On Short- and Long-Term Traffic Prediction in Optical Networks Using Machine Learning

Michał Aibin, Nathan Chung, Tyler Gordon, Liam Lyford, Connor Vinchoff
Department of Computing, British Columbia Institute of Technology, Canada
Email: maibin@bcit.ca

***Abstract*—In this paper, we formulate the problem of traffic prediction in optical networks. We then design a machine learning approach based on Graph Convolutional Network and the Generative Adversarial Network to enable efficient network states forecasting. The main focus is on detecting the peak traffic in networks that can affect the routing decisions. We validate our results using pseudorealistic datasets generated in a custom simulator and real networks provided by the network operator. The findings confirm our approach's efficiency for optimizing both the real-time routing and long-term network design decisions.**

I. Introduction

Because of its ease of use and high fault tolerance, cloud computing has attracted a large number of companies in recent years. As a result, demand for high-data transmission is increasingly growing [1]. Major cloud providers, such as Google, Amazon, and Microsoft, actively invest and vying for a more significant share of the market. However, the current Internet infrastructure is not suited to scale with this growth in demand. New technologies, such as Spectrally-Spatially Flexible Optical Networks (SS-FONs), have been proposed as a possible solution for handling the increased traffic [2]. The 'elasticity' of an SS-FON is determined by two dimensions: the dynamic spectrum and the space assignment. SS-FON is the advanced version or next-generation Dense Wavelength Division Multiplexing (DWDM) technology [3]. The dimensions we can optimize include space, bandwidth, and wavelength, thanks to independent spectral and spatial fibre resource management. The space dimension in fibres is mainly implemented to allow for space-division multiplexing (SDM) and versatile wavelength allocation to improve overall transmission power.

Moreover, even by introducing new technologies, we cannot avoid reaching physical limits. Studies suggest that by 2030 optical networks will meet the problem of capacity crunch [4]. Instead of looking into updating the technology, we can look into more advanced methods to control it. These new models are needed to extract valuable information from a comprehensive set of network data and create a concept of a cognitive network [5]. A cognitive network is a type of network that utilizes advanced analytical solutions from several research areas (i.e., deep learning, data analytics, knowledge representation, telecommunication, network management) to solve modern problems in communication networks [6]. We can define a cognitive optical network as a transport network that uses a cognitive process to perceive current network conditions to plan, decide, act on those conditions, learn from the historical data, and forecast future events to achieve end-to-end goals. The cognitive processes, which learn from historical data to improve performance, apply various data analytics solutions typically utilizing machine learning techniques. In particular, Data Analytics, Machine Learning and Deep Learning concepts are regarded as promising methodological areas to enable cognitive network data analysis, thus enabling advanced strategies of resource allocation. The key performance metrics of optical networks that we want to improve using cognitive methods are capital and operational expenditures (CAPEX and OPEX), network resources and energy consumption.

Current research suggests that network resource provisioning algorithms that implement cognitive networks via traffic prediction, such as Monte Carlo Tree Search (MCTS), result in the most efficient overall optical resource usage. Such algorithms are unfortunately relatively sensitive to traffic load pattern changes, i.e., burst data [7]. MCTS converges to a terminal state through a random selection of action; accuracy is thus directly correlated with the number of calculation cycles processed within the computational budget allocated to the network prediction algorithm. We believe that reducing the selection space by introducing data pattern sensitivity will increase the probability that a viable path will be found within the computational budget. To do so, we will use the advantages of the graph convolutional network (GCN) [8] and the generative adversarial network (GAN) [9], [10]. More specifically, we will first use the GCN to capture each single graph snapshot's topological structure characteristics. The studied network features are then going to be used in GAN, a class of machine learning systems that pits two neural networks against each other. In our model, one network, called the generator, will generate network traffic to create burst traffic. Then the other network, called the discriminator, will try to predict the appearance of future burst phenomena from current state data in order to decrease Request Blocking Percentage (BP) by adjusting the network resource provisioning strategy.

