KICS The Korean Institute of Communications and Information Sciences

Available online at www.sciencedirect.com
ScienceDirect

ICT Express 9 (2023) 247–252

www.elsevier.com/locate/icte

LSTM-based throughput prediction for LTE networks

Hyeonjun Na[a], Yongjoo Shin[b], Dongwon Lee[b], Joohyun Lee[b,*]

[a] Department of Applied Artificial Intelligence, Hanyang University, Ansan, South Korea
[b] Department of Electrical and Electronic Engineering, Hanyang University, Ansan, South Korea

Received 15 September 2021; received in revised form 12 November 2021; accepted 5 December 2021
Available online 13 December 2021

Abstract

Throughput prediction is crucial for reducing latency in time-critical services. We study the attention-based LSTM model for predicting future throughput. First, we collected the TCP logs and throughputs in LTE networks and transformed them using CUBIC and BBR trace log data. Then, we use the sliding window method to create input data for the prediction model. Finally, we trained the LSTM model with an attention mechanism. In the experiment, the proposed method shows lower normalized RMSEs than the other method.

Keywords: Machine learning; Deep learning; Throughput prediction; LSTM; Attention method

1. Introduction

Throughput prediction plays an important role in real-time video and audio streaming services that require low-latency network protocols.[1] For example, the throughput prediction can be a solution that can improve quality in video streaming applications such as YouTube or NetFlix [2]. Current video streaming service determines video quality depending on the network quality. Therefore, the throughput prediction is required to guarantee stable Quality of Service and Experience (QoS/QoS). In addition, the throughput prediction can assist file downloading by efficiently scheduling downloads [3]. Previous throughput estimation methods used exponential moving average, arithmetic mean, and the last sample. However, these methods are not sufficiently accurate and thus we need a novel method [4].

The throughput in wireless networks quickly fluctuates by numerous random factors. The factors that cause fluctuation are very diverse, and it is too difficult to find the relationship between the factors and throughput using traditional models [5]. Therefore, various methods are being studied to improve the prediction performance, e.g., integrating the data-based and image-based prediction models [6].

In this paper, we investigate how machine and deep learning can assist precise throughput prediction and classify the previous studies that use learning models such as random forest (RF), support vector regression (SVR), and long short-term memory (LSTM). Then, we propose a prediction system for network throughput using LSTM with the attention method. Our prediction system consists of three processes: Collection, Training, and Evaluation. We first collect the TCP log data traces for two congestion control algorithms, BBR and CUBIC, and convert them into throughput data. The throughput data is divided into training and test sets. In training, we use LSTM using the attention method as the prediction model that approximates a mapping function from previous samples to a future throughput. Here, we use the sliding window method for efficient learning and prediction. To evaluate the learning models, we calculate the root mean squared error (RMSE) between the true throughput values and the predicted values in the test set. The prediction results show that our prediction system achieves lower RMSE in BBR and CUBIC throughput traces, compared to the existing methods, Seq2Seq and LSTM without the attention mechanism.

2. Background

2.1. Related work on throughput prediction

In this subsection, we describe several algorithms that are mainly used for throughput prediction in the literature. We

* Corresponding author.
E-mail addresses: hfdjkfkl@hanyang.ac.kr (H. Na), superpika@hanyang.ac.kr (Y. Shin), dw2689@hanyang.ac.kr (D. Lee), joohyunlee@hanyang.ac.kr (J. Lee).

[1] This paper is an extended version of our ICUFN paper [1].
Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2021.12.001

summarize the prediction models used in each literature into three categories in Table 1.

(1) Random Forest (RF): The RF algorithm is an ensemble/boosting learning method for regression and classification tasks [7]. In [8], Minovski et al. selected RF and SVR for throughput prediction. In terms of R^2 and the mean squared error (MSE), RF achieved better performance compared to those of SVR. In [9], Schmid et al. used RF with important features such as Reference Signal Strength Indicator (RSSI), and Round Trip Time (RTT). Khan et al. proposed an RF model for real-time throughput prediction with different numbers of features [10].

