Using Artificial Neural Networks to Evaluate the Capacity and Cost of Multi-Fiber Optical Backbone Networks

Alexandre Freitas and João Pires *

Posted Date: 18 October 2024

doi: 10.20944/preprints202410.1299.v1

Keywords: multi-fiber optical networks; artificial neural networks; machine learning; network capacity and cost; fiber assignment

Article

Using Artificial Neural Networks to Evaluate the Capacity and Cost of Multi-Fiber Optical Backbone Networks

Alexandre Freitas [1], João Pires [2,*]

[1] Department of Electrical and Computer Engineering, Instituto Superior Técnico, Universidade de Lisboa, Avenida Rovisco Pais 1, 1049-001 Lisboa, Portugal; alexandre.costa.freitas@tecnico.ulisboa.pt
[2] Department of Electrical and Computer Engineering and Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, Avenida Rovisco Pais 1, 1049-001 Lisboa, Portugal; jpires@lx.it.pt
* Correspondence: jpires@lx.it.pt

Abstract: A possible solution to address the enormous increase in traffic demands faced by network operators is to rely on multi-fiber optical backbone networks. These networks use multiple optical fibers between adjacent nodes, and, when properly designed, they are capable of handling petabits of data per second (Pbit/s). In this paper, an artificial neural network (ANN) model is investigated to estimate both the capacity and cost of a multi-fiber optical network. Furthermore, a fiber assignment algorithm is also proposed to complement the network design, enabling the generation of datasets for training and testing the developed ANN model. The model consists of three layers, including one hidden layer with 50 hidden units. The results show that for a large network, such as one with 100 nodes, the model can estimate performance metrics with an average relative error of less than 0.4% for capacity and 4% for cost, while achieving a computation time nearly 800 times faster than the heuristic approach used in network simulation. Additionally, the network capacity is around 5 Pbit/s.

Keywords: multi-fiber optical networks; artificial neural networks; machine learning; network capacity and cost; fiber assignment

1. Introduction

In recent years, data traffic has increased significantly, a trend expected to continue due to the growth of applications and services that require high bandwidth and generate large amounts of data. Examples include video streaming services, cloud computing, machine-to-machine applications, online gaming and the adoption of emerging technologies like 5G and beyond and advanced artificial intelligence applications [1]. This evolving scenario places special requirements on the backbones of network operators, which could experience traffic flows between their nodes reaching tens of Tb/s in the medium term, and even up to hundreds of Tb/s in the long term [1]. This situation presents a significant challenge for the design of future optical networks, particularly their backbone segments.

Optical networks are communication infrastructures, owned by telecommunication operators (telcos) or internet companies (e.g., Google, Microsoft, Meta), that utilize light for transmission, processing, and routing information and rely on optical fibers as their transmission medium. A fundamental technology in the field of optical networking is Wavelength Division Multiplexing (WDM). WDM allows the simultaneous transmission of multiple optical signals (also designated as optical channels) on the same optical fiber, with each channel using a different wavelength. The number of optical channels that can be transmit over an optical fiber is limited to about 100 when using the traditional C-band, restricting the maximum WDM transmission capacity to well below 100 Tb/s for significant distances [2]. To greatly increase the number of optical channels to cope with the

 doi:10.20944/preprints202410.1299.v1

enormous growth in bandwidth demand, one can rely on space division multiplexing (SDM) techniques. This approach can be implemented using a multi-fiber (MF) solution, i.e., multiple standard single-mode fiber pairs per link instead of just one, as it is typical or, alternatively, advanced fibers such as multicore fibers or few mode fibers, with both solutions still operating in the C-band [2]. By relying on these solutions, it is feasible to design petabit-class optical networks, which are networks capable of handling data at speeds reaching or exceeding one petabit per second (Pb/s) [3].

For designing MF networks, it is crucial to define, in addition to the traditional routing and wavelength assignment solutions, a strategy for allocating fibers to the network, specifically a fiber assignment strategy. In [4], two approaches were proposed to optimize the networks capacity by adding extra fibers. In the first approach, fibers were added to links supporting the maximum number of traffic demands, while in the second, fibers were added to links exhibiting the highest number of adjacent demands. Furthermore, in [5], the idea is to add extra fibers to links that are responsible for blocking traffic demands due to spectrum exhaustion, with the goal of minimizing the number of fibers added.

