TAILING: Tail Distribution Forecasting of Packet Delays using Quantile Regression Neural Networks

Ralf Lübben
Flensburg University of Applied Sciences, Germany
ralf.luebben@hs-flensburg.de

Amr Rizk
University of Duisburg-Essen, Germany
amr.rizk@uni-due.de

***Abstract*—Major building blocks of communication networks such as flow control and congestion control rely on fresh estimates of the network state to control data traffic injection into the network. These measured metrics are usually implicitly considered as estimates of the future network state until updated. In this paper, we propose to directly and explicitly estimate packet-based predictive QoS metrics from network measurements. As many applications possess strict QoS requirements, we focus here on bounding packet delay quantiles. Our approach is based on training neural networks to predict the quantile of the delay distribution observed by future packets given some observations of packet delays. We validate our approach through recovering classical closed-form delay quantiles that are obtained from analytical models of simple queueing systems. We show that our approach goes beyond these simple models in that it provides quantile estimates for complex scenarios and under various traffic patterns including empirical data traffic traces.**

I. Introduction

Flow and congestion control are essential building blocks for data communication networks. End host often require robust mechanisms to adapt their packet sending rates to the available service of communication links to satisfy application requirements. For example, distributed applications such as Cooperative Adaptive Cruise Control (CACC) [1] that combine trajectory planning and sensory information fusion, possess not only delay, throughput and loss requirements but also establish requirements on the spatial fulfillment of these conditions. Evidently, providing QoS guarantees at dense intersections or complex traffic situations is fundamentally different from counterparts in open highway scenarios. Such distributed applications with strict QoS constrains motivate the goal of this work, i.e., *to predict QoS conditions from network measurements.* In particular, we are interested in predicting tails of QoS conditions, not averages, due to the strict nature of the application constraints.

In this paper, we consider the problem of providing timely QoS metric predictions and provide corresponding preliminary results. To enable a robust QoS forecasting method, we evaluate in this work the applicability of a machine learning model, specifically, quantile regression neural networks, see [2], [3]. These are used to achieve reliable estimates for packet delays in form of forecasting quantiles such as $P[\text{delay} > x] \leq \varepsilon$, whereas ε is typically small, but depends on the application. We note that such probabilistic expressions may be used in future work within flow and congestion control or traffic engineering and optimization in software defined networks (SDN). Here, we first evaluate the first building block, namely, the forecasting of delay quantiles of some measured data traffic flow. Our evaluation relies on synthetic traffic traces and latency measurements to benchmark against well-known performance bounds, additionally, we also apply our prediction approach to real-world traffic estimates provided by the measurement lab (MLAB) [4]. Besides end-to-end delay prediction, our approach can be flexibly applied to scenarios in which traffic and delay measurements are available, which may comprise software defined networks with integrated probing nodes and thereby 5G and upcoming 6G mobile networks that rely on softwaritization of networks and related sophisticated traffic monitoring solutions. For details on related monitoring frameworks, we refer to [5], [6].

Our approach is related to adaptation schemes that derive their transmission rate from system models such as the one used in TCP BBR and congestion control algorithms for real-time communication [7], [8], [9]. The approach is also related but goes beyond methods that use asymptotic models that describe the steady state of a network link of a path. For example, analytical models describe the steady state behavior of certain link or path metrics such as the sojourn and waiting times. Similar models can be found in [10], [11] to describe the available bandwidth as an average value. Here, we use such analytical models to verify that the provided data-driven approach delivers congruent results, however, we go beyond such models to obtain tail forecast values that are very hard to obtain in an analytical fashion.

In this work, we are bridging time-series prediction with the description of the forecast as a probabilistic bound. Essentially, we estimate an upper bound for the probability that the delay of the $n + k$th packet overshoots a delay quantile $W^{\varepsilon}(n + k)$ conditioned on prior observed packet delays. The use of neural networks for the prediction allows for a non-parametric modeling, which drops assumptions on concrete packet arrivals, the link services processes as well as on the error terms in a classical time series model. The model further allows the evaluation of input features beyond the past packet delays, e.g., given a prediction of future packet arrival or path information to improve the prediction.

