

Two Use Cases of Machine Learning for SDN-Enabled IP/Optical Networks: Traffic Matrix Prediction and Optical Path Performance Prediction [Invited]

Gagan Choudhury, David Lynch, Gaurav Thakur and Simon Tse

Abstract—We describe two applications of machine learning in the context of IP/Optical networks. The first one allows agile management of resources at a core IP/Optical network by using machine learning for short-term and long-term prediction of traffic flows and joint global optimization of IP and optical layers using colorless/directionless (CD) flexible ROADMs. Multilayer coordination allows for significant cost savings, flexible new services to meet dynamic capacity needs, and improved robustness by being able to proactively adapt to new traffic patterns and network conditions. The second application is important as we migrate our metro networks to Open ROADM networks, to allow physical routing without the need for detailed knowledge of optical parameters. We discuss a proof-of-concept study, where detailed performance data for wavelengths on a current flexible ROADM network is used for machine learning to predict the optical performance of each wavelength. Both applications can be efficiently implemented by using a SDN (Software Defined Network) controller.

Index Terms — Machine Learning; Traffic Matrix Prediction; Multi-Layer Optimization; Routing; Open ROADMs; Optical Transport Network; SDN.

I. INTRODUCTION

There is great recent interest in applying machine learning techniques in the networking context. See [1] for a recent survey. In this paper, we provide two initial applications of machine learning to more efficiently manage our IP/Optical networks in conjunction with a SDN controller.

First Application – Predicting Network Traffic Matrix: The traffic management of a core IP/Optical backbone of a large Internet Service Provider (ISP) has to deal with dynamic traffic changes under various network conditions including scheduled and unscheduled outages, and makes efficient use of network resources while also

satisfying the loss and latency requirements of each class of traffic type it carries [2, 4]. It also needs to be flexible enough to provide new services demanding dynamic capacity [2, 17]. The IP layer of the network consists of IP links connected among IP devices such as router ports or white-box switch ports. The IP links are routed over a path in the optical layer using flexible ROADMs (Reconfigurable Optical Add/Drop Multiplexers), transponders at endpoints, and optical signal regenerators along the path when it is too long. Improving efficiency means reducing the totality of IP resources (IP ports) and optical resources (ROADMs, transponders and regenerators). If there are N traffic endpoints and K Quality of Service (QoS) classes, the totality of traffic flows can be specified by a traffic matrix of $KN(N-1)$ elements and each such element represents the traffic from a specific source to a specific destination and belonging to a specific QoS class. The elements of the traffic matrix are usually highly correlated and their variability over time may be characterized by complex, nonlinear oscillations and seasonal periodicities at different time scales. We use machine learning for accurate short-term and long-term prediction of all elements of the traffic matrix and combine that with joint global optimization of IP and optical layers [3] using flexible Colorless/Directionless (CD) ROADMs [14]. This results in significant cost savings, flexible new services to meet dynamic capacity needs with better accuracy, and increased robustness by being able to proactively adapt to new traffic patterns and network conditions. The methodology can be efficiently implemented by using a combined Packet and Optical layer or Multi-Layer SDN (Software Defined Network) controller [4, 13].

Second Application – Predicting Optical Path Performance in a Multi-Vendor Network: Large ISPs typically operate single-vendor Layer 0 ROADM networks for optical transport. Before provisioning new wavelengths, the ISP should verify that the proposed physical routes meet optical performance standards. Usually, this evaluation is conducted using closed vendor-proprietary tools which

incorporate detailed analysis of the various vendor specific optical components. We started, initially in metro areas, an Open ROADM network architecture initiative [5-7] where the ROADMs and other optical plug-ins will be model-driven with open standard interfaces, thus allowing interoperability among different vendor equipment. The introduction of Open ROADM and the SDN controller technologies will allow ISPs to more effectively and uniformly leverage network performance data to set up optimal wavelength paths that meet optical performance standards. Because Open ROADM will integrate equipment from multiple vendors, single-vendor performance evaluation tools will no longer be suitable for evaluating new wavelength paths. Instead, we propose a new machine learning model which will use network data to predict optical performance of new wavelengths in a multi-vendor environment. We describe a proof-of-concept study, where we collect detailed information for wavelengths on a current flexible ROADM network, and then use machine learning to predict the optical performance of each wavelength, specifically the bit error rate. The machine learning model is able to predict the bit error rate with a mean squared error value of less than 1.0. It can be incorporated into the Path Compute Engine (PCE) within the SDN Controller to verify that all new Open ROADM wavelengths meet optical performance standards. The model can also monitor the performance of existing wavelengths and proactively move and/or groom them to better paths as conditions evolve. It should be noted that as Open ROADM technology matures, the same methodology can be extended to core long-haul ROADM network.

II. MACHINE LEARNING FOR TRAFFIC MATRIX PREDICTION AT A CORE IP/OPTICAL NETWORK

A. Framework for Closed Loop Optimization using Machine Learning

Figure 1 depicts the framework for self-optimizing an IP/Optical network in a closed loop manner where future traffic prediction from machine learning, real-time network and traffic measurements, and knowledge based feedback on traffic changes and failures will collectively drive a joint global optimization engine for both the packet and optical layers. A multi-layer SDN controller collects long-term and short-term traffic and failure data to facilitate these three types of feedback, implements the global optimization algorithms, and pushes the required changes to the packet and/or optical layers of the network. The framework significantly reduces the network cost, improves robustness and facilitates offering new services that require accurate short-term traffic projections. Optimization needs to be done in at least two different time scales. In the short time scale (seconds, minutes and hours) the available network resources are fixed and we have to use them optimally. In this setting, network changes (traffic matrix and network failures) can be detected and the network is re-configured (either at the packet or at the optical level) in a reactive mode based on real-time feedback. Short-term traffic prediction based on machine learning allows us to respond to these changes in a way that is much more resource-efficient and less disruptive to network operators. In the longer term (days, weeks and

months), we need to do a network design exercise including simulation of many potential traffic change and failure scenarios to determine just the optimal level of resources to ensure that the network, with the agility of SDN control, can flexibly cope with all possible traffic change and failure scenarios. Here long-term traffic prediction based on machine learning will play a key role.