Network capacity is a key performance metric in optical networks. This capacity can be defined as the maximum amount of data that the entire network can handle per unit of time, and it is closely related to the concept of channel capacity introduced by Claude Shannon in 1948 [6].The estimation of network capacity is a challenging task because it depends not only on physical layer aspects related to optical fibers and other optical devices but also on networking aspects such as physical and logical topology, routing, as well as wavelength and modulation assignment. Consequently, it suffers from the hurdle of long computation times, especially when dealing with large-scale networks. Although the problem of predicting optical network capacity has been the focus of many studies, (see [7–10]), to the best of the authors' knowledge, none of the published research has relied on machine learning (ML) techniques for this purpose, despite these techniques being widely used in the context of optical networks to address other problems [11–13]. The closest study is reported in [14], where a routing and wavelengths assignment (RWA) problem is treated using ML techniques by transforming it into a multi-classification problem, which is then solved using logistic regression and deep neural network techniques. However, the network capacity estimation problem although also involving RWA calculations, is more general than this. Furthermore, the complexity of the problem for MF-networks is even higher due to the necessity of using fiber assignment techniques.

In this paper, we investigate the utilization of a ML solution, specifically an artificial neural network (ANN) model [12], to estimate both the capacity and cost metrics of an MF-based optical network capable of handling Pb/s of data, with the cost being defined as the total length of optical fiber required in the network. The goal is to determine whether it is possible to significantly speed up the computations of these two metrics in comparison with heuristic methods, while still achieving accurate results.

To generate the large sets of synthetic data needed to train and test the model, we used a tool previously developed by the authors [10].This tool not only generates random networks topologies that aim to mimic real optical backbone networks but also performs routing and fiber assignment operations on these networks using heuristics developed specifically for this purpose, including the fiber assignment algorithm that is described in this work, which is a crucial component of our methodology.

The rest of the paper is organized as follows: Section 2 reviews important aspects of network modeling and random network generation and explains how both network capacity and cost can be computed. It describes also the fiber assignment algorithm proposed here for allocating fibers in MF-networks. Section 3 details the ANN model introduce in this work. Section 4 presents some simulation results and, finally, Section 5 summarizes and concludes the paper.

2. Network Aspects and Data Set Generation

2.1. Network Modelling

In an abstract way, an optical network can be described as an undirected weighted graph $G(V,E)$, with $V = \{v_1, \dots, v_N\}$ denoting a set of nodes and $E = \{e_1, \dots, e_K\}$ denoting a set of links,

 doi:10.20944/preprints202410.1299.v1

where $N = |V|$ is the number of nodes and $K = |E|$ is the number of links. In transparent optical networks, all node functionalities take place in the optical domain, and the nodes are built using reconfigurable optical add-drop multiplexers (ROADMs), which are responsible for switching optical channels between different fibers, among other functions. Meanwhile, an optical link represents a physical interconnection between two nodes, implemented using a pair of optical fibers, or in the case of MF networks multiple pairs of fibers. Furthermore, each optical fiber supports WDM signals, meaning it carries a specific number of optical channels. Each link $(v_i, v_j) \in E$ is characterized by three attributes: $l_{i,j}$, the link length in kilometers between the nodes v_i and v_j; $nf_{i,j}$, the number of optical fiber pairs in the link; $u_{i,j}$, the link capacity measured in terms of the number of optical channels denoted as N_{ch}. In this work, we assume that fiber transmission takes place in the extended C band, which has a bandwidth of 4800 GHz, enabling the support of $N_{ch,max}$=75 channels, with a channel spacing of 64 GHz corresponding to a baud rate of 64 Gbaud.

Besides N and K, other important parameters of the graph G are the node degree $\delta(G)$, the network diameter $d(G)$, and the algebraic connectivity $a(G)$. $\delta(G)$ defines the number of links connected to a given node, $d(G)$ is the length of the longest shortest path between any two nodes, while $a(G)$ is the second smallest eigenvalue of the graph's Laplacian matrix [15].

In the context of ANNs, it is necessary to have very large datasets for training and testing purposes. To achieve this, it is useful to be able to generate numerous network topologies, which can be done through random graphs designed to adequately describe the characteristics of real-world optical networks. In [10], we described a tool that we developed to generate random networks appropriate for describing optical backbone networks. The tool is based on a modified Waxman model and can generate networks that are resilient to single-link failures. In a simplified way, this model works by dividing a two-dimensional (2D) square plane with area $A = L^2$ (L is the side length of the plane) into a set of regions. In these regions, N nodes are randomly placed, and then the nodes are interconnected with links according to the Waxman probability, which is characterized by the α and β parameters, both in the range [0,1].