The outline of the paper is as follows: Section II illustrates our system model and the approach towards the estimation of delay quantiles. Section III links the approach to simple system models for which closed-form analytical results are

Accepted IEEE ICC 2023

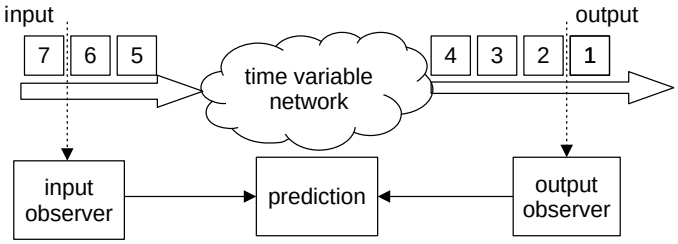


Fig. 1. System model for delay prediction: packet timestamps are observed at the input and output of the system to predict the delay quantile for future arriving packets.

known. Section IV shows and discusses delay predictions for complex communication systems fed with synthetic as well as real-world traffic data. Section V reviews the related work. Sec. VI concludes this paper with a discussion of strengths and limitations.

II. Problem Description and Approach

We consider the system depicted in Fig. 1, where observers obtain timestamps at ingress and egress of a end-to-end network path or segment and hence delays for individually transmitted packets. The realization of such a system can be implemented, for example, by smart network interface cards that provide time-synchronization and high-speed packet capturing, through in-network telemetry capable switches or, as we demonstrate in Sec. IV-B, by round trip time (RTT) measurements. The delay of packet n is described as

$$W(n) = T_D(n) - T_A(n) \tag{1}$$

where $T_D(n), T_A(n)$ are the departure time and the arrival time of this packet, respectively. The task of the observers and the prediction is to take these delay samples and provide a conditional upper bound on the delay quantile $W^\varepsilon(n+k)$ for the $n+k$th packet of the form

$$P[W(n+k) > W^\varepsilon(n+k)|W(n),\ldots,W(1)] \leq 1-\varepsilon \tag{2}$$

for $k \geq 1$. Note that the estimate sought here is pointwise in the sense that it holds for a predefined future packet with index k.

In the following, we take an empirical approach towards the delay quantile estimation problem above. In contrast to purely analytical approaches as discussed e.g. in [12], we design a supervised and offline machine learning approach for the prediction of $W^\varepsilon(n)$. In detail, we train a neural network (NN)[1] to predict the delay quantile from past delays and information on arrival traffic. We differentiate two input feature sets and train a neural network in which only delay measurements are known and a second network in which future arrivals times are known in addition. We expect that knowing or estimating the future arrivals to the network helps improving the delay quantile prediction.

[1]The code is available at https://gitlab.com/ralfluebben/tailing

TABLE I
Input features

Parameter	Values
delays of pkt#	$[[300:400),[399:400)]$
interarrival time of pkt#	$[\{\emptyset\},[400:600)]$

TABLE II
Hyperparameters used for training of the DNN

Parameter	Value/Setting
number of layers	$[1,2,3]$
neurons per layer 1	$[10,20,30,40,100,200]$
neurons per layer 2	$[0,10,20,30,40,100]$
neurons per layer 3	$[0,10,20,30,40]$
learning rate	$[\text{adaptive},0.01,0.001,0.0001]$
l2 regularization	$[0.01,0.001]$
drop out	$[0.0,0.5]$
optimizer	adam
epochs	600 with early stopping
batch size	2048

To estimate the packet delay quantiles, we train a quantile regression neuronal network using the pinball loss function

$$L(y,\hat{y}) = \begin{cases} (1-\varepsilon)(\hat{y}-y) & \text{if } y < \hat{y} \\ \varepsilon(y-\hat{y}) & \text{if } y \geq \hat{y} \end{cases} \tag{3}$$

We minimize the expectation of this loss function with respect to the unknown delay distribution that provides the ε-quantile [13], [2], [3]. By the use of this loss function, the neural network learns to predict the ε-quantile of the delay. We perform a hyperparameter optimization using the parameters listed in Tab. II for a deep feed forward neural network (DNN) architecture and parameters given in Tab. III for a long short term memory (LSTM) neural network. The selection of the hyperparameters is based on the Hyperband search [14]. Since the search algorithm selects hyperparameters based on the progress of the optimization criteria on a small number of epochs, we perform the optimization for each learning rate individually. The rationale here is that a small learning rate may have a slower progress than a high learning rate but may still perform better for a high number of epochs. To benchmark the predictions of the trained NN, we use, in addition to the pinball loss function that leads to the prediction of the statistical upper bound, two further metrics. Firstly, the mean absolute error, in the following denoted as *distance*, which returns the absolute distance (difference) between the predicted bound and the measured delay values as we seek a tight bound, i.e., a small mean absolute error. Secondly, we compare the predicted quantile to empirical quantiles, i.e., the empirical quantile is extracted packet-wise for each packet number from all available sample paths of delay traces.

III. Comparison with analytical results

In the following, we empirically show that the estimates of the quantile estimation approach from the previous section coincide with analytical results obtained for tractable examples.