A major challenge facing dynamic routing algorithms is the effect of spontaneous peak network traffic phenomena (i.e. FIFA World Cup, Olympics) [11]. Furthermore, as background network traffic increases, the effect of such peak phenomena becomes more significant, as fewer network resources are available to accommodate the load. The ability of a dynamic routing algorithm to handle routing traffic peaks (in other words "flash events") is highly dependent upon its traffic prediction mechanisms. Routing algorithms that do not include traffic prediction regarding the overall network state inefficiently capture unusual peak data characteristics. We argue that implementing a GCN-GAN model in traffic prediction to increase pattern sensitivity in network resource provisioning will decrease BP.

A. Key Insight

This paper's main novelty is the implementation of a deep learning technique, the GCN-GAN method, to solve the

978-3-903176-33-1

problem of short and long-term traffic prediction in real optical networks. Telus PureFibre provided data, and thanks to that, we can verify the effectiveness of our resource allocation method in dynamic routing in real-life scenarios. We compare the GCN-GAN to the Long-Short Term Memory method. This paper is follow-up research based on the work presented in [12], where we used GCN-GAN for traffic prediction. The main differences are: more widely studied scenarios with various traffic matrices; the study of the network planning phase, where we can use our approach to identify parts of the network to be updated/expanded; validation of our results using the real deployed optical networks.

B. Contributions

The following are the main contributions of this paper.

- **Approach:** We study the optical network resource allocation problem. We use the datasets based on the Telus PureFibre historical data and real network simulations (based on Canadian PureFibre networks).
- **Implementation:** We implement the GCN-GAN method to solve the problem mentioned above. We then compare it with Long-Short Term Memory as a reference method from the literature. We are not comparing those two approaches to other baseline solutions, such as Shortest Path First, as in other works, we proved that our approach is much more efficient than the one in [13]. Our implementation achieves fast computation time, and it is learning about the traffic in real-time, which means it can be easily deployed to real-time optical network operations.
- **Evaluation:** We evaluate our algorithms by utilizing a pre-existing CEONS simulator [14]. We then deploy the algorithms to real networks to confirm our findings.

The remainder of the paper is divided as follows. In Section III, we introduce the optimization problem and describe the network model. Section IV contains information about the algorithms used in this paper. In Section V, we present a simulation setup, followed by the presentation of results, and finally, the conclusion in Section VII.

II. Related Works

Traffic prediction algorithms have received considerable attention in other network system domains, such as urban physical traffic. Deep learning models in these domains often suffer from blurry prediction due to the difficulty of accurately averaging possible future problem states and often consider only the next iteration of possible states for prediction, which does not fully capture the spatial-temporal features of the data. Newer models implement adversarial training to minimize blurry prediction errors in multi-step traffic prediction and graph convolution to generate future states that more accurately reflect spatial-temporal features and have been shown to surpass non-adversarial models in this domain [15].

In [16], the authors propose and evaluate a model called Pattern-Sensitive Networks (PSNs) to solve the problem of handling traffic fluctuations through unsupervised learning. Overall, the authors conclude that PSNs better capture the variation patterns implicit in historical traffic flow and, therefore, better identify potential future traffic fluctuations while significantly reducing the model's training time. However, the PSN performed slightly worse than LSTM-based models under normal traffic conditions and only 3.14% better in extreme conditions.

In [11], the authors propose an approach to dealing with traffic fluctuations by applying a novel light-splitting scheme in EONs. The authors show that significant throughput improvement can be achieved via dynamic bandwidth reconfiguration for degraded service provisioning during network congestion. The authors also noted that optical path re-routing and wavelength defragmentation are the limiting factors for significant performance improvement when optimizing network resources. While the authors focused on the latter, our focus will be on the former as we look to optimize the prediction of viable candidate paths.