2.2. Routing, Fibre Assignment, Capacity and Cost

Network capacity refers to the maximum amount of data that the network can theoretically handle per unit of time, typically measured in bits per second (bit/s). This metric depends on many network parameters, including the physical topology defined by the graph $G(V, E)$ and the logical topology, which describes the way how the information flows between all the networks nodes. The logical topology is defined by the traffic matrix $T = [t_{s,d}]$, where each entry $t_{s,d}$ represents a traffic demand, or in other terms, the volume of traffic flowing from a source node s to a destination node d, with $s, d \in V$. For each traffic demand $t_{s,d}$, it is necessary to find a path in the graph $G(V, E)$, between node s and node d. This is the role of the routing process. The routing process can be implemented using rigorous mathematical techniques, such as integer linear programing (ILP), or heuristic, as an alternative [16]. As ILPs become computationally infeasible for large-scale networks, we have to rely on heuristics in this work, as the analysis of such networks is paramount.

When the number of channels N_{ch} per fiber is limited, as in this work, the routing process is known as constrained routing and can lead to traffic demand blocking whenever no channels (wavelengths) are available on one or more links of the path. To overcome such a limitation, one can add more pairs of optical fiber per link as needed, as it is the case of MF-networks. This leads to a new process referred to as unconstrained routing plus fiber assignment. This process can be implemented using the heuristics proposed in [10].

To generate the datasets required to train and test the ANN we have applied the referred heuristics to the randomly generated networks using the modified Waxman model assuming a uniform traffic demand between all the network node pairs, which can be defined as

$$t_{s,d} = \begin{cases} 1 & s \neq d \\ 0 & s = d \end{cases}. \quad (1)$$

Taking into account the traffic matrix $T = [t_{s,d}]$ of size $N \times N$, and assuming that $u_{i,j} = \infty$, we can apply unconstrained routing to each network graph $G(V, E)$ to compute the list of established

 doi:10.20944/preprints202410.1299.v1

paths $P = [\pi_{s,d}]$, with the path $\pi_{s,d}$ having the length $l(\pi_{s,d}) = \sum_{i,j} l_{i,j}$. Additionally, we compute the link wavelength matrix, $W = [w_{i,j}]$,which is also a $N \times N$ matrix, where $w_{i,j}$ is the list of all the wavelengths λ_k present in the link (i,j), $i.e., w_{i,j} = [\lambda_k]$. As referred before, fiber assignment is central process in MF networks. To implement this process, we propose Algorithm 1, which allows us to obtain the fiber matrix $NF = [nf_{i,j}]$, representing the number of fibers per link, taking into account that the maximum number of wavelengths per link is $N_{\lambda,max}$=$N_{ch,max}$.

Algorithm 1: Fiber Assignment

Input: graph $G(V,E)$; wavelength matrix $W = [w_{i,j}]$, number of wavelengths $N_{\lambda,max}$.
Output: fiber matrix $NF = [nf_{i,j}]$.

Initialize NF,with $nf_{i,j} = 0, \forall (i,j) \in E$.
for each pair of nodes (i,j) in W **do**
  if G has an egde (i,j) **then**
    if there are no wavelengths used in(i,j), i.e. $w_{i,j} = 0$ **then**
      $nf_{i,j} \leftarrow 1$:At least one fiber is required
    else
      normalized wavelengths $\leftarrow w_{i,j}$ mapped into the range 1 to $N_{\lambda,max}$
      num_fibres ←maximum number of wavelengths repetitions in normalized wavelengths
      $nf_{i,j} \leftarrow$ num_fibres
    end if
  else
    $nf_{i,j} \leftarrow 0$: Case there is no edge *(i,j)*
  end if
end for
return NF

Note that with the unconstrained routing, the number of wavelengths in each link is not limited, so the value assigned to a given λ_k can be any natural number, in contrast to constraint routing, where it is bounded by $N_{\lambda,max}$. In the algorithm, to determine the number of fibers needed in each link, the maximum number of "repeated wavelengths" in that link must be determined. A wavelength is considered a "repeated wavelength" when its value modulo $N_{\lambda,max}$ (where the modulo operation returns the remainder after division) is equal to that of another wavelength also present in that link. For instance, if $N_{\lambda,max}$is 75, then wavelengths 1 and 76 are "repeated" because 76 modulo 75 equals 1. This implies that both wavelengths would occupy the same channel in a link, hence they are "repeated". This concept is crucial in determining the number of fibers needed for a link, ensuring that each "repeated" wavelength has its own fiber. Finding the maximum count of "repeated wavelengths" will ensure that there are enough fibers to accommodate all the wavelengths, thus assuring that there are no channels with the same wavelength on the same fiber.