TABLE III
HYPERPARAMETERS USED FOR TRAINING OF THE LSTM NN

Parameter	Value/Setting
number of cells	$[10, 20, 30, 40, 100, 200]$
learning rate	[adaptive, $0.01, 0.001, 0.0001$]
drop out	$[0.0, 0.5]$
optimizer	adam
epochs	200 with early stopping
batch size	2048

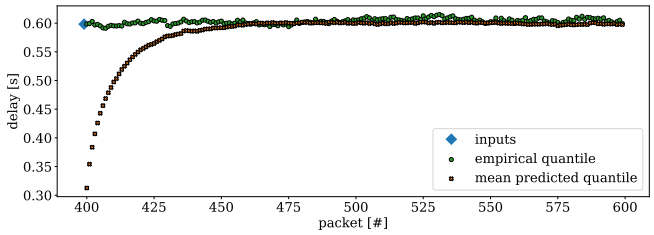


Fig. 2. M/M/1 system: Empirical delay quantile vs. mean predicted quantile for delay input sample of packet # 399.

A. M/M/1 System

We start with an example of a well understood queueing system, the M/M/1 system, with one server having exponentially distributed service times with parameter μ, as well as, exponentially distributed inter-packet arrival times with parameter λ. It is known that an M/M/1 queueing system has a steady-state response time distribution of

$$P[W > a\overline{W}] = e^{-a} \tag{4}$$

with expected response time $\overline{W} = \frac{1}{1-\rho}\frac{1}{\mu}$ and the shorthand notation $\rho := \lambda/\mu$ for $a \geq 0$. Now, fixing the violation probability $e^{-a} = 1 - \epsilon$, our approach estimates the corresponding delay quantile $a\overline{W}$, which we denoted above as W^{ϵ}. For $1 - \epsilon = 0.95$, the related response time quantile is 0.599.

To validate the empirical quantile estimation approach we use training data generated from simulations and compare the quantile estimate $W^{\epsilon}(n)$ of packet n to the analytical delay quantile $a\overline{W}$. We obtain simulation data using the discrete event simulator Omnet++ where we simulate an M/M/1 queueing system and record arrival and departure times $T_A(n), T_D(n)$ to compute packet delays.

We simulate $3 \cdot 10^5$ packet traces for different utilizations. We split the data into a training, validation, and test set. The training set comprises 80%, the validation set 10%, and the test set 10% of the traces. We use the delay of packets in the range $[n - r, n)$, where $r \geq 1$ specifies the length of the input feature sequence, to predict future delay quantiles for the packets in the range $[n, n+f)$ for $f > 0$. Note that the quantile prediction provided in Sec. II is point-wise, i.e., we obtain one prediction for a certain packet index. Hence, the comparison with the analytical steady state quantile $a\overline{W}$ from (4) is strictly only meaningful for packets far enough in future such that they can be considered in steady state. We select $n = 400$ to assume steady state delays, $f = 200$ for a sufficiently large prediction interval, and $r = 1$ and $r = 100$, respectively.

For our estimation approach, we use the input features shown in Tab. I, i.e., solely the packet delays and in comparison the combination of the packet delays and known future packet arrival time points. Tab. V in the appendix shows the optimal hyperparameters, related quantiles, and distance metrics of the validation and test sets.

Fig. 2 compares the empirical packet-wise quantiles W^{ϵ} to the mean predicted quantile for the test data set. Recall that the analytical value for this example accounts to $a\overline{W} = 0.599$. The prediction series converges to the analytical and empirical quantile quickly. We only show results for the deep neural network architecture here since the LSTM network shows a very similar performance.

Overall, we find that the training creates neural networks that are able to predict packet delays for that the quantile condition Eq. (2) holds. We also note that the prediction improves if knowledge or estimates of future arrivals, i.e., the next packet interarrival times, are included for the prediction (not shown here). The delay quantile is still correctly predicted according to the given ε, but the distance decreases, i.e., the prediction returns a tighter bound. The improvement using this additional information is relevant for applications which influence future packet arrivals, e.g., through selecting videos qualities to be transmitted, selecting sensor status sending times, or encoder settings in video streaming scenarios.

IV. DELAY PREDICTIONS: QUEUEING SYSTEMS AND EMPIRICAL TRACES

Next, we evaluate the delay quantile prediction approach described before on different systems, starting with synthetic queueing systems to empirical network data traces.

A. Synthetic queueing systems

First, we show the predictions of packet delays for queueing systems with service and interarrival times drawn from a Weibull distribution (denoted as W), i.e., we perform experiment for a M/W/1 system and a W/W/1 system. The Weibull distribution leads to a slower than exponential tail of the service or interarrival times leading to an intuitive increase of the delay quantiles. The optimal parameters for the neural networks after training are given in Tab. VI and Tab. VII in the appendix for the M/W/1 and W/W/1 system, respectively. Again, only results for the DNN architecture are shown since the LSTM architecture performs similarly. We observe that for all variants of the input features, the training generates neural networks that provide an empirically valid prediction. Again, the inclusion of information of the arrivals leads to tighter bounds.

