Combining Long-Short Term Memory and Reinforcement Learning for Improved Autonomous Network Operation

Fatemehsadat Tabatabaeimehr, Sima Barzegar, Marc Ruiz, and Luis Velasco*

Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
e-mail: luis.velasco@upc.edu

Abstract: A combined LSTM and RL approach is proposed for dynamic connection capacity allocation. The LSTM predictor anticipates periodical long-term sharp traffic changes and extends short-term RL knowledge. Numerical results show remarkable performance.

1. Introduction

The traffic generated by many 5G services is complex and hard to model due to the presence of multiple periodicities ranging from few hours to several days [1]. Although 24h is still the dominant periodicity for most services, other periodicities with shorter or larger period can introduce *sharp* traffic changes that significantly distort the typical daily profile. This fact makes impractical applying traditional predictive approaches in many multilayer optical network automation scenarios, e.g., dynamically allocating capacity to a traffic flow according to traffic prediction [2].

Aiming at modelling complex time series, such as network traffic, Long-Short Term Memory (LSTM) artificial neural networks were proposed in [3] due to its ability to learn data with long-term sequential dependencies and indefinite duration. Together with LSTMs, specific loss functions have been recently proposed to minimize prediction error in the presence of sharp changes [4]. Nevertheless, although the overall accuracy of LSTMs can be higher than that of alternative traffic prediction methods, they still incur in occasional errors that can reduce robustness for autonomous network operation. Particularly, traffic under-prediction can lead to connection capacity under-provisioning, which translates into traffic loss.

In contrast, Reinforcement Learning (RL) has proved to autonomously adapt connection capacity to complex traffic evolution, by penalizing those actions that cause traffic loss while reducing capacity overprovisioning (see, e.g., [5], [6]). However, capacity management based on low-complex RL does not perform that well in the event of sharp traffic changes, which might require incrementing their complexity to deal with both short- and long-term dependencies. Note that complex RL algorithms present some drawbacks, such as unstable and long-time learning process.

In this work, we propose combining a LSTM-based traffic prediction model and a low-complex RL algorithm for dynamic connection capacity allocation, aiming at outperforming both methods when used separately. In particular, we consider that traffic flows show some periodical components that produce: *i*) a traffic evolution with smooth changes that can be easily learnt by a RL algorithm; and *ii*) some periodical long-term sharp traffic changes, which are detected and quantified by the LSTM-based predictor. The latter feeds the RL algorithm to extend its knowledge without adding extra complexity.

2. Traffic prediction and RL for autonomous operation

In this section, we present our proposal for autonomous capacity management of connections transporting packet traffic flows. Without loss of generality, we consider a connection to be either a customer connection transporting a few Gb/s traffic flow or a virtual link supporting flow aggregation with traffic of hundreds Gb/s. The objective is to allocate, at every time t, the minimum capacity to the connection to support the flow for the next period, while meeting the intended performance, e.g., avoiding loss due to capacity under-provisioning.

Fig. 1 illustrates the three considered approaches for implementing autonomous capacity management: *i*) *threshold-based* (Fig. 1a); *ii*) a *low-complex RL* algorithm (Fig. 1b); and *iii*) a *LSTM-based* prediction (Fig. 1c). Fig. 1d sketches the expected evolution of the allocated capacity (solid-colored lines) for a traffic flow (dotted line) that experiences a sharp traffic increase. Under the threshold-based approach, future capacity is reactively adjusted according to the current traffic. A threshold value is statically defined, which must be set low enough to guarantee no loss during sharp traffic changes, which leads to high overprovisioning. In contrast, a low-complex RL-based algorithm allows a finer capacity allocation by learning fast the right margin of capacity that needs to be allocated to guarantee no loss. Here, the environment computes the state and reward according to short-term monitoring data and feeds an agent that does both model learning and decision making by taking the most proper action (increase or decrease) capacity according to the current state. However, long-term periodical sharp traffic changes will be poorly learnt with a short-term perspective and some losses can be expected. Finally, LSTM can be used to predict long-term periodic traffic. Notwithstanding the good overall accuracy, LSTM underestimates traffic at the beginning of the peak, which can lead to loss, and largely overestimates traffic during and after the sharp change, which produces capacity overprovisioning.

