IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 18, 2019, accepted July 10, 2019, date of publication July 17, 2019, date of current version August 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2929480

Deep Learning-Based Dynamic Bandwidth Allocation for Future Optical Access Networks

JOHN ABIED HATEM, AHMAD R. DHAINI, (Senior Member, IEEE), AND SHADY ELBASSUONI

American University of Beirut, Beirut 1107 2020, Lebanon

Corresponding author: John Abied Hatem (jmh28@aub.edu.lb)

This work was supported by the University Research Board (URB), American University of Beirut, under Grant 103367.

ABSTRACT Over the last decade, Passive Optical Networks (PONs) have emerged as an ideal candidate for next-generation broadband access networks. Meanwhile, machine learning and more specifically deep learning has been regarded as a star technology for solving complex classification and prediction problems. Recent advances in hardware and cloud technologies offer all the necessary capabilities for employing deep learning to enhance Next-Generation Ethernet PON's (NG-EPON) performance. In NG-EPON systems, control messages are exchanged in every cycle between the optical line terminal and optical network units to enable dynamic bandwidth allocation (DBA) in the upstream direction. In this paper, we propose a novel DBA approach that employs deep learning to predict the bandwidth demand of end-users so that the control overhead due to the request-grant mechanism in NG-EPON is reduced, thereby increasing the bandwidth utilization. The extensive simulations highlight the merits of the new DBA approach and offer insights for this new line of research.

INDEX TERMS Deep learning, dynamic bandwidth allocation, machine learning, NG-EPON, optimization, PON, simulations.

I. INTRODUCTION

According to Cisco's Visual Networking Index forecast, the global Internet traffic, which amounted to approximately 27 Tbps in year 2016, will reach a whopping 106 Tbps by year 2021 [1]. This unprecedented growth in Internet traffic combined with the advancement in the backbone network, have accentuated the bottleneck in the first/last mile. Over the last decade, Passive Optical Network (PON) has been viewed as the most promising solution to the access bottleneck problem. More notably, the Ethernet PON family (i.e., 1G-EPON, 10G-EPON, and 100G-EPON) has been considered the most promising PON variant, due to its cost effectiveness, high bandwidth capacity, and ability to efficiently support quality-of-service (QoS) [2].

Next-Generation EPONs (NG-EPON) are expected to support an increasing number of users, provide much higher data rates, and support stringent QoS requirements for new and diverse range of applications (e.g., Tactile Internet). Accommodating all these requirements would increase the complexity of NG-EPON systems; thus, ''efficiently customized'' NG-EPONs can become the new trend. That is, traditional ''one-size-fits-all'' approaches will not work, and optimizing the different aspects of data transmission in NG-EPONs to provide better services to larger number of users while being energy and bandwidth-efficient is not an easy task. One of the most powerful tools that could help address the foregoing challenges and which has recently begun to be adopted in optical networks in general [3]–[7] and PONs in specific is machine learning [8]–[15]. In particular, deep learning has been gaining popularity as a stellar approach for solving complex classification and prediction problems. Given sufficiently-large training data, deep neural networks can be trained to optimize some objective function by exploiting hidden or implicit patterns in the training data. The huge amount of data that can be collected by monitoring different parameters in NG-EPON make machine learning a great candidate for network performance optimization. Moreover, recent advances in GPU technology and Cloud Computing provide the processing and storing capabilities needed for training computationally-expensive machine learning models such as the deep learning ones [16].

The associate editor coordinating the review of this manuscript and approving it for publication was Zhaoxiang Zhang.

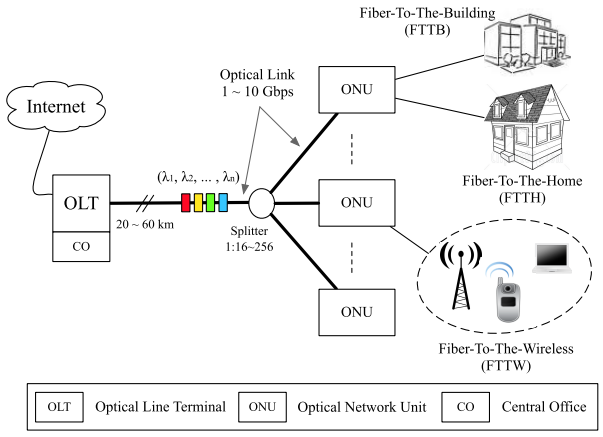

FIGURE 1. Typical NG-EPON architecture [17].

As illustrated in Fig. 1, a typical NG-EPON has a tree topology that comprises an Optical Line Terminal (OLT) residing at the Internet Service Provider (ISP)'s central office (CO), and connecting via a passive optical splitter a set of Optical Network Units (ONUs) located at the end-users' premises. The transmission in the downstream direction is typically performed by the OLT via broadcasting data to the ONUs using time division multiplexing (TDM); whereas in the upstream direction, TDM and/or wavelength division multiplexing (WDM) can be employed, coupled with a dynamic bandwidth allocation (DBA) scheme. The communication between the OLT and the ONUs is conducted via the Multi-Point Control Protocol (MPCP), which enables the ONU to send a REPORT message to the OLT, informing it about its upstream bandwidth demands. In return, the OLT would respond with a GATE message granting the ONU a time slot. This exchange happens in a cyclic manner, which causes a control bandwidth overhead that may be large when the network is provisioned with a large number of ONUs and long EPON distances [18]. The overhead may vary depending on the employed DBA scheme. For instance, with *online* schedulers (i.e., where the OLT schedules grants "on-the-fly" without waiting until all the REPORT messages are received; e.g., IPACT [2]), the idle time and subsequently the packet delay are reduced. However, ensuring fairness among ONUs and supporting QoS is not easily attainable as the OLT would lack a holistic view of all the ONU demands. This problem is resolved using *offline* schedulers, where the OLT waits until all the REPORT messages are received, and then preforms DBA and schedules grants accordingly. This enables the OLT to support QoS and enables fairness among ONUs, at the expense of decreased channel utilization due to the control overhead between transmission cycles [18], [19].

In this paper, we propose a novel DBA scheme that employs deep learning (thus-called *Deep-DBA*) with the *offline* scheduler to improve the network bandwidth utilization so as to provision more users and/or push more services into the network. More specifically, we build a Long-Short Term Memory (LSTM) Recurrent Neural Network (RNN) [20] and train it to predict the bandwidth requests of ONUs for a certain number of future cycles based on past cycles. This gives the OLT a holistic view of the requests of all the ONUs for the predicted future cycles, and allows it to schedule future time slots without requiring the ONUs and the OLT to exchange REPORT/GATE messages in every cycle. Extensive simulation results highlight the merits of the new approach and show how Deep-DBA is able to preserve the advantages of *offline* DBAs, while achieving higher bandwidth utilization compared to *online* schemes. More specifically, the main contribution of Deep-DBA is reducing the bandwidth overhead. This bandwidth gain can be used to provision more users and/or services in the network. The other contribution is that Deep-DBA is an offline scheme but at the same time exhibits a similar performance to online schemes in terms of throughput and delay.

The rest of the paper is organized as follows. In Section II, we present a survey of the related state-of-the-art works. Section III presents the proposed Deep-DBA scheme. Extensive simulation and experimental results are presented in Section IV. Finally, we conclude the paper and discuss future extensions in Section V.

II. RELATED WORK

In this section, we review the state-of-the-art works that employ machine learning to enhance the operation and performance of bandwidth allocation in PONs.

To offer better QoS support to PON subscribers, the authors of [8] presented a service level agreement (SLA)-based proportional-integral-derivative (PID) controller. The PID is enhanced with an online neural network to tune the parameters of the PID controller. The input layer of the machine learning model has three neurons representing the past three error-readings of the PID. Similarly, the output layer has three neurons representing the three parameters used to tune the PID controller.

To optimize the upstream bandwidth allocation in PONs, the authors of [9] proposed to dynamically (re)allocate the SLA parameters, which are represented by the Committed Information Rate (CIR, which is the guaranteed bandwidth provided to the user), the Excess Information Rate (EIR, which is some additional bandwidth that may be provided to the user), and the Peak Information Rate (PIR, which is the maximum bandwidth that can be assigned to a user), based on the user profile. Namely, using K-means clustering, users are classified into three different groups: *heavy*, *light*, and *flexible* for specific periods of the day. Subsequently, excess bandwidth is allocated to the EIR of heavy users to improve their QoS. The limitation of this scheme is that the majority of users were classified as *flexible*. This work is then extended in [10], such that user groups are further classified based on the bandwidth usage during weekdays and weekends. Furthermore, a "Grey Forecasting Model" is employed to predict the future bandwidth demand trend of users in the flexible group, that is, whether they will shift or not to the

heavy or light groups to have a more balanced distribution of the excess bandwidth.