Fig. 3 shows a strong congruence of the mean of the predicted delay quantiles and the empirical quantiles for a W/W/1 queueing system. In comparison to the light tailed service and interarrival times in the example in Fig. 2, we observe that the predicted quantiles here converge slower to their empirical steady state counterparts. Note that, the

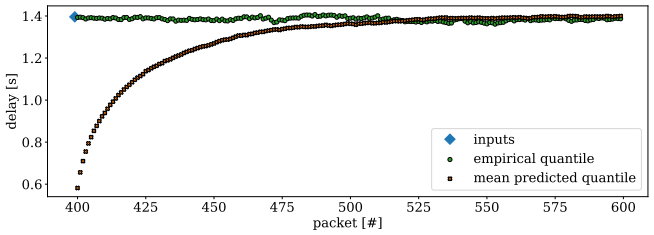


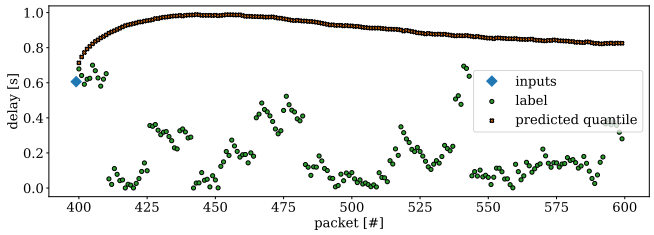
Fig. 3. W/W/1 system: Empirical delay quantile vs. mean predicted quantile. Input features are the delay samples for packets # $[399 : 400)$. Model parameters: Scale σ and shape k as $\sigma = 1.5, k = 0.6647$ (interarrival times) and $\sigma = 0.0375, k = 0.6647$ (service times). The utilization is $\varrho = 0.75$ and the violation probability for the quantile estimates is $\varepsilon = 0.05$.

distance values given in Tab. V to Tab. VII allow only for a comparison between results related to one specific queueing system, i.e., to compare the tightness of the predicted bound of that specific system. Since different queueing systems such as M/M/1 and W/W/1 exhibit a different burstiness and thereby different upper bounds for the same value of ε the distance metric is not comparable between different systems.

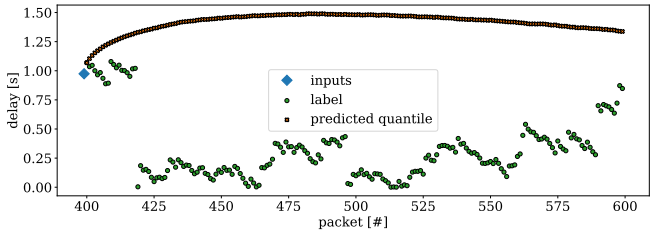
Next, we extend the prediction to systems where the packet arrivals are not independently and identically distributed (iid.) but come from an ON-OFF Markov source where in the ON state (state 1) the source produces packets at a constant rate P and in the OFF state (state 2) the packet arrivals stop. The packet arrival stream is characterized through three parameters, i.e., the probability to change states P_{ij}, i.e., moving from state i to state j and the mean time T to change states twice [12]. The latter metric is usually considered as proxy for the burstiness of the flow.

We train the neural networks and optimize the hyperparameters from Tab. II and Tab. III, except for the learning rate which we fix to the adaptive rate. We configure the packet arrival stream such that the packet rate in the ON state is 20 packets per second and the probability to be in the ON state $p_{ON} = \frac{p_{21}}{p_{12}+p_{21}} = 0.75$. The mean time to change states twice is set to 5 seconds. The service time increments are exponentially distributed with mean of 0.05 seconds. As input feature sequence, we select $r = 100$ and $r = 1$, i.e. the delay sample ranges are from packets $[300 : 400)$ and $[399 : 400)$ for comparison. For both input features we obtain valid quantiles of 0.051 and corresponding test performance of 0.05 and 0.055, respectively. We omit the hyperparameters here, and again the DNN and LSTM network perform similarly good. Fig. 4 shows examples of sampled delay traces in comparison to the predicted bound. Note that we previously presented in Fig. 2 and Fig. 3 the empirical delay quantile of the traces in comparison to the predicted bound, here we present the sampled delays. The samples in theses sub-figures differ in their burst durations. The neural network predicts delay series that incorporate these different burst period length, i.e., the rising curve for a larger burstiness lasts longer in Fig. 4b.