(2) Support Vector Regression (SVR): SVR is a version of a support vector machine (SVM) for regression [11]. It follows linear regression, where the output value has a linear relationship with the input value. Wei et al. [12] applied SVR to model the relationship between the future throughput and channel quality indicator (CQI). They concluded that the SVR model with parameters of throughput and RSSI predicts better than the SVR model with a single parameter of throughput or RSSI.

(3) Long Short-Term Memory (LSTM): LSTM is an improved model of recurrent neural network (RNN). LSTM achieves a lower value of absolute value of relative error (ARE) than other two algorithms, RF and SVR, indicating better learning ability for patterns, where ARE is the ratio of the absolute prediction error to the actual throughput value [13]. Schmid et al. [14] applied a two-layer bidirectional LSTM model with the input features of throughput . They compared the performance of the proposed LSTM model with SVR, RF, and feedforward neural network models. They found that their LSTM model is robust to dynamic fluctuation and thus results in the best prediction accuracy.

In the previous work [14], the LSTM outperforms all other algorithms, RF and SVR. In [8], it is shown that RF achieved a better outcome compared to SVR in terms of MSE and R^2. Narayanan et al. [15] identified the different factors that affected the prediction performance and proposed a context-aware throughput prediction framework based on gradient decision boosted trees (GDBT) and sequence-to-sequence (Seq2Seq). In this study, Seq2Seq performed better than GDBT. In [16], Kasongo et al. proposed method for Wireless IDS was based on deep long short-term memory (DLSTM) using a filter based algorithm based on information gain (IG). The DLSTM approach was compared to RF and SVM. The results indicated that the DLSTM model performs better than RF and SVM. Therefore, we can conclude that the LSTM and Seq2Seq models performed better than other methods such as GDBT, RF, and SVR.

2.2. Performance metrics

There are various metrics to evaluate the prediction accuracy of the model. We summarize which metric is used in each literature in Table 2. These metrics are generally based on the size of the error, the difference between the estimated and the true value. In general, there exist a lot of estimated values in one prediction set, so it is better to summarize each error into a single value. One common solution is averaging these errors, such as MAE and RMSE. For output value y and estimated value $\hat{y}$, they are defined as follows:

$$\text{MAE} = \frac{1}{n}\sum_{n}^{i=1}\left|y_i - \hat{y}_i\right|, \qquad \text{RMSE} = \sqrt{\frac{1}{n}\sum_{n}^{i=1}\left|y_i - \hat{y}_i\right|^2} \tag{1}$$

The reason for using squared error is that it is easy to detect large errors. The other metrics denote error as a ratio or percentage such as the mean absolute percentage error (MAPE) and R^2 score. They are defined as follows:

$$\text{MAPE} = \frac{1}{n}\sum_{n}^{i=1}\left|\frac{y_i - \hat{y}_i}{y_i}\right|, \qquad R^2 = 1 - \frac{\text{MSE}}{{\sigma_y}^2}, \tag{2}$$

where MSE is the mean squared error, the square of RMSE, and ${\sigma_y}^2$ is the variance of y. Since MAE, RMSE, and MAPE compute the amount of error, these metrics should be smaller for higher accuracy. On the other hand, a higher R^2 score implies the higher accuracy of the model. In addition to these metrics, relative error is also used for performance metrics. In [14], the mean relative error (MRE) and mean squared relative error (MSRE) are also used. MRE and MSRE are suitable for finding severe outliers in the prediction data.

Table 1
Summary of prediction models for network throughput.

Reference	Prediction Model			
	RF	SVR	LSTM	GDBT
[4]	○			
[8–10]	○	○		
[13]	○		○	
[14,16]	○	○	○	
[15]			○	○

Table 2
Summary of performance metrics in the literature.

