International Journal of Advances in Applied Sciences (IJAAS)
Vol. 6, No.4, December 2017, pp. 283~292
ISSN: 2252-8814

Comparative Study of Various Neural Network Architectures for MPEG-4 Video Traffic Prediction

J. P. Kharat*
DKTE Ichalkaranji, India

Article Info

Article history:

Received Sep 15, 2017
Revised Nov 10, 2017
Accepted Nov 20, 2017

Keyword:

Cascaded feed forward neural
Feed forward neural network
Linear prediction
MPEG-4 video traffic
network
Nonlinear prediction
Time delay neural network

ABSTRACT

Network traffic as it is VBR in nature exhibits strong correlations which make it suitable for prediction. Real-time forecasting of network traffic load accurately and in a computationally efficient manner is the key element of proactive network management and congestion control. This paper comments on the MPEG-4 video traffic predictions evaluated by different types of neural network architectures and compares the performance of the same in terms of mean square error for the same video frames. For that three types of neural architectures are used namely Feed forward, Cascaded Feed forward and Time Delay Neural Network. The results show that cascade feed forward network produces minimum error as compared to other networks. This paper also compares the results of traditional prediction method of averaging of frames for future frame prediction with neural based methods. The experimental results show that nonlinear prediction based on NNs is better suited for traffic prediction purposes than linear forecasting models.

Corresponding Author:

J. P. Kharat,
DKTE Ichalkaranji, India.

1. INTRODUCTION

Many multimedia applications, such as video-conferencing and video-based entertainment services, rely on the efficient transmission of live or stored video. These videos are compressed before transmission in order to reduce the size of videos. For this purpose MPEG-4(Moving Picture Expert Group) video compression standard is used. This MEG-4 standard generates variable bit rate traffic which is very bursty in nature, due to the frame structure of the encoding scheme and natural variations within and between scenes [1-7]. If one has to transmit this traffic over network, this traffic will be transmitted by peak rate. But as MPEG-4 traffic is VBR in nature, I-frame is large in size as compared to P and B frame sizes. Hence efficient utilization of bandwidth will not take place. Also Variable-bit-rate (VBR) traffic complicates the design of efficient real-time storage, retrieval, transport, and provisioning mechanisms capable of achieving high resource utilization

To overcome this problem, generally VBR traffic is smoothed. For smoothing of traffic, various approaches have been suggested. Mainly there are two types: Linear Method and Non-linear method [8].
In linear method, various approaches have been suggested. Few of them are: The first approach is to convert VBR to CBR. In this approach, first few frames are taken from source in buffer.

Then the average of these frames is taken. Finally frames will be transmitted at this average rate. But this scheme introduces finite delay in playback at receiver. Also there are chances that few frames might get lost due to buffer fullness, and may affect the quality of picture and hence QOC.

The second approach is as frames come from the source, the rate is updated per frame. But this approach also introduces playback delay at receiver. The most common nonlinear forecasting methods involve neural networks (NN) [9-11]. Although some articles state that linear prediction models are unable to describe the characteristics of network traffic [11], other studies confirm the practical usability of linear predictors for real-time traffic prediction [12]. Thus, it remains unclear which predictors provide the best performance, being in the same time simple, adaptable and accurate.

The rest of paper is organized as below: Section I describes the three different types of neural architectures used for prediction. Section II focuses on the training and Testing of Neural networks. Section III deals with simulation results of three architectures. Section IV presents the comparative analysis of all the three architectures. Section V compares the Neural Network results with the conventional methods and finally, section V makes the final conclusion.

2. NEURAL NETWORK

A NN has multiple interconnected processing elements grouped into layers. Each layer has several nodes. The inputs to one node in a layer are the outputs of all other nodes in the previous layer. The nodes algebraically sum these weighted signals and pass them through a nonlinear squashing function to produce a net output. The function is usually a sigmoid function or a hyperbolic tangent. Based on the structure of the network or the way the nodes are interconnected, there are two broad categories of NNs; the feed forward multi-layer perceptron (FMLP) and the recurrent multi-layer perceptron (RMLP). FMLP is different from RMLP in the sense that there is no cross talk between the nodes of a given layer. Each layer in a multilayer neural network has its own specific function. The input layer accepts input signals from the outside world and redistributes these signals to all neurons in the hidden layer. The input layer rarely includes computing neurons, and thus does not process input patterns. The output layer accepts output signals, a stimulus pattern, from the hidden layer and establishes the output pattern of the entire network. Any continuous function can be expressed with one hidden layer. Two hidden layers can predict discontinuous functions too [13].