In the previous experiments, we predicted the quantiles for networks with iid. samples or memoryless queueing systems.



(a) Burstiness T=50



(b) Burstiness T=100

Fig. 4. Delay quantile predictions for a exemplary trace of Markov ON-OFF packet arrivals with burstiness T: The quantile estimator learns the burstiness property. The rising curve for a larger burstiness lasts longer.

Intuitively, a neural network cannot improve its prediction from longer input sequences, since no additional information is contained in longer input sequences. Additional experiments, not shown in this work, with more complex, non-iid., or non-memoryless traffic and service patterns show an improvement using a LSTM neural network architecture and longer input trains.

B. Trace-based Evaluation

The experiments in the previous sections use synthetic stationary data traces. In contrast, empirical data traces do not show classical statistical properties regarding distribution and load. Hence, we consider next a data set containing empirical traces from the MLAB. Instead of one-way delays, we use round trip time measurements as training input since clock synchronization cannot be guaranteed in these empirical data sets.

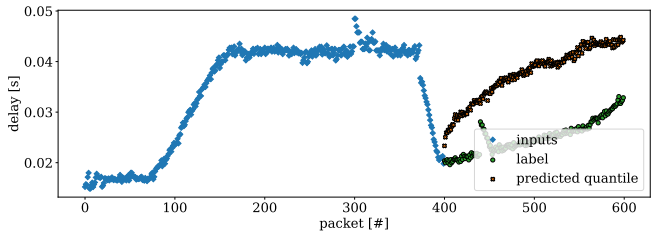
We extract $2 \cdot 10^5$ RTT samples from MLAB pcap-files from Nov. 2020 from one MLab measurement server in Hamburg, Germany. To obtain non-client specific predictions we train a delay quantile prediction neural network based on measurements from all clients connected to this server. We train a DNN and LSTM network as explained above to predict the RTT. Tab. IV presents the results after hyperparameter optimization. Both architectures (DNN and LSTM) provide predictions that fulfill the quantile definition. We note that using longer input trains tightens the prediction. Fig. 5 shows delay quantile predictions for exemplary traces using the LSTM model for round-trip time delays obtained from the MLAB traces.

V. Related Work

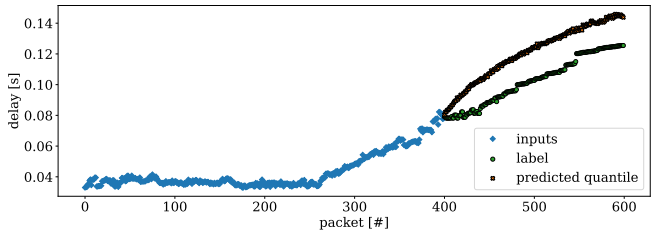
Forecasting the traffic behavior in computer networks using machine learning is successfully applied in various works for

TABLE IV
MLAB DATA

input features		hyperparameters							distance		quantile	
delay	interarrivals	arch	units 1	units 2	units 3	dropout	lear. rate	l2 reg	validation	test	validation	test
[399 : 400)	∅	Dense	200	0	0	0.0	adaptive	0.001	0.227	0.23	0.05	0.049
[300 : 400)	∅	Dense	200	0	0	0.0	adaptive	0.001	0.219	0.22	0.052	0.053
[0 : 400)	∅	Dense	30	40	0	0.0	adaptive	0.001	0.218	0.218	0.054	0.055
[399 : 400)	∅	LSTM	200	0	0	0.0	adaptive	0.001	0.225	0.228	0.052	0.05
[300 : 400)	∅	LSTM	70	0	0	0.0	adaptive	0.001	0.219	0.219	0.05	0.049
[0 : 400)	∅	LSTM	70	0	0	0.0	adaptive	0.001	0.219	0.219	0.051	0.052



(a) Example trace 1



(b) Example trace 2

Fig. 5. Delay quantile predictions using the LSTM model for round-trip time delays obtained from MLAB traces.

which we refer to the surveys [15], [16]. Here, we bring into focus approaches that target forecasting the QoS parameters on end-hosts related to our end-to-end prediction of delay quantiles and examples in which the prediction of delays is applied in non end-to-end use cases.

Often, forecasting or estimation are applied to throughput prediction or related to adaptive bitrate (ABR) algorithms in streaming applications to select variations of chunks in a video stream of different quality and thereby size, i.e. low quality chunks require less throughput. Typically, the optimization goal is to transmit the highest quality chunk so that it arrives before the content is played out to avoid audio or video stalls. In [17], an ABR algorithm for video streaming is proposed using reinforcement learning which was based among other parameters on download times and measured network throughput. Also in [18] a reinforcement learning approach is applied to ABR to optimize the quality of experience. In [19], a neural network is used to predict the transmission time, which is used to select a suitable chunk in an ABR algorithm. These works go essentially back to [7] in which different throughput predictors are proposed that are not based on machine learning.