Ref.	Performance Metric				
	MAE	MSE (RMSE)	MRE	MSRE (RMSRE)	R^2
[4]					○
[8]		○			○
[9]	○		○	○	
[6,10]	○	○			○
[13]		○			
[14]	○	○	○	○	
[15]	○	○			

3. Proposed prediction system

3.1. Prediction problem

We denote the network throughput data as $Y = \{y_1, y_2, y_3, \ldots, y_T\}$, which is a series of non-negative values for time slots $\mathcal{T} = \{1, 2, 3, \ldots, T\}$. To use data for model training and evaluation, we define the training set as $X = \{y_1, y_2, y_3, \ldots, y_k\}$, where $k \in \mathcal{T}$, and the test set as $Z = \{y_{k+1}, y_{k+2}, y_{k+3}, \ldots, y_T\}$. The goal of the prediction problem is to estimate Z and minimize the prediction error.

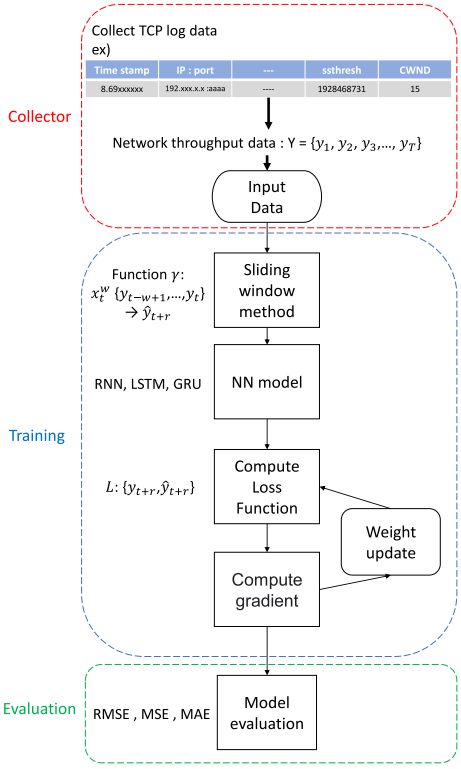

Fig. 1. Data prediction system using a neural network model.

3.2. Prediction method using sliding window

Fig. 1 shows the schematic representation of the prediction system using a recurrent neural network. To train and predict the network throughput efficiently, we use the sliding window method. We first denote $x_t^\omega = \{y_{t-\omega+1}, y_{t-\omega+2}, y_{t-\omega+3}, \ldots, y_t\}$ as ω consecutive samples at each time $t = \omega, \ldots, T - r$. We define a mapping function γ such that $\gamma(x_t^\omega) = y_{t+r}$, for $t = \omega, \ldots, k - r$. In other words, this function outputs the future throughput at time $t+r$ using the throughput data from $t-\omega+1$ to t, where r is the length of time slots until the target time slot. Our goal is to obtain an approximated function $\hat{\gamma}$ using the training set X or x_t^ω for $t = \omega, \ldots, k - r$, using a neural network. We also denote $\hat{y}_{t+r} = \hat{\gamma}(x_t^\omega)$.

The function $\hat{\gamma}$ is modeled by a neural network with ω input values or x_t^ω and one output value $\hat{y}_{t+r}$. For the training process, we first define the loss function L that calculates the error (e.g., MSE) between the true throughputs y_{t+r} and the estimated throughputs $\hat{y}_{t+r}$ for $t = \omega, \ldots, k - r$ in the training set. Using the loss function and the prediction errors, we compute the gradient with backpropagation and update the weights of the neural network. We repeat this process until the loss is below a given threshold.

After the training process, we evaluate the performance using the test set Z. To evaluate the prediction performance of the approximated function $\hat{\gamma}$ using the test set Z, we calculate RMSE as Eq. (1).