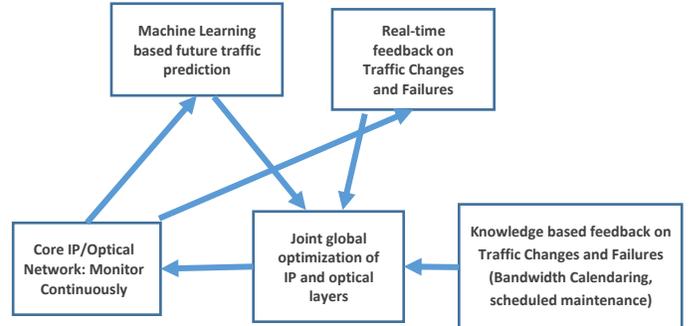


Fig. 1. Closed Loop Optimization with Machine Learning and other Feedbacks.

B. Routing of traffic over Packet/Optical Network

Figure 2 illustrates an example of an integrated IP/Optical network and its interaction with the SDN controller. E_i represents the IP edge routers, B_i represents IP core or backbone locations and O_i represents optical nodes (ROADMs). A subset of the optical nodes is collocated with an IP core location. We show two core routers, A and B, per core location but in general the number can be variable. All unicast traffic originates/terminates at the edge routers and each such router is connected to at least two core routers (same or different locations) using physically diverse paths. In addition, there may also be point-to-multipoint multicast traffic and all the endpoints of such traffic are also edge routers (it is of course possible for the same router to perform both edge and core functions).

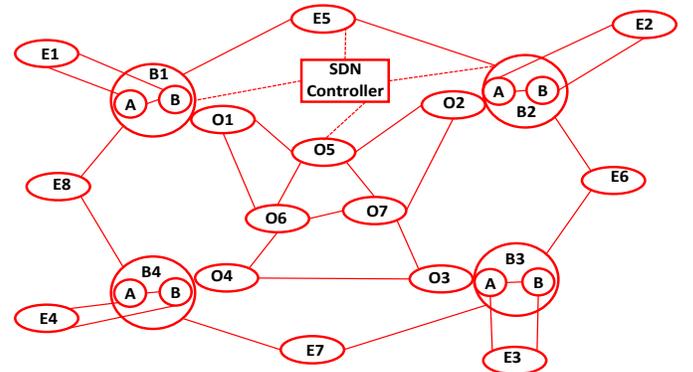


Fig. 2. Example IP/Optical Network with Backbone (B) and Edge (E) locations.

A subset of all possible pairs of IP routers is connected via IP links to form an IP network. An IP link between two different core locations needs to be routed over the optical network. As an example, the IP router A in location B2 may be connected to the IP link B in location B4 over the sequence of optical nodes O2-O7-O6-O4. The SDN controller can control the edge routers, the core routers and the optical

nodes. The SDN controller is logically shown as a single centralized entity but it may be functionally separated into one controlling the IP network and one controlling the optical network. Furthermore, for the purpose of reliability and disaster recovery, it makes sense to have one active SDN controller and one or more standby SDN controllers located geographically in different places.

Figure 3 explains various levels of routing in the network. The IP links are routed over the ROADM layer and the MPLS TE tunnels carrying end-to-end traffic are routed over the IP layer.

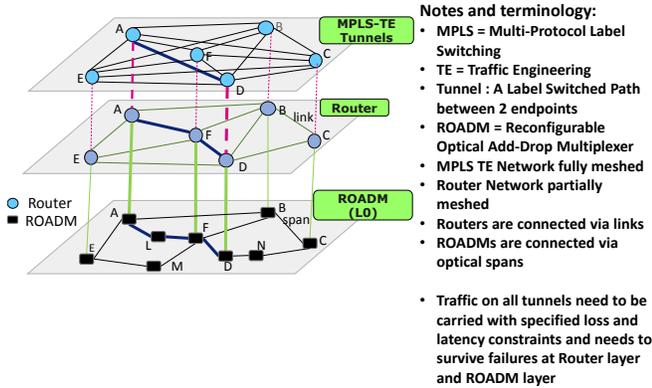


Fig. 3. Various Levels of Routing.

C. Flexibility of Resource Management with CD ROADMs and Digital Fiber Cross Connect (DFCC) devices

Figure 4 shows an end-to-end routing of an IP link over the optical layer. R1 and R2 represents router ports in two different geographical locations. T1 is a ROADM transponder port connecting to R1 and T2 is a ROADM transponder port connecting to R2 (T1 and R1 are typically collocated and so are T2 and R2). RE1 is a regenerator (needed only if the route-miles from T1 to T2 is beyond the optical reach distance from T1 to T2). The connected combination of a router port and ROADM transponder in the same location is called a Tail and we have two tails in this illustration: Tail1 and Tail2.

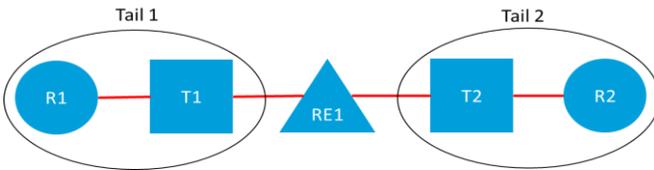


Fig. 4. End-to-end Routing of IP Link over Optical Layer.