2.1. Feedforward Neural Network

The network is composed of an input layer, a series of hidden layers and an output layer. In this network, the signals from each node are transmitted to all the nodes in the next layer, and only the hidden layers have a sigmoid-type discriminatory function. In this work, a hyperbolic tangent has been used as the discriminatory function. The input and the output layers have linear discriminatory functions and the input layer has no biases. FMLPs with appropriate signals in the input layer are good at approximating static nonlinearities, i.e. memory-less nonlinear functions. Each of the processing elements of an FMLP network is governed by the following equation.

$$x_{ij}=\sigma_{[L,i]}\left(\sum_{j=1}^{N_{|t-1|}} w_{[l-1,j][l,j]}\, x_{[l-1,j]} + b_{[l,j]}\right) \tag{1}$$

Where x [l, i] is the i^{th} node output of the l^{st} layer for sample t, w [l−1, j] [l, i] is the weight, the adjustable parameter, connecting the j^{th} node of the (l − 1) th layer to the i^{th} node of the l^{th} layer, b [l, i] is the bias, also an adjustable parameter, of the i^{th} node in the l^{th} layer.

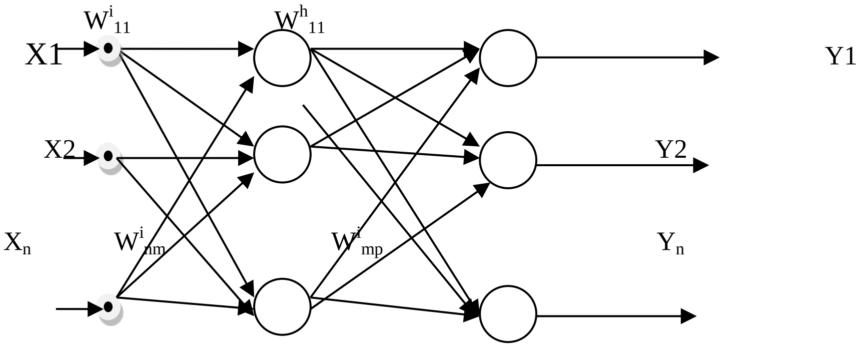

Figure 1. Feed-forward Neural Network

2.2. Cascade-Forward Neural Network (CF)

This network is a Feed-Forward network with more than one hidden layer. Multiple layers of neurons with nonlinear transfer functions allow the network to learn more complex nonlinear relationships between input and output vectors. This network can be used as a general function approximator. It can approximate any function with a finite number of discontinuities, arbitrarily well, given sufficient neurons in the hidden layer. Figure 2 shows the architecture of a cascade-forward Network with two hidden-layers.

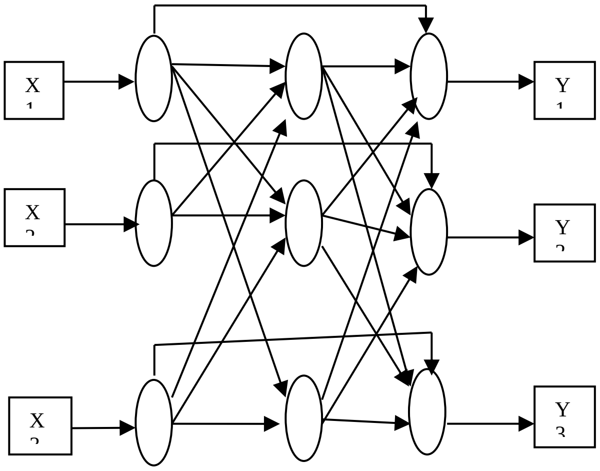

Figure 2. Cascaded Feed-Forward Neural Network

2.3. Feed-Forward with Tapped Time Delays (FFTD)

A tapped delay line can be used with linear neurons to modify only the input layer (to allow for delayed inputs). The tapped delay line sends the current signal, in addition to a number of delayed versions, to the weight matrix. The rest of the network, beyond the input layer, is the same as the feed-forward network. Figure 3 shows a tapped delay line applied to a single input. The same concept can be applied to all network inputs.

