Machine Learning based Call Admission Control Approaches: A Comparative Study

Abul Bashar, Gerard Parr, Sally McClean and Bryan Scotney
School of Computing and Information Engineering
University of Ulster
Coleraine, UK BT52 1SA
{bashar-a, gp.parr, si.mcclean, bw.scotney}@ulster.ac.uk

Detlef Nauck
Research and Technology
British Telecom, Adastral Park
Ipswich, UK IP5 3RE
detlef.nauck@bt.com

***Abstract*— The importance of providing guaranteed Quality of Service (QoS) cannot be overemphasised, especially in the NGN environment which supports converged services on a common IP transport network. Call Admission Control (CAC) mechanisms do provide QoS to class-based services in a proactive manner. However, due to the factors of complexity, scale and dynamicity of NGN, Machine Learning techniques are favoured to analytical approaches for providing autonomous CAC. This paper is an effort to compare the performance of two such approaches - Neural Networks (NN) and Bayesian Networks (BN), to model the network behaviour and to estimate QoS metrics to be used in the CAC algorithm. It provides a way to find the optimum model training size for accurate predictions. Performance comparison is based on a wide range of experiments through a simulated network in Opnet. The outcome of this comparative study provides some interesting insights into the behaviour of NN and BN models and how they can be utilised for better CAC implementations.**

***Index Terms*— Call Admission Control, Machine Learning, Quality of Service, Neural Networks, Bayesian Networks.**

I. Introduction

THE problem of providing guaranteed Quality of Service (QoS) was first witnessed in the ATM networks which supported class-based services. Due to the advances in the network architectures and the demand for multimedia rich services and applications, this problem has received much attention from the researchers. The solution to this problem assumes even more significance in the recent times because of the rapid deployments of IP-based Next Generation Networks (NGN) which promises guaranteed QoS to converged services that share the common transport layer [1]. Call Admission Control (CAC) mechanism plays a proactive role in providing QoS by limiting the entry of traffic at the edges of the network. However, as the number of services, their classes and size of the network grows, the CAC problem becomes difficult and intractable to solve through conventional analytical methods.

Machine Learning (ML) approaches have been successful in solving the CAC problem, since they are capable of modelling the system behaviour through the process of *learning*, based on observation of performance data over a period of time [2]. Once appropriately *trained*, they can automatically estimate and predict future system behaviour and subsequently make admission decisions with high accuracy and speed. However, to our best knowledge, there is no research work which specifically compares the application of existing ML approaches to solve the problem of CAC. This paper is an effort to address the following research questions:

- How much training data is sufficient to construct an accurate CAC model?
- Which system QoS metric(s) need to be estimated for automated admission decisions?
- What are the unique feature(s) of existing ML approaches and how do they compare to one another?

To answer the above research questions, we take into consideration the existing ML approaches which have been applied in the telecommunications domain to solve various network management problems [3]. Based on this study, we consider two prominent approaches for our comparative study, namely, *Neural Networks* (NN) and *Bayesian Networks* (BN) [4]. We formulate the CAC process as a classification problem, which in essence has to classify a call request into two classes: *Admit* or *Deny*. We provide the answers through simulation results obtained from experiments performed in *Opnet Modeler*, *Hugin Researcher* and *IBM SPSS Modeler* [5] [6] [7].

The rest of this paper is structured as follows. In Section II we provide the required background and survey the existing related work. In Section III we present the details of a generic ML-based CAC framework and a comparative theoretical analysis of NN and BN approaches. Simulation results and discussions will be presented in Section IV. Section V concludes the paper by suggesting possible future work.

II. Background And Related Work

QoS is a prime concern for both the service providers and subscribers. CAC is a major preventive technique to provide guaranteed QoS to various class-based services, as recommended by the ITU-T for NGN [8]. Its job is to make a decision to admit or deny a new call into the network, based on

978-1-4244-8909-1/$26.00 ©2010 IEEE

This paper was peer reviewed at the direction of IEEE Communications Society by subject matter experts for publication in the CNSM 2010 proceedings.

the condition that the QoS of the existing calls and also the new call can be simultaneously satisfied.