Traditionally, if any component along the path of the IP link fails (or there is a fiber-cut), the entire IP link fails and no non-failed component can be reused. However, with SDN controller managing both the packet and the flexible CD ROADM networks [14], the three components, namely Tail1, Tail2 and RE1 are disaggregated, interoperable and the non-failed components can be reused by the controller. Furthermore, if a Dynamic Fiber Cross-connect (DFCC) device [15] is used in connecting the two components of the

Tail then it is also disaggregated and one of its components can be reused to form a new Tail which in turn can be used to create a new IP link. The real-time SDN controller can leverage this resource disaggregation capability to provide numerous resource reuse/sharing opportunities to proactively overcome traffic fluctuations and network failures.

D. Machine Learning-Based Future Traffic Prediction

If there are N traffic endpoints and K QoS classes then there are $T = KN(N - 1)$ elements in the traffic matrix. As an example, if $K = 2$ and $N = 50$ then $T = 4900$. We assume that each element of the traffic matrix is routed over the packet network as a TE (Traffic Engineering) tunnel. In general, for routing flexibility, each TE tunnel may be split into multiple ones but we ignore it here for simplicity and illustration purpose. Typically, the TE tunnel traffic at a large ISP network is characterized by complex, nonlinear oscillations and seasonal periodicities at different time scales, reflecting customer usage of the network. The traffic on the highest activity tunnels contains a strong daily oscillation, a less prominent weekly oscillation (reflecting different usage patterns on weekends), along with occasional, sharp jumps that correspond to the network dynamically shifting traffic between tunnels. An example of the total traffic and the traffic on a particular TE tunnel is shown in generic bandwidth units in Figure 5. Our goal is to develop a machine learning based, real-time prediction of the traffic load for each of the TE tunnels at future time horizons of minutes, hours, days and weeks, although we primarily concentrate on hours or higher time scales.

We denote a given TE tunnel's traffic by $\{x_0, x_1, \dots, x_N\}$. For each TE tunnel, the goal is to form a statistical model that forecasts the traffic on that TE tunnel for a given forward time horizon a_T , such as the next hour or next 24 hours. We use a nonlinear autoregressive-like model of the form $x_t = f(x_{t-a_1}, x_{t-a_2}, \dots, x_{t-a_T}) + b + ct + \epsilon_t$, where $\{a_1, \dots, a_T\}$ are pre-specified time lags, $b + ct$ is a linear trend and ϵ_t is Gaussian white noise. We estimate the mapping f by applying Gaussian process regression (GPR), a Bayesian nonlinear regression model where the number of parameters estimated grows with the amount of data. GPR, also known as Kriging, models f as a realization of a Gaussian process with a covariance kernel function formed from the observed data. The posterior estimate $E(x_t|f)$ under the model has an explicit formula in terms of the training data, and can be used to make out-of-sample predictions. GPR is used extensively in different fields and has been found to perform well in situations with limited data available, although the standard form of the algorithm can become computationally intensive as the data size grows. GPR can also be viewed a probabilistic formulation of kernel regression and provides Bayesian credible intervals (error bars) on any forecasts, which are helpful in interpreting the results. In comparison to a classical, linear autoregressive (AR) or moving average (MA) model, this type of model is better able to capture the asymmetry between the rising and falling parts of the daily oscillation, as seen in Figure 5. More details of GPR can be found in [11].

We apply GPR by first de-trending the time series with a

linear regression, and then regressing $\{x_{t-a_1}, x_{t-a_2}, \dots, x_{t-a_T}\}$ on x_t for all t such that both t and $t - a_T$ lie within the training period. In the machine learning literature, GPR is typically applied to time series in a different manner than this, by regressing t on x_t for all times t in the training dataset. However, this approach requires more detailed prior knowledge of the data to specify a good kernel function (a key part of the GPR model), and also lacks a direction of time or notion of causality in the model. In practice, we found that it performed worse than the lagged approach described above. We also applied several other regression models that are standard in the machine learning literature, including penalized linear models, boosted decision trees and random forests (see [9] for details), but GPR was found to have better out-of-sample prediction accuracy than these other methods. This is likely explained by the fact that our training data size is limited but has a relatively high signal-to-noise ratio and the mapping f is stationary over time, which is well suited for GPR.

In practice, the choice of lags $\{a_1, \dots, a_M\}$ has a large impact on the model's accuracy. Specifying too few lags fails to capture longer term dependencies in the model, while having too many lags results in a large parameter space where f cannot be estimated efficiently. It is known that GPR generally becomes less accurate for data with a large number of features. To choose the lags, we apply a heuristic based on the partial autocorrelation ρ_t of the data. The partial autocorrelation has been widely studied in classical time series analysis and is used in the Box-Jenkins methodology [8] to find the number of lags in a classical AR model. It is defined by $\rho_t = \text{corr}(x_0, x_t | \{x_1, x_2, \dots, x_{t-1}\})$, and represents the amount of extra correlation at time lag t after accounting for the correlation at all smaller lags. An example of ρ_t for the tunnel data is shown in Figure 6. We compute ρ_t over the training period, and choose only those lags where $\rho_t > a_T^{1/6}/15$. The intuition behind this choice is that for small a_T (say, one hour ahead), the mapping f is easy to estimate and we can learn a higher dimensional model with more lags, while for large a_T (one week ahead), the mapping f is much noisier, and we estimate a lower dimensional model to compensate for it.