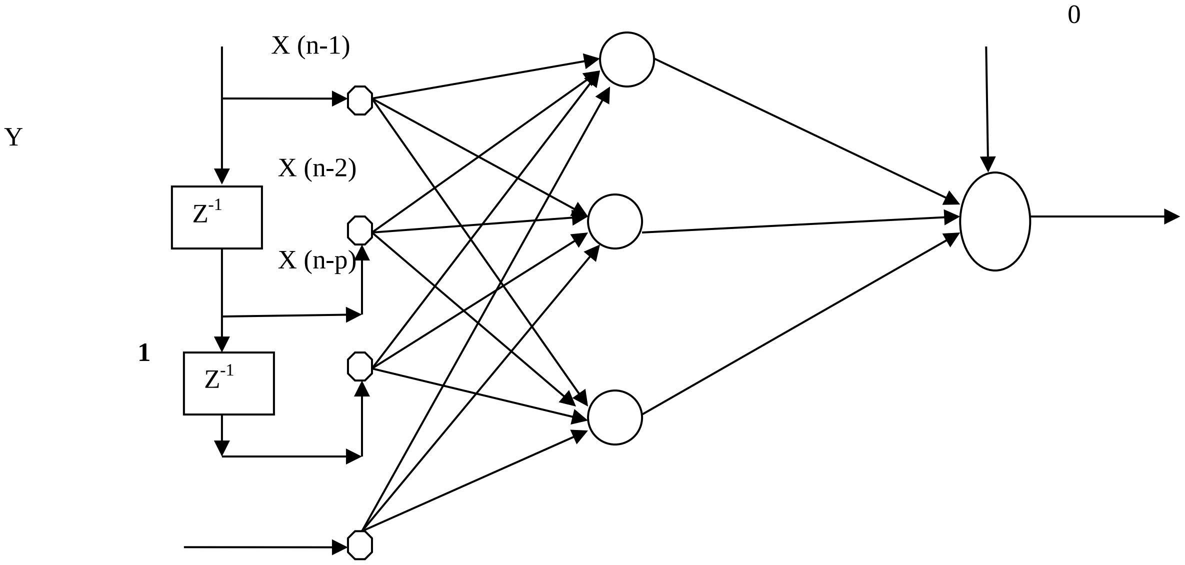

Figure 3. Feed-Forward with Tapped Time Delays (FFTD)

3. PREDICTION OF MPEG-CODED VIDEO TRACES

3.1 Approach

Neural networks are generally considered to be one of the most effective tools for prediction. Due to their analogy with biological neural networks (human brain), they seems to be suitable to solve prediction related tasks. Our predictions were based on taking *N* previous patterns to predict one following pattern.

Since we used a supervised learning paradigm, our networks worked with desired values of target patterns during the training process.

3.2. Methodology Used

For neural networks, it is not a challenge to predict patterns existing on a sequence with which they were trained. The real challenge is to predict sequences or movies that the network did not use for training. However, the part of the sequence to be used for training should be “rich enough” to equip the network with enough power to reconstruct or extrapolate patterns that may exist in other sequences or movies. This requires a proper selection of the movie to be used for training in addition to the proper selection of the number of training points. Then, the trained networks are tested with a different portion of the training movie in addition to a number of segments from the remaining movies (not used during training). The other issue that needs to be addressed is how long the training sequence should be in order to capture “just enough” useful patterns from the training movie. This issue is important because a longer-than-necessary training sequence will give good results for the movie used during training and poor results for the remaining movies.

For all predictors designed in this study, only a segment of the Jurassic Park video sequence is used for training and cross-validation, to determine the stopping point for the training process. After fixing all of the network parameters, the developed predictors are tested on the remaining segment of the Jurassic Park. All time-series are scaled by a single scaling factor so that they lie mostly in the range from 1 to -1, making them suitable for processing by the neural network. Some of the details of the video data traces used in the current research, along with the segments used in training and cross-validation, are as follows:

Training Sequence consist of first 3000 frames from movie Jurassic park encoded in high quality mode. Testing Sequence consists of First 1000, Middle 1000 and Last 400 frames from the same movie sequence. In order to achieve good results, probably one of the most important problems is to choose the appropriate configuration of neural network. During training of MLPs, we tried many types of configurations for our predictions. There were notable differences of prediction errors among them. We chose the RMSE (root mean square error) as an objective criterion to compare them. We made experiments with the number of input neurons changing from 1 to 20 and also experiments with various number of hidden neurons and number of hidden layers. We achieved the best results of training the MLP network using network configuration 20-15-10-1 (which means: 20 input neurons, 10 and 15 neurons in hidden layer, 1 output neuron), Levenberg-Marquardt training algorithm and learning-rate parameter 0.001.We also made experiments with the learning rate and number of iterations while training the network. We got the best result for Lr (l earning rate)=0.01 and Epochs=1500.Also we tried the prediction for various window sizes. The window size variation, we preferred is 5 to 12. The results of the prediction for the training and test set are shown in next section.

3.3. Performance Metrics

Here we define the performance metrics used to compare the performance of the different models developed in this research. Three types of errors can be used as performance metric for the prediction schemes developed in this work. The three performances metric are defined as:

a. Mean Square Error (MSE): MSE is the ratio between the sum of the square of the prediction error and the sum of the square of the input data. It is represented by the following Equation:

$$\text{MSE} = \frac{\sum_{j=1}^{N}\left(x_{MA}(j)-x_{\widehat{MA(J)}}\right)^{2}}{\sum_{j=1}^{N}x_{MA}(J)2} \times 100 \tag{2}$$

Where N is the length of the moving average time-series, X_{MA} is the actual size of the j-th element of the moving average time-series and X_{MA} is the prediction of the j-th element. MSE is an indicator of the overall quality of the prediction.

b. Maximum Absolute Error (MAE): MAE is the maximum error between the actual moving average of the VOP sizes and the predicted moving average of the VOP sizes. It is given by the following Equation:

$$MAE = \max_{1\leq j\leq N}\left|x_{MA}(j) - \widehat{x_{MA}}(j)\right| \tag{3}$$

MAE is the maximum prediction error and provides the information about the worst case of failure of the prediction model.

c. Maximum Relative Error (MRE): MRE is the maximum of the ratio between the prediction error and the actual input data and is given by the equation:

$$\text{MRE}= \max_{1\le j\le N} \frac{|x_{MA}(j)-\widehat{x_{MA}}(j)|}{\left|x_{MA(j)}\right|} \tag{4}$$

MRE is a measure of the relative comparison between the prediction error and the corresponding actual moving average value of the VOP size.