A bandwidth prediction approach that outputs quantiles of the expected bandwidth at geolocations in automotive scenarios using physical layer, data link layer, speed, traffic, and weather information is described in [20]. Similarly, passive probing parameters from lower network layers are used in [21] to predict the mean end-to-end latency in automotive scenarios.

An online throughput prediction for ABR selection in cellular networks is illustrated in [22] that further includes a prediction of the user's environment such as public transport, indoor, and open air environments, since characteristics of cellular networks differ strongly in these environments.

The application and advantage of machine learning to the field of available bandwidth estimation is shown in [23], [24], [25]. Further applications may be to estimate the link service as it can be inferred from delay measurements, see e.g. [26], [27].

Often a forecast is directly integrated into transport layer protocols, typically, for congestion control. In [9], the sender predicts a sending rate, so that packets arrive with a delay below a certain value with high probability. Also the congestion control designed in [28] uses machine learning to optimize the sending rate under delay constraints. In [29] congestion control algorithms are designed automatically by training.

For prediction and optimization, machine learning is applied to SDNs, the survey [30] classifies and summarizes various approaches. Here, we highlight approaches that focus on the delay prediction. The authors in [31] envision the prediction of delays for the optimization of SDNs topology and show that delay can be predicted with a small error, especially the mean end-to-end delay is predicted for various scenarios including variations in topology, network size, traffic distribution, traffic intensity, and routing configurations. Also in [32], [33], [34] neural networks, specifically, graph neural networks, are trained to predict performance indicators such as throughput, delay, and jitter for network topologies with input parameters such as traffic, topology, and routing configuration.

Closely related to our work is [35] in which QoS distributions for, e.g., the delay are predicted from traffic samples by a conditional variational autoencoder neural network. We also envision a stochastic description of the delay, but instead of using the distribution as input feature during the training, we rely on quantile regression to directly return a stochastic estimate of the delay.

The reviewed related work shows the advantages of the application of machine learning to the estimation of end-to-

end QoS parameters. Often, the approaches comprise throughput estimation in conjunction with ABR algorithms or with congestion control protocols for end-to-end approaches. Delay prediction is among other performance parameters used for optimization in SDNs. Complementary, we present the prediction of delay quantiles for end-to-end traffic flows using quantile regression.

VI. Conclusion and Discussion

This work uses quantile regression neural networks to reliably estimate quantiles of packet sojourn times in communication systems. In general, the empirical results show that a neural network is able to provide valid delay quantile estimates. First, we show that neural networks recover classical results from queueing theory on delay quantiles. Further, we show that they are also able to provide results in more complex scenarios with varying load, mixed arrival, and service processes. This underpins the applicability of the presented approach to computer networks, which typically feature more complex structures with unknown arrival and service process parameters. We finally show an application of these estimates in the context of real-world data traces obtained from MLAB.

Limitations of this work include that the quantile predictions are point-wise and not sample path predictions. Future work comprises the illumination of the relationships between structures of the queueing elements in the network to provide more complex forecasting for traffic optimization.

Acknowledgements

The work of A. Rizk is part of the 5G-IANA project that has received funding from the European Union Horizon 2020 research and innovation program under grant agreement No. 101016427.

Appendix

Tab. V, Tab. VI, and Tab. VII display the selected parameters after hyperparameter tuning and the distance as well as the empirical quantile for the queueing systems M/M/1, M/W/1, and W/W/1. For each system four experiments were conducted with a short and long delay sequence as well as inter-arrivals of upcoming packets. For all systems a valid quantile is predicted. Note that information on upcoming packet arrivals improve the prediction. The experiments were conducted with a DNN and LSTM architecture, which perform similarly, where we only show the results of the DNN architecture.

References

[1] D. F. Külzer, M. Kasparick, A. Palaios, R. Sattiraju, O. D. Ramos-Cantor, D. Wieruch, H. Tchouankem, F. Göttsch, P. Geuer, J. Schwardmann, G. Fettweis, H. D. Schotten, and S. Stanczak, "AI4Mobile: Use Cases and Challenges of AI-based QoS Prediction for High-Mobility Scenarios," in *Proc. IEEE VTC Spring*, Apr. 2021.

[2] R. Koenker, A. Chesher, and M. Jackson, *Quantile Regression*, ser. Econometric Society Monographs. Cambridge University Press, 2005.