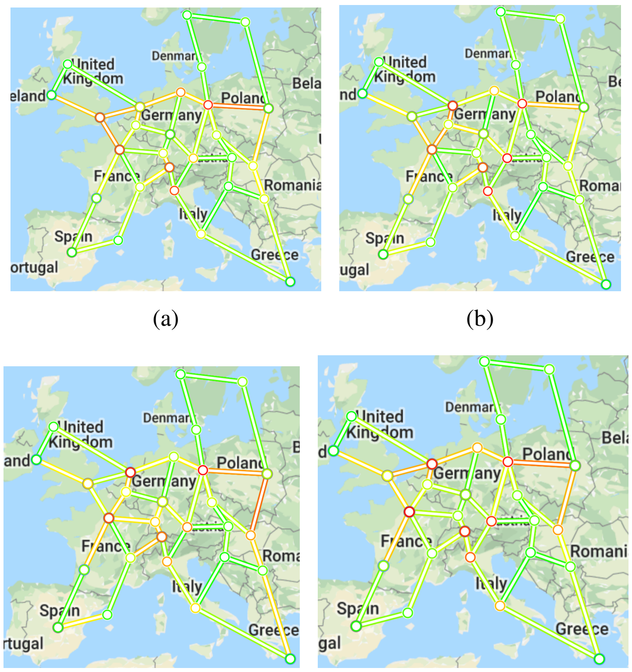
In [17], the authors propose and evaluate a non-linear model called Graph Convolutional Network Generative Adversarial Network (GCN-GAN) designed to predict temporal link dynamics in weighted dynamic networks. Their model combines GCNs and Long Short-Term Memory (LSTMs) networks to capture evolving patterns in successive graph time slices, which is used to train a generative-adversarial model to create plausible future states. The authors conclude that the GCN-GAN outperforms six competitors, including a generic LSTM, especially in networks with sparse edge weights but have yet to implement a concrete deployment. Please note that this work uses a similar approach, but we study a different situation. We want to predict the "flash events" (defined later in this paper) to allow network operators (such as Telus) to redirect the traffic to use the existing resources efficiently. Moreover, we are using GCN-GAN to allow network operators to expand further the network, based on the existing bottlenecks discovered thanks to our approach.

In [13], the authors propose and evaluate the effectiveness of applying machine learning techniques for traffic prediction. They compared AMRA, Genetic Algorithm (GA), and MNC with and without Monte Carlo Tree Search (MCTS) and Artificial Neural Network (ANN) traffic prediction. The authors also compared SPF for medium to high traffic loads and IBM CPLEX Solver for lower traffic loads. The authors concluded that MCTS handles traffic changes in a shorter period. The GA+MCTS outperformed AMRA+MCTS on low traffic conditions, but AMRA+MCTS outperformed on higher traffic loads. They concluded that it was because of the number of decisions to be made in a short amount of time. AMRA+MCTS performed well in all scenarios.

III. Problem Definition

Given that the optical network can be defined as a dynamic network [18], we can consider the network state (network snapshot) at a given time t as an undirected graph $G_t(V, E)$, where V denotes a set of vertices (network nodes) and E is a set of directed edges (fibre links). Given the snapshots of previous time slices and current time slice

$\{G_{t-(x+1)}(V,E), G_{t-x}(V,E), ..., G_t(V,E)\}$ (with x equal to the total snapshots number), the goal of is to predict the topology state of the next time slice $(t+1)$. In other words, the proposed GCN-GAN model predicts the $G_{t+1}(V,E)$, based on all historical states $\sum_0^x G_{t-x}(V,E)$. Let us look at the Figure 1. Part 1a refers to network state $G_{t-1}(V,E)$, and part 1b to $G_t(V,E)$. Based on those two snapshots, we generate the prediction of state $G_{t+1}(V,E)$, as seen in Figure 1c. It is then validated against the actual result, as seen in Figure 1d.



(a) (b) (c) (d)

Figure 1: Network states: (a) $G_{t-1}(V,E)$ (b) $G_t(V,E)$. Prediction (c) $G_{t+1}(V,E)$ Ground Truth (d) $G_{t+1}(V,E)$.

