AI-Based TCP Performance Modelling

K. Mahmoud and B.V. Ghita

Centre for Security, Communications and Network Research
Plymouth University, United Kingdom
e-mail: info@cscan.org

Abstract

Different mathematical models exist for modelling TCP algorithms and interrelations between TCP and network parameters. In this research, two artificial neural network models were developed to model TCP performance of both lossless and lossy traffic flows. A mathematical base line was defined for accuracy comparison in terms of regression and MSE of estimated throughput. The presence of idle time in TCP flows was investigated and accounted for in the models, in addition to the consideration of non-standard flows and statistical outliers. Neural models developed had outperformed the mathematical modelling of TCP throughput along all stages of this research. Finally, it was suggested to revise the available mathematical model to take idle time into consideration.

Keywords

TCP Performance, Throughput, History-based Modelling, Neural Networks, Idle Time, Slow Start, Bulk TCP Transfer, Robust AI-Based Modelling

1 Introduction

The importance of TCP performance is due its 90% representation of the Internet traffic and hence its reflection on the overall performance of IP networks (Shah et al., 2007). The need to provide realistic performance modelling of the TCP throughput and find relationships between network conditions and this throughput is essential. Traditional mathematical models do not provide accuracy as expected despite their complexity, especially for short-lived TCP connections (Ghita and Furnell, 2008).

This paper adopts an AI-based approach using neural networks in MATLAB to model TCP throughput (i.e. transmission time) for both lossless and lossy TCP connections, comparing results obtained with mathematical modelling and results from previous research. During the modelling, various TCP parameters are investigated in terms of their effect and relationship to the actual throughput. Conclusions are made on whether mathematical models and TCP algorithm may be revised and modified based on these observations

2 Previous Research

A research was made by He et al. (2007) to develop a model for predicting the TCP throughput for bulk TCP transfers in particular. As a testbed, their research made use of an architecture of 50-60 nodes distributed in universities, research labs and ISPs in the US, Europe and Asia. In their research, they have initially strengthened on the difference between performance estimation evaluated during TCP transfer, and performance prediction which is acquired using probing prior to the actual transfer of data. He et al. (2007) have classified the models used to evaluate the performance of TCP for TCP transfers into two classifications; formula-based or mathematical models, and history-based models, each approach having its own advantages and drawbacks.

2.1 Mathematical TCP Models

Formula-based models depend on mathematical expressions to evaluate the expected TCP throughput from the TCP parameters. A mathematical model was proposed by Cardwell et al. (2000) describing each stage of a TCP connection: slow start, segment loss, congestion avoidance and delayed acknowledgement. This model was considered as a reference and baseline in this research in order to evaluate the performance results obtained from the AI-based model.

$$E[T] = E[T_{SS}] + E[T_{loss}] + E[T_{ca}] + E[T_{delack}]$$

2.2 History-Based Models

History-based models mainly depend on the previous knowledge acquired from historical TCP transfers. The models use adaptive learning in order to form relationships between observed path characteristics and the resulted TCP throughput of each transfer. Accordingly, history-based models are independent of the TCP implementation used at the server and the receiver ends. As per (He et al., 2007), the prediction accuracy of their history-based model gave better accuracy with a RMSRE less than 0.4 for 90% of the traces.

Another research was made by Mirza et al. (2010) in which they adopted a machine learning approach to predict TCP throughput. They have used Support Vector Regression (SVR). The measurements used in their models were the available bandwidth on the congested link, the queuing at the bottleneck node, and the loss rate. They have used both passive and active path measurements. For the passive measurements, parameters (available bandwidth, queuing, and loss rate) were obtained from pre-captured TCP flows, and for active measurements the same parameters were obtained from the active monitoring cards. Their results obtained from their experiments indicated that for bulk TCP transfer, the predicted TCP throughput was within 10% of the actual value 87% of the time.

A research approach for estimating TCP performance using neural networks was adopted by (Ghita et al., 2008). They have used both synthetic and real network traffic. In their research they have divided their training data sets into lossless and lossy flows. The results obtained from the neural model have revealed significant improvement with nearly a ten-fold improvement of the relative error, while that was not the case for traffic with segment losses.