3.4.Scaling of the Data

For the purpose of non-linear modeling using NNs, the input data must be scaled to lie within certain bounds. The scaling of input data is a very important aspect of training the network. In order to prevent the saturation of nodes in NNs, the input data is forced to lie between −1 and 1.

4. SIMULATION RESULTS

4.1. Feed-forward Neural Network

a.Prediction Results for Various Window Sizes

In this section we will discuss the simulation results. In this study 3 neural networks were tested for the same input patterns. We have taken the previous frame sizes to predict future single frame size. This can be referred as single step ahead prediction. The number of previous data points used to predict the next data point is termed as Window Size. For our experimentation we have varied the window size from 5 frames to 12 frames and compared the results for various architectures. In this work, five empirical MPEG-4 video traces generated by Fitzek et al. and available on the public domain [14] are used.

Figure 4. represents the training set for 12 window size. Here 12 window sizes represents that first 12 frames in the sequence have been taken to predict the 13th frame in the sequence. Such first 3000 frames from the movie Jurassic Park have taken as a training set to train the neural network. From Figure 4 it is clear that the predicted frames and actual frames are very much closer. In this case the prediction error is very small.

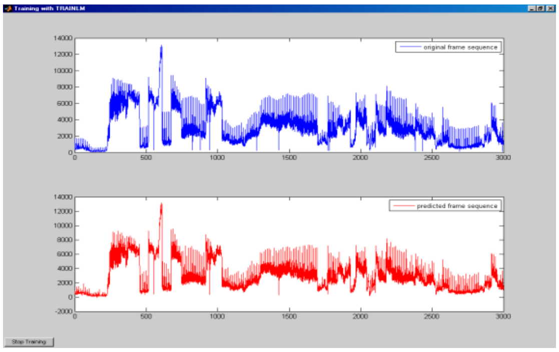

Figure 4. Training Sequence for 12 Window Size

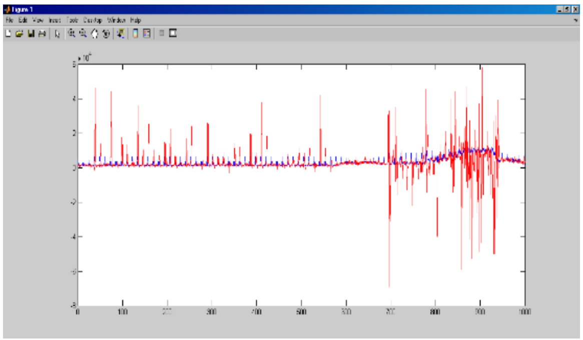

Figure 5. Test Sequence for First 1000 Frames

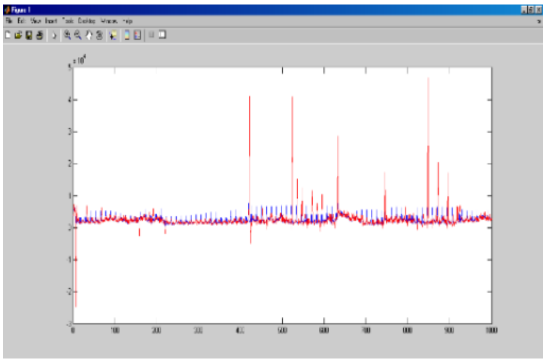

Figure 6. Test Sequence for Middle 1000 Frames

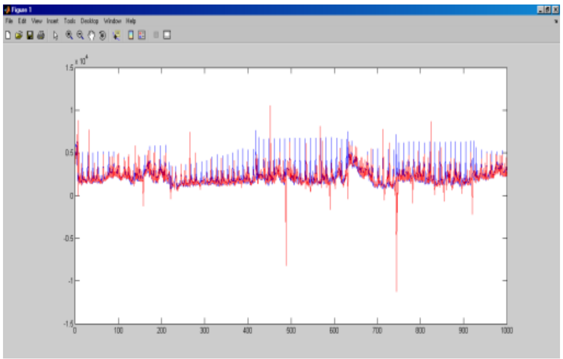

Figure 7. Test Sequence for Last 400 Frames

Figure 5 to 7 represents the test results. The different frame sequences from the same movie have been taken to test the trained network. Figure 5 represents the test result for first 1000 frames in the movie “Jurassic Park”. The Red indicates predicted frames while blue indicates original frame sequence. Figure 6 represents the test result for the middle 1000 frames in the movie “Jurassic Park”. Figure 7 represents the test results for the last 400 frames in the movie “Jurassic Park”.

In the same way the network is evaluated for the same test sequences simply by changing the window sizes. For each window size the root mean square error is calculated. As the window size changes, the prediction error also changes.