[3] J. W. Taylor, "A Quantile Regression Neural Network Approach to Estimating the Conditional Density of Multiperiod Returns," *Journal of Forecasting*, vol. 19, no. 4, pp. 299–311, 2000.

[4] Measurement Lab, "The M-Lab NDT data set," https://measurementlab.net/tests/ndt.

[5] D. Bhamare, A. Kassler, J. Vestin, M. A. Khoshkholghi, J. Taheri, T. Mahmoodi, P. Öhlén, and C. Curescu, "IntOpt: In-band Network Telemetry optimization framework to monitor network slices using P4," *Computer Networks*, vol. 216, 2022.

[6] D. Scano, F. Paolucci, K. Kondepu, A. Sgambelluri, L. Valcarenghi, and F. Cugini, "Extending P4 in-band telemetry to user equipment for latency- and localization-aware autonomous networking with AI forecasting," *IEEE J. Opt. Commun. Netw.*, vol. 13, pp. D103–D114, 2021.

[7] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 325–338, Aug. 2015.

[8] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-Optimal Bitrate Adaptation for Online Videos," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1698–1711, 2020.

[9] K. Winstein, A. Sivaraman, and H. Balakrishnan, "Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks," in *Proc. USENIX NSDI*, Apr. 2013, pp. 459–471.

[10] X. Liu, K. Ravindran, and D. Loguinov, "A Queueing-Theoretic Foundation of Available Bandwidth Estimation: Single-Hop Analysis," *IEEE/ACM Trans. Netw.*, vol. 15, no. 4, pp. 918–931, Aug. 2007.

[11] ——, "A Stochastic Foundation of Available Bandwidth Estimation: Multi-Hop Analysis," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 130–143, Feb. 2008.

[12] M. Fidler and A. Rizk, "A Guide to the Stochastic Network Calculus," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 92–105, 2015.

[13] R. Koenker and G. Bassett, "Regression Quantiles," *Econometrica*, vol. 46, no. 1, pp. 33–50, 1978.

[14] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization," *Journal of Machine Learning Research*, vol. 18, no. 185, 2018.

[15] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," *Journal of Internet Services and Applications*, vol. 9, no. 1, 2018.

[16] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 2019.

[17] H. Mao, R. Netravali, and M. Alizadeh, "Neural Adaptive Video Streaming with Pensieve," in *Proc. SIGCOMM*, Aug. 2017, pp. 197–210.

[18] M. Gadaleta, F. Chiariotti, M. Rossi, and A. Zanella, "D-DASH: A Deep Q-Learning Framework for DASH Video Streaming," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 703–718, 2017.

[19] F. Y. Yan, H. Ayers, C. Zhu, S. Fouladi, J. Hong, K. Zhang, P. Levis, and K. Winstein, "Learning in situ: a randomized experiment in video streaming," in *Proc. USENIX NSDI*, Feb. 2020, pp. 495–511.

[20] D. Schäufele, M. Kasparick, J. Schwardmann, J. Morgenroth, and S. Stańczak, "Terminal-Side Data Rate Prediction For High-Mobility Users," in *Proc. IEEE VTC*, Apr. 2021.

[21] D. F. Külzer, F. Debbichi, S. Stańczak, and M. Botsov, "On Latency Prediction with Deep Learning and Passive Probing at High Mobility," in *Proc. IEEE ICC*, Jun. 2021.

[22] C. Qiao, G. Li, J. Wang, and Y. Liu, "NEIVA: Environment Identification based Video Bitrate Adaption in Cellular Networks," in *Proc. IEEE/ACM IWQoS*, 2019.

[23] S. K. Khangura, M. Fidler, and B. Rosenhahn, "Machine learning for measurement-based bandwidth estimation," *Computer Communications*, vol. 144, pp. 18–30, 2019.

[24] S. K. Khangura and S. Akin, "Measurement-Based Online Available Bandwidth Estimation Employing Reinforcement Learning," in *Proc. IEEE ITC*, 2019, pp. 95–103.

[25] S. K. Khangura and S. Akın, "Online Available Bandwidth Estimation using Multiclass Supervised Learning Techniques," *Computer Communications*, vol. 170, pp. 177–189, 2021.

[26] R. Lübben, M. Fidler, and J. Liebeherr, "Stochastic Bandwidth Estimation in Networks with Random Service," *IEEE/ACM Trans. Netw.*, vol. 22, no. 2, pp. 484–497, Apr. 2014.

[27] A. Rizk and M. Fidler, "On the identifiability of link service curves from end-host measurements," in *Network Control and Optimization, Euro-NF Workshop, NET-COOP*, ser. Lecture Notes in Computer Science, vol. 5425, 2008, pp. 53–61.