3 Methodology and Model Design

Three sources of captured traffic were used for training and validating during modelling: traffic collected from Brescia University, Plymouth University, and MAWI Research Group. Figure 1 demonstrates the process diagram for this research. Backpropagation feed forward neural networks were considered for the modelling process. Backpropagation, or propagation of error, is a supervised learning method, implementing the Delta rule to update the weights of the networks to reach convergence (Freeman and Skapura, 1991).

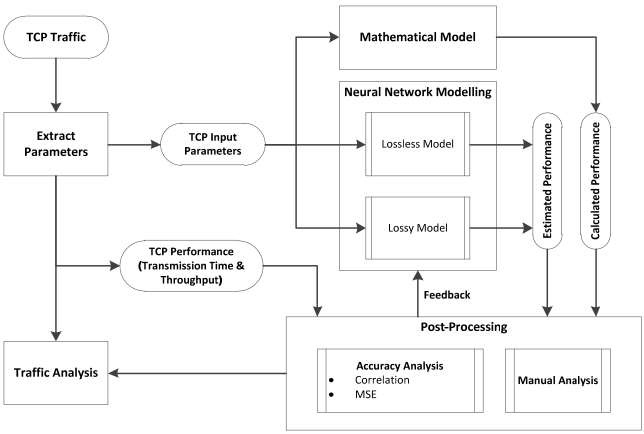

Figure 1: Process diagram of research stages.

Two separate neural models were developed for both lossless and lossy TCP traffic. Six input variables were fed to the lossless model: the actual data sent, the average RTT, the average and maximum segment size, the maximum congestion window, and the initial sender window size. The same were used for the lossy model in addition to the loss rate as evaluated by the count of triple duplicate ACK, and the average retransmission time. For both models, the actual transmission time was considered as the target during training, while the output being the estimated transmission time estimated by the models.

From the statistical analysis performed at early stages of the research, the existence of prolonged periods of idle time within the lifespan of TCP connection compared the average RTT and total transmission time, as show in Figure 2. Hence, further pre-processing and filtering was applied based on the maximum idle time values, and neural model performance was re-evaluated after this exclusion in samples, while comparing them to Cardwell's mathematical model.

The effect of excluding statistical outliers (i.e. 2^{nd} and 98^{th} percentile) of all TCP variables used was investigated. Additionally, TCP flows with non-standard conditions according to the following conditions were excluded and investigated:

1. RST packets sent in either direction.
2. More than a single SYN/FIN sequence exchanged in each direction.
3. Data packets in the reverse direction.

4. Non-HTTP traffic.
5. Flows with small average MSS (less than 1400 bytes).

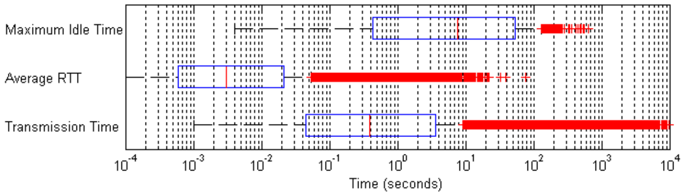

Figure 2: Box-and-whisker diagrams of TCP time parameters (Plymouth University Traffic

4 Results and Analysis

The following sections demonstrate all results obtained from both the mathematical and neural network models developed in MATLAB, for lossless and lossy traffic.

4.1 Lossless Dataset

When considering all valid flows, the regression value obtained from the mathematical model was 0.3216, and from the neural model was 0.7680. As shown in Figure 3, the distribution of scattered actual and estimated transmission time for the mathematical model shows a relatively small subset of samples following the ideal fit line (Y=T), while the majority of scattered samples are well distributed below this line with high residual values, which indicates no account for any possible additional time (idle time) within the lifespan of the connection. On the other hand, the neural model seems to be accounting for this possible additional time and provided an evenly distributed scattering above and below the ideal fit line (Y=T). The MSE of performance estimation was 11.8910 and 1.8253 for the mathematical and neural model respectively, which are considered relatively high.