While the graph nodes and links directly correspond to the nodes and links in the network topology, the weight set used to derive the adjacency matrix has no similar directly corresponding data. In order to define the weight set, we must first consider how the network traffic is generated. Our model considers the network traffic to be composed of a set of requests D that originate from a source node s and terminate at a destination node f. As requests represent the allocation of bandwidth for data transfer, a request $d \in D$ contains the volume of bandwidth required and the time required for the request to finish transmitting data, indicated by $c(d)$. Thus, our model defines the network traffic in a given time slice t to be the combined bandwidth volume of all requests with remaining time to live in the network, leading to a weight set corresponding to the total Erlang value (traffic intensity value) in each link. The goal is to predict the weight set of each link in the next time slice $t+1$.

IV. Algorithms

In summary, the GCN-GAN approach works by predicting a potential future network state from current input and historical network state data, as shown in Figure 1. After predicting $G_{t+1}(V,E)$ is generated, it is then compared with actual data coming in the next iteration/network state. The complete GCN-GAN model is made up of three parts: a graph convolutional network (GCN), a generational adversarial network (GAN), and a simple feedforward neural network. As shown in Figure 2, the first two act as a generative network, while the last one acts as a discriminative network.

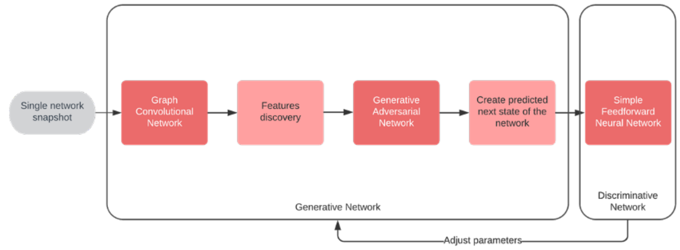


Figure 2: GCN-GAN flowchart

The execution steps are as follows. First, the GCN captures the characteristics of a single graph snapshot (network state). Then, the resulting representations of the snapshot are used to discover network features. As a result, we have a feature set $F \in \mathbb{R}^{\mathbb{N}}$ and adjacency matrix $A \in \mathbb{R}^{\mathbb{N}\times\mathbb{N}}$ of each network snapshot. Next, the GAN uses those sets to generate the network's predicted state. In the adversarial process, the generative part of the GAN will construct a realistic graph. Simultaneously, the discriminator will identify the most likely future graphs, which will imply future network states. The generator will continue to adjust in this process until it achieves a satisfactory prediction result. Please keep in mind that this procedure is repeated for each and every network snapshot that is available. This is a similar process to the one described in more detail in [17]. However, we use the model in our procedure to detect burst traffic and change resource allocation strategies to lower the Blocking Percentage of requests and plan potential network expansions by identifying the network bottlenecks.

V. Simulation Setup

The authors of [19] characterize a flash event (which occurred during the 1998 FIFA World Cup) that we will refer to as a "double-burst" scenario (DB). DB events can be defined as a large increase in traffic followed by a slight decrease. A secondary higher-traffic burst occurs once the decline is present (caused, e.g., by unpredictable events, such as a penalty shootout). We can then identify "single-burst" flash events (SB), which do not have secondary escalated bursts. After the initial traffic spike, single-burst events have a long and steady traffic rate decline. A final event, "plateau-burst" (P), has a high similarity to the initial traffic spike seen in both single and double bursts, but it is designed to produce false-positives as the traffic increase stops rapidly and randomly. Thus, if the GCN-GAN can correctly determine that a plateau-burst scenario is neither a single-burst nor a double-burst, the model can correctly recognize these scenarios' distinguishing characteristics.