To predict the additional packets that may arrive during the polling period, the authors of [11] proposed a data mining forecasting DBA, so-called DAMA, which employs an enhanced k-nearest neighbor (k-NN) algorithm. Results show that predicting the additional bandwidth improves the network performance in terms of latency and jitter.

In [12], the authors proposed an artificial neural network (ANN) decision-making model to predict the bandwidth demand of an ONU. The ANN model is trained to predict uplink latency under different network scenarios so as to dynamically allocate bandwidth to meet low latency requirements.

To support Tactile services, the authors of [13] employed a Bayesian estimation to approximate the packet inter-arrival time for Poisson-distributed Tactile traffic in a WDM-PON. For Pareto-distributed traffic, the authors used a maximum-likelihood sequence estimation to approximate the *On* and *Off* durations. The estimations are performed at the ONU, and are then sent to the OLT using a REPORT message. Consequently, the OLT evaluates the average bandwidth demand of each ONU and maintains the low latency constraint by dynamically varying the number of active wavelength channels.

Finally, the authors of [14] proposed a machine learning based DBA (MLP-DBA), where an ANN model is deployed at the OLT to identify the *On* and *Off* periods of bursty Internet traffic for the next polling cycle of every ONU. Based on this prediction, the bandwidth demand during the waiting time is evaluated. Consequently, if the sum of the requested bandwidth plus the predicted bandwidth is greater than the maximum bandwidth allowed for each ONU, an extra cycle is introduced by generating additional GATE messages for these ONUs at the beginning of the next polling cycle. This would offer lower latency and enable the support of Tactile services.

As can be seen, none of the above reviewed works addressed the challenge of improving the bandwidth utilization in PONs. In our work, we tackle this problem by making use of deep learning (more specifically using a Long-Short Term Memory Recurrent Neural Network) with the *offline* scheduler to predict the user demands (i.e., the bandwidth requested using REPORT messages) so as to reduce the amount of exchanged control messages between the OLT and ONUs, and eliminate the idle time of the *offline* scheduler between predicted cycles. This would in turn reduce the network latency and allow for provisioning more services and/or users in the network.

III. DEEP LEARNING-BASED DBA (DEEP-DBA)

A. SYSTEM MODEL

In NG-EPON systems, in every polling cycle, each ONU sends to the OLT a REPORT message depicting its buffering queues' occupancies, which reflects the end-users' bandwidth demands. Consequently, the ONUs are granted time slots by the OLT in the next cycle using GATE messages; these time slots are sized depending on the DBA discipline.

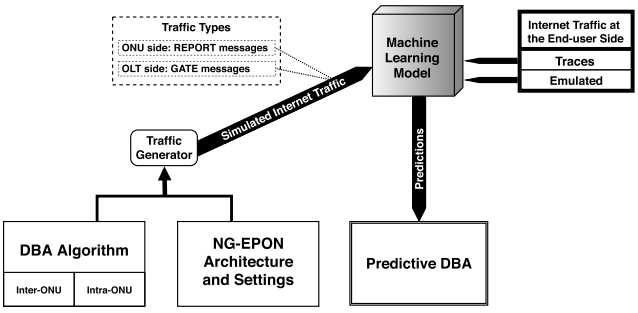

FIGURE 2. Proposed machine-learning based system model.

As illustrated in Fig. 2, to reduce the control messaging overhead (which can be significant depending on the number of connected ONUs, the distance between the OLT and ONUs, and the channel speed), we propose to employ a machine learning model to predict the bandwidth demand of an ONU for the next Q cycles based on its demands in the past P cycles. Here, bandwidth demands can be collected in three forms: 1) incoming traffic flows/streams at the end-user side; 2) REPORT messages in every polling cycle, which depict the bandwidth demands; and 3) GATE messages issued by the OLT in every polling cycle, which indirectly reflect the bandwidth demands of the ONUs. The latter two depend on the employed DBA algorithm and the network architecture and settings, as these affect the behavior of the network and thus the bandwidth included in the REPORT and GATE messages. Consequently, the machine learning model is trained on the collected data, and the obtained model is saved and embedded as a module in the DBA so as to perform predictive bandwidth allocation.

B. BANDWIDTH DEMAND PREDICTION USING DEEP LEARNING

Internet traffic in NG-EPON can be seen as time series, which corresponds to the bandwidth demand in every cycle. To predict Internet traffic, several linear prediction techniques based on statistical learning have been proposed [21]–[23]. Although these methods can learn the linear correlation structure of the time series, they fail to learn nonlinear patterns [24]. Recently, non-linear prediction has been performed using neural networks, which have been widely used due to their ability to approximate any linear or non-linear patterns in an accurate manner even when the underlying data relationships are unknown. Results in [25]–[27] show that by using neural networks, better prediction outcome in terms of accuracy can be obtained compared to previous methods such as AutoRegressive Moving Average (ARMA), AutoRegressive Integrated Moving Average (ARIMA), AutoRegressive AutoRegressive (ARAR) and HoltWinter algorithm.

While Feed-Forward Neural Networks are able to provide accurate prediction and fast response time, they fail to handle sequence data, and they are only limited to

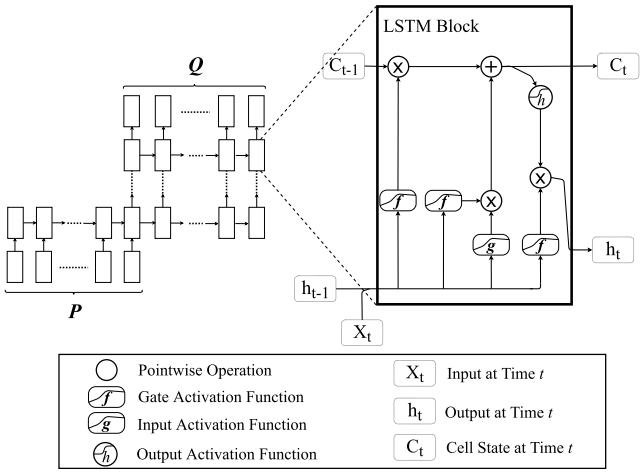

FIGURE 3. Employed LSTM RNN model.

data within a fixed-size window. In contrast, RNNs take into consideration past-seen input to make predictions in the current time step making it a good candidate for *sequence-to-sequence* predictions. However, conventional RNN models suffer from vanishing and/or exploding gradient problems, which limits the RNN's capability to model long-term dependencies [28]. Consequently, LSTM networks have been designed to address these foregoing issues [20], and they were demonstrated to outperform neural networks and traditional RNNs for many applications [27]–[30]. Moreover, recent studies suggest that Convolutional Neural Networks can be used for time series forecasting as shown in [31], [32].

We note that any of the aforementioned models that can handle *sequence-to-sequence* time series predictions can be used to perform predictive DBA. In this work, based on similar findings in related problems [28]–[30] and extensive experimental results on our dataset, we choose to employ an RNN LSTM model. Note that only one model is needed by the proposed scheme and it is employed at the OLT. Fig. 3 depicts the general LSTM architecture that we employ to predict future sequences from previous ones. As illustrated, the LSTM network is fed a sequence of P cycles as input, such that each cycle $p \in P$ includes the ONU's reported queue size. The output of the model would be the next sequence of Q cycles, where each cycle $q \in Q$ includes the predicted total queue size for this ONU in this cycle. This is an improvement over previous works [30], [33] since not only we are predicting the traffic in the next time step $t+1$ (next cycle), but we are also predicting the bandwidth demands for the upcoming $t+Q$ time steps.

C. OPERATION OF DEEP-DBA

The key principle behind Deep-DBA is to make use of the predictions made by a machine learning model using the past P REPORTs, so as to allocate bandwidth for the next Q cycles without requiring any further REPORT messages within those cycles. Thus, a Deep-DBA cycle would typically comprise two sets of cycles; namely the *reporting* cycles $\{p_1, p_2, \cdots, p_P\}$ and the *prediction* cycles $\{q_1, q_2, \cdots, q_Q\}$.

For simplicity and without loss of generality, we illustrate in Fig. 4 the operation of the proposed Deep-DBA for $P = 2$, and $Q = 8$. As observed, during the reporting cycles, every ONU sends a REPORT message requesting bandwidth based on its queue size (just like with regular DBA approaches). Consequently, the OLT runs the DBA algorithm (i.e., T_{DBA}) and responds with a GATE message that includes a grant for the next cycle **only**, based on the employed scheduling discipline (i.e., Limited, Gated, etc.). However, the OLT keeps record of the foregoing request to be used for prediction. As such, during the last reporting cycle p_P, as soon as the OLT receives the P^{th} REPORT message from an ONU, it uses the P saved requests of this ONU as input to the deep learning model, so as to predict its request sizes for the next Q cycles; thereby marking the start of DBA prediction time $T_{Deep-DBA}$.