The research leading to these results has received funding from the Spanish MINECO TWINS project (TEC2017-90097-R) and by the ICREA institution.

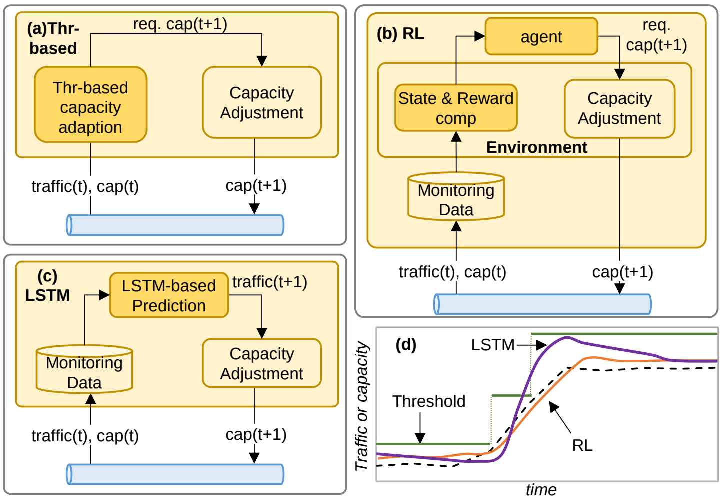

Fig. 1. (a) Thr-based, (b) RL, (c) LSTM, and (d) expected capacity

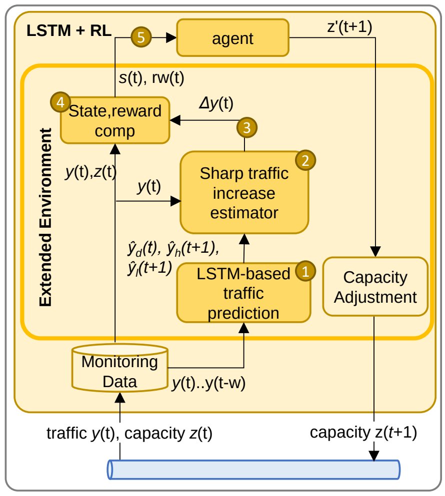

Fig. 2. Cooperative LSTM + RL approach

In view of the above, we propose combining LSTM-based traffic prediction model and low complex RL-based capacity allocation; the approach (LSTM+RL) is illustrated in Fig. 2. This approach includes an *extended environment* that integrates LSTM-based traffic prediction to generate inputs that add knowledge to be considered in the state definition and reward computation without increasing RL model complexity. In this approach, LSTM-based model training is tuned to produce unbiased (larger than expected) traffic estimation by means of a dedicated *loss function*. Once trained, the LSTM-based model is used to generate several predictions (labeled 1 in Fig. 2) that, jointly with the current traffic, are then analyzed by a module that estimates whether it is likely that the traffic will sharply increase (2). If so, the expected traffic increase is computed by combining different estimations and sent for state and reward computation (3). Such expected extra traffic is combined with the monitored traffic and processed as actual current measurements (4); this directly affects the inputs that the agent needs to learn and take actions (5). The formal details of the extended environment blocks are presented in the next section.

3. Extended RL Environment with LSTM-based traffic prediction

Table 1 summarizes the used notation. We consider that a monitoring data set is available for every connection, which consists of samples <$y(t)$, $z(t)$>, with the traffic and capacity at time t, respectively. For the sake of simplicity, let us consider that a LSTM model is represented by eq. (1); it receives as input the last w traffic monitoring samples and produces traffic predictions $\hat{y}$ with different upper confidence intervals to be used at time t. It is of paramount importance, as mentioned above, training the model to benefit over-prediction instead of typical centered (unbiased) prediction. To this aim, we consider a *truncated* mean square error (trMSE) component in the loss function (see eq. (2)) that returns the MSE if and only if the relative error between real and prediction is above a given β value. To avoid extremely large and meaningless over-prediction, the proposed loss function weights trMSE and MSE with parameter α, as defined in eq. (3). Once trained, LSTM model f is used to generate three predictions based on different confidence intervals at every time t: *i*) $\hat{y}_d(t)$ used for sharp increase detection, *ii*) $\hat{y}_h(t+1)$ for traffic prediction only during sharp increase, and *iii*) $\hat{y}_l(t+1)$ for traffic prediction when no sharp increase has been detected. The sharp traffic estimation module takes the set of predictions, as well as current and last monitored traffic, and it estimates the expected traffic increase $\Delta y(t)$ that will be considered for state and reward computation, following eq. (4). Note that the traffic prediction is based on different traffic estimations depending on whether a sharp increase has been detected or not.