TABLE V
M/M/1: OPTIMAL HYPERPARAMETERS AND RELATED EMPIRICAL QUANTILES AND DISTANCES

input features		hyperparameters						distance		quantile	
delay	interarrivals	units 1	units 2	units 3	dropout	lear. rate	l2 reg	validation	test	validation	test
$[399:400)$	$\emptyset$	100	0	0	0.0	adaptive	0.001	0.601	0.601	0.049	0.051
$[300:400)$	$\emptyset$	200	0	0	0.0	adaptive	0.001	0.602	0.602	0.049	0.051
$[399:400)$	$[400:600)$	1200	0	0	0.0	adaptive	0.001	0.505	0.506	0.051	0.053
$[300:400)$	$[400:600)$	200	0	0	0.0	adaptive	0.001	0.508	0.509	0.052	0.053

TABLE VI
M/W/1: OPTIMAL HYPERPARAMETERS AND RELATED EMPIRICAL QUANTILES AND DISTANCES

input features		hyperparameters						distance		quantile	
delay	interarrivals	units 1	units 2	units 3	dropout	lear. rate	l2 reg	validation	test	validation	test
$[399:400)$	$\emptyset$	1200	0	0	0.0	adaptive	0.001	1.009	1.012	0.05	0.049
$[300:400)$	$\emptyset$	200	0	0	0.0	adaptive	0.001	1.008	1.01	0.05	0.049
$[399:400)$	$400:600$	40	0	0	0.0	adaptive	0.001	0.934	0.936	0.05	0.049
$[300:400)$	$400:600$	1200	0	0	0.0	adaptive	0.001	0.939	0.942	0.05	0.05

TABLE VII
W/W/1: OPTIMAL HYPERPARAMETERS AND RELATED EMPIRICAL QUANTILES AND DISTANCES

input features		hyperparameters						distance		quantile	
delay	interarrivals	units 1	units 2	units 3	dropout	lear. rate	l2 reg	validation	test	validation	test
$[399:400)$	$\emptyset$	1200	0	0	0.0	adaptive	0.001	1.327	1.328	0.051	0.049
$[300:400)$	$\emptyset$	1200	0	0	0.0	adaptive	0.001	1.324	1.326	0.051	0.05
$[399:400)$	$400:600$	1200	0	0	0.0	adaptive	0.001	1.137	1.139	0.051	0.05
$[300:400)$	$400:600$	1200	0	0	0.0	adaptive	0.001	1.119	1.123	0.053	0.053

[28] M. Dong, T. Meng, D. Zarchy, E. Arslan, Y. Gilad, B. Godfrey, and M. Schapira, "PCC Vivace: Online-Learning Congestion Control," in *Proc. USENIX NSDI*, Apr. 2018, pp. 343–356.

[29] A. Sivaraman, K. Winstein, P. Thaker, and H. Balakrishnan, "An experimental study of the learnability of congestion control," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 479–490, Aug. 2014.

[30] R. Etengu, S. C. Tan, L. C. Kwang, F. M. Abbou, and T. C. Chuah, "AI-Assisted Framework for Green-Routing and Load Balancing in Hybrid Software-Defined Networking: Proposal, Challenges and Future Perspective," *IEEE Access*, vol. 8, pp. 166 384–166 441, 2020.

[31] A. Mestres, E. Alarcón, Y. Ji, and A. Cabellos-Aparicio, "Understanding the modeling of computer network delays using neural networks," in *Proc. ACM Big-DAMA Workshop*, Aug. 2018, pp. 46–52.

[32] K. Rusek, J. Suárez-Varela, P. Almasan, P. Barlet-Ros, and A. Cabellos-Aparicio, "RouteNet: Leveraging Graph Neural Networks for Network Modeling and Optimization in SDN," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2260–2270, Oct. 2020.

[33] M. Ferriol-Galmés, J. Suárez-Varela, J. Paillissé, X. Shi, S. Xiao, X. Cheng, P. Barlet-Ros, and A. Cabellos-Aparicio, "Building a Digital Twin for network optimization using Graph Neural Networks," *Computer Networks*, vol. 217, 2022.

[34] B. Jaeger, M. Helm, L. Schwegmann, and G. Carle, "Modeling TCP Performance Using Graph Neural Networks," in *Proc. ACM Workshop on Graph Neural Networking (GNNet)*, 2022, pp. 18–23.

[35] S. Xiao, D. He, and Z. Gong, "Deep-Q: Traffic-Driven QoS Inference Using Deep Generative Network," in *Proc. ACM Workshop NetAI*, Aug. 2018, pp. 67–73.