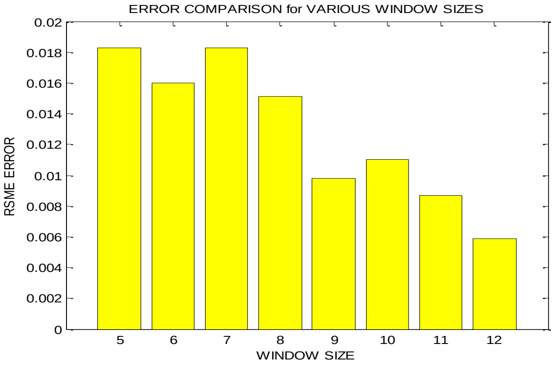

Figure 8. Bar Chart Representation of Errors of Various Window Sizes

Figure 8 represents the bar char representation of errors of various window sizes. While transmission of the movie, in order to reduce the delay time, it is required to take the minimum number of frames to predict the future frames. Minimum the window size, lesser will be the delay. Hence we evaluated the performance of the feed forward neural network for various frame sizes. The window size, we considered is from 5 to 12. From above fig. it can be concluded that, network gives minimum error for 12 window size. Generally in MPEG-4 standard one GOP consist of 12 frames. If further window size is increased, the prediction error will be decreased further. But it will increase the delay time. By considering both these parameters, we can say that network gives the best prediction results for 12 window size.

4.2. Cascaded Feed-forward Neural Network

4.2.1. Prediction Results for Various Window Sizes

Figure 9. represents the training set for 12 window size. Such first 3000 frames from the movie Jurassic Park have taken as a training set to train the neural network.

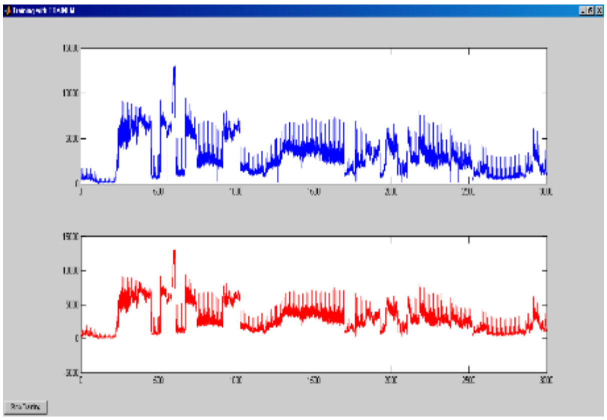

Figure 9. Training Sequence for 12 Window Size

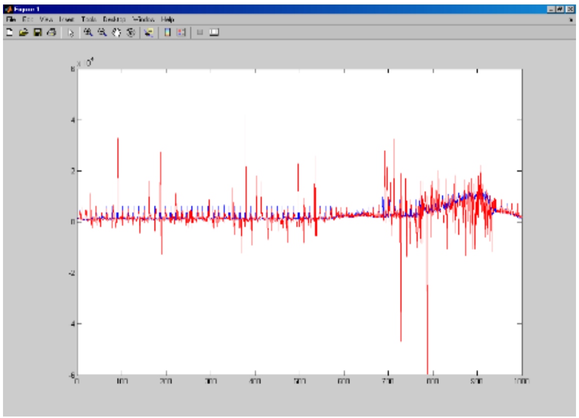

Figure 10. Test Sequence for First 1000 Frames

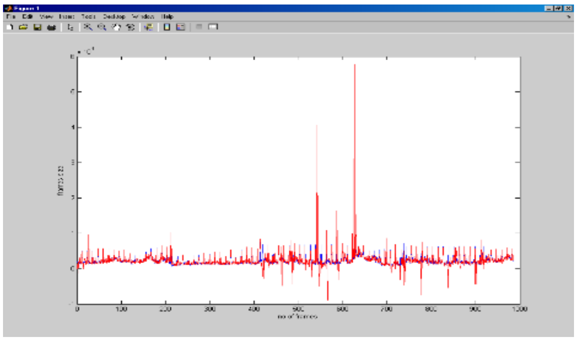

Figure 11. Test Sequence for Middle 1000 Frames

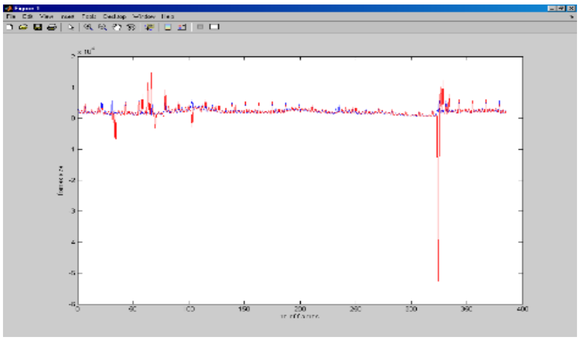

Figure 12. Test Sequence for Last 400 Frames

Figure 10 to 12 represents the test results. The frame sequences from the same movie have been taken to test the trained network. Figure 10 represents the test result for first 1000 frames in the movie “Jurassic Park”. The Red indicates predicted frames while blue indicates original frame sequence. Figure 11 represents the test result for the middle 1000 frames in the movie “Jurassic Park”. Figure 12 represents the test results for the last 400 frames in the movie “Jurassic Park”.