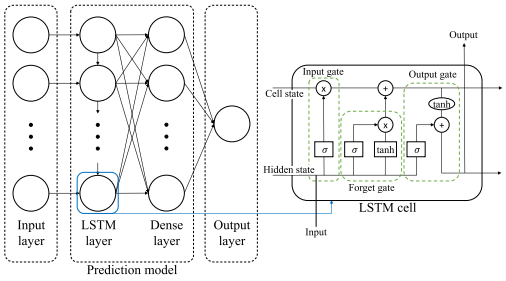

Fig. 2. LSTM neural network model [17].

3.3. Neural network models for prediction

The Recurrent neural network (RNN) is a sequence neural network model required for the data prediction system mentioned in Section 3.2. This model has the characteristic of sending the result value from the activation function in the hidden layer to the output layer and sending it back to the input of the next calculation in the hidden layer. This feature has the advantage of learning data patterns compared to the multi-layer perceptron model (MLP) in which values are transmitted feedforward. However, as the time step gets longer, the activation function is applied redundantly and the value is lost, causing the vanishing gradient problem.

LSTM [18] is one of the RNN models designed to overcome the disadvantage of the original RNN, which is illustrated in Fig. 2. This model has a structure in which the cell state is added to the hidden layer of the RNN, and it solves the vanishing gradient problem by controlling the cell state with three gates: input gate, forget gate, and output gate [17] and transferring the state to the next layer while maintaining the existing state.

The Seq2Seq model [19] is a combination of the RNN series models, including the LSTM model. The input part acts as an encoder that compresses time series data into vectors, and the output part acts as a decoder that converts the compressed vectors back into time-series data. In the process of transformation, the model learns the data distribution in the latent vector space. This Seq2Seq model is mainly used in natural language data and is also used to detect outliers in time series data. However, there is a major problem with the Seq2Seq model. When the model tries to compress all the information into one fixed-size vector, it loses information. In other words, the longer the input data, the lower the output quality.

As an alternative, the Attention mechanism [20] has emerged to compensate for the loss of accuracy of the output sequence when the input sequence has various lengths. The basic idea of attention method is that when the decoder predicts the output at each time step, instead of referring to all the inputs at the same rate, it focuses on the part of the input that is related to the output to be predicted at that time. The structure of LSTM using the attention method is shown in Fig. 3. The encoder layer is a stack of recurrent units which accept a single element of the input sequence. Each hidden state in the encoder layer is computed as an output of the

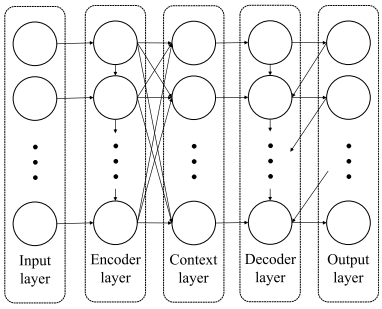

Fig. 3. Structure of LSTM using the Attention method [20].

weighted sum of the previous hidden state in the layer and the current input. The context layer, which is made of context vectors, is the output of the encoder layer and becomes the input for the decoder. It contains the information for the input sequence to allow the decoder to estimate the final output sequence. The decoder layer contains a stack of recurrent units which accept context vectors as the input sequence of the decoder.

4. Experiment

4.1. Dataset

We use TCP log data in LTE networks for two congestion control algorithms, TCP CUBIC and BBR, collected in [21]. TCP CUBIC [22] is a congestion control algorithm that simplifies window control, and while maintaining the advantages of the existing binary increase congestion control (BIC) TCP, it overcomes the disadvantages of occupying excessive bandwidth or generating overhead. BBR [23] is a congestion control algorithm by estimating the bandwidth of a network by reducing packet latency through start-up, drain, probe-bandwidth, and probe-RTT modes.

We convert TCP log data received in the LTE network into throughput trace data using RTT and congestion window (CWND) in the following equation:

$$\text{Throughput} = \text{CWND}/\text{RTT}. \tag{3}$$

By using this, we obtain the network throughput at 100 ms intervals. We divide the network throughout data into a training set and a test set with a 7:3 ratio (i.e., the training set is 70% and the test set is 30%). The training set is used for prediction model training and weight update, and the test set is used for evaluating the performance metrics.