We created three simulation scenarios: short-term pseudorealistic data generated with the Complex Elastic Optical Network Simulator (CEONS) [14] and used for training and testing of our model; short-term real-life fibre network data and long-term real-life fibre network data obtained from Telus PureFibre for validation of our model in larger, real-life optical networks. Each request was characterized by the destination node, source node, Time To Live (TTL), and traffic volume in Gb/s. The TTL, in this case, indicates how long the request will be in the network, decreasing by one in each simulation iteration before it reaches zero. We then generated batches of data with SB, DB, and P traffic characteristics. We created one snapshot of the topology per hour, with each simulation tick representing a minute, resulting in $x = 168$ snapshots for short-term data (one week) and 15000/45000/90000 snapshots for long-term data (3/6/12 months of requests, respectively). After that, we used 70% of the short-term results for training and 30% for testing and validation. Finally, we tested our learned model on existing networks to see how it could be used in different contexts without retraining. The real network consists of 32 nodes and is located in British Columbia, Canada. Since we have 32 nodes in the network, the GCN-GAN model layer configuration is as follows. We use 32×32 neurons as an input layer (referring to $A \in \mathbb{R}^{\mathbb{N} \times \mathbb{N}}$), 32 neurons in the hidden layer and 32×32 neurons as an output layer. The primary evaluation metric that we used is the Mean Squared Error (MSE). MSE is the average squared difference between the estimated values and the actual value. We define our MSE as:

$$MSE = \frac{1}{t} \sum_{1}^{T} (|g_t - \tilde{g}_t|)^2$$

where g_t is the real traffic, while $\tilde{g}_t$ is predicted one.

VI. Performance Evaluation

The authors of [17] used four datasets to compare their model to six baseline approaches. After running simulations, they found that the LSTM model provided better results than the other ones. We opted to compare the GCN-GAN to the LSTM using three simulation scenarios since the LSTM performed substantially better than all other approaches. In the first evaluation, we trained our GCN-GAN separately for every single type of event, i.e., Single-Burst (SB), Double-Burst (DB) and Plateau-Burst (P). For example, we trained the LSTM and GCN-GAN on the SB training set and then validated its performance using the SB testing set. Similarly, the process was repeated for other datasets, training a new model using each one of them. Each scenario dataset consisted of pseudorealistic data generated through the CEONS simulator. Because of that, our GCN-GAN could identify each event type's main features, so later, we could use the same model with traffic that included all types of events. As we can see in Table I, the GCN-GAN achieved better results than the LSTM, offering lower MSE.

Table I: Short-Term Pseudorealistic Data Performance Evaluation Results

Methods	Average MSE		
	SB	DB	P
LSTM	0.0052	0.0061	0.0108
GCN-GAN	**0.0040**	**0.0060**	**0.0089**

Once the model was trained, we performed three additional performance evaluations, using real short-term traffic (one week of data), without retraining the model (as seen in Table II). First, we needed to find the specific "time windows" in the Telus PureFibre networks in which SB, DB, and P burst types happened. In each of the datasets, we required at least one SB or DB or P event. We then run simulations using ten datasets per event and presented the average MSE of each category (Table II). The GCN-GAN was still able to outperform the LSTM method. Its superior performance was evident in detecting a P-type event.

Table II: Short-Term Real Data Performance Evaluation Results

Methods	Average MSE		
	SB	DB	P
LSTM	0.0056	0.0054	0.0093
GCN-GAN	**0.0034**	**0.0028**	**0.0028**

Finally, we used the same trained model and applied it to real-life networks long-term data (Table III) with all types of events, i.e. SB, DB and P. This time, we did not look for specific time windows; instead, we just run simulations using different scenarios, in each trying to predict the overall 3/6 and 12 months of optical network traffic that consisted of each (SB, DB and P) type of bursts. First, let us discuss the results for the prediction of 3 months of traffic. The performance of GCN-GAN was superior in comparison to the LSTM, achieving 15 times smaller MSE. Then, we proceed with a comparison of six months of real traffic prediction. The trends showed that the results were slightly deteriorating. The main reason is that the forecasting length is much larger, and thus, there might be more specific events that affect traffic prediction. Finally, let us focus on the results for the one-year traffic prediction. As we can see, the GCN-GAN still achieved better results than the LSTM. In general, we can conclude that we can accurately predict up to 3 months of traffic, so network operators can use it to make short-term software and hardware configuration changes. On the other hand, we can use longer prediction to detect bottlenecks in the networks and plan future expansions in the longer horizon.