By knowing the length of the path $\pi_{s,d}$, $l(\pi_{s,d})$, it is possible to compute its maximum capacity value, $C(\pi_{s,d})$, also denoted as Shannon capacity, measured in bits per second. This calculation uses the optical reach values of the path (see Table 2 of [10]), where optical reach is defined as the maximum length of the path for which a certain value of the capacity can be achieved assuming a baud rate of 64 Gbaud. Furthermore, after obtaining the capacity of all the established paths one arrives at the network capacity, which is given by

$$C_{net} = \sum_{s,d} C(\pi_{s,d}). \quad (2)$$

Another important metric is the network's cost. For simplification purposes, we assume that the transponder cost can be neglected in comparison with the fiber cost, which seems to be a reasonable assumption for optical backbone networks [17]. In this case, the network cost is given by

 doi:10.20944/preprints202410.1299.v1

$$\Lambda_{net} = \sum_{i,j} l_{i,j} \times nf_{i,j}. \quad (3)$$

3. Neural Network Design

An artificial neural network is a network of units, also called neurons, which are organized in multiple layers, including an input layer, a variable number of hidden layers and an output layer. These layers operate in a fully connected way, meaning that each neuron of a given layer is connected to all the neurons of the next layer. Each neuron has a variable weight per input, denoted as $\omega_{m,i}$,with m defining the neuron position in a layer and i its input, which are summed together along with a bias term b_m. The result of this operation is then passed through an activation function to obtain the output of that neuron. The activation function used in this study for the hidden layers is the ReLU (Rectified Linear Unit) function which is given by

$$g(x) = \max(0, x) \quad (4)$$

while for the output layer we have the linear activation function, that is

$$g(x) = x. \quad (5)$$

Note that, both activation functions are commonly used in regression problems, such as the one we are considering here [18].

The training of neural networks consists of determining the values for all $\Omega = [\omega_{m,i}]$ matrices and bias vectors $B = [b_m]$ that minimize a given loss function with a given iterative method (optimizer algorithm). For the training process, it is necessary to randomly generate a large number of datasets, using the procedures described previously. Each dataset includes an array of inputs $X = [x_1, x_2 \ldots . x_n]$, called features, and an array of outputs $Y = [y_1, y_2]$, obtained by network simulation, called labels. The features include the number of nodes, the number of links, the network diameter, the algebraic connectivity, and quantities such as the maximum, minimum, average and variance of both link length and node degree. Furthermore, the labels include the network capacity $y_1 = C_{net}$, given by (2), and the network cost $y_2 = \Lambda_{net}$, given by (3).

In the training process each dataset is split into a training set (the data used to determine the model's parameters), a validation set (used to make an unbiased evaluation of the model's performance during training) and a test set (used to assess the model's performance after the training is complete). Before the data is split into these three sets, it needs to be pre-processed and shuffled. Data pre-processing consists in preparing the data to make it more suitable for the training process.

The loss function is used to measure the difference between the value predicted by the ANN and the actual value obtained by simulation. In other words, it measures the error associated to the model's predictions. For regression problems, the mean squared error (MSE) is commonly used as the loss function [18]. MSE can be expressed as follows:

$$MSE = \frac{1}{M} \sum_{i=1}^{M} (\hat{y}_i - y_i)^2 \quad (6)$$

where M is the number of data values being considered, $\hat{y}_i$ are the estimated values, and y_i are the actual values.

The optimizer algorithm is the method that determines how the weight matrices and bias vectors are updated during the training process. Common optimizers include the Stochastic Gradient Descent (SGD) or the Adaptive Momentum Estimation (Adam), with the former being used in this work. The update of the network parameters requires the computation of the gradient of the loss function, a task performed by the backpropagation algorithm [19]. An important parameter related to the optimizer is the learning rate. This parameter determines the magnitude of the updates applied to the weights and biases during each iteration. Another important parameter is the batch size, which refers to the size of subsets into which the training data is divided. The dropout regularization can

 doi:10.20944/preprints202410.1299.v1

also be used to prevent overfitting which occurs when a model learns the training data too closely but fails to make accurate predictions on the testing data.

A key aspect of training an ANN is optimizing the hyperparameters. Hyperparameters are the variables that configure how the model learns from the data. This includes the number of hidden layers, the number of units in each hidden layer, the learning rate, the batch size, and dropout regularization. During training, various hyperparameters combinations are tested to achieve the best performance on the validation set. This operation is called hyperparameter tuning.

In this work, the tuning operation is performed using the R^2score metric, which is defined as

$$R^2 = 1 - \frac{\sum_{i=0}^{M}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{M}(y_i - \bar{y}_i)^2} \quad (7)$$

where y_i is the actual value, $\hat{y}_i$is the predicted value, $\bar{y}$ is the mean of the actual values, and M is the number of data values being considered. The R^2 score will take values between 0 and 1, where a value of 1 indicates that the model fits the data perfectly and 0 that it does not fit the data at all. That means that the closer the values are to 1, the better the model is performing [20].