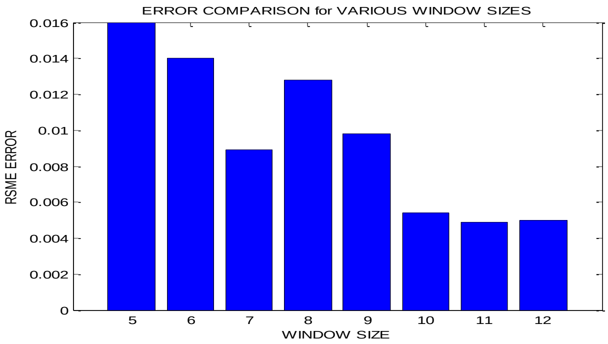

Figure 13. Bar Chart Representation of Errors of Various Window Sizes

Figure 13. shows the error comparison for various window sizes. From above Figure it can be concluded that, network gives minimum error for 12 window size.

4.3. Time Delay Neural Network

4.3.1. Prediction Results for Various Window Sizes

Figure 14 represents the training set for 12 window size. Such first 3000 frames from the movie Jurassic Park have taken as a training set to train the neural network.

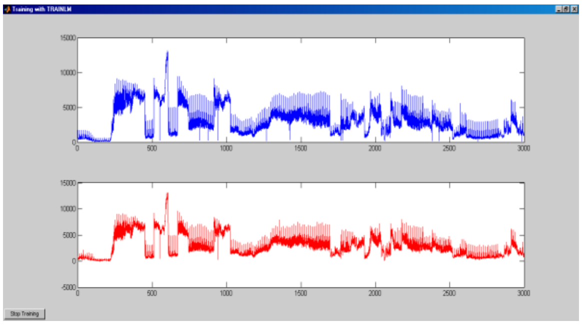

Figure 14. Training Sequence for 12 Window Size

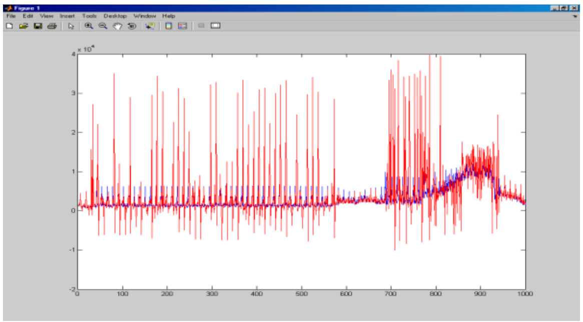

Figure 15. Test Sequence for First 1000 Frames

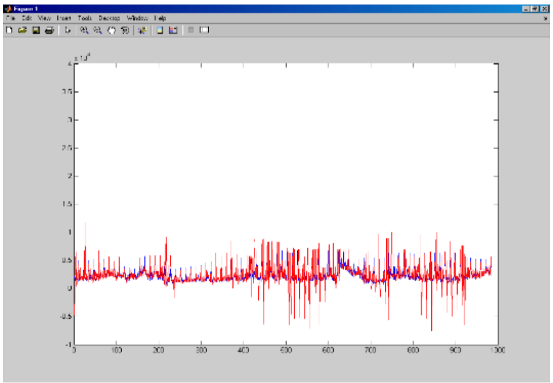

Figure 16. Test Sequence for Middle 1000 Frames

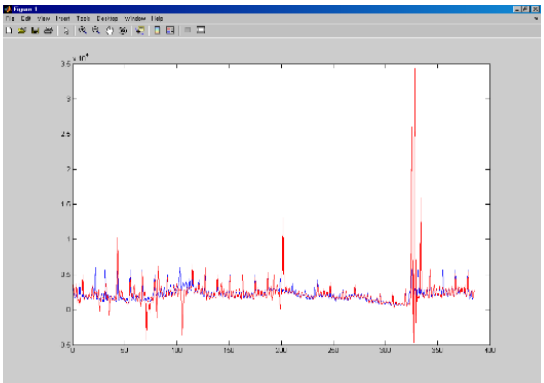

Figure 17. Test Sequence for Last 400 Frames

Figure 15 to 17 represents the test results. The frame sequences from the same movie have been taken to test the trained network. Figure 15 represents the test result for first 1000 frames in the movie “Jurassic Park”. The Red indicates predicted frames while blue indicates original frame sequence. Figure 16 represents the test result for the middle 1000 frames in the movie “Jurassic Park”. Figure 17 represents the test results for the last 400 frames in the movie “Jurassic Park”.

Figure 18 shows the error comparison. From above figure it can be concluded that, network gives minimum error for 12 window size.