Table III: Long-Term Real Data Performance Evaluation Results - 3/6/12 months

Methods	Average MSE		
	3 months	6 months	12 months
LSTM	0.886	8.69	32.21
GCN-GAN	**0.057**	**2.81**	**12.51**

Let us now discuss those results in more detail by presenting the link states' adjacency matrices during the Short-Term Real

Data and Long-Term Real Data traffic scenarios, in Figures 3-5. We show the Ground Truth (optimal result - used as a reference) and result generated by GCN-GAN and LSTM. The axes refer to nodes, and the colours indicate the traffic intensity in a specific link between them - the closer the colour to the ground truth, the better the prediction.

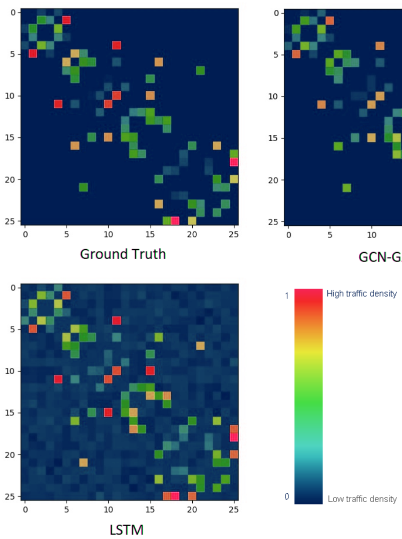


Figure 3: Short-term real traffic - link states prediction.

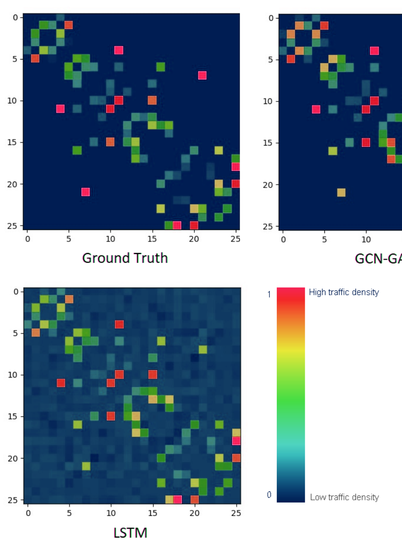


Figure 4: Long-term real traffic (3 months) - link states prediction.

First, let us discuss the performance of our solution in predicting short-term, real optical traffic (as shown in Figure 3), based on an average of all burst types. As we can see, the LSTM appears to have a 'noise' in the links. It indicates the inaccurate prediction of the link states between nodes. In this scenario, the GCN-GAN underestimated the network's traffic density (colours are closer to green/blue, which indicates lower traffic density). Next, we want to discuss the Long Term optical traffic prediction (3 months of traffic), presented in Figure 5. Again

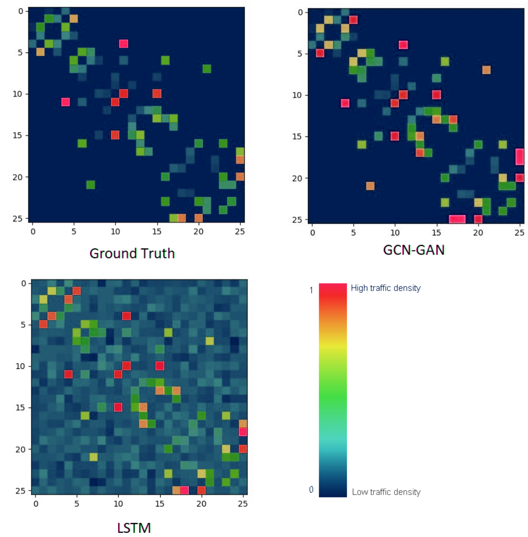


Figure 5: Long-term real traffic (12 months) - link states prediction.