Finally, the Q-learning algorithm presented in [5] has been extended in two ways: *i*) the state s is computed not only based on the current monitored traffic $y(t)$, but also on the expected increase $\Delta y(t)$, as defined in eq. (5), where n^s represents the number of discrete states; and *ii*) the reward function component related with capacity overprovisioning (r) has been modified to avoid penalizing when $\Delta y(t)$ is different than 0; a large capacity

Table 1. Notation

$y(t)$	Actual traffic at time t.
$z(t)$	Allocated connection capacity at time t.
$\hat{y}$	Traffic prediction
f	LSTM model
α, β	Loss function weight and minimum relative error
$\hat{y}_d(t)$	Traffic prediction for sharp increase detection
$\hat{y}_h(t)$	Traffic prediction during sharp increase
$\hat{y}_l(t)$	Traffic prediction during no sharp increase
$\Delta y(t)$	Expected traffic increase
r, s	RL reward and state

$$\{\hat{y}(t+i), i = 1..m\} \sim f(\{y(t-j), j = 0..w\}) \qquad (1)$$

$$trMSE(t;\beta) = \begin{cases} \left(y(t) - \hat{y}(t)\right)^2, & (y(t) - \hat{y}(t))/y(t) \geq \beta \\ 0, & otherwise \end{cases} \qquad (2)$$

$$loss(t;\alpha,\beta) = \alpha \cdot trMSE(t;\beta) + (1-\alpha) \cdot MSE(t) \qquad (3)$$

$$\Delta y(t) = \begin{cases} \hat{y}_h(t+1) - y(t), & y(t) > \hat{y}_d(t) \\ \hat{y}_l(t+1) - y(t), & otherwise \end{cases} \qquad (4)$$

$$s(t) = \min\left(\lceil n^s \cdot (y(t) + \Delta y(t))/z(t) \rceil\right) \qquad (5)$$

$$r(t) = [z(t) - y(t) > \Delta y(t)] \qquad (6)$$

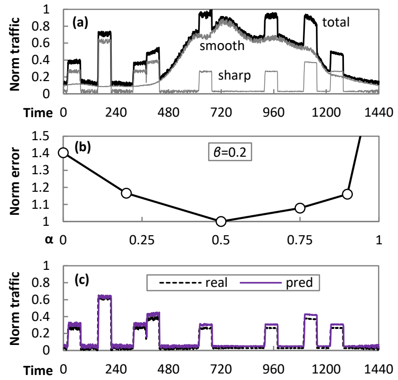

Fig. 3. Traffic and LSTM performance

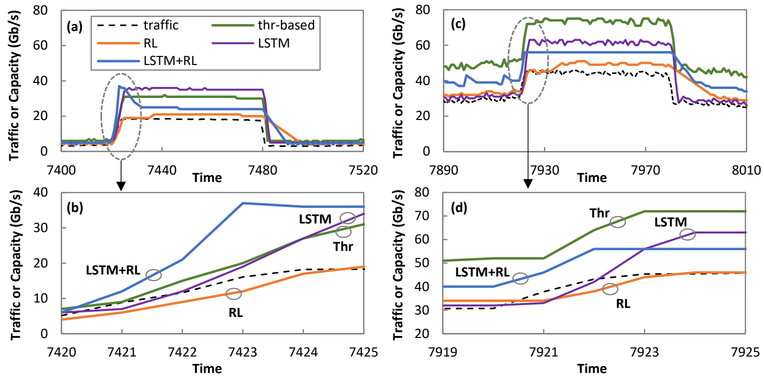

Fig. 4. Approaches comparison for low (a, b) and high (c, d) traffic

provisioning caused by an extra traffic added is not penalized, provided that such overprovisioning is within the magnitude of the extra traffic (see eq. (6)). In this way, the effect of the sharp change is not considered for learning background smooth traffic, which is the main target of the RL algorithm.