From the hyperparameter tuning process the ANN's structure was defined (see Figure 1). The model that achieved the best performance on the validation set has 1 hidden layer with 50 hidden units ($m = 50$), considering the number of features equal to 12 ($n = 12$). The learning rate was optimized to 0.1, the batch size was set to 64, and no dropout regularization was needed.

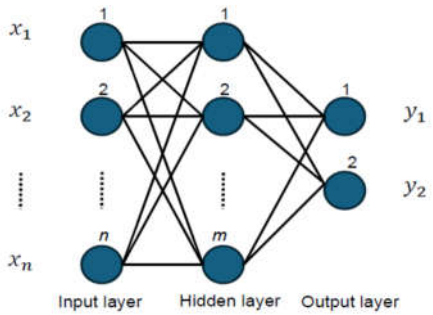


Figure 1. Model of the ANN network with 1 hidden layer.

This model structure and learning rate resulted in a relatively high R^2 scores: 0.9994 for y_1 and 0.9962 for y_2. Additionally, it has a relatively low number of trained parameters (the total number of weights and biases), with 752 parameters, which represents a good balance between model complexity and performance. To build and optimized the ANN model we used the PyTorch framework [21].

4. Simulation Results and Discussion

To train the ANN model, a set of 8480 networks was used. These networks were generated with the tool described in [10] considering a 2D square plane with side lengths varying from 1000 km to 5000 km in increments of 1000 km, number of regions in the plane set to 4, number of nodes varying from 5 to 100, number of links varying from 5 to 231 and an average node degree varying from 2 to around 5. The Waxman parameters chosen were α = β = 0.4. Furthermore, the maximum number of channels per links is set to $N_{ch,max}$= 75.

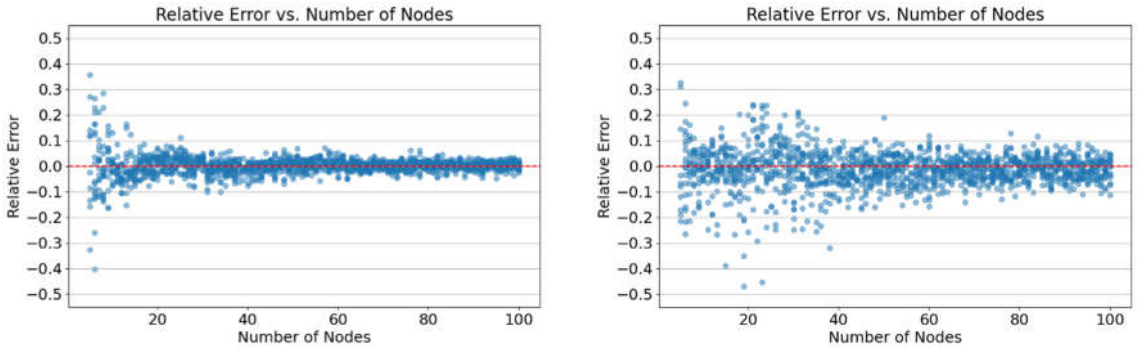
Once the model is trained, the final step is to evaluate its performance through testing. For this purpose, a dataset of 1440 random networks was generated under the same conditions as those used to train the model. The network simulation took around 1 hour and 16 minutes for the entire dataset, while the prediction time for the ANN model was just 11 milliseconds.

The mean relative errors for this test dataset, as defined by (8), are: 2.47% for the network capacity (y_1) and 5.29% for the total fibre cost (network cost) (y_2) predictions. Figure 2 shows the scatter plot of the relative errors against the number of nodes for both outputs. Each dot represents the relative error (RE) for each individual network in the set, given by:

 doi:10.20944/preprints202410.1299.v1

$$RE = \frac{y_i - \hat{y}_i}{y_i} \tag{8}$$

with y_i being the value determined from the simulation solution and $\hat{y}_i$ the prediction made with the ANN model.

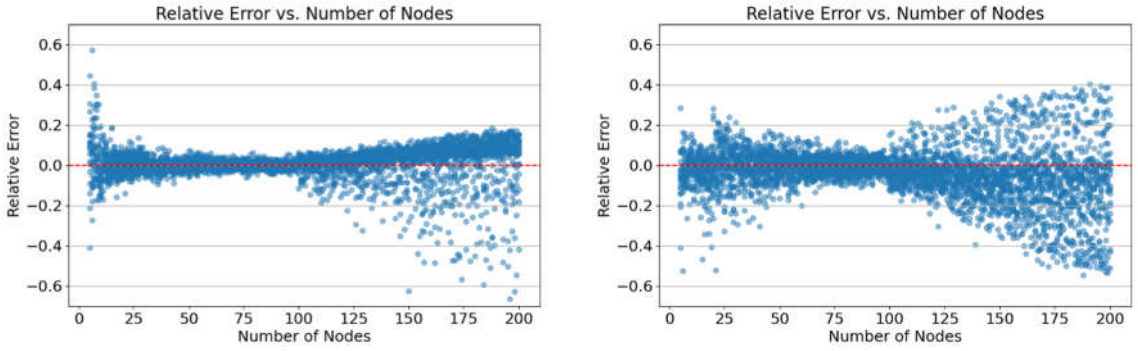


(**a**) (**b**)