When predictions are obtained by the deep learning model and the OLT has the predicted request sizes for all ONUs, these predicted request sizes are considered as if they are REPORT messages received from the ONUs for cycles $\{q_1, q_2, \cdots, q_Q\}$. After the prediction, the OLT will apply the same DBA scheme used to grant transmission windows for each ONU for cycles $\{q_2, \cdots, q_Q, p_1\}$ without requiring any further REPORT messages. This reduces the effective cycle time, and increases the network utilization. Subsequently, the OLT informs the ONUs of their transmission windows for cycles $\{q_2, \cdots, q_Q, p_1\}$ using GATE messages sent during the first prediction cycle q_1. This can be accomplished in three different ways:

1) The OLT sends $Q \times N$ GATE messages in a contiguous manner, where N is the number of ONUs.
2) The OLT incorporates 4 grants in one GATE message (which adheres to the default GATE message structure), so that $\frac{Q \times N}{4}$ GATEs are sent in a contiguous manner.
3) The format of the GATE message is modified so that it can include Q grants, which enables the OLT to send N GATEs in a contiguous manner.

As observed in Fig. 4, the GATE message based on the predicted REPORT for the first ONU to start upstream transmission should arrive before the start of the second prediction cycle q_2. Hence, there is sufficient time to do the machine learning prediction which starts in the beginning of the last reporting cycle p_P, when the REPORT of the 1^{st} ONU arrives, and continues through the first prediction cycle q_1. Therefore, the time from the start of $T_{Deep-DBA}$ until the latest moment for the first GATE to reach its corresponding ONU is almost equal to the duration of 2 cycles, which is more than sufficient given the instantaneous output that is normally produced by a trained deep learning model [16].

In the next Deep-DBA cycle, during the first reporting cycle p_1, the ONUs start data transmission immediately; however, they also send REPORT messages at the end of their transmission window marking the start of the reporting

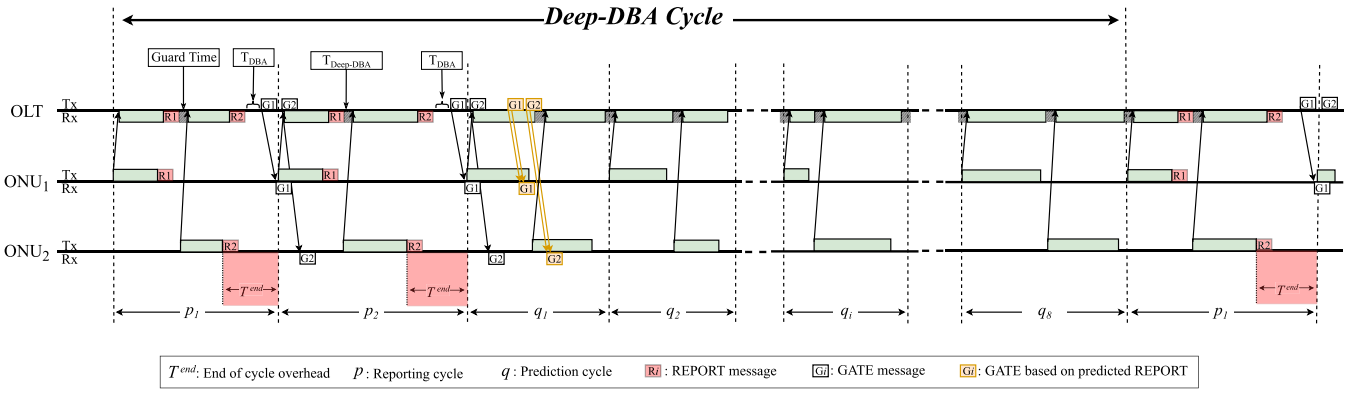

FIGURE 4. Operation of the proposed deep-DBA scheme.

TABLE 1. Summary of notations.

Notation	Description
N	The number of ONUs
$T_{G_i^j}$ / $T_{R_i^j}$	Total delay for GATE / REPORT of ONU j in cycle i
T_G^{proc} / T_R^{proc}	GATE / REPORT processing time
T_G^{trans} / T_R^{trans}	GATE / REPORT transmission time
T_j^{prop}	Propagation time of ONU j's packet
T_{DBA} / $T_{Deep-DBA}$	DBA computation/prediction time
T_i^{end}	End-of-cycle overhead in cycle i
R_i	REPORTs overhead in cycle i
$O_{Deep-DBA}$	Total overhead using Deep-DBA
O_{REG}	Total overhead using a regular NG-EPON

cycles. The total number of cycles in one Deep-DBA cycle K, can be obtained as:

$$K = P + Q. \tag{1}$$

To highlight the merits of the offline Deep-DBA, we calculate its gain compared to regular offline and online DBA schemes. For the reader's convenience, we summarize the notations used in Table 1.

The total delay caused by a GATE message destined for ONU j in cycle i is computed as follows:

$$T_{G_i^j} = (2 \times T_G^{proc}) + T_G^{trans} + T_j^{prop}. \tag{2}$$

Similarly, the total delay caused by a REPORT message from ONU j in cycle i is computed as follows:

$$T_{R_i^j} = (2 \times T_R^{proc}) + T_R^{trans} + T_j^{prop}. \tag{3}$$

As illustrated in Fig. 4, the reporting cycles with Deep-DBA are similar to the offline cycles of a legacy DBA (that is, an ONU sends a request message in cycle $i-1$, and receives a grant from the OLT for cycle i). However with Deep-DBA, two salient overhead periods that occur in these cycles are eliminated in the prediction cycles, namely the end of the cycle idle time, T_i^{end}, and the REPORT messages transmission time. The overhead T_i^{end} comprises the time taken by the N^{th} REPORT message to be transmitted and processed, the time taken to compute the DBA, T_{DBA}, and the time taken by the first GATE message to be transmitted and processed. Thus, T_i^{end} can be computed as follows:

$$T_i^{end} = \begin{cases} T_{R_i^N} + T_{DBA} + T_{G_i^1} & i \in \{p_1, p_2, \ldots, p_P\} \\ 0 & i \in \{q_1, q_2, \ldots, q_Q\} \end{cases} \tag{4}$$

As such, the total overhead caused by REPORT messages during a Deep-DBA cycle i, R_i, would be obtained by:

$$R_i = \begin{cases} (N-1) \times T_R^{trans} & i \in \{p_1, p_2, \ldots, p_P\} \\ 0 & i \in \{q_1, q_2, \ldots, q_Q\} \end{cases} \tag{5}$$

Here, the transmission delay of the N^{th} ONU is accounted for in (4). Thus, the control overhead would only be incurred in the reporting cycles. Therefore, the total overhead using Deep-DBA can be computed as follows:

$$O_{Deep-DBA} = \sum_{i=1}^{P} (T_i^{end} + R_i). \tag{6}$$

Conversely, in a regular NG-EPON model, the total overhead using offline scheduling would be computed as follows:

$$O_{REG} = \sum_{i=1}^{K} (T_i^{end} + R_i). \tag{7}$$

Consequently, the total gain G obtained via Deep-DBA can be estimated as follows:

$$G = O_{REG} - O_{Deep-DBA} = \sum_{i=1}^{Q} (T_i^{end} + R_i). \tag{8}$$

For example, for $N = 128$ ONUs, a maximum cycle time of 2 ms, and an upstream speed of 10 Gbps, the *REPORT overhead* would amount to a data rate of 43 Mbps [34]. In contrast, for $P = 2$ and $Q = 6$, Deep-DBA would lower this value to 11 Mbps. This is a gain of 32 Mbps in throughput, which allows to provision more users and/or more services in the network. Moreover, increasing Q and/or decreasing P may further decrease the control overhead; however, this may be

TABLE 2. Simulation parameters.

Number of ONUs	16
Channel speed	1 *Gbps*
Link speed between ONU and user	65 *Mbps*
Distance from OLT to ONU	20 *km*
Guard time	1 μs
Processing time (T_R^{proc} and T_G^{proc})	10 *ns*
Maximum cycle time	2 *ms*
ONU buffer size	10 *MB*
$T_{DBA}, T_{Deep-DBA}$	$\approx$ 0 (negligible)

at the expense of the model prediction accuracy, which might affect the overall network performance as we show in the next section.