We use data collected from a large ISP network over 2017, with the data up to 11:00 PM, July 31 used for training the model, and the data from (12:00 AM, August 1) + (a_T hours) onward used for testing the model's performance, where a_T is the desired forecast horizon. The gap between the training and testing periods ensures that there is no overlap in the data used to train the model and the data used to test it. This is done for each tunnel separately, using its own past history, and for a range of different a_T from 1 hour to 168 hours (one week). The choice of lags is determined separately for each tunnel and value of a_T . The tunnels with the highest activity have quite different characteristics than the ones with lower activity, and typically result in a different choice of lags. We use the scikit-learn implementation of GPR [10] with a squared-exponential kernel in GPR (a standard choice; see [11]), with a bandwidth parameter $\theta = 0.01$ and a noise power $\text{var}(\epsilon_t) = 0.01$. For a given model, we measured the error using the relative median absolute error (MAE) on the test period of the data, as well as the relative MAE over only the

period of peak activity in the network, 1:00 AM to 5:00 AM GMT, which is important for capacity planning purposes. The MAE is a more appropriate metric than the standard mean-squared error since it is less sensitive to error contributions from short impulses or bad data points. We train the models for each TE tunnel from May to July and test them over August. The error metrics are 1.61% and 1.12% for the total traffic. For individual TE tunnels, they are 2-10% for the high activity ones and 5-30% for lower activity ones (where the traffic often consists of random impulses that are not predictable). An example of the forecasted total traffic for several different a_T is shown in Figure 7, with the different models combined to form a forecasted trajectory over four days.

Several extensions and improvements of this forecast model are possible. The model can be extended to account for the dependencies between multiple tunnels, where the current value depends not only on the same TE tunnel's past values but also the past values of all other TE tunnels. Such a model would have a very large parameter space and would need additional penalization or model selection to work well in practice. Another generalization is an online version of this model, where the model is retrained and updated on every sample. We also train separate models at each forecast horizon, but these models are interlinked and it may be more efficient to train a joint model that accounts for dependencies between them. We did not pursue these extensions in this paper, since the model described here already gives a good fit and runs quickly enough for our purpose.

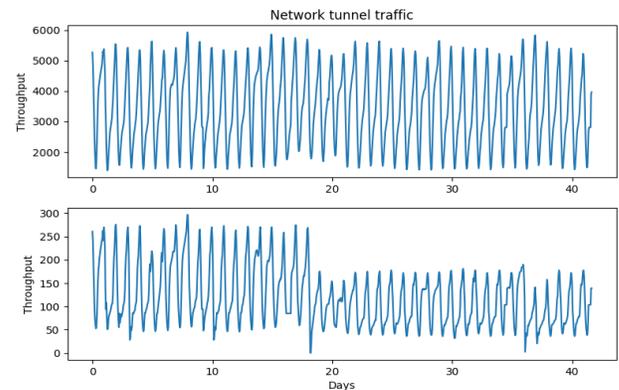


Fig. 5. The tunnel traffic across the entire network (top) and an individual TE tunnel (bottom) using generic bandwidth units.

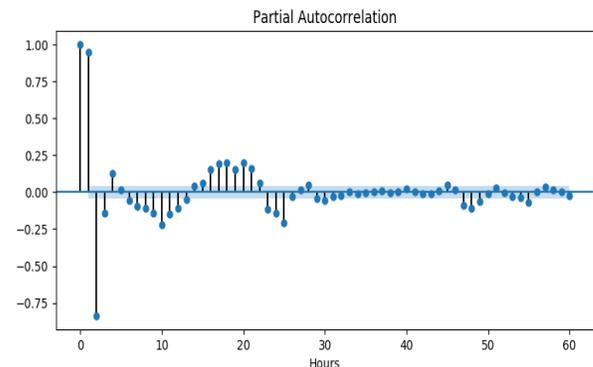


Fig. 6. Partial autocorrelation of total network traffic over three months. The shaded region is a 95% confidence interval.

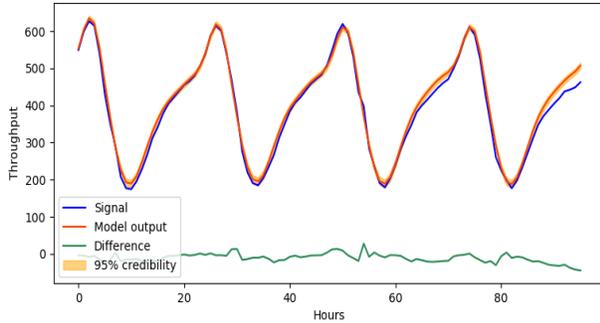


Fig. 7. Comparison of the total traffic (blue) and a 4-day forecast formed by multiple models (red) trained over the previous three months with $\mathbf{a}_T \in \{1, 2, \dots, 96\}$. The relative MAE is 1.61% and the PMAE is 1.12%. Generic BW units used.

E. Using Machine Learning and other Feedback to Optimize at Various Time Scales

SDN controller implementation brings along real-time network data and the capability of data-driven analytics for proactive closed-loop network management. Further, machine learning techniques can be applied in different time scales depending on the network states and scenarios.

Sub-Second Time Scale: When an abrupt network failure occurs, the SDN controller typically cannot make any traffic routing changes fast enough except for relying on a Fast Reroute (FRR) mechanism to temporarily bypass a failed path to a pre-computed backup path [12] within an order of tens of milliseconds of detecting a failure. In a pre-SDN environment, a back-up tunnel or FRR bypass path is typically reserved for every link bundle in the network. During static network planning and design phase, it is important to select a shortest possible FRR bypass path that has enough capacity. In SDN environment, one can use machine learning based timely traffic prediction to periodically re-adjust and optimize these FRR bypass paths.

Seconds-to-minutes Time Scale: We have demonstrated recently [13] that SDN controller can retrieve real-time network data, make and execute optimal layer 3 TE tunnel changes over fixed IP links in a sub-minute interval. Real-time machine learning can make the changes proactively and is therefore less disruptive to customers.