Figure 2. Relative errors as a function of the number of nodes for both outputs of the ANN model (*N* ranging from 5 to 100) (**a**) Total network capacity; (**b**) Total fiber cost.

It was also shown that for the total network capacity (Figure 2(a)) 89.45% of the examples have a relative error below 5%, and 96.67% of the examples have a relative error below 10%. In the case of the total fiber cost (Figure 2 (b)) 87.02% of the examples have a relative error below 10%, and 94.24% of the examples have a relative error below 15%. It can be seen that the model tends to perform better on networks with a higher number of nodes, while its performance is more irregular on networks with fewer nodes. A possible explanation for this behavior is that smaller networks might exhibit more variability in their features as well as in the relationships between features and labels, which makes it more challenging for the model to make predictions accurately. On the other hand, larger networks could be more homogeneous, exhibiting more uniform and consistent patterns that the model can learn and predict more effectively.

In order to analyse how the ANN model behaves with testing datasets that have a number of nodes outside the training range, we generated 3920 additional networks under the same conditions as the previous sets, but with the number of nodes ranging from 5 to 200. Generating this set took 55 hours and 31 minutes, while the ANN model predicted the corresponding set in only 79 milliseconds. Figure 3 shows a scatter plot comparing the relative errors as a function of the number of nodes for this set of networks. The plots in Figure 3 show that the results are identical to those of Figure 2 when the number of nodes ranges from 5 to 100. However, outside this range, the model's performance become unreliable, although it still performs quite well for up to about 115 nodes. This behaviour is expected, as the model was trained on a specific range of data (number of nodes ranging from 5 to 100) and extrapolating beyond this range can lead to less reliable predictions.



(**a**) (**b**)

 doi:10.20944/preprints202410.1299.v1

Figure 3. Relative errors as a function of the number of nodes for both outputs of the ANN model (*N* ranging from 5 to 200) (**a**) Total network capacity; (**b**) Total fiber cost.

The capability of a model to perform well in the presence of unseen data, which it was not trained on, is known as Out-of-Distribution (OOD) generalization [22]. This is a challenge that conventional supervised learning methods (such as ANNs) often find difficult to handle effectively as these types of models fundamentally assume that the training and test datasets originate from the same distribution. Note that addressing the OOD generalization problem is an active area of research in the field of ML [22].

Table 1 compares the results predicted by the ANN model for the total network capacity ($\hat{y}_1$) and total fibre cost ($\hat{y}_2$) with the corresponding results obtained by applying a heuristic approach to different random networks, using the tool described in [10], as well as Algorithm 1 for the fiber assignment task. These results show that the ANN models tend to have a good performance in the generated networks within this range of nodes, with the relative errors generally being low. Furthermore, the prediction times with the ANN are always significantly faster than the computation times obtained with the network simulation tool. For example, for a network with 100 nodes the prediction time is about 17.1 milliseconds, whereas the computation time is about 13.5 seconds. This means that the ANN model is roughly 800 times faster than the heuristics approach, while achieving low relative errors of about 0.4% for network capacity and about 4% for network cost.

Table 1. Accuracy of ANN prediction: y_1: capacity; y_2: cost.

N	y_1 [Tb/s]	$\hat{y}_1$[Tb/s]	RE(%)	y_2 [10^3 km]	$\hat{y}_2$ [10^3 km]	RE(%)
10	48.0	45.4	5.44	24.47	24.08	1.59
20	303.2	317.9	4.86	14.71	14.00	4.85
30	708.0	705.2	0.39	27.05	26.62	1.57
40	803.0	820.1	2.13	122.27	123.21	0.77
50	1244.2	1214.4	2.40	231.63	257.36	11.1
60	1837.2	1937.3	5.45	267.02	247.97	7.14
70	3432.8	3393.9	1.13	128.44	131.79	2.61
80	3185.0	3197.8	0.40	626.00	597.00	4.63
90	5898.6	5864.0	0.59	189.51	194.54	2.66
100	5394.6	5376.5	0.34	718.09	690.02	3.91

Remarkably, the network capacity for a number of nodes greater than or equal to 50 nodes exceeds 1 Pb/s, reaching about 5 Pb/s for 100 nodes. However, this comes at the cost of significantly increasing the required optical fiber length in the network, as this work is based on the MF paradigm, where additional fibers are added whenever a link reaches its maximum supported number of optical wavelengths.