One of the earliest call admission control schemes were first applied to ATM networks [9]. These schemes were based on analytical methods like equivalent bandwidth, heavy traffic approximations and upper bounds on cell loss probabilities. The problem with these approaches was that they need to make simplifying assumptions about traffic distributions, as otherwise they would become analytically involved. This resulted in reduced accuracy and also to over provisioning of network resources.

This led to ML based CAC schemes which had improved performance in terms of reduced Cell Loss Ratio and Delay. One of the solutions is based on NN, which used back-propagation NN for learning relation between offered traffic and service quality [10]. CAC scheme has been developed based on combined use of Particle Swarm Optimization and Fuzzy logic for next generation mobile communication networks [11]. Support Vector Machine (SVM) based CAC algorithm utilises service vector and network vector to predict admission state for admission decisions [12]. A more recent BN-based CAC framework has been proposed by us which implements delay prediction (about 97% prediction accuracy) based call admission decisions [13].

All the approaches mentioned above formulate the CAC process as a classification problem. Even though there is a paper which compares NN and BN for response time modelling in service-oriented systems [14], there is no evidence in the literature which compares these ML approaches for CAC problem. The uniqueness of this paper lies in the fact that, it compares two most prominent approaches, NN and BN, for their classification accuracy for CAC decisions, using estimates of QoS metrics, packet loss and delay. The next section describes in detail the ML-based framework for CAC.

III. ML-BASED CAC FRAMEWORK

CAC system generally resides on an edge router, whose function is to allow controlled traffic into the core network. A generic CAC framework based on the ML approach is shown in Fig. 1. The input to such a system consists of customer call requests and the output is a decision to either admit or deny the call. Usually the call request consists of its characteristics defined in terms of *traffic descriptors* and *desired QoS* during its service time in the network. Traffic descriptors can include parameters like peak rate, average rate, maximum burst duration and type of application, which are generally supplied by the caller. In the absence of such information, they can even be measured through network monitoring tools. Qos requirements include some measure of metrics like packet loss, average delay or delay variation (jitter). In addition to these inputs, a set of *system state* parameters like available link bandwidth and buffer occupancy are also provided as inputs to CAC. The output of the ML module (NN or BN in our case)

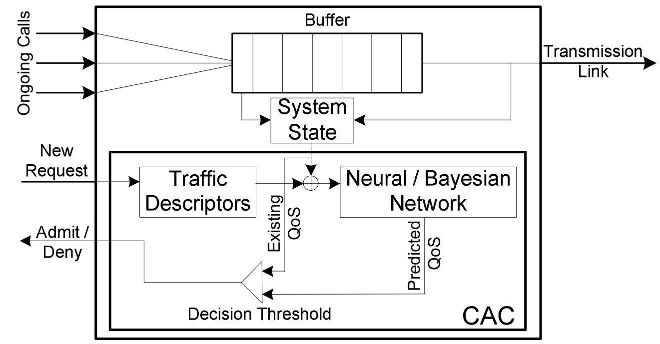

Fig. 1. Machine Learning based CAC.

can be either the final Admit/Deny decision, or the value of any QoS metrics like packet loss, delay or jitter. Based on the choice of inputs and outputs, the ML module is *trained offline* with a set of data which is observed in the system over a period of time. The training data set consists of cases where both the inputs and outputs are known. However, when the trained model is in the online mode, it provides the estimate of the output for a particular input combination.

In the event of a new request, the traffic descriptors and the system state information are given as an input to the NN/BN module. The output of the NN/BN module, which is the predicted QoS, is compared to the existing QoS and an admission decision (Admit/Deny) is made based on some predefined thresholds. It is clear that the overall CAC performance is dependent on the prediction accuracy of the model. Prediction accuracy depends on how well the model estimates the unknown output, when presented with unseen cases not present in the training set. In our study, we compare the NN/BN approaches on the classification accuracy of QoS metrics, packet loss and delay.

Comparison of NN and BN: NN and BN belong to a class of machine learning approach called as *supervised* learning. As the name suggests, the models are *trained* on a set of cases and then used (or *tested*) on another set of new cases. The performance of these models is generally evaluated in terms of training speed, model complexity, prediction accuracy and model interpretability [14]. A summary of salient features of NN and BN approaches and their comparison is presented in Table I.