IV. PERFORMANCE EVALUATION

To validate the effectiveness of the proposed approach, we generate training data and conduct extensive simulations using OMNET++ [35]. The simulation parameters, as in [36], are shown in Table 2. The 95% confidence interval of the simulation results gave $\approx$ 2% variation, which is statistically insignificant; hence it is not shown in the figures.

A. DATASET

Without loss of generality, to validate the feasibility of the proposed Deep-DBA scheme, in this work, we generate traffic for the Gated and Limited scheduling disciplines, which are the most widely used legacy disciplines for predictive DBA schemes [19]. Namely, we implement a traffic generator at each ONU, which generates Poisson-distributed and Pareto-distributed traffics; the latter has 2000 alternating (i.e., ON/OFF periods) sources to emulate the long-range dependence and self-similarity of bursty Internet traffic [37]. Our proposed scheme does not depend on a particular traffic distribution; rather, it is designed to predict the user demands independent of the traffic arrival distribution. The selection of Poisson and Self-Similar traffics is for simulations purposes, as these are the most commonly adopted distributions in the literature. We note that conducting performance evaluation using real captured datasets may be more persuasive and insightful. However, as detailed in [27], [29], [30], LSTM models (which we also employ in our work) are shown to be efficient and have highly accurate predictions when real traffic traces are used. Thus, evaluating Deep-DBA using LSTM models with a real captured dataset, albeit being more insightful, may not provide different results, and thus may not change the conclusions made in this work. Furthermore, different from previous works [14], we only make use of the request sizes, which are already included in the REPORT messages sent from the ONUs to the OLT, and no extra features are used to train the LSTM model so as not to add additional information in the REPORT messages. This also experimentally proved to be sufficient for predicting

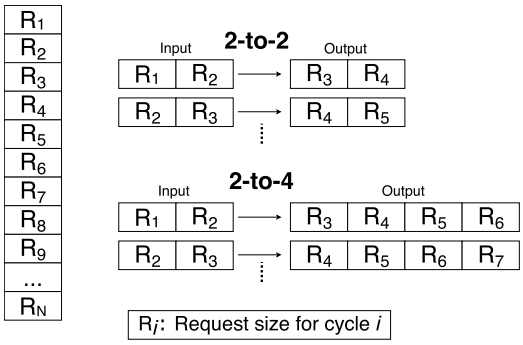

FIGURE 5. Dataset preparation.

requested bandwidth without any added value for extra features.

Overall, our dataset consists of 8 million REPORT messages collected at all the different network loads, which is large enough to build robust LSTM models that generalize well. For different *P*-to-*Q* values, different datasets are prepared as shown in Fig. 5 to train, validate and test the corresponding *P*-to-*Q* LSTM model. For example, if a 2-to-2 model is employed, two REPORT messages are used as input, and the next two REPORT messages are used as output, and so on. As is custom in such settings, 80% of the dataset is used for training, 10% for validation, and 10% for testing.

B. TRAINING THE LSTM MODEL

The proposed scheme employs one LSTM model at the OLT to predict the bandwidth demands of all ONUs at all network loads. The dataset is normalized by dividing each request value by the maximum queue size, and the LSTM model is trained on the entire dataset which consists of the bandwidth demands collected from REPORT messages at all the different network loads. Since our model is trained to predict Internet traffic for all network loads, it does not matter when the peak hour happens and how the traffic volume is changing during the day [38] because the model would be able predict the traffic accordingly. As the loss-function, we use the Mean-Squared Error (MSE) between the predicted queue sizes and the actual queue sizes. The optimizer used to train our models is "AdaGrad" with a learning rate 0.01. For different values of *P* and *Q*, a defined DBA scheme (i.e., Gated, Limited, etc.), NG-EPON architecture, and traffic distribution, the hyper-parameters of the LSTM network are tuned accordingly. The LSTM models had 2 to 3 hidden layers and training each model took between 10 to 50 epochs. We have also considered both the Pareto and Poisson traffic distributions. However, since the results were very similar for both traffic types, we only report the ones for the Pareto distribution as it captures the bursty nature of Internet traffic [37]. We use the Tensorflow backend to build and train our LSTM models [39]. The training was performed on a machine with Intel XEON processor, Nvidia Quadro P2000 GPU card, and 64 GB of RAM. Training each LSTM model took on average about 4 to 8 hours.

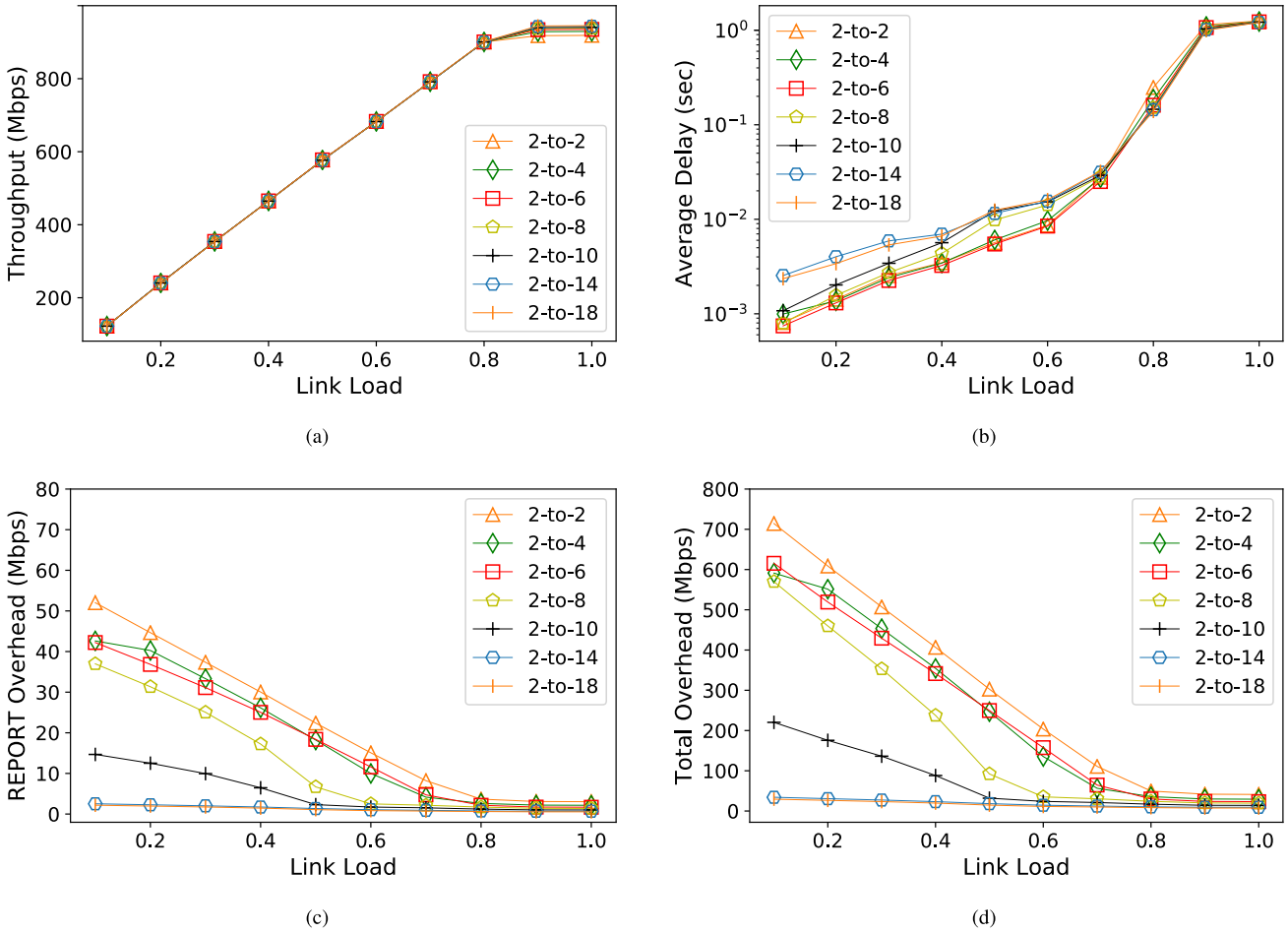

FIGURE 6. Comparison of different *P*-to-*Q* LSTM models: a) Throughput, b) Average delay, c) REPORT overhead, d) Total overhead.