A key point in the analysis is understanding how the ANN model performs on real optical network topologies, despite being trained on synthetic data generated from random networks. To address this point, Table 2 provides results for four real reference networks: COST239 ($N = 11, K = 26, \bar{l} = 462.6$ km) [23], DTAG ($N = 14, K = 23, \bar{l} = 236.5$ km) [9], NSFNET ($N = 14, K = 21, \bar{l} = 1211.3$ km) [23], and UBN ($N = 24, K = 43, \bar{l} = 993.2$ km) [23], with $\bar{l}$ being the average link length.

Table 2. Accuracy of DNN predictions in reference networks. y_1: capacity; y_2: cost.

Network	y_1 [Tb/s]	$\hat{y}_1$[Tb/s]	RE(%)	y_2 [10^3 km]	$\hat{y}_2$ [10^3 km]	RE(%)
COST239	81.2	82.8	2.01	24.06	23.07	4.11
DTAG	147.4	145.9	1.01	10.88	10.95	0.69
NSFNET	98.0	104.1	6.18	45.39	38.63	14.87
UBN	272.8	269.9	1.10	85.42	101.42	18.73

 doi:10.20944/preprints202410.1299.v1

The results show that the ANN model predicts both outputs with low relative errors in the majority of cases and achieves computation times approximately 10 times faster than the heuristic method. However, there are instances where higher relative errors have been observed, with two cases exceeding a 10% relative error for the network cost: the NSFNET and UBN cases. Interestingly, these two cases correspond to the networks with larger average link lengths. An explanatory hypothesis for this behavior is that these networks exhibit significant variability in their features, making it more difficult for the ANN model to accurately capture the relationships between features and labels, a trend similar to the one shown in Figure 2(b) for networks with fewer than 40 nodes.

5. Conclusions

In this paper, the problem of estimating the capacity and cost of multi-fiber optical networks was addressed using for this purpose an ANN model. These networks, by using multiple fiber pairs per link, can achieve very high network capacities, even on the order of petabits per second.

To generate the datasets required to train and test the ANN, we applied an appropriate heuristic that relies on a fiber assignment algorithm, which was also proposed in the context of this work.

The implemented model was an ANN with 12 inputs (parameters related to the physical topology of the optical network), 2 outputs, and 1 hidden layer. The outputs correspond to two metrics: the network capacity, measured in Tbit/s, and the network cost, quantified by the total length of optical fiber deployed in the network, measured in km.

The ANN was trained with a number of nodes varying between 5 and 100, and it was extensively tested within the same range. The results showed good performance with a mean relative error of 2.47% and 5.29% for the first and second metric, respectively. The ANN model also showed significantly faster performance compared to a heuristic method, with the ANN predictions never taking more than a few tens of milliseconds, while the network simulation could take up to tens of seconds to reach the results in larger networks.

Remarkably, the network capacity for 50 or more nodes exceeds 1 Pb/s, reaching about 5 Pb/s for 100 nodes. However, this comes at the cost of a significant increase in the length of the total optical fiber required in the network.

Author Contributions: Conceptualization, A.F. and J.P.; methodology, A.F. and J. P.; software, A.F.; validation, A.F.; formal analysis, A.F and J. P.; investigation, A.F. and J. P.; writing—original draft preparation, J.P.; writing—review and editing, J.P.; visualization, A.F.; supervision, J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ruiz , M. *et al*. Network traffic analysis under emerging beyond-5G scenarios for multi-band optical technology adoption. *J. Opt. Commun. Netw*. **2023**, 15, F36-F47, doi: 10.1364/JOCN.492128.
2. Winzer, P. J. The future of communications is massively parallel. *J. Opt. Commun. Netw.* **2023**, 15, 783-787, doi: 10.1364/JOCN.496992.
3. Furukawa, H.; Luís, R. S. Petabit-class optical networks based on spatial-division multiplexing technologies. In Proceedings of the 2020 International Conference on Optical Network Design and Modelling (ONDM), Barcelona, 18-21 May 2020, doi: 10.23919/ONDM48393.2020.9132998.
4. Parker, M. C.; Wright, P.; Lord, A. Multiple fiber, flexgrid elastic optical network design using MaxEnt optimization. *J. Opt. Commun. Netw*. **2015, 7**, B194-B201, doi: 10.1364/JOCN.7.00B194.
5. Lopez, V. *et al*. Optimized design and challenges for C&L band optical line systems. *J. Lightw. Technol*. **2020**, 38, 1080-1091, doi: 10.1109/ JLT.2020.2968225.
6. Shannon, C. E. A mathematical theory of communication. *The Bell System Technical J*. **1948,** 27, 379-423, doi: 10.1002/j.1538-7305.1948.tb01338.x.