TABLE I
Salient features of NN and BN ([15][16])

Feature	NN	BN
Building block	Neuron	Node
Model Structure	Feed Forward Structure	Directed Acyclic Graph
Parameter Learning Algorithm	Back propagation Algorithm	Expectation Maximisation (EM) Algorithm
Structural Learning Algorithm	Fixed structure with arbitrary hidden nodes	Necessary Path Condition (NPC) algorithm
Domain knowledge Incorporation	Possible with little effort	During parameter and structure learning phase
Model Interpretability	Represents a black box	Intuitive graphical presentation

IV. SIMULATION RESULTS

A. Experimental setup

The network topology for our CAC experiments was simulated using *Opnet Modeler* is shown in Fig. 2. The four sources (S1-S4) connect to four destinations (D1-D4) via the Router A. The CAC algorithm is running at Router A, which is connected to Router B through the transmission link.

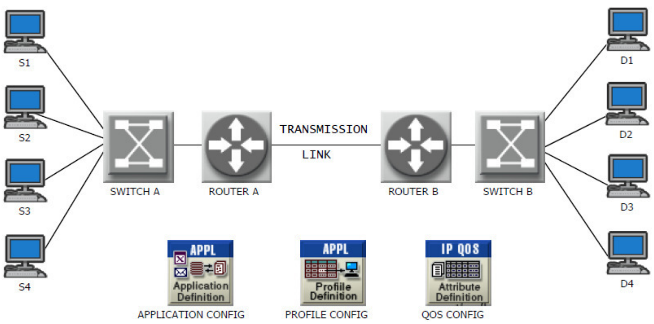

Fig. 2. Network topology in Opnet Modeler.

The traffic sources are modelled as Poisson processes for the call arrivals. The traffic characteristics of individual calls are bursty in nature and modelled as ON-OFF processes with generation parameters as shown in Table II. The sources model a converged traffic scenario as found in the NGN. The router buffer is modelled as a FIFO queue of size 400 packets and constant service rate of 100 kbits/sec.

TABLE II
Characteristics of traffic sources

Parameter	Value
Traffic ON state time (s)	Exponential (10.0)
Traffic OFF state time (s)	Exponential (90.0)
Packet Interarrival Time (s)	Exponential (0.1)
Packet Size (bytes)	Exponential (1024)

The simulation was run (with multiple seeds) over a simulation time of 3000 minutes to capture data for offline training purpose. The statistics were captured at an interval of 60s. The statistics used in our study were: aggregate incoming *Traffic* (traffic descriptor), *Queue* occupancy (system state), queuing *Delay* and packet *Loss* (QoS metrics), described in Table III. For the purpose of model interpretability, we discretise these statistics into five levels (*VLO, LO, MED, HI, VHI*) of equal width. It is to be noted that both BN and NN are capable of accepting discrete as well as continuous inputs.

TABLE III
Training data description for NN/BN models

Neuron/Node name	Description	Type
Traffic	Aggregate traffic (bits/s)	Input
Queue	Queue occupancy (bits)	Input
Delay	Packet queuing delay (s)	Input
Loss	Number of dropped packets	Output

B. Model construction

The training data obtained from the previous step is used to learn the NN/BN model. The NN model was obtained using the default optimal back propagation algorithm implemented in *IBM SPSS Modeler*. The BN model was built using the NPC & EM algorithms of *Hugin Researcher*. The classification mode of operation of NN/BN requires us to specify the target to be estimated. We experimented with two targets, *Delay* and *Loss*, to decide which one performs better for model building.

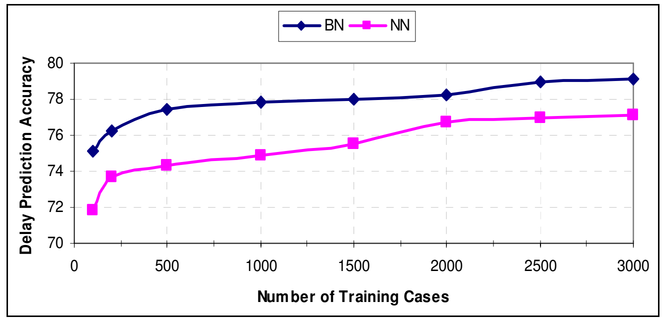

Fig. 3. Prediction accuracy plot for *Delay*.