4.2. Experiment setup

To use the prediction system presented in Section 3.2, we select LSTM as the neural network of the prediction model. The learning rate (lr) is set to 0.01. In the training process, we use the hyperbolic tangent, $tanh$ as the activation function of the LSTM model, the mean squared error (MSE) as the loss function, and ADAM [24] as the weight update algorithm. Then we apply the attention method to LSTM. The encoder and decoder layers are set as LSTM. In addition, we combine throughputs in four consecutive time slots into one set in the sliding window method (i.e., $\omega = 4$) and use it as an input of each model. Finally, we run simulation over the TCP log data to compare the proposed system with the existing methods, Seq2Seq and LSTM without the attention model.

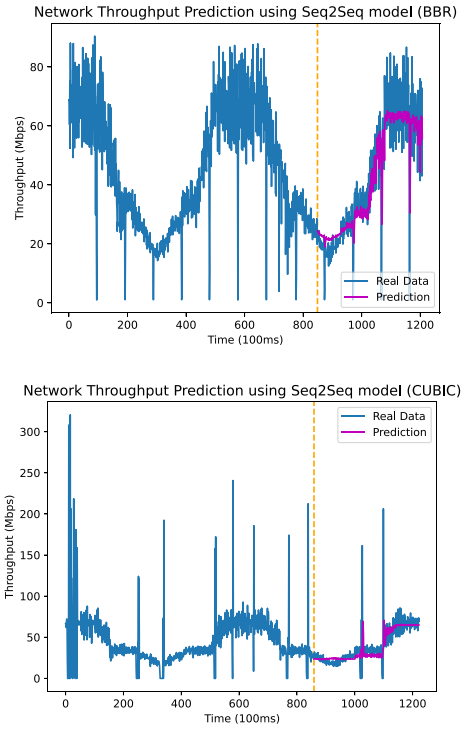

Fig. 4. Network throughput prediction using Seq2Seq.

5. Result

We use RMSE in Eq. (1) to verify the performance of the model. In this case, the closer the RMSE value is to 0, the higher the prediction accuracy, indicating good prediction performance. Also, we define normalized RMSE (nRMSE) as follows:

$$\text{nRMSE}\ (\%) = \frac{100 \times \text{RMSE}}{\max(Z) - \min(Z)}. \tag{4}$$

The throughput prediction results of each model are shown in Figs. 4, 5, and 6. In the test set of BBR traced log data, the maximum throughput is 86.64 Mbps, the minimum throughput is 1.08 Mbps, and the average throughput is 42.48 Mbps. The test set of CUBIC traced log data shows maximum throughput of 147.97 Mbps and a minimum throughput of 0 Mbps, and the average throughput is 40.4 Mbps. The RMSE and nRMSE for the prediction models are summarized in Table 3. LSTM with the attention method shows higher throughput prediction accuracy than Seq2Seq and LSTM without the attention method. The RMSE of LSTM with the attention method is reduced by 16.3% and 10.2% on average compared to Seq2Seq and LSTM without the attention method, respectively.

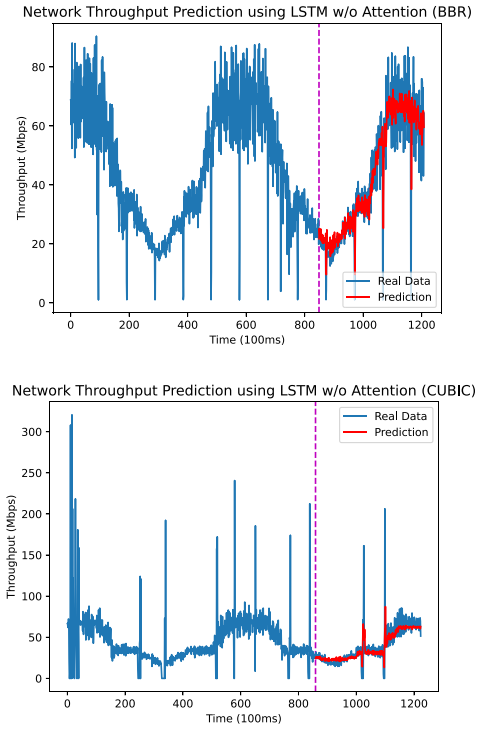

Fig. 5. Network throughput prediction using LSTM w/o Attention.