the LSTM appears to overestimate the traffic density in the network, as well it has worse 'noise' in this scenario compared to short-term traffic prediction. The GCN-GAN model benefits from a longer "learning curve" and results in more accurate prediction, only overestimating the traffic in certain links. Finally, we present the adjacency matrices for the 12 months prediction in Figure 4. As we can notice, the GCN-GAN is still much better than the LSTM. The LSTM has much more noise in its traffic prediction. This is also related to the results presented in Tables I, II and III. This follows the fact that the LSTM has a lower learning rate that may allow the model to learn a globally optimal set of weights but may take significantly longer to train. While such learning might bring optimal results, flash events in the network cause abnormal information towards the LSTM, thus negatively affecting the performance. While LSTM can regain a very accurate prediction level, it takes time to readjust, resulting in short-time periods of deterioration of performance. We observe that the actual value ranges of the average traffic and the fluctuation through nodes or links increase rapidly with the longer predictions that need to be performed. One of the potential solutions to overcome this problem is to use the Forget Gates, as in [7].

To validate our solution's efficiency in network design and planning, we conducted a study on the existing Telus network to check how our approach can improve the routing. First, we focused on how much more bit-rate per second can be transmitted via the network, using our forecasting method. As we can notice in Figure 6, thanks to the GCN-GAN method's usage, we achieve a 10-15% better average transmission bit-rate per second. This is justified by the fact that GCN-GAN can preserve sufficient resources for predicted traffic in advance, reducing the number of rejected requests. This is especially visible in short-term forecasting. Once the forecast is more than a month, the accuracy of it starts to deteriorate. To further follow-up on this, we measured the BP in the network over the same periods, as seen in Figure 7. We measured the Blocking Percentage in the Metro Vancouver part of the Telus PureFibre

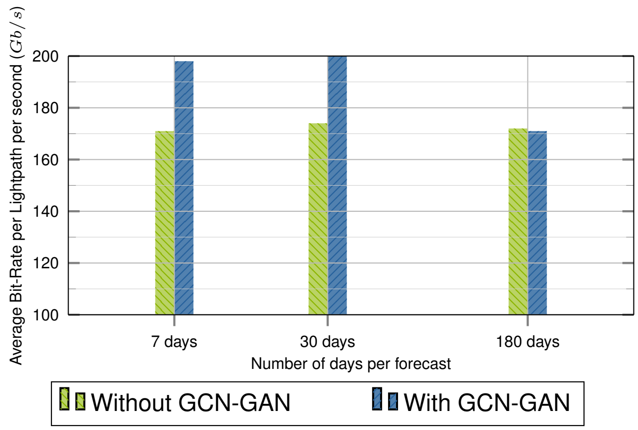


Figure 6: The comparison of the average transmitted bit-rate in a lightpath with and without forecasting.

network for this part of the observation. As we can see, with forecasting, we were able to reduce the BP to less than 1% and keep it within this limit for up to 30 days of forecast. Once the forecasting was longer than one month, the performance was worse. This confirms the previous findings from Tables I, II and III. To sum up, we should enable short-term traffic prediction in real-time networks, allowing BP to be reduced and an increased number of clients to be served in the network. The long-term traffic prediction allows us to detect future bottlenecks that will allow us to design and plan the network expansion in the future, but it is not needed in the network's everyday, real-time operations.

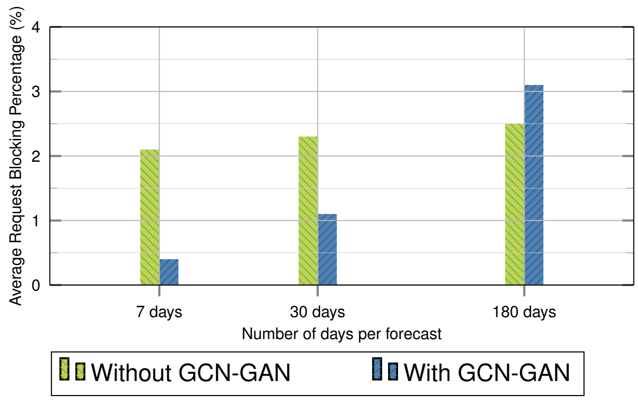


Figure 7: BP of requests with traffic prediction enabled/disabled.