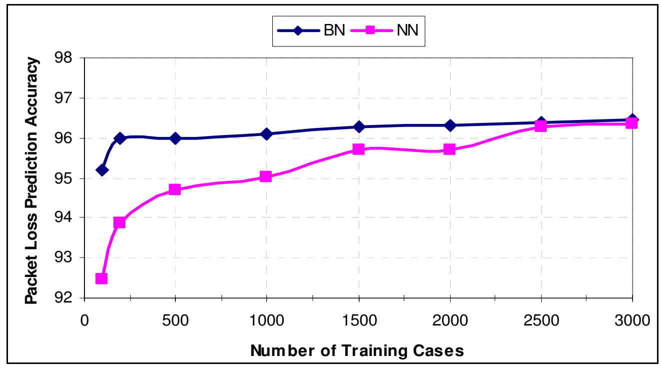

Fig. 4. Prediction accuracy plot for *Loss*.

We also experimented with various training set sizes and observed the prediction accuracy. Fig. 3 provides the variation of prediction accuracy as a function of training data set size for *Delay* estimation. Fig. 4 shows a similar plot for *Loss*. These results were obtained using the process of 10-fold cross validation, where the collected data is partitioned into training and test sets and the prediction accuracy is averaged over 10 such iterations.

It is observed from Fig. 3 and Fig. 4 that, in both the cases BN outperforms NN in terms of prediction accuracy. This is because BN can model discrete datasets better than NN. This leads us to conclude that BN can provide better prediction accuracy than NN and hence is a competitive approach for CAC problem. Another observation is that, both the approaches provide at least 20% more accurate predictions of *Loss* as compared to *Delay*. This is because the probability distribution of *Loss* data was found to have lesser variance. Hence, we conclude that it is wise to use the NN/BN models to predict *Loss* and then base the admission decisions on this QoS estimate as opposed to *Delay*. So, we decided to choose *Traffic, Queue* and *Delay* as the NN/BN inputs and *Loss* as output in the final models.

The choice of training data set size can also be inferred from Fig.4. We observe that the BN model approaches saturation region earlier as compared to NN. So, the best model for BN is obtained at the training size of 2000 cases, as there is only a small improvement of 0.15 % in prediction accuracy for more cases. Similarly, for the NN case the best model is obtained for the training size of 3000 cases. This leads us to the conclusion that, BN approach converges faster to give a model with similar prediction accuracy as that of NN.

C. NN and BN Models

Based on the observations of the previous section, we now arrive at the final NN and BN models. Fig. 5 presents the NN model obtained from the training data set size of 3000 cases. It consists of 15 input neurons (3 input variables of 5 states each), 3 neurons in the hidden layer and 5 neurons in the output layer. It can be observed that, the NN model is not very convenient graphical representation and provides little information about the relationships between the variables.

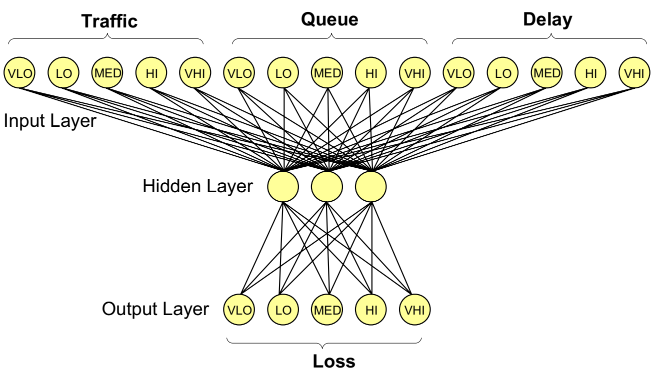

Fig. 5. Neural Network model.

Fig. 6 shows the BN model obtained from the training data set of 2000 cases. As can be seen, it is a Directed Acyclic Graph (DAG) with the structure consisting of four nodes (along with their marginal probability windows). It is up to the system modeller to make use of this model to estimate the target variable (in our case, it is the *Loss* node). It is observed that the BN model representation appeals to the human intelligence and is very intuitive to work with. However, it needs to be understood that the BN model is internally represented by Conditional Probability Tables (CPTs), whereas the NN model is represented by the link weights.