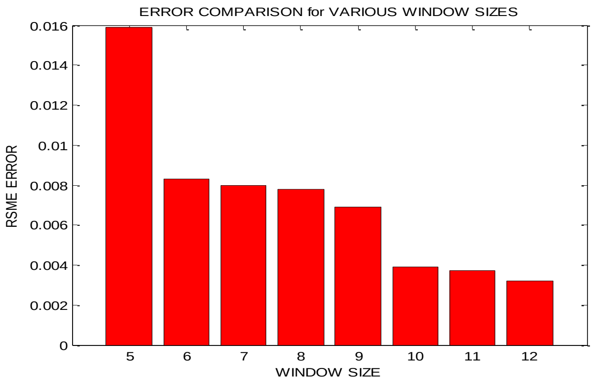

Figure 18. Bar Chart Representation of Errors of Various Window Sizes

5. EVALUATION OF THE PERFORMANCES OF 3 NEURAL ARCHITECTURES

Figure 19 represents the error wise comparison of three neural architectures. For the said problem three neural network architectures are considered namely feed-forward, cascaded feed-forward and time delay neural network. All the three architectures are trained and tested for the same frame sequences. The training and testing data for all the architectures are same. Also the layers and no of neurons in each layer is also kept constant. The layer wise and neuron wise all architectures are having same configuration. The training parameters, such as no of epochs, learning rate and performance goal are also same for all the three architectures. All the architectures are evaluated in terms of root mean square error. The error is calculated for all the window sizes. From the above figure, it is clear that, for cascaded feed-forward neural network the error per window size is very small as compared to remaining architectures. The performance of the feed-forward network is superior to that of time delay neural network.

Hence by referring the Figure 4, we can conclude that, the out of the three architectures, the performance of cascaded feed-forward network is highest as compared to remaining two architectures. Also we can say that the neural networks can be used as the effective tool for video traffic prediction.

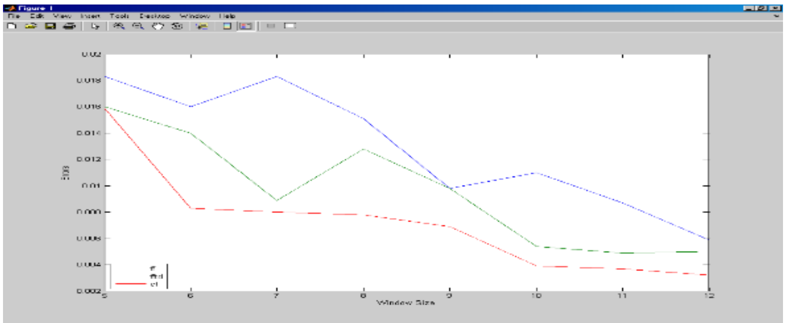

Figure 19. Performance Evaluation of Architectures In Terms of Error

6. COMPARISON OF EURAL APPRAOCH WITH CONVENTIONAL APPRAOCH

Generally the conventional method of prediction is Averaging method. This is called as linear method of prediction. In this method the average of previous frame sizes is taken to predict the next frame size. In our problem of prediction, we had taken this linear approach for comparing results and for evaluating the performance of neural network. First for averaging method, we considered 12 window sizes. By this approach, we predicted the frames.

Figure 20 indicates the comparison between NN approach and Averaging approach. Blue indicates the error between actual frames and predicted frames by NN approach. While Red indicates the error between the actual frames and predicted frames by averaging method. From the above figure, it is clear that the less error is given by neural network as compared to Averaging method error output.

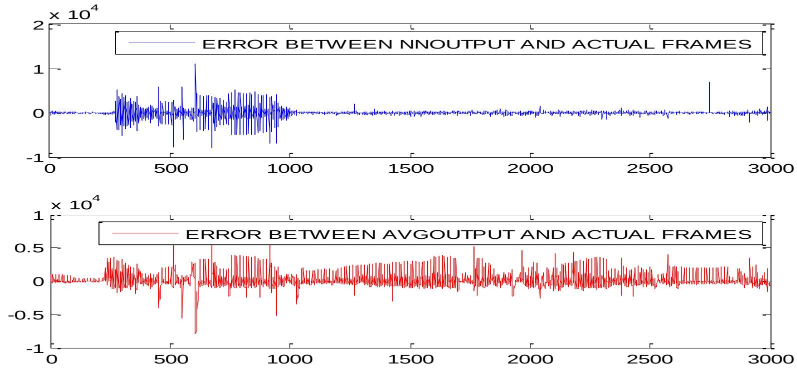

Figure 20. Error Comparison for NN Approach and Traditional Approach

7. CONCLUSIONS

Traffic prediction models using neural networks are useful tools for traffic management algorithms, flow control mechanisms, congestion control schemes, dynamic bandwidth allocation, and QOS control of live VBR videos and other real-time video applications where the video stream is not known in advance. While several papers have dealt with traffic prediction models that can capture some statistical characteristics of the traffic and the inherent non-stationarities and nonlinearities associated with MPEG video traffic, a detailed study of different neural networks techniques has not been done.