7. Vincent, R. J.; Ives, D. J.; Savory S. J. Scalable capacity estimation for nonlinear elastic all-optical core networks. *J. Lightw. Technol*. **2019**, 37, 5380-5391, doi: 10.1109/JLT.2019.2942710.
8. Matzner, R.; Semrau, D.; Luo R.; Zervas, G.; Bayvel, P. Making intelligent topology design choices: understanding structural and physical property performance implications in optical networks. *J. Opt. Commun. Netw*. **2021**, 13, D53–D67, doi: 10.1364/JOCN.423490.
9. Ives, D. J.; Bayvel, P.; Savory, S. J. Routing, modulation, spectrum and launch power assignment to maximize the traffic throughput of a nonlinear optical mesh network. *Photon. Netw. Commun***. 2015**, 29, 244-256, doi: 10.1007/s11107-015-0488-0.
10. Freitas, A.; Pires, J. Investigating the impact of topology and physical impairments on the capacity of an optical backbone network. *Photonics*, **2024**, 11, 342, doi: 10.3390/photonics11040342.
11. Musumeci, F.; Rottondi, C.; Nag, A.; Macaluso, I.; Zibar, D.; Ruffini, M.; Tornatore, M. An overview on application of machine learning techniques in optical networks. *IEEE Commun. Surveys Tuts.*, **2019**, 1383-1408, doi: 10.1109/COMST.2018.2880039.
12. Gu, R.; Yang Z.; Ji Y. Machine learning for intelligent optical networks: a comprehensive survey. *arXiv:2003.05290v1*, **2020**, doi:10.1016/j.jnca.2020.102576.
13. Morais, R. M.; Pedro, J. Machine learning models for estimating quality of transmission in DWDM Networks. *J. Opt. Commun. Netw.*, **2018**, 10, D84-D99, doi: 10.1364/JOCN.10.000D84.
14. Martín, I., *et al*. Machine learning-based routing and wavelength assignment in software-defined optical networks. *IEEE Trans. Netw. Service Manag*. **2019**, 16, 871-883, doi: 10.1109/TNSM.2019.2927867.
15. Châtelain, B.; Bélanger, M. P.; Tremblay, C.; Gagnon, F.; Plant, D. V. Topological wavelength usage estimation in transparent wide area networks. *J. Opt. Commun. Netw*., **2009**, 1, 196-203, doi: 10.1364/JOCN.1.000196.
16. Santos, J.; Pedro, J.; Monteiro, P.; Pires, J. Optimized routing and buffer design for optical transport networks based on virtual concatenation. *J. Opt. Commun. Netw.* **2011**, 3, 725-738, doi: 10.1364/JOCN.3.000725.
17. Çetinkaya, E. K.; Alenazi, M. J. F.; Cheng, Y; Peck, A. M.; Sterbenz, P. G. A comparative analysis of geometric graph models for modelling backbone networks. *Optical Switch. and Networ.* **2014**, 14, 95-106, doi: 10.1016/j.osn.2014.05.001.
18. Chugh, S.; Ghosh, S.; Gulistan, A.; Rahman, B. Machine learning regression approach to the nanophotonic waveguide analyses. *J. Lightw.Technol.* **2019**, 37, 6080–6089, doi: 10.1109/JLT.2019.2946572.
19. Le Cun, Y. Theoretical framework for back-propagation. In Proceedings of the 1988 Connectionist Models Summer School, Pittsburg, USA (pp. 21-28), Morgan Kaufman.
20. Coefficient of determination, r-squared. Available online: https://www.ncl.ac.uk/ webtemplate/ask-assets/external/maths-resources/statistics/regression-and-correlation/ coefficient-of-determination-r-squared.html (accessed on 7 July 2024).
21. Pytorch. Available online: PyTorch | About (archive.org) (accessed on 20 April 2024).
22. Liu, J.; Shen, Z.; He, Z. Y.; Zhang, X.; Xu, R.; Yu, H; Cui, P. Towards out-of-distribution generalization: a survey. *arXiv:2108.13624*, **2023**. Available online: https://arxiv.org/abs/2108.13624, doi:10.48550/arXiv.2108.13624 (accessed on 3 October 2024).
23. Pires, J. J. O. On the capacity of optical backbone networks. *Network* **2024**, 4(1), 114–132, doi:10.3390/network4010006.