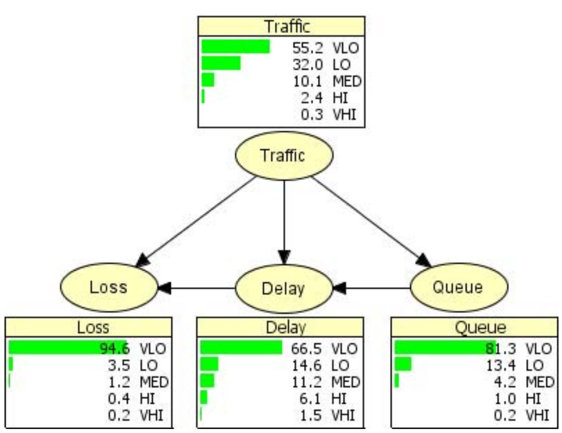

Fig. 6. Bayesian Network model.

The results presented here were based on the assumption that the QoS estimates provided by the NN/BN models will be further utilised to make CAC decisions. This is part of our ongoing work and we plan to evaluate these approaches for call blocking probabilities as a function of call arrival rates.

V. Conclusion

This paper presented a framework for evaluating ML-based approaches for CAC. More specifically, it compared the performance of two major ML approaches, Neural Networks and Bayesian Networks, for QoS prediction. It demonstrated that training data size for BN model was relatively smaller than the NN model. Further, it showed that a better CAC algorithm can be based on packet loss QoS metric as opposed to delay QoS metric. In addition, it brought out the relative merits and demerits of using NN and BN models through a comprehensive comparison. Future work involves a detailed performance analysis of NN and BN models for testing call blocking probability & computational speed in an online setup.

Acknowledgment

The authors would like to acknowledge the support of University of Ulster for providing Vice Chancellors Research Studentship and IU-ATC for funding the joint work with BT.

References

[1] *General overview of NGN,* ITU-T Recommendation Y.2001, Dec 2004.

[2] E. Alpaydin, *Introduction to Machine Learning*, MIT Press, 2004.

[3] J. Qi *et. al*, "Artificial intelligence applications in the telecommunication industry," in *Expert Systems*, vol. 24, pp. 271-291, Sep. 2007.

[4] T.T.T. Nguyen, G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol.10, no.4, pp.56-76, Fourth Quarter 2008.

[5] OPNET Modeler 16.0, http://www.opnet.com

[6] Hugin Researcher 7.3, http://www.hugin.com

[7] IBM SPSS Modeler 13.0, http://www.spss.com

[8] *Resource and admission control functions in next generation networks,* ITU-T Recommendation Y.2111, Nov 2008.

[9] H. G. Perros, K. M. Elsayed, "Call admission control schemes," *IEEE Communications Magazine*, pp. 82-91, Nov 1996.

[10] A. Hiramatsu, "ATM communications network control by neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 122-130, Mar 1990.

[11] C. Huang, Y. Chuang, and D. Yang, "Implementation of call admission control scheme in next generation mobile communication networks using particle swarm optimization and fuzzy logic systems," *Expert Systems with Applications*, vol. 35, vo. 3, pp. 1246-1251, Oct 2008.

[12] P. Guo, M. Zhang, Y. Jiang, J. Ren, "Policy-based QoS control using call admission control and SVM," in *Proc. of Pervasive Computing and Applications, ICPCA 2007,* pp. 685-688, July 2007.

[13] A. Bashar, G. Parr, S. McClean, B. Scotney, and D. Nauck, "Knowledge discovery using Bayesian network framework for intelligent telecommunication network management," in *Proc. of 4th International Conference on Knowledge Science, Engineering and Management, (KSEM 2010)*, Springer LNAI, vol. 6291, pp. 518-529, Sep 2010.

[14] R. Zhang, and A. Bivens, "Comparing the use of Bayesian networks and neural networks in response time modeling for service-oriented systems," in *Proc. of ACM Workshop on Service-Oriented Computing Performance, SOCP 2007*, pp. 67-74, June 2007.

[15] S. Haykin, *Neural networks and learning machines*, 3rd Ed., Pearson Co., NJ, USA, 2009.

[16] F. Jensen, *Bayesian Networks and Decision Graphs*, 2nd Ed., Springer Co., NY, USA, 2007.