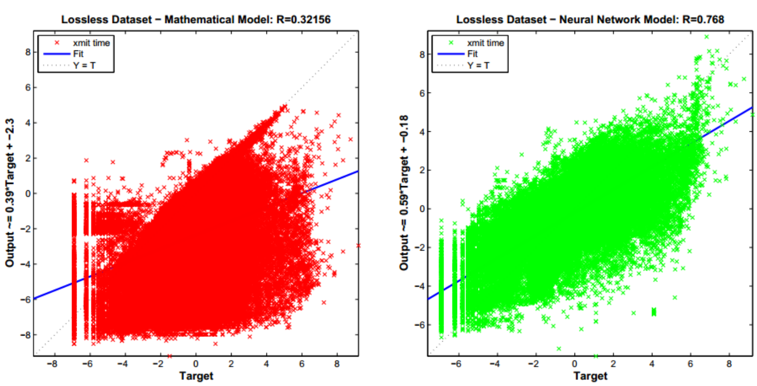

Figure 3 Regression obtained for lossless connections for the combined dataset using both mathematical and neural network model.

Gradually excluding flows with high idle time improved estimation accuracy, as demonstrated in Table 1. The regression and MSE results obtained when filtering connections with maximum idle time less than twice the average RTT were 0.9892 and 0.0229 respectively for the neural network model and 0.9067 and 0.2220 respectively for the mathematical model. The regression analysis for both models is shown in Figure 4. The uniform scattering of estimated transmission time by both model along the idle fitting line(Y=T) had clearly improved. The CDF for absolute relative error is show in Figure 5. At these near ideal conditions, the neural network model is still providing better accuracy performance.

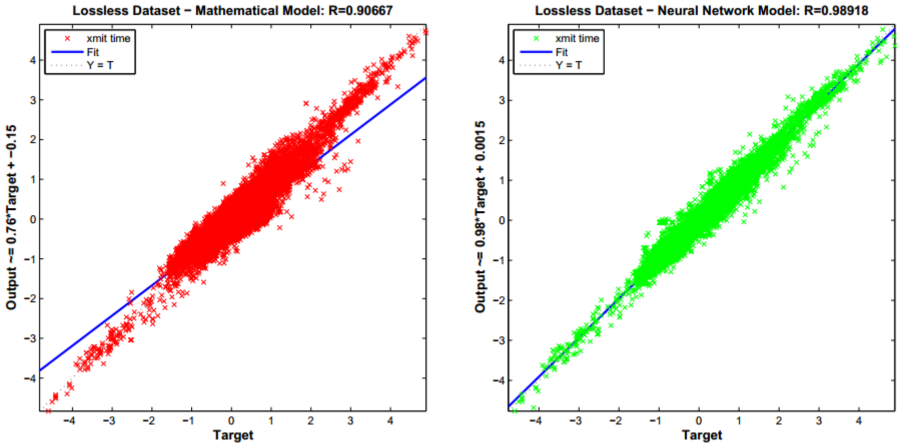

Figure 4: Regression obtained for lossless connections for the combined dataset using both mathematical and neural network model, after filtering connections with maximum idle time larger than twice the average RTT.

Maximum Idle Time to Average RTT Ratio	Number of Samples	Neural Network Model		Mathematical Model	
		MSE	regression	MSE	regression
2	19458	0.0229	0.9892	0.2220	0.9067
6	27342	0.0699	0.9759	0.2945	0.9200
10	30913	0.1108	0.9649	0.4286	0.9019
14	33338	0.1588	0.9510	0.5629	0.8857
18	35627	0.1675	0.9496	0.6982	0.8717
22	37840	0.1995	0.9409	0.8549	0.8574
26	39487	0.2181	0.9363	0.9684	0.8468
30	40848	0.2421	0.9299	1.0821	0.8371

Table 1: Results obtained by gradually excluding flows with relatively high idle time, compared to average RTT for each flow.