4. Numerical Results

For evaluation purposes, we have generated 6 months of synthetic traffic data for one traffic flow, which includes: *i*) one main traffic component with daily variation. For the sake of comparability, we have used the same traffic as in [5], and *ii*) an on-off traffic profile with several periodicities ranging from 6h to 5 days. One day of traffic with the defined components is presented in Fig. 3a, where traffic is normalized to the maximum value. The first half of the traffic (3 months) has been used for training and validating the proposed LSTM-based traffic prediction. To this aim, we have focused only on the sharp traffic component and evaluated several configurations of the LSTM model to tune loss function parameters. Fig. 3b shows the prediction error (normalized to the minimum value) as a function of α, for β=0.2. As it can be observed, balancing trMSE and normal MSE, i.e., α in (0,1), provides the best performance, which validates the usefulness of the proposed loss function. Specifically, α=0.5 was selected to train and test the final LSTM model, which consists of 3 layers and 3 neurons per layer (each with a tanh() activation function) and a final dense layer to generate the final output. With that model, we evaluated the formulation in Section 3; the results are shown in Fig. 3c. For the sake of clarity, the prediction curve is the sum of the predicted $\Delta y(t)$ and previous traffic $y(t\text{-}1)$. We observe a tight correlation of real and predicted traffic that supports concluding that the proposed sharp increase estimator clearly captures the evolution of sharp traffic, which is key for the success of the proposed combined approach.

The performance of the LSTM+RL approach has been evaluated using the last 3 months of traffic data, which has been scaled to emulate a flow having daily variations ranging between 3 and 45 Gb/s. The extended environment was implemented in the simulator presented in [5] and used to evaluate the proposed approach, as well as the other considered methods for comparison purposes. Fig. 4 highlights the behavior of all the approaches at some intervals of the simulation, where different traffic load is observed. For low load (Fig. 4a-b), LSTM and RL approaches are not able to anticipate enough the sharp traffic increase, which leads to capacity under-provisioning and traffic loss, whereas the threshold-based approach fits capacity remarkably close to actual traffic, resulting into a very good performance in terms of capacity over-provisioning. For high loads (Fig. 4c-d), the bad performance of LSTM and RL is confirmed, while threshold-based allocates more capacity than our proposed combined solution.

Table 2 summarizes the simulation results, where the combined LSTM and RL approach provides no losses. In contrast, the LSTM and RL approaches, when working separately, are unable to anticipate the sharp traffic changes and thus, they produce significant traffic loss. Note also that those approaches require slightly lower over-provisioning than the combined one, and thus the overprovisioning of the latter seems to be close to optimal.

Table 2. Summary

Approach	max loss (Gb/s)	max cap (Gb/s)	Avg over-prov (%)
Threshold	0	75	73%
RL	10	51	20%
LSTM	5	63	36%
LSTM+RL	0	67	49%

5. Conclusions

Autonomous connection capacity management has been proposed based on extending short-term RL knowledge with LSTM for long-term traffic prediction. The approach exhibited noticeable performance under daily traffic profiles with on-off sharp changes, which resulted in no losses and moderate capacity overprovision.

References

[1] C. Marquez *et al.*, “Identifying Common Periodicities in Mobile Service Demands with Spectral Analysis,” IEEE MedComNet 2020.
[2] F. Morales *et al.*, “Dynamic Core VNT Adaptability based on Predictive Metro-Flow Traffic Models,” IEEE/OSA JOCN 2017.
[3] T. Guo *et al.*, “Robust Online Time Series Prediction with Recurrent Neural Networks,” IEEE DSAA, 2016.
[4] V. Le Guen *et al.*, “Shape and Time Distortion Loss for Training Deep Time Series Forecasting Models,” NeurIPS, 2019.
[5] S. Barzegar *et al.*, “Reinforcement Learning -based Autonomous Multilayer Network,” ECOC 2020.
[6] F. Tabatabaeimehr *et al.*, “Supporting Beyond 5G Applications by Coordinating AI-based Intent Operation. An Example for Multilayer Metro Networks,” ECOC, 2020.