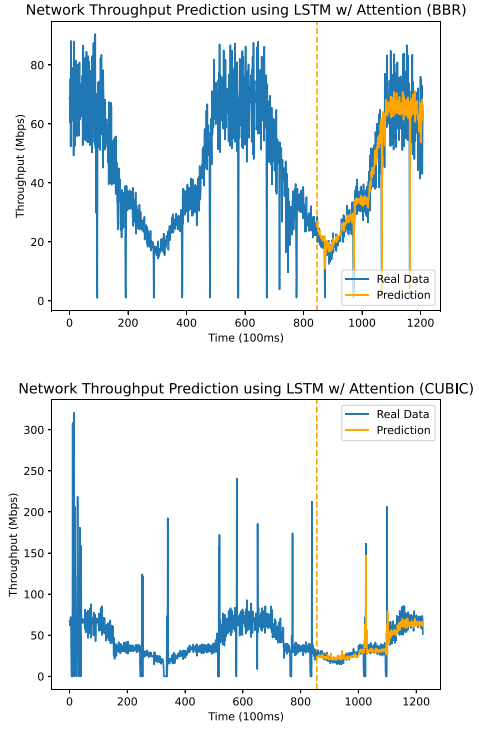

Fig. 6. Network throughput prediction using LSTM w/o Attention.

6. Conclusion

We proposed a system that predicts network throughput using a recurrent neural network, LSTM using the attention method. The proposed prediction system can be used in time-critical applications such as streaming services, to reduce the latency. We applied the sliding window method that can reduce the amount of computations in training. We then obtained prediction results on throughput data compared with the existing methods, and LSTM using the attention method showed higher prediction accuracy.

Table 3
RMSE and normalized RMSE for three learning models.

(BBR)	Seq2seq	LSTM	LSTM w/ Attention
RMSE (Mbps)	9.98	9.45	8.33
nRMSE (%)	11.18	10.59	9.33
(CUBIC)	Seq2Seq	LSTM	LSTM w/ Attention
RMSE (Mbps)	14.68	13.46	12.32
nRMSE (%)	4.59	4.2	3.85

CRediT authorship contribution statement

Hyeonjun Na: Conceptualization, Software, Writing – original draft. **Yongjoo Shin:** Investigation, Writing – original draft, Visualization. **Dongwon Lee:** Conceptualization, Resources, Writing – original draft. **Joohyun Lee:** Methodology, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by Samsung Research Funding & Incubation Center of Samsung Electronics, South Korea under Project Number SRFC-TB1803-05.

References

[1] D. Lee, J. Lee, Machine learning and deep learning for throughput prediction, in: Proc. of International Conference on Ubiquitous and Future Networks, ICUFN, 2021.
[2] T. Mangla, N. Theera-Ampornpunt, M. Ammar, E. Zegura, S. Bagchi, Video through a crystal ball: Effect of bandwidth prediction quality on adaptive streaming in mobile environments, in: Proc. of the International Workshop on Mobile Video, 2016.
[3] C.E. Andrade, S.D. Byers, V. Gopalakrishnan, E. Halepovic, M. Majmundar, D.J. Poole, L.K. Tran, C.T. Volinsky, Managing massive firmware-over-the-air updates for connected cars in cellular networks, in: Proc. of ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services, 2017.
[4] D. Raca, A.H. Zahran, C.J. Sreenan, R.K. Sinha, E. Halepovic, R. Jana, V. Gopalakrishnan, B. Bathula, M. Varvello, Empowering video players in cellular: Throughput prediction from radio network measurements, in: Proc. of ACM Multimedia Systems Conference, 2019.
[5] Q. He, C. Dovrolis, M. Ammar, On the predictability of large transfer TCP throughput, ACM SIGCOMM Comput. Commun. Rev. 35 (4) (2005) 145–156.
[6] S.P. Sotiroudis, S.K. Goudos, K. Siakavara, Deep learning for radio propagation: Using image-driven regression to estimate path loss in urban areas, ICT Express 6 (3) (2020) 160–165.
[7] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[8] D. Minovski, N. Ogren, C. Ahlund, K. Mitra, Throughput prediction using machine learning in lte and 5g networks, IEEE Trans. Mob. Comput. (2021).