Minutes-to-hours Time Scale: In addition to being able to reroute TE tunnels over fixed IP links, we can also use the flexibility of CD ROADMs and Dynamic Fiber Cross Connect (DFCC) to create new IP links, delete or reroute an existing IP link based on changing network and traffic conditions. Machine learning predictions allow us to do this in a proactive manner rather than in a reactive manner based on real-time feedback.

Days, weeks and months Time Scale: The introduction of a multi-layer SDN controller fundamentally changes the network capacity augmentation process from relatively disjointed L0 and L3 capacity planning to an integrated multi-layer planning; and from a single long-term planning horizon to include a much shorter planning timescale. Here the main need is to ascertain how much extra resources are needed within days in addition to weeks and months of

ordering interval. The resources include router ports, ROADM transponders and Regenerators. We need to simulate many failure scenarios in an integrated multi-layer fashion based on future traffic predictions with a timescale of days and weeks. Machine learning plays a critical role here by accurately predicting the entire Traffic Matrix in time scales of days and weeks.

Optimization Methodology: For a fixed capacity of IP links, the most efficient routing of TE tunnels can usually be achieved by multi-commodity flow [16] which requires arbitrary splitting of TE tunnels. Since we have a practical constraint on how TE tunnels can be split and different latency constraints for different traffic types, an alternative method is to use Constrained Shortest Path (CSPF) routing. Careful orchestration of the order of tunnel routing can achieve almost the same efficiency as multi-commodity flow but runs much faster. We have also developed efficient heuristics to optimize over many different failure scenarios and joint global optimization of optical and IP layers using the flexibility of CD ROADMs and DFCC [3].

F. Comparison of traditional design and operation of IP/Optical Networks with that based on Machine Learning and Joint Multilayer Optimization

Improved Efficiency: Table I shows improved efficiency and cost reduction with machine learning based traffic prediction and joint multilayer optimization at a large ISP network. All numbers shown are generic normalized values and are only to be used to compare among the different scenarios.

Table I. Normalized View of Efficiency Gains with Machine Learning and Optimization

Scenario	# of 100 GE Tails	# of 100 GE Regens	Cost
1. No Machine Learning, IP Layer Optimization Only, fixed IP to Optical mapping	1,000	100	1,040
2. Addition of Machine Learning at long time-scale	910	90	946 (-9%)
3. Addition of Joint Multilayer Optimization with a Fixed IP Layer Topology	810	80	842 (-19%)
4. Addition of Dynamically Changing IP Layer Topology as traffic changes	640	110	684 (-34%)

We have two types of resources, Tails and Regens. The cost is given in units of 100 GE Tails and for the purpose of illustration, it is assumed that the cost of a 100 GE Regen is 40% that of a 100 GE Tail. We see that if we just do Machine Learning (ML) providing more accurate traffic prediction over days and weeks of time-frame but no multi-layer optimization, we have about 9% saving. Next if we combine ML with joint multi-layer optimization where the mapping between IP links and optical resources may be readjusted but with the same set of IP links (i.e., fixed IP layer topology), we get about 19% saving. Finally if we combine ML with full multi-layer global optimization (mapping of IP links to optical resources can change and IP

links themselves can change as network condition changes), we get about 34% saving.

Less disruption to customer traffic with proactive Machine Learning Based Approach: We considered a tight and highly optimized network design of a large ISP network, simulated failures and traffic surges and made the following observations:

- If the same static Fast Reroute (FRR) Backup path is used irrespective of traffic changes, one can find some traffic losses when a failure happens near the peak traffic period as the pre-defined FRR backup path may not have sufficient capacity. It was possible to avoid these losses by proactively changing the FRR backup paths based on traffic changes predicted by machine learning. Alternatively, if our goal is to always avoid traffic loss then the static FRR paths would be more expensive compared to dynamically changing FRR paths based on future traffic changes predicted by ML.
- With ROADM network controller, one can add and/or re-arrange a wavelength much faster than traditional manual and static methods but it still takes about 2-3 minutes to complete. Therefore, if one tries to change IP layer topology during a peak-traffic period based on reactive real-time-based traffic observation, one will experience some traffic loss. Using machine learning prediction, this traffic loss could be avoided by making the IP layer topology changes about 20 minutes before the traffic surge. Again, if we want to avoid traffic loss but with static reactive method, we will need more resources, thereby increasing cost.

Ability to offer more efficient Bandwidth Calendaring Service: Due to temporal variation and asymmetry of traffic matrix, there is usually significant amount of spare capacity left in the network that can be used for offering a flexible service with temporary capacity need such as a bandwidth calendaring service [2, 17]. As this type of service may be offered in a matter of hours and days that is much shorter than any capacity planning cycle, the knowledge of near-term traffic pattern through machine learning can significantly improve the feasibility and efficiency of offering such service.

III. USING MACHINE LEARNING TO PREDICT OPEN ROADM OPTICAL PATH PERFORMANCE

A. Problem Description

Figure 8 shows a typical wavelength connection over a ROADM (Reconfigurable Optical Add/Drop Multiplexer) network. ROADMs support Layer 1 services, such as private lines, and provide transport for higher layer services. Each ROADM is connected to one or more other ROADMs with one or more pairs of fibers. A Layer 1 wavelength can be set up between two transponders. Each transponder is connected to a nearby ROADM, and the wavelength is then routed through the ROADM network.

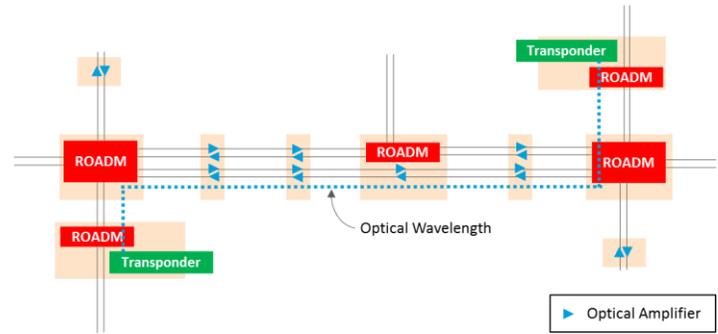


Fig. 8. Portion of a Sample ROADM Network.