C. SETTING P-TO-Q

To achieve the highest gain using Deep-DBA, P must be set as small as possible, and Q must be set as large as possible. However, the selection of these values must not be at the expense of high prediction error and poor network performance (in terms of packet latency and network throughput). Thus, we built different LSTM models for different P and Q values and compared the obtained MSE by running the models on the test set. For $P = 1$ (i.e., the smallest possible value for P), the LSTM models had significantly high errors; whereas for $P \geq 2$, the errors were adequate. Consequently, we varied the values of P and Q and measured the performance of each built model. As shown in Table 3a, we first built LSTM models with equal values of P and Q starting from 2. Results show that when P and Q are smallest, the lowest MSE is obtained. Next, to check the impact of increasing P, we built LSTM models with $Q = 2$ for different values of P. Results in Table 3b show that increasing P does not yield lower MSE and therefore will not have a positive impact on the performance. Finally, as shown in Table 3c, we set $P = 2$ and increase Q. As expected, the MSE increases as the value of Q increases.

Given these findings, we set $P = 2$, since increasing P would not offer lower MSE values. In addition, increasing P would entail increasing Q to even higher values, which in turn will cause higher prediction errors. For example, setting $P = 2$ and $Q = 8$ means for every $K = 10$ cycles, 8 cycles are prediction cycles, which sums up into 80% of all cycles being prediction cycles. Thus, to obtain the same percentage when $P = 4$, Q must be set to 16.

To choose the best value of Q, we compare the network performance under Deep-DBA with the Limited discipline for different values of Q. As shown in Fig. 6a, the throughput on high loads increases with increasing Q values, since increasing the number of prediction cycles will decrease both the REPORT and T_i^{end} overheads, leaving the gained bandwidth to be used by the ONUs. Fig. 6b shows small difference in delay for different models. However, the lowest delay is obtained for $Q \leq 6$, whereas the delay increases for higher values of Q. This is caused by the mis-predictions of these models; this behavior is related to the obtained MSE errors corresponding to each of these models. Fig. 6c and Fig. 6d highlight how both the REPORT and total overhead (i.e., $T_i^{end} + R_i$) decrease as the value of Q increases.

TABLE 3. Mean square error with: (a) $P = Q$, (b) $P \geq Q$, (c) $P \leq Q$.

P-to-Q	MSE
2-to-2	4.3×10^{-6}
4-to-4	8.8×10^{-6}
6-to-6	9.9×10^{-6}

(a)

P-to-Q	MSE
2-to-2	4.3×10^{-6}
4-to-2	5.7×10^{-6}
6-to-2	6.9×10^{-6}

(b)

P-to-Q	MSE
2-to-2	4.3×10^{-6}
2-to-4	4.8×10^{-6}
2-to-6	8.1×10^{-6}
2-to-8	8.8×10^{-6}
2-to-10	1.1×10^{-5}
2-to-12	1.3×10^{-5}
2-to-14	1.5×10^{-5}
2-to-16	1.7×10^{-5}
2-to-18	1.9×10^{-5}
2-to-20	2×10^{-5}

(c)

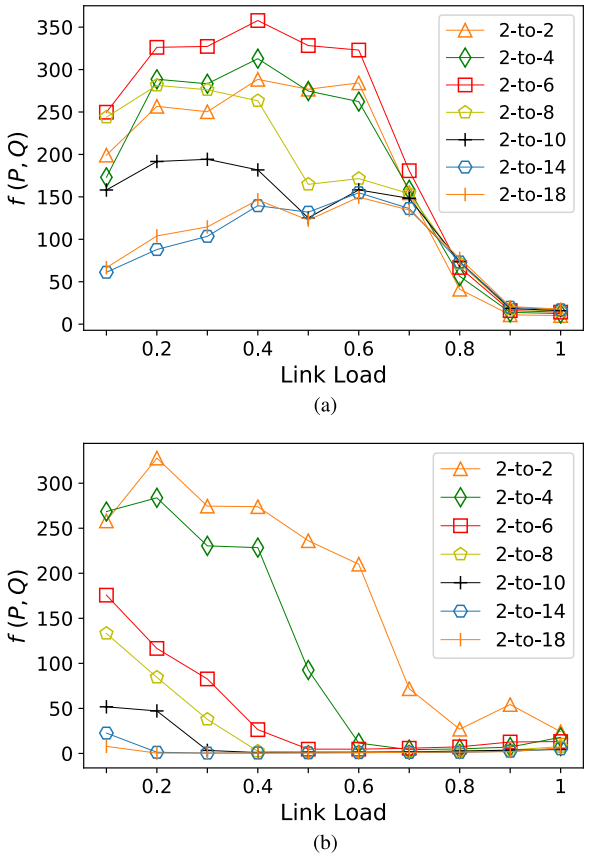

FIGURE 7. Choosing P-to-Q for: a) Limited scheme, b) Gated scheme.

However, for high values of Q, the low REPORT and total overhead bandwidth are due to the long cycle times caused by over-predicting ONU bandwidth demands by the LSTM model. These over-predictions cause the OLT to grant larger transmission windows compared to what is actually needed by the ONUs, which results in wasted bandwidth. Thus, choosing the best Q value would be equivalent to maximizing the throughput, meanwhile minimizing the average delay and total wasted bandwidth (which comprises both the total overhead and prediction error wasted bandwidth). This is equivalent to maximizing the following objective function:

$$f(P, Q) = \frac{throughput(P, Q)}{delay(P, Q) \times waste(P, Q)}. \quad (9)$$

Therefore, to choose the best P-to-Q ratio, we normalize the different parameters, which are obtained from simulations, and plot $f(P, Q)$ in Fig. 7. Results show that the best P-to-Q under the Limited discipline is 2-to-6, with an MSE of 8.1×10^{-6}. In contrast, the best P-to-Q value under the Gated scheme is 2-to-2, with an MSE of 1.7×10^{-4}. Yet, it can be observed that the 2-to-4 model could also achieve a "good-enough" trade-off between an "acceptable" $f(P, Q)$ value and higher network utilization. We note that the MSE under the Gated scheme is higher compared to the Limited scheme due to the high fluctuations of queue sizes, especially at higher loads, making training of such models more difficult.

We validate the performance of the best P-to-Q models under the Limited and Gated schemes in Fig. 8 (i.e., with the 2-to-6, and 2-to-2 models, respectively), by comparing the predicted Internet traffic versus the actual Internet traffic. Here, "Link Load" corresponds to the load on the access link (i.e., between the user and the ONU). We observe that with the Limited scheme, the predicted traffic misses some very short bursts; however, it closely tracks the actual traffic overall. On the other hand, as expected, for the Gated scheme, the margin of mis-predicted queue size and incoming bursts is slightly larger (except at Load 1.0, which makes the system no longer in steady state). Moreover, we deduce that as Q increases and P decreases, the accuracy of the predictions decreases, and vice versa. Hence, there is a trade-off between the number of predicted cycles and the accuracy of predictions.

D. COMPARISON WITH OTHER SCHEMES

Fig. 9 compares the performance of the proposed Deep-DBA scheme under the Limited discipline (i.e., using the 2-to-6 LSTM network) with the prediction-based IPACT with Grant Estimation (IPACT-GE) scheme that predicts the size of incoming requests between two successive cycles [40], the legacy offline Limited (i.e., *Lim-Offline*) DBA scheme, the legacy online Limited (i.e., *Lim-Online*), and the most recent machine learning based predictive DBA (i.e., *MLP-DBA*) [14]. High loads beyond 0.8 are included in the figures because stress-testing the PON performance at high loads is common practice in the literature, and is usually

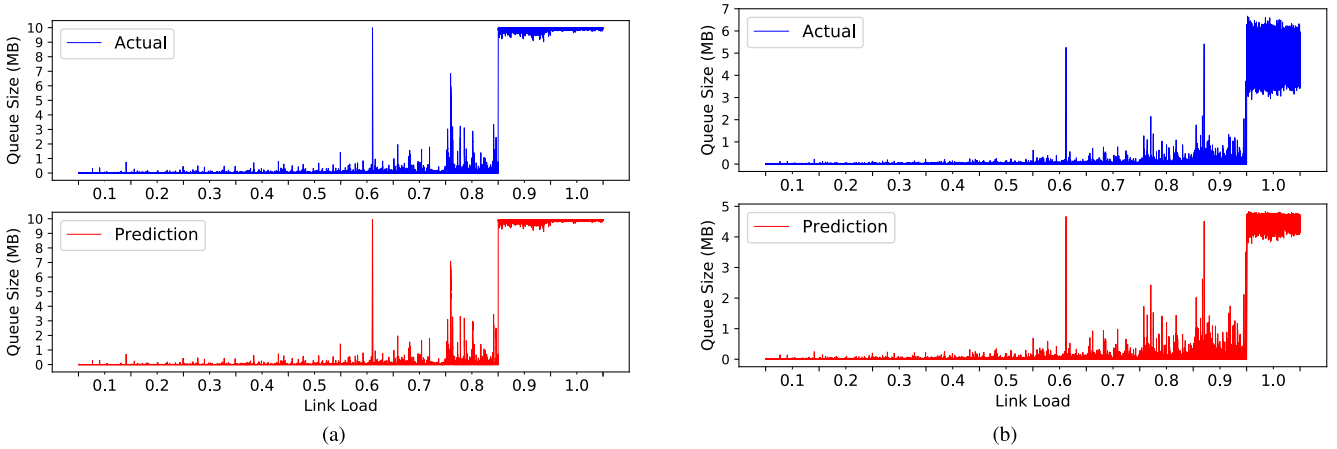

FIGURE 8. Predicted vs. actual bandwidth demand: a) Limited scheme (with 2-to-6), b) Gated scheme (with 2-to-2).