[9] J. Schmid, M. Schneider, A. Höß, B. Schuller, A comparison of ai-based throughput prediction for cellular vehicle-to-server communication, in: Proc. of International Wireless Communications & Mobile Computing Conference, IWCMC, 2019.

[10] M.A. Khan, R. Hamila, N.A. Al-Emadi, S. Kiranyaz, M. Gabbouj, Real-time throughput prediction for cognitive Wi-Fi networks, Elsevier J. Netw. Comput. Appl. 150 (2020) 102499.

[11] H. Drucker, C.J. Burges, L. Kaufman, A. Smola, V. Vapnik, et al., Support vector regression machines, Adv. Neural Inf. Process. Syst. 9 (1997) 155–161.

[12] B. Wei, W. Kawakami, K. Kanai, J. Katto, A history-based TCP throughput prediction incorporating communication quality features by support vector regression for mobile network, in: Proc. of IEEE International Symposium on Multimedia, ISM, 2017.

[13] D. Raca, A.H. Zahran, C.J. Sreenan, R.K. Sinha, E. Halepovic, R. Jana, V. Gopalakrishnan, On leveraging machine and deep learning for throughput prediction in cellular networks: Design, performance, and challenges, IEEE Commun. Mag. 58 (3) (2020) 11–17.

[14] J. Schmid, M. Schneider, A. HöB, B. Schuller, A deep learning approach for location independent throughput prediction, in: Proc. of IEEE International Conference on Connected Vehicles and Expo, ICCVE, 2019.

[15] A. Narayanan, E. Ramadan, R. Mehta, X. Hu, Q. Liu, R.A. Fezeu, U.K. Dayalan, S. Verma, P. Ji, T. Li, et al., Lumos5g: Mapping and predicting commercial mmwave 5g throughput, in: Proc. of ACM Internet Measurement Conference, IMC, 2020.

[16] S.M. Kasongo, Y. Sun, A deep long short-term memory based classifier for wireless intrusion detection system, ICT Express 6 (2) (2020) 98–103.

[17] L. Salmela, N. Tsipinakis, A. Foi, C. Billet, J.M. Dudley, G. Genty, Predicting ultrafast nonlinear dynamics in fibre optics with a recurrent neural network, Nat. Mach. Intell. 3 (4) (2021) 344–354.

[18] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[19] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.

[20] A.M. Rush, S. Chopra, J. Weston, A neural attention model for abstractive sentence summarization, 2015, arXiv preprint arXiv:1509.00685.

[21] S. Park, J. Lee, J. Kim, J. Lee, S. Ha, K. Lee, ExLL: An extremely low-latency congestion control for mobile cellular networks, in: Proc. of ACM Conference on Emerging Networking EXperiments and Technologies, CoNEXT, 2018.

[22] S. Ha, I. Rhee, L. Xu, CUBIC: a new TCP-friendly high-speed TCP variant, Oper. Syst. Rev. 42 (5) (2008) 64–74.

[23] N. Cardwell, Y. Cheng, C.S. Gunn, S.H. Yeganeh, V. Jacobson, BBR: Congestion-based congestion control: Measuring bottleneck bandwidth and round-trip propagation time, ACM Queue 14 (5) (2016) 20–53.

[24] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.