Various factors can affect the quality of the optical signal. Imperfections in the fiber can add noise, and there will be signal distortions as the signal passes through equipment such as ROADMs and amplifiers and over distance. Thus, it is important to verify that a new wavelength will meet performance standards before putting it into service. Because Open ROADM networks can include equipment from multiple vendors, we cannot use a proprietary single-vendor tool to analyze new wavelength paths. Instead, we propose a machine learning model to predict optical performance.

B. Model Features

In order to construct a machine learning model, we compile all available data for every optical wavelength in an existing ROADM network. We then distill this data into a set of input features for each wavelength, or data sample. These features include fiber type, frequency, length of path, margin, measured fiber loss, measurement date, number of amplifiers in the path, number of pass-through ROADMs, ORL (Optical Return Loss), OSNR (Optical Signal to Noise Ratio), PMD (Polarization Mode Dispersion), and speed.

As a measure of service quality, we wish to predict the Pre-FEC (forward error correction) Bit Error Rate for each wavelength in each direction. Since the BER values span several orders of magnitude, we use $\log_{10}(\text{BER})$ as the quantity to be estimated.

C. Machine Learning Analysis

We apply a variety of machine learning algorithms and compare their performance, but the dataset is smaller than typical machine learning application, and the quality and quantity of the data turns out to be more of a limiting factor than the statistical methods we use. We focus on penalized and ensemble regressions, which are well suited for small-scale data such as this. We do not consider more sophisticated models like deep neural networks, as they are likely to overfit due to the small sample size and diverse types of features in the data. We use scikit-learn [10], a free open-source Python library that contains industry-standard implementations of these and many other machine learning models.

The machine learning methods we consider broadly fall into three categories: penalized linear regressions, nonlinear regressions and ensembles of regression trees. We denote the set of features by X and the output values ($\log_{10}(\text{BER})$) by y .

In the first category, we consider ridge regression and LASSO. These are traditional models that are easy to interpret and serve as a baseline for the other methods. They

both estimate coefficients β in a model of the type $\min_{\beta} \|y - \beta X\|_2^2 + a\|\beta\|_p^p$, with $p = 2$ (ridge) or $p = 1$ (LASSO). The latter encourages sparsity in β , as would be expected if most features have no impact on the BER, while the former reduces instability in estimating β when the features are highly correlated.

Among nonlinear regression models, we look at the performance of quadratic LASSO, Gaussian process regression and the multilayer perceptron. Quadratic LASSO is a simple variant of LASSO using features of the form $x_i x_j$ for every pair (i, j) , giving a total of 676 features. Gaussian process regression was described in Section D, while the multilayer perceptron is a classical, fully-connected neural network.

Among ensemble models, we apply gradient boosted regression trees and random forests. A regression tree is a piecewise linear regression that iteratively splits the data according to an error criterion and fits separate regressions to each portion of the split data. Both ensemble methods train several different regression trees over different subsets of the features, and take an average over all of the trees to obtain a final estimate.

More details of all of these algorithms can be found in [9].

D. Model Performance

For each machine learning model, we consider 50 random splits of the data, each with 2/3 of the data used for training the model and 1/3 used for testing the model. Various hyper-parameters of each model (e.g., the penalty factor in ridge and LASSO regressions) are optimized by choosing random values on each split and taking the best one. To measure the performance of an algorithm, we compute the mean squared error (MSE) across all points, the MSE for only the points with high BERs above some threshold (which we denote HMSE), and the MSE across the points with the 10% worst errors (denoted WMSE), averaged over all 50 splits of the dataset. In practice, we want the model to have good ballpark estimates of BER (not necessarily very precise ones) and are especially interested in the measurements with higher BERs, so the HMSE and WMSE are more important than the MSE.

Table II shows the performance of the different machine learning models according to these error criteria.

Table II. Performance of Different Machine Learning Models

Model	MSE	HMSE	WMSE
Ridge regression	1.06	3.32	5.80
LASSO regression	1.15	3.63	6.25
LASSO with quadratic features	0.83	2.30	5.19
Multilayer perceptron	0.94	2.91	6.12
Gaussian process regression	0.90	1.87	5.90
Gradient boosted regression trees	0.81	2.08	5.18
Random forest regression trees	0.81	1.86	5.14

In general, the ridge and LASSO regressions both perform well on MSE, but less so on the other two metrics. The quadratic features yield some improvement especially on HMSE, and the Gaussian process regression improves the results further. The random forest consistently achieves the best overall error rates, at the cost of a higher model

complexity and less interpretability. We also found that standardizing each feature before applying the model, a common data preprocessing technique, does not improve the performance due to the diverse mix of continuous and discrete features. For the random forest, the predicted and actual BER across one of the training/testing splits is shown in Figure 9.

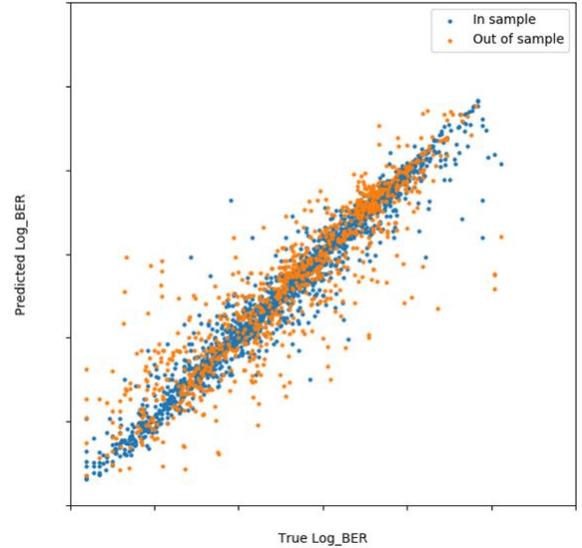


Fig. 9. Machine Learning Model Performance.