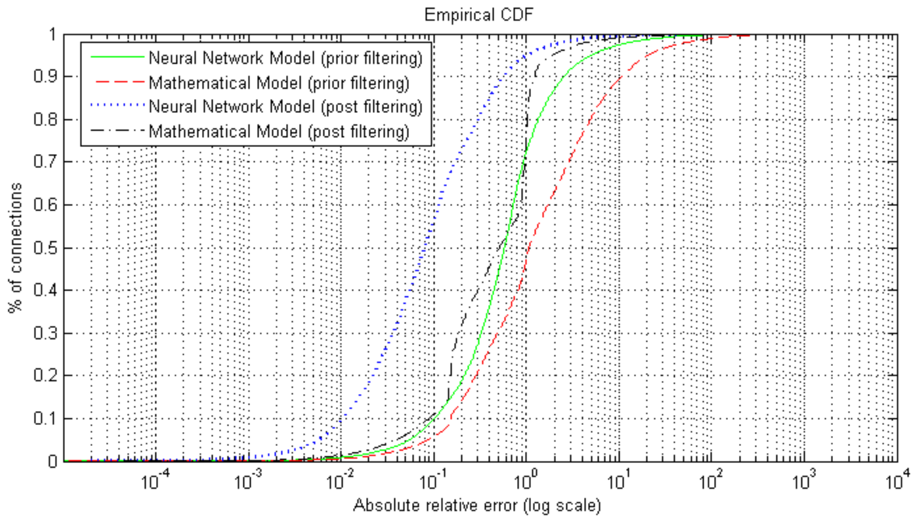

Figure 5: CDF of absolute relative error for lossless connections for the combined dataset, after filtering connections with maximum idle time larger than twice the average RTT.

Filtering statistical outliers and non-standard connections improved performance even more. MSE measure of the neural model decreased from 0.0646 to 0.0325 (50.31%), and regression increased from 0.9778 to 0.9881 (101.05%). While for the mathematical model, MSE decreased from 0.2945 to 0.0921 (31.27%), and regression improved from 0.9200 to 0.9707 (105.51%). The filtering criterion with the most positive effect was to exclude non-HTTP flows. Regression analysis is shown in Figure 6, and the CDF of absolute relative errors in Figure 7.

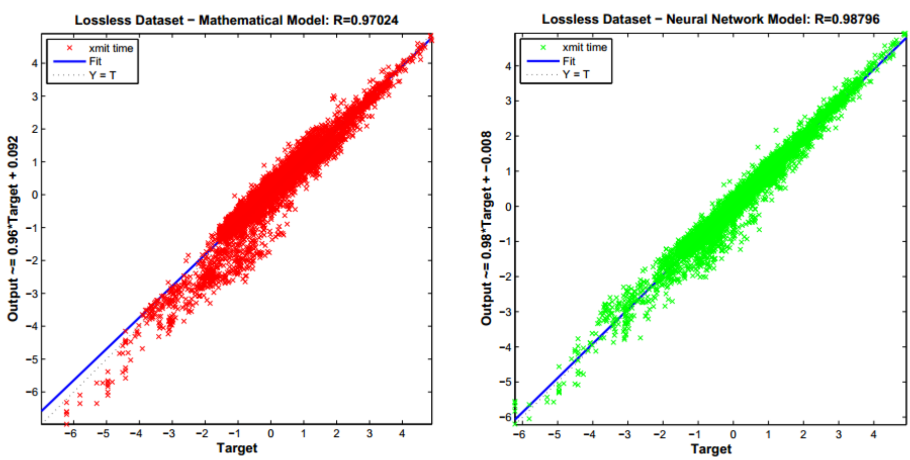

Figure 6: Regression obtained for lossless connections for the combined dataset using both mathematical and neural network model, post filtering non-standard TCP connections.