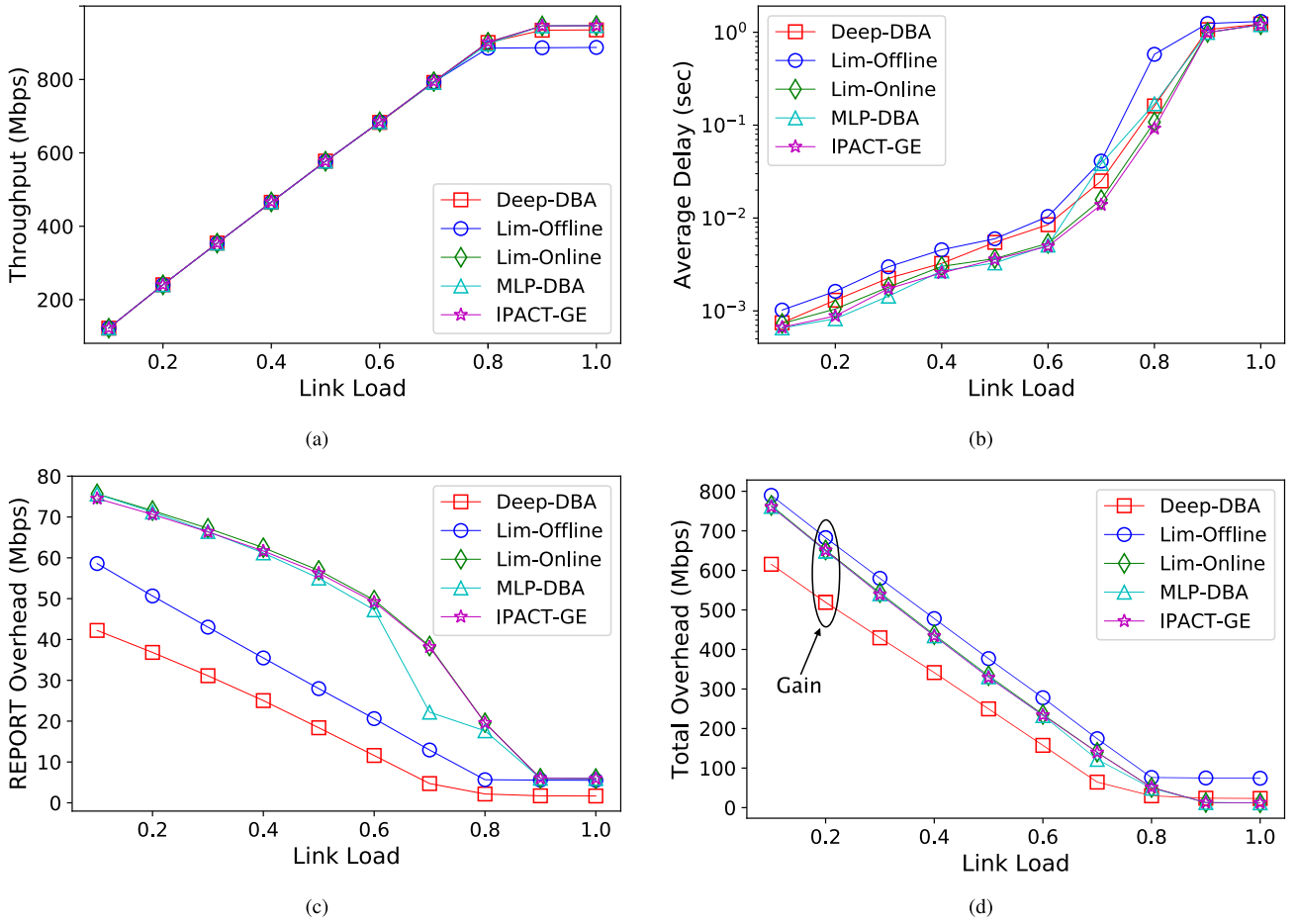

FIGURE 9. Comparison of schemes under the limited discipline: a) Throughput, b) Average Delay, c) REPORT Overhead, d) Total Overhead.

conducted to capture a comprehensive performance measurement, which aids at performing network engineering and user provisioning (which typically accounts for peak utilization).

As shown in Fig. 9a, *Lim-Offline* exhibits the lowest throughput due to the control and T_i^{end} overheads. *Lim-Online*, *MLP-DBA*, and *IPACT-GE* exhibit higher throughput since they are online schemes and thus do not incur the T_i^{end} overhead. Deep-DBA exhibits increased throughput similar to the online schemes even though it is an offline scheme. This improvement is due to the reduction of the control and the T_i^{end} overheads.

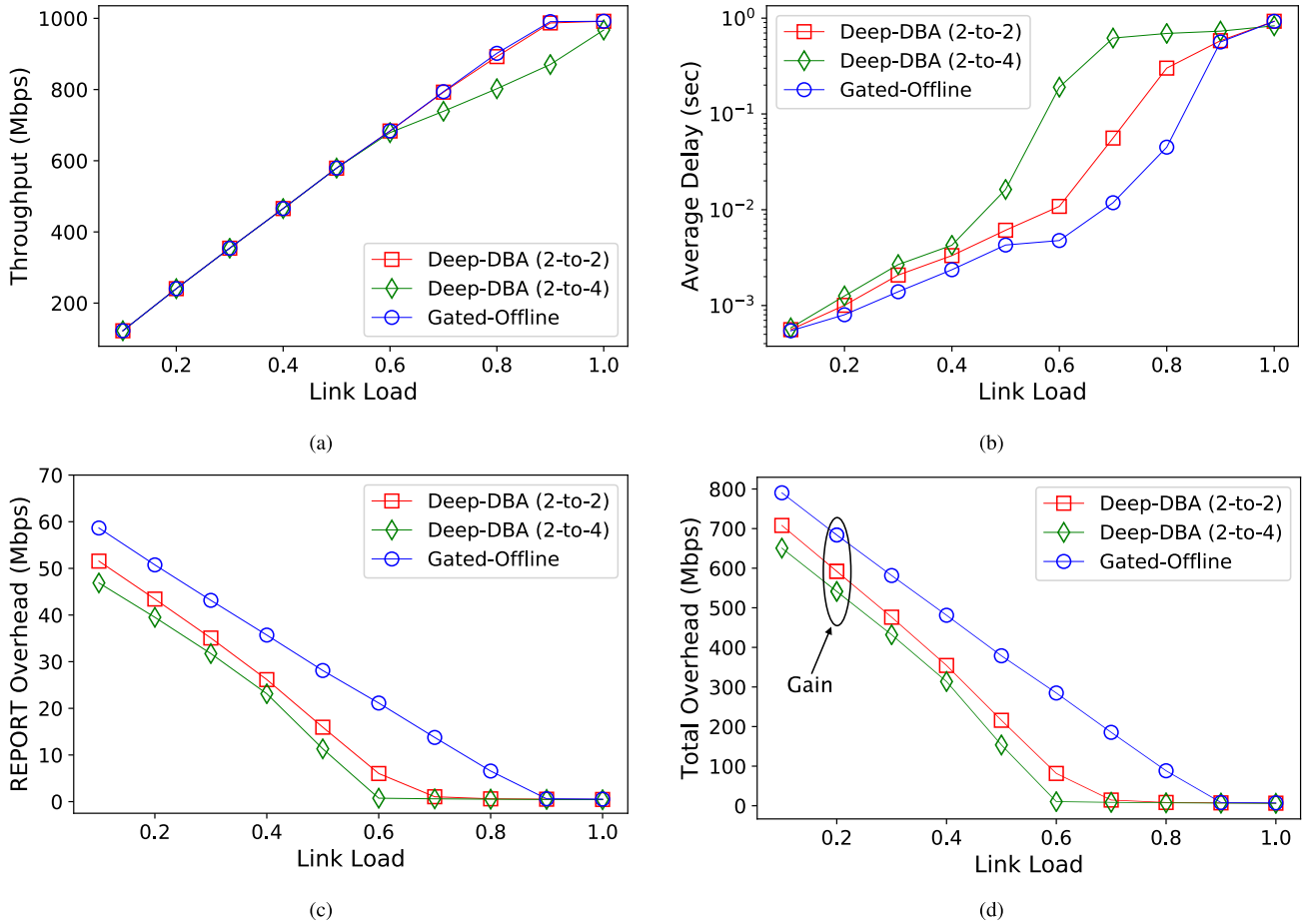

FIGURE 10. Comparison of schemes under the gated discipline: a) Throughput, b) Average Delay, c) REPORT Overhead, d) Total Overhead.