E. Importance of Features

In a random forest model, the importance of a given feature can be measured by randomly permuting values of the feature and measuring how much the regression error increases. This is used to form a score for each feature known as the Gini importance (see [9] for details). We apply this methodology here, with the importance scores averaged over all 50 models and splits of the data and normalized on a scale from 0 to 100, with 100 indicating the most important feature.

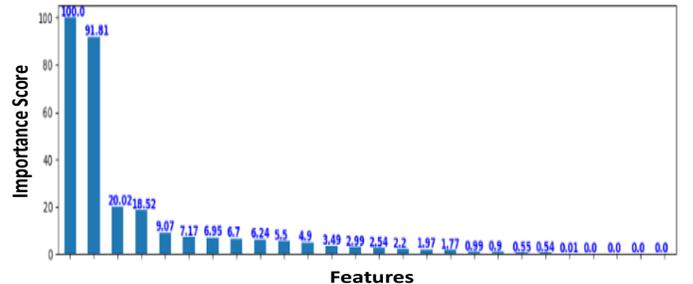


Fig. 10. Illustration of Relative Importance of Machine Learning Features.

As seen in Figure 10, the top two features are significantly more important than the others, which have little effect on the BER. We train the random forest with only the top 10 features (importance score over 2.00) and obtain MSEs of 0.86/1.87/5.40, which are not far off from the errors in Section 3.4.

Figures 11 and 12 show the BER as a function of the speed,

miles and total loss, as well as the speed, OSNR, and slot number. These plots demonstrate the complex, nonlinear structure of the data and indicate why the random forest is able to outperform a simple linear regression model.

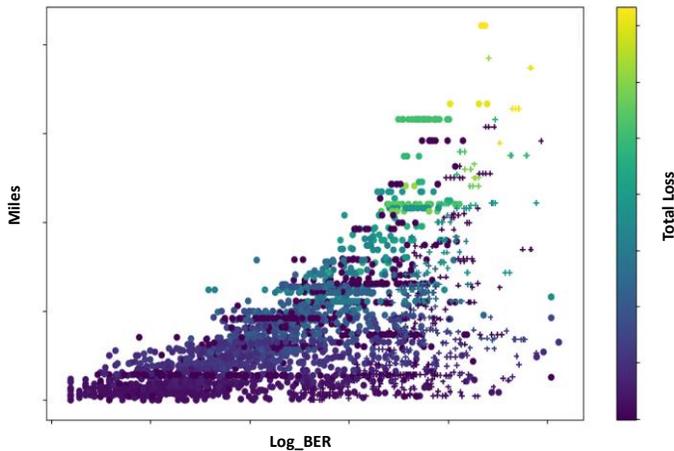


Fig. 11. Effect of Speed, Miles and Loss on BER.

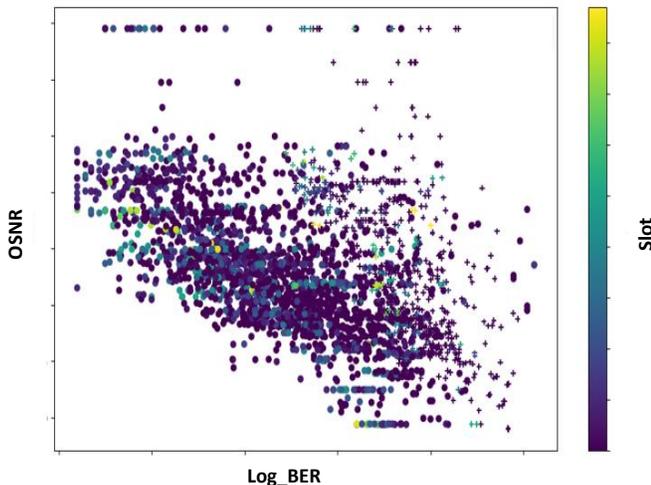


Fig. 12. Effect of Speed, OSNR and Slot Number on BER.

F. Application

The model can be incorporated as a feature of the Path Compute Engine (PCE) within the SDN Controller to verify that all new Open ROADM wavelengths meet optical performance standards. The PCE can generate a proposed path for each new wavelength request, and then invoke the Machine Learning model to predict the optical performance of the proposed path. If the path meets optical performance standards, then the wavelength can be deployed into the Open ROADM network. Otherwise, the PCE can generate an alternative path and try again. Periodically, the SDN Controller can use the latest network data to retrain the Machine Learning model and then update the model parameters.

G. Observations

The proof of concept study demonstrates that it is possible to create a machine learning model to predict the optical

performance of ROADM wavelengths, specifically pre-FEC bit error rate, with reasonably good accuracy without knowing many of the details of the optical line or fiber. In particular, the model is able to do this with fewer features and far less data than typical machine learning applications.

The next step will be to extend the machine learning model to predict optical performance of wavelengths in the new Open ROADM network. This model can be implemented as a microservice as part of the Path Compute Engine within the SDN Controller. The Open ROADM version of the model may have slightly different features due to differences in the new network, but the general approach should be similar. While the data supporting this study comes from a single vendor network, other vendor equipment should provide similar data with similar interpretation and thus the methodology can be applied in a multi-vendor environment. In addition, as Open ROADMs are model-driven, performance data from other optical plug-ins can be included, if needed, to further enhance the model and predictability.