VII. Conclusion

The GCN-GAN model can effectively solve the challenging optical traffic prediction problem by correctly representing edge weights' sparsity in each network snapshot. By allowing forecasting in the resource allocation process, we have accepted more traffic into the network and thereby reduced the network's operating expenditure (OPEX).

References

[1] Cisco, "Global Cloud Index: Forecast and Methodology, 2016–2021 (white paper)," CISCO, Tech. Rep., 2018.

[2] D. Klonidis, F. Cugini, O. Gerstel, M. Jinno, V. Lopez, E. Palkopoulou, M. Sekiya, D. Siracusa, G. Thouénon, and C. Betoule, "Spectrally and spatially flexible optical network planning and operations," *IEEE Communications Magazine*, 2015.

[3] B. Mukherjee, "WDM optical communication networks: progress and challenges," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1810–1824, 2000.

[4] A. D. Ellis, N. Mac Suibhne, D. Saad, and D. N. Payne, "Communication networks beyond the capacity crunch," 2016.

[5] J. Mata, I. de Miguel, R. J. Durán, N. Merayo, S. K. Singh, A. Jukan, and M. Chamania, "Artificial intelligence (AI) methods in optical networks: A comprehensive survey," *Optical Switching and Networking*, vol. 28, pp. 43–57, 4 2018.

[6] M. Chen, Y. Hao, H. Gharavi, and V. C. Leung, "Cognitive information measurements: A new perspective," *Information Sciences*, vol. 505, no. July, pp. 487–497, 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0020025519306528

[7] M. Aibin and K. Walkowiak, "Monte Carlo Tree Search with Last-Good-Reply Policy for Cognitive Optimization of Cloud-Ready Optical Networks," *Journal of Network and Systems Management*, 2020.

[8] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017.

[9] I. J. Goodfellow, J. Pouget-abadie, M. Mirza, B. Xu, and D. Warde-farley, "Generative Modeling Generative Modeling," *Annual Plant Reviews*, 2016.

[10] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative Adversarial Networks: An Overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 1 2018.

[11] Z. Zhong, N. Hua, M. Tornatore, J. Li, Y. Li, X. Zheng, and B. Mukherjee, "Provisioning Short-Term Traffic Fluctuations in Elastic Optical Networks," in *International Conference on Transparent Optical Networks*, 2019.

[12] C. Vinchoff, N. Chung, T. Gordon, L. Lyford, and M. Aibin, "Traffic Prediction in Optical Networks Using Graph Convolutional Generative Adversarial Networks," in *International Conference on Transparent Optical Networks*, 2020, pp. 3–6.

[13] M. Aibin, "Traffic prediction based on machine learning for elastic optical networks," *Optical Switching and Networking*, vol. 30, pp. 33–39, 11 2018.

[14] M. Aibin and M. Blazejewski, "Complex Elastic Optical Network Simulator (CEONS)," in *17th International Conference on Transparent Optical Networks (ICTON)*, Budapest, Hungary, 2015, pp. 1–4.

[15] Y. Zhang, S. Wang, B. Chen, and J. Cao, "GCGAN: Generative Adversarial Nets with Graph CNN for Network-Scale Traffic Prediction," in *Proceedings of the International Joint Conference on Neural Networks*, 2019.

[16] Y. Lin, X. Dai, L. Li, and F. Y. Wang, "Pattern Sensitive Prediction of Traffic Flow Based on Generative Adversarial Framework," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[17] K. Lei, M. Qin, B. Bai, G. Zhang, and M. Yang, "GCN-GAN: A Non-linear Temporal Link Prediction Model for Weighted Dynamic Networks," *Proceedings - IEEE INFOCOM*, vol. 2019-April, no. July, pp. 388–396, 2019.

[18] M. Aibin, "Dynamic Routing Algorithms for Cloud-Ready Elastic Optical Networks," Ph.D. dissertation, Wroclaw University of Science and Technology, 2017.

[19] A. Bhandari, A. L. Sangal, and K. Kumar, "Characterizing flash events and distributed denial-of-service attacks: an empirical investigation," *Security and Communication Networks*, 2016.