As shown in Fig. 9b, the online schemes exhibit the lowest delay, which is expected since they have no T_i^{end} overhead, which reduces the idle and cycle times. *MLP-DBA* and *IPACT-GE* show slightly better results compared to the legacy *Lim-Online* since they use bandwidth prediction with the specific aim of decreasing the packet delay. The *Lim-Offline* scheme exhibits higher delays due to the overhead T_{REG} calculated in (7). On the other hand, even though Deep-DBA is an offline scheme, its performance is much better than the *Lim-Offline* scheme and is closer to the online schemes. This is due to the gain achieved as per (8). However, as previously mentioned, lower packet delays could be attained using Deep-DBA for different P-to-Q ratios. Nevertheless, these may either affect the prediction accuracy and/or may not achieve the most optimal bandwidth utilization.

The control overhead due to REPORT messages can be observed in Fig. 9c. The online schemes have a higher REPORT overhead than the offline scheme. This is because the online schemes do not have the T_i^{end} overhead, which results in a shorter cycle time compared to the offline schemes. Typically, at lower loads, the cycle time is shorter, which causes more control messages to be exchanged in short periods of time; as such the control overhead decreases as the load increases. However, we notice that Deep-DBA achieves the lowest control overhead over all loads (e.g., around 42 Mbps with Deep-DBA, compared to 60 Mbps for the *Lim-Offline* scheme, and around 75 Mbps for the online schemes). At higher loads, the cycle time is equal to the maximum cycle time; hence, the control overhead reaches its lowest value for all schemes (e.g., 5.5 Mbps with IPACT-GE, offline Limited, and offline Limited-GE), whereas it is equal to 1.5 Mbps with Deep-DBA. This highlights the advantages of Deep-DBA, which enables the OLT and ONUs to only exchange control messages in reporting cycles every K cycles, as opposed to every cycle.

Fig. 9d shows the total overhead observed in the network (which is the control messaging overhead plus the cycle idle time) under all schemes. As noticed, even though online schemes are optimized to reduce the idle time, Deep-DBA is still able to achieve the lowest total overhead. This bandwidth gain can be used so that more users can be provisioned in the network, and better QoS support can be attained. This again highlights the merits of Deep-DBA over existing approaches.

In Fig. 10, we compare the performance of Deep-DBA under the Gated discipline (i.e., using the 2-to-2 and 2-to-4 LSTM models) with the offline Gated DBA scheme (*Gated-Offline*). Due to their nature, the prediction of IPACT-GE in [40] and MLP-DBA in [14] cannot be directly

applied to an offline Gated scheme. As shown in Fig. 10a, the 2-to-2 Deep-DBA provides the same throughput as the offline Gated discipline, whereas the 2-to-4 provides lower throughput, which is captured by $f(P, Q)$ in (9).

In Fig. 10b, the average delay with Deep-DBA is a bit higher than with *Gated-Offline*. This is due to the ''mis-predictions'' of the LSTM model, which will make the ONU buffer more packets (thus, request more bandwidth), thereby making the OLT grant the ONUs larger transmission windows even at low loads. This effect is caused by the nature of the Gated discipline, which unlike the Limited discipline, does not bound the bandwidth demand by a maximum value; thus, the mis-prediction would have a snowballing effect.

Finally, Fig. 10c and Fig. 10d show how Deep-DBA exhibit lower REPORT and total overheads than *Gated-Offline*. More importantly, the results here show how the chosen model (i.e., 2-to-2 or 2-to-4) presents a tradeoff between higher bandwidth utilization (that is, higher bandwidth gain) and downgraded network performance (i.e., throughput and delay).

All these foregoing experiments highlight the merits of Deep-DBA in combining the advantages of offline and online schemes. They also demonstrate its ability to be adaptive to any network configuration and settings. In other words, as long as the dataset (whether it is real traces or generated via simulations) is available, the right machine learning model can be built, and consequently an efficient Deep-DBA scheme can be employed. More importantly, the results provide insights into the design of future machine learning-based NG-EPONs.

V. CONCLUSION

In this paper we proposed Deep-DBA, a novel DBA scheme for NG-EPONs, which employs deep learning to predict the future bandwidth demands of ONUs by peep-holing only a few previous ONU demands, so as to reduce the overhead due to the request-grant mechanism. Results demonstrate how Deep-DBA is able to combine the advantages of both the online and offline schemes, thereby improving the network utilization achieved with online schemes, and at the same time maintaining the properties of fairness and QoS support that offline schemes enable, without impairing the network's performance. The fast progress in the field of machine learning promises new and better architectures and techniques that will be able to increase the number of prediction cycles and decrease the prediction error. The proposed method has the flexibility to employ any current or future sequence-to-sequence machine learning model. Moreover, Deep-DBA can operate with any scheduling scheme; thus, future works can employ more schemes to further improve the network performance. In our future extensions of this work, we will use real traffic traces and/or traffic emulators to provide a more comprehensive study.

REFERENCES

[1] ''The zettabyte era: Trends and analysis,'' Cisco, San Jose, CA, USA, White Paper 1551296909190103, 2017.

[2] G. Kramer, *Ethernet Passive Optical Networks*. New York, NY, USA: McGraw-Hill Professional, 2005.

[3] F. Musumeci, C. Rottondi, A. Nag, I. Macaluso, D. Zibar, M. Ruffini, and M. Tornatore, ''An overview on application of machine learning techniques in optical networks,'' Mar. 2018, *arXiv:1803.07976*. [Online]. Available: https://arxiv.org/abs/1803.07976

[4] J. Mata, I. De Miguel, R. J. Duran, N. Merayo, S. K. Singh, A. Jukan, and M. Chamania, ''Artificial intelligence (AI) methods in optical networks: A comprehensive survey,'' *Opt. Switching Netw.*, vol. 28, pp. 43–57, Apr. 2018.

[5] Y. Zhao, B. Yan, D. Liu, Y. He, D. Wang, and J. Zhang, ''SOON: Self-optimizing optical networks with machine learning,'' *Opt. Express*, vol. 26, no. 22, pp. 28713–28726, 2018.

[6] B. Yan, Y. Zhao, X. Yu, W. Wang, Y. Wu, Y. Wang, and J. Zhang, ''Tidal-traffic-aware routing and spectrum allocation in elastic optical networks,'' *J. Opt. Commun. Netw.*, vol. 10, no. 11, pp. 832–842, 2018.

[7] J. Yuan, Y. Fu, R. Zhu, X. Li, Q. Zhang, J. Zhang, and A. Samuel, ''A constrained-lower-indexed-block spectrum assignment policy in elastic optical networks,'' *Opt. Switching Netw.*, vol. 33, pp. 25–33, Jul. 2019.

[8] N. Merayo, D. Juárez, J. C. Aguado, I. de Miguel, R. J. Durán, P. Fernández, R. M. Lorenzo, and E. J. Abril, ''PID controller based on a self-adaptive neural network to ensure QoS bandwidth requirements in passive optical networks,'' *J. Opt. Commun. Netw.*, vol. 9, no. 5, pp. 433–445, May 2017.

[9] N. E. Frigui, T. Lemlouma, S. Gosselin, B. Radier, R. L. Meur, and J.-M. Bonnin, ''Dynamic reallocation of SLA parameters in passive optical network based on clustering analysis,'' in *Proc. 21st Conf. Innov. Clouds, Internet Netw. Workshops (ICIN)*, Feb. 2018, pp. 1–8.

[10] N. E. Frigui, T. Lemlouma, S. Gosselin, B. Radier, R. L. Meur, and J.-M. Bonnin, ''Optimization of the upstream bandwidth allocation in passive optical networks using Internet users' behavior forecast,'' in *Proc. Int. Conf. Opt. Netw. Design Modeling (ONDM)*, May 2018, pp. 59–64.