In addition to planning paths for new wavelengths, the SDN Controller can also use the Machine Learning model to monitor the optical performance of existing wavelengths and move them to better paths as conditions evolve.

IV. SUMMARY

We have described two applications of machine learning for managing IP and Optical networks. The first application allows significant cost saving by combining machine-learning-based long-term traffic prediction with global optimization of IP/Optical layers using CD ROADMs and DFCC devices. It also uses machine-learning-based short-term traffic prediction to allow proactive network changes to reduce customer traffic disruptions and opens an opportunity to offer flexible services based on dynamic capacity needs. The second application enables the selection of improved metro ROADM paths based on the latest optical performance data. Both applications can be efficiently implemented using an SDN controller.

These methodologies can be extended to different network settings depending on technology evolution, network data availability, and maturity of machine learning.

ACKNOWLEDGMENT

The authors would like to thank their colleagues, Martin Birk and Kathy Tse, for providing their expert knowledge in Open ROADM and Optical Transport Network as well as suggesting the optical path performance use case. They would also like to thank Kathy Meier-Hellstern for carefully reviewing the paper and providing many valuable comments.

REFERENCES

- [1] Z. Fadlullah et al., "State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow's Intelligent Network Traffic Control Systems," *IEEE Commun. Surveys & Tutorials*, 2017.
- [2] J. Donovan and K. Prabhu, "Building the Network of the Future", CRC Press, First Edition, ISBN 1138631523, September 2017.
- [3] G. Choudhury, M. Birk, B. Cortez, A. Goddard, N. Padi, K. Meier-Hellstern, J. Paggi, A. Raghuram, K. Tse, S. Tse and A.

- Wallace, "Software Defined Networks to greatly improve the efficiency and flexibility of Packet IP and Optical Networks", ICNC 2017, January 26-29, 2017, Silicon Valley, CA, USA.
- [4] M. Birk, G. Choudhury, B. Cortez, A. Goddard, N. Padi, A. Raghuram, K. Tse, S. Tse, A. Wallace and K. Xi, "Evolving to an SDN-Enabled ISP Backbone: Key Technologies and Applications", in IEEE Communications Journal, October 2016.
 - [5] Open ROADM MSA, <http://openroadm.org>
 - [6] TransportPCE, <https://wiki.opendaylight.org/view/TransportPCE:Main>
 - [7] M. Birk, "AT&T's Direction Towards a Whitebox ROADM", Open Network Summit 2015, Session: SDN for Service Provider Networks: Transport SDN, June 17th 2015, Santa Clara, CA.
 - [8] J. D. Hamilton, "Time Series Analysis", Princeton University Press, Princeton, NJ, 1994
 - [9] T. Hastie, R. Tibshirani, J. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition", Springer, New York, NY, 2009
 - [10] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," The Journal of Machine Learning Research, 12:2825–2830, 2011
 - [11] C. E. Rasmussen, C. K. I. Williams, "Gaussian Processes for Machine Learning," MIT Press, Cambridge, MA, 2005
 - [12] P. Pan et al., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels," IETF RFC 4090, May, 2005.
 - [13] S. Tse and G. Choudhury, "Real-Time Traffic Management in AT&T's SDN-Enabled Core IP/Optical Network," Invited presentation at OFC 2018, March 11-15, San Diego, CA, USA.
 - [14] Y. Li et al., "Impact of ROADM colorless, directionless, and contentionless (CDC) features on optical network performance," Invited paper, IEEE/OSA Journal of Optical Communications and Networking (Volume: 4, Issue: 11, Nov. 2012).
 - [15] G. Zervas et al., "Multi-Granular Optical Cross-Connect: Design, Analysis and Demonstration," IEEE Journal on Selected Areas in Communications, vol. 27, no. 3, April 2009.
 - [16] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, "Introduction to Algorithms (3rd ed.)". MIT Press and McGraw–Hill, 2009.
 - [17] L. Gkatzikis et al. "Bandwidth calendaring: Dynamic services scheduling over Software Defined Networks," IEEE ICC 2016.

BIOGRAPHIES

Gagan Choudhury (gchoudhury@att.com) is a lead inventive scientist at AT&T Labs, Middletown, New Jersey. He received a Ph.D. in electrical engineering from the State University of New York at Stony Brook in 1982. His research interests are in the optimization, analysis, and design of software defined networks and mobility networks. He is an IEEE Fellow (2009) and became an AT&T Fellow in 2009 for "outstanding contributions to performance analysis and robust design and their application to improving the performance, reliability and scalability of AT&T's networks."

David Lynch (dflynch@att.com) is a lead inventive scientist at AT&T Labs, Middletown, New Jersey. He received an A.B. in mathematics and computer science from Susquehanna University and an M.S. and Ph.D. in operations research from Cornell University. Since joining AT&T in 1984, he has worked on a variety of network optimization applications, including private and public, voice and data, WAN and metro, and Layers 1 through 3. His research interests include development and implementation of telecommunications network optimization algorithms.

Gaurav Thakur (gt6510@att.com) is a principal inventive scientist at AT&T Labs, Middletown, New Jersey. He received a B.S. in mathematics from the University of Maryland in 2007 and a Ph.D. in applied mathematics from Princeton University in 2011. His research interests are in statistical signal processing, machine learning and harmonic analysis, as well as applications of these techniques to other fields.

Simon Tse (stse@att.com) is a director of inventive science at AT&T Labs, Middletown, New Jersey. He received a B.S. in engineering from Brown University, an M.S. and a Ph.D. from Harvard University in applied sciences, and an M.B.A. from the Wharton School of the University of Pennsylvania. He began his career in 1985 with AT&T Bell Laboratories. He currently manages a group of technical professionals in network topology designs, network traffic management, and software defined network controllers for multi-layer network resource and routing optimization.