In this work, we have examined three different neural network techniques (CF, FF, and FFTD) and evaluated them in terms of their performance in predicting MPEG-4 video traffic.

This work can be concluded with the following statements.

1. We have tried many configurations and types of neural networks for video stream data prediction. First, we tried to find suitable network configurations. This process led us to the network architecture 20-10-6-1.

2. For comparison purposes, we tried to predict the data using various input patterns. We chose 5 to 12 input patterns. From figs.6.5, 7.5, 8.5, the best suitable pattern was 12 input patterns. In order to make the prediction more effective, it is possible to take also the character of the time series into account. For our data, approximately each 1^{2th} pattern forms a peak (in other words, the distance of the consecutive peaks is mostly 12 patterns). This is why for 12 input pattern; we get the minimum prediction error as compared to other input patterns.
3. We have evaluated three architectures. The best performance we get for cascaded feed-forward and feed-forward neural network in terms of error measure. The performance of time delay neural network is fairly low as compared to previous two architectures. But if we compare in terms of simulation time, time delay neural network is simulated in less time as compared to previous two architectures.
4. By considering all above points we can conclude that, accurate traffic prediction using neural networks is indeed possible. This is especially true for MPEG-4 video traffic which is more difficult to predict than other MPEG video standards because it is burstier over a wide range of time scales and has higher degree of self-similarities.

8. TRACKS FOR FEAATURE WORK

Some recommendations for future work are:

1. Use of more than one model for multi-step-ahead prediction of the source video traffic. This requires the design of a scheme which switches between the predictions models depending on the bit rate of the video traffic.
2. Design of non-linear prediction models which can be adapted online. Till now researchers have used linear, non-linear and adaptive linear models for the prediction of MPEG-coded video source traffic. The domain non-linear modeling techniques which can be adapted online for the prediction of MPEG-coded video source traffic has not been explored.
3. Design of a control scheme for efficient delivery of multimedia traffic using the output of the empirical models described in this research work.

REFERANCES

[1] E. P. Rathgeb, "Policing of Realistic Vbr Video Traffic in an Atm Network," *International Journal on Digital and Analog Communication Systems*, Vol. 6, Pp. 213–226, October–December 1993.

[2] W. E. Leland, M. S. Taqqu, W. Willinger, And D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," *Ieee/Acm Trans. Networking*, Vol. 2, Pp. 1–15, February 1994.

[3] A. R. Reibman And A.W. Berger, "Traffic Descriptors For Vbr Video Teleconferencing Over Atm Networks," *Ieee/Acm Trans. Networking*, Vol. 3, Pp. 329–339, June 1995.

[4] M. Grossglauser, S. Keshav, And D. Tse, "Rcbr: A Simple And Efficient Service For Multiple Time-Scale Traffic," *Ieee/Acm Trans. Networking*, December 1997.

[5] M. Krunz And S. K. Tripathi, "*On The Characteristics Of Vbr Mpeg Streams*," In *Proc. Acm Sigmetrics*, Pp. 192–202, June 1997.

[6] M. Garrett And W. Willinger, "*Analysis, Modeling and Generation of Self-Similar Vbr Video Traffic*," In *Proc. Acm Sigcomm*, September 1994.

[7] M. Krunz And S. K. Tripathi, "*On The Characteristics of Vbr Mpeg Streams*," In *Proc. Acm Sigmetrics*, Pp. 192–202, June 1997.

[8] Adel Abdennour "Evaluation of Neural Network Architectures For Mpeg-4 Video Traffic Prediction", *Ieee Transaction on Broadcasting*, Vol.52, No 2, June 2006.

[9] P. Cortez, M. Rio, M. Rocha, P. Sousa, Internet Traffic Forecasting Using Neural Networks, International Joint Conference on Neural Networks, Pp. 2635–2642. Vancouver, Canada, 2006.

[10] V. B. Dharmadhikari, J. D. Gavade, An Nn Approach For Mpeg Video Traffic Prediction, 2nd International Conference on Software Technology and Engineering, Pp. V1-57–V1-61. San Juan, Usa, 2010.

[11] H. Feng, Y. Shu, Study on Network Traffic Prediction Techniques, International Conference on Wireless Communications, Networking and Mobile Computing, Pp. 1041–1044. Wuhan, China, 2005.

[12] L. Cai, J. Wang, C. Wang, L. Han, A Novel Forwarding Algorithm Over Multipath Network, International Conference on Computer Design and Applications, Pp. V5-353–V5-357. Qinhuangdao, China, 2010.

[13] S. Haykin, Neural Networks - A Comprehensive Foundation. Upper Saddle River, New Jersey: Prentice Hall, 1994.

[14] Mpeg-4 And H.263 Video Traces For Network Performance Evaluation, Http://Www-Tkn.Ee.Tu Berlin.De/Research/Trace/Trace.Htm1.