[11] P. Sarigiannidis, D. Pliatsios, T. Zygiridis, and N. Kantartzis, ''DAMA: A data mining forecasting DBA scheme for XG-PONs,'' in *Proc. 5th Int. Conf. Modern Circuits Syst. Technol. (MOCAST)*, May 2016, pp. 1–4.

[12] L. Ruan and E. Wong, ''Machine intelligence in allocating bandwidth to achieve low-latency performance,'' in *Proc. Int. Conf. Opt. Netw. Design Modeling (ONDM)*, May 2018, pp. 226–229.

[13] E. Wong, M. P. I. Dias, and L. Ruan, ''Predictive resource allocation for tactile Internet capable passive optical LANs,'' *J. Lightw. Technol.*, vol. 35, no. 13, pp. 2629–2641, Jul. 1, 2017.

[14] L. Ruan, S. Mondal, and E. Wong, ''Machine learning based bandwidth prediction in Tactile heterogeneous access networks,'' in *Proc. IEEE INFOCOM-IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2018, pp. 1–2.

[15] Y. Wu, M. Tornatore, Y. Zhao, and B. Mukherjee, ''Traffic classification and sifting to improve TDM-EPON fronthaul upstream efficiency,'' *J. Opt. Commun. Netw.*, vol. 10, no. 8, pp. C15–C26, 2018.

[16] 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,'' *J. Internet Serv. Appl.*, vol. 9, p. 16, Jun. 2018.

[17] A. R. Dhaini, P.-H. Ho, and G. Shen, ''Toward green next-generation passive optical networks,'' *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 94–101, Nov. 2011.

[18] J. Zheng and H. T. Mouftah, ''A survey of dynamic bandwidth allocation algorithms for Ethernet passive optical networks,'' *Opt. Switching Netw.*, vol. 6, no. 3, pp. 151–162, 2009.

[19] M. P. Mcgarry, M. Reisslein, and M. Maier, ''Ethernet passive optical network architectures and dynamic bandwidth allocation algorithms,'' *IEEE Commun. Surv. Tuts.*, vol. 10, no. 3, pp. 46–60, 3rd Quart., 2008.

[20] S. Hochreiter and J. Schmidhuber, ''Long short-term memory,'' *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, ''Sketch-based change detection: Methods, evaluation, and applications,'' in *Proc. 3rd ACM SIGCOMM Conf. Internet Meas.*, 2003, pp. 234–247.

[22] A. Sang and S.-Q. Li, ''A predictability analysis of network traffic,'' *Comput. Netw.*, vol. 39, no. 4, pp. 329–345, 2002.

[23] K. Papagiannaki, N. Taft, Z.-L. Zhang, and C. Diot, ''Long-term forecasting of Internet backbone traffic,'' *IEEE Trans. Neural Netw.*, vol. 16, no. 5, pp. 1110–1124, Sep. 2005.

[24] A. Esposito, M. Faundez-Zanuy, F. C. Morabito, and E. Pasero, *Multidisciplinary Approaches to Neural Computing*, vol. 69. Cham, Switzerland: Springer, 2018.

[25] P. Cortez, M. Rio, M. Rocha, and P. Sousa, "Multi-scale Internet traffic forecasting using neural networks and time series methods," *Expert Syst.*, vol. 29, no. 2, pp. 143–155, May 2012.
[26] M. Barabas, G. Boanea, A. B. Rus, V. Dobrota, and J. Domingo-Pascual, "Evaluation of network traffic prediction based on neural networks with multi-task learning and multiresolution decomposition," in *Proc. IEEE 7th Int. Conf. Intell. Comput. Commun. Process.*, Aug. 2011, pp. 95–102.
[27] A. Azzouni and G. Pujolle, "NeuTM: A neural network-based framework for traffic matrix prediction in SDN," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2018, pp. 1–5.
[28] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," Feb. 2014, *arXiv:1402.1128*. [Online]. Available: https://arxiv.org/abs/1402.1128
[29] R. Vinayakumar, K. Soman, and P. Poornachandran, "Applying deep learning approaches for network traffic prediction," in *Proc. Int. Conf. Adv. Comput., Commun. Inform. (ICACCI)*, Sep. 2017, pp. 2353–2358.
[30] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *Proc. IEEE Int. Conf. Smart City/SocialCom/SustainCom (SmartCity)*, Dec. 2015, pp. 153–158.
[31] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," Mar. 2017, *arXiv:1703.04691*. [Online]. Available: https://arxiv.org/abs/1703.04691
[32] A. Mozo, B. Ordozgoiti, and S. Gómez-Canaval, "Forecasting short-term data center network traffic load with convolutional neural networks," *PLoS ONE*, vol. 13, no. 2, 2018, Art. no. e0191939.
[33] A. Azzouni and G. Pujolle, "A long short-term memory recurrent neural network framework for network traffic matrix prediction," May 2017, *arXiv:1705.05690*. [Online]. Available: https://arxiv.org/abs/1705.05690
[34] R. Roy, G. Kramer, M. Hajduczenia, and H. J. Silva, "Performance of 10G-EPON," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 78–85, Nov. 2011.
[35] A. Vargas. *Omnet++*. Accessed: Jul. 2019. [Online]. Available: http://www.omnetpp.org/
[36] A. R. Dhaini, P.-H. Ho, G. Shen, and B. Shihada, "Energy efficiency in TDMA-based next-generation passive optical access networks," *IEEE/ACM Trans. Netw.*, vol. 22, no. 3, pp. 850–863, Jun. 2014.
[37] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Trans. Netw.*, vol. 2, no. 1, pp. 1–15, Feb. 1994.
[38] M. Feknous, T. Houdoin, B. Le Guyader, J. De Biasio, A. Gravey, and J. A. T. Gijón, "Internet traffic analysis: A case study from two major European operators," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2014, pp. 1–7.
[39] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, and S. Ghemawat. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://www.tensorflow.org/
[40] Y. Zhu and M. Ma, "IPACT with grant estimation (IPACT-GE) scheme for Ethernet passive optical networks," *J. Lightw. Technol.*, vol. 26, no. 14, pp. 2055–2063, Jul. 15, 2008.

JOHN ABIED HATEM received the B.S. degree in computer science from Haigazian University, Beirut, Lebanon, in 2016, and the M.S. degree from the American University of Beirut (AUB), Beirut, in 2019, where he is currently a Research Assistant with the Computer Science Department. His current research interests include AI, machine learning, big data, passive optical networks, the IoT, routing protocols, and RPL.

AHMAD R. DHAINI received the B.Sc. degree in computer science from the American University of Beirut (AUB), in 2004, the M.Sc. degree in electrical and computer engineering from Concordia University, Canada, in 2006, with a best thesis award nomination, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Canada, in 2011.

From 2006 to 2007, he was a Software Analyst and Consultant with TEKSystems, Canada. From 2007 to 2008, he was a Software Designer at Ericsson, Canada. From 2011 to 2012, he was a Research Associate with the University of Waterloo, Canada, and as a Consultant at KAUST, Saudi Arabia. From 2012 to 2014, he was a Postdoctoral Scholar with the Photonics and Networking Research Laboratory (PNRL), Stanford University, after being awarded the prestigious NSERC postdoctoral fellowship. He also completed the Stanford Ignite program for entrepreneurship and innovation, which teaches scientists how to convert an idea into a business. In 2017, he was a Visiting Assistant Professor with the University of Waterloo, while on leave from American University of Beirut (AUB). He is currently an Assistant Professor of computer science with AUB. He is a co-inventor of two US patents. He has also authored/coauthored more than 50 highly cited research articles in top IEEE journals and conferences. His research interests include several themes of integrated networks such as fiber-wireless (FiWi) broadband access networks, mission-critical networks, green communications, edge computing, and software-defined networking. He has also co-PIed several projects related to biotechnology, more specifically in the areas of mobile health and medical image analysis.

Dr. Dhaini was a reviewer and technical program committee (TPC) member of several major IEEE journals and conferences. He was granted several awards, such as the Ontario Graduate Scholarship in Science and Technology (OGSST), and other various teaching and research awards from the University of Waterloo. He serves as an Associate Editor of IEEE Access, and an Editor of *Photonics Networks Communications* journal (Springer). He is a reviewer of NSF, NSERC, and several US universities' internal grants.

SHADY ELBASSUONI received the Ph.D. degree from the Max-Planck Institute for Informatics (MPII), Germany. He was a Postdoctoral Researcher with the Qatar Computing Research Institute (QCRI). He is currently an Assistant Professor with the Computer Science Department, American University of Beirut. His current research spans multiple areas, including machine learning, information retrieval and crowd sourcing, and their applications for public health and digital humanities.