (2000)
13. Markopoulou, A., Tobagi, F., Karam, M.: Loss and delay measurements of internet backbones. Computer Communications 29(10), 1590–1604 (2006)
14. Moon, S., Kurose, J., Towsley, D.: Packet audio playout delay adjustment: performance bounds and algorithms. Multimedia Systems 6(1), 17–28 (1998)
15. Botta, A., Pescapé, A., Ventre, G.: Quality of service statistics over heterogeneous networks: Analysis and applications. European Journal of Operational Research 191(3), 1075–1088 (2008)
16. Tresp, V., Hofmann, R.: Nonlinear time-series prediction with missing and noisy data. Neural Computation 10(3), 731–747 (1998)
17. Chatfield, C.: Time-series forecasting. Chapman and Hall (2001)
18. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180(10), 2044–2064 (2010)