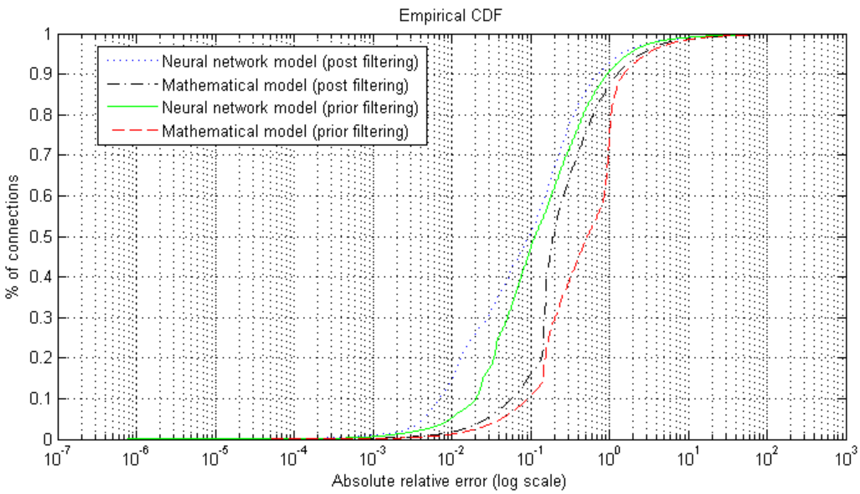

Figure 7: CDF of absolute relative error for lossless connections for the combined dataset, prior and post filtering non-standard TCP connections.

4.2 Lossy Dataset

For the lossy subset, filtering all sorts of outliers and idle time has improved the MSE of the neural model from 0.0371 to 0.0301 (81.13%), and slightly increased the regression from 0.9857 to 0.9863 (100.06%). As for the mathematical model, MSE decreased from 0.5990 to 0.5381 (89.83%). At this stage of filtering, the variation and inconsistency in performance was observed due to the reduced number of training samples to only 3341 TCP flows, which may have led to over-fitting the model.

4.3 Results from Plymouth University Dataset

The results obtained using the dataset from Plymouth University under same testing conditions are summarised in Table 2.

Filtering Criteria	Number of Samples	Neural Network Model		Mathematical Model	
		MSE	Regression	MSE	Regression
All Valid lossless flows	100000	3.3045	0.7722	19.5918	0.3408
Excluding high idle times and non-standard flows	100000	0.5102	0.8697	0.9951	0.7428
All valid lossy flows	57299	1.3533	0.8697	12.2968	0.5886
Excluding high idle times and non-standard flows	1913	0.1066	0.9828	0.7173	0.9419

Table 2: Summary of results obtained using the dataset from Plymouth University

5 Conclusions and Future Research

At all stages of modelling and testing, the neural models have provided better accuracy in estimating TCP throughput with respect to the mathematical model.

The observation made to the estimation of transmission time as resulted from the neural model and how the model in way anticipated for the idle time periods in TCP connections suggests the modification of available mathematical models to possibly include an additional average additional time to the total transmission time. This average could result from a function of average RTT, loss rate and congestion window. The application of such modified model could be implemented and evaluated in a simulated environment such as (NS2) to study the effect on congestion window when resuming data transfer after an idle time period. In this research, only the maximum idle time as calculated by tcptrace was considered. Although this value may give a good representation of total idle time, specially using AI-based methods, more research can be done to modify the output from tcptrace to iteratively evaluate the total idle time during the complete lifetime of a TCP connection, and consider this value as input at neural network modelling stages. This is expected to provide better estimation accuracy.

It was found difficult to identify TCP algorithm for manual analysis and traces investigation. A proposed approach is to identify the implemented TCP congestion algorithm used in captured TCP traffic in order to investigate how each implementation deals with the presence of idle time, and how congestion window is modified after the occurrence of an idle time period. Hence, a comparison could be done between different TCP congestion implementations and evaluate how each implementation performs in finding the ideal congestion window after an idle time, which should result in faster transmission after these idle time periods.

6 References

Cardwell, N., Savage, S. and Anderson, T. (2000), Modeling tcp latency, pp. 1742–1751.

Freeman, J. A. and Skapura, D. M. (1991), Neural networks: algorithms, applications, and programming techniques, Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.

Ghita, B. and Furnell, S. (2008), ‘Neural network estimation of tcp performance’, International Conference on Communication Theory, Reliability, and Quality of Service 0, 53–58.

He, Q., Dovrolis, C. and Ammar, M. (2007), ‘On the predictability of large transfer tcp throughput’, Comput. Netw. 51(14), 3959–3977.

Mirza, M., Sommers, J., Barford, P. and Zhu, X. (2010), ‘A machine learning approach to tcp throughput prediction’, IEEE/ACM Trans. Netw. 18(4), 1026–1039.

Shah, S., Rehman, A., Khan, A. and Shah, M. (2007), Tcp throughput estimation: A new neural networks model, in ‘Emerging Technologies, 2007. ICET 2007. International Conference on’, pp. 94 –98.