

# Pattern recognition model to aid the optimization of *Dynamic Spectrally-Spatially Flexible Optical Networks*

Paweł Ksieniewicz<sup>1</sup>[0000-0001-9578-8395], Róża Goścień<sup>1</sup>[0000-0003-4346-0763],  
Miroslaw Klinkowski<sup>2</sup>[0000-0002-8375-2827], and Krzysztof  
Walkowiak<sup>1</sup>[0000-0003-1686-3110]

<sup>1</sup> Wrocław University of Science and Technology  
Wyb. Wyspińskiego 27, 50-370 Wrocław, Poland  
{pawel.ksieniewicz, roza.goscien, krzysztof.walkowiak}@pwr.edu.pl

<sup>2</sup> National Institute of Telecommunications  
Szachowa 1, 04-894 Warszawa  
m.klinkowski@itl.waw.pl

**Abstract.** The following paper considers pattern recognition-aided optimization of complex and relevant problem related to optical networks. For that problem, we propose a four-step dedicated optimization approach that makes use, among others, of a regression method. The main focus of that study is put on the construction of efficient regression model and its application for the initial optimization problem. We therefore perform extensive experiments using realistic network assumptions and then draw conclusions regarding efficient approach configuration. According to the results, the approach performs best using *multi-layer perceptron* regressor, whose prediction ability was the highest among all tested methods.

**Keywords:** spectrally-spatially flexible optical networks · routing, space and spectrum optimization · network optimization · ensemble learning · pattern regression

## 1 Introduction

According to *Cisco* forecasts, the global consumer traffic in the Internet will grow on average with annual compound growth rate (CAGR) of 26% in years 2017–2022 [3]. The increase in the network traffic is a result of two main trends. Firstly, the number of devices connected to the internet is growing due to the increasing popularity of new services including *Internet of Things (IoT)*. The second important trend influencing the traffic in the internet is popularity of bandwidth demanding services such as video streaming (e.g., *Netflix*) and cloud computing. The Internet consists of many single networks connected together, however, the backbone connecting these various networks are optical networks based on fiber connections. Currently, the most popular technology in optical networks is WDM (*Wavelength Division Multiplexing*), which is expected to be

not efficient enough to support increasing traffic in the nearest future. In last few years, a new concept for optical networks has been deployed, i.e., architecture of *Elastic Optical Networks* (EONs). However, in the perspective on the next decade some new approaches must be developed to overcome the predicted "capacity crunch" of the Internet.

One of the most promising proposals is *Spectrally-Spatially Flexible Optical Network* (SS-FON) that combines *Space Division Multiplexing* (SDM) technology [14], enabling parallel transmission of co-propagating spatial modes in suitably designed optical fibers such as multi-core fibers (MCFs) [1], with flexible-grid EONs [4] that enable better utilization of the optical spectrum and distance-adaptive transmissions [15]. In MCF-based SS-FONs, a challenging issue is the inter-core crosstalk (XT) effect that impairs the quality of transmission (*QoT*) of optical signals and has a negative impact on overall network performance. In more detail, MCFs are susceptible to signal degradation as a result of the XT that happens between adjacent cores whenever optical signals are transmitted in an overlapping spectrum segment. Addressing the XT constraints significantly complicates the optimization of SS-FONs [8].

Besides numerous advantages, new network technologies bring also challenging optimization problems, which require efficient solution methods. Since the technologies and related problems are new, there are no benchmark solution methods to be directly applied and hence many studies propose some dedicated optimization approaches. However, due to the problems high complexity, their performance still needs a lot of effort to be put [8, 6]. We therefore observe a trend to use *artificial intelligence* techniques (with the high emphasis on *pattern recognition* tools) in the field of optimization of communication networks. According to the literature surveys in this field [11, 13, 10, 2], the researchers mostly focus on discrete labelled *supervised* and *unsupervised learning* problems, such as traffic classification. Regression methods, which are in the scope of that paper, are mostly applied for traffic prediction and estimation of quality of transmission (*QoT*) parameters such as delay or bit error rate.

This paper extends our study initiated in [7]. We make use of pattern recognition models to aid optimization of dynamic MCF-based SS-FONs in order to improve performance of the network in terms of minimizing bandwidth blocking probability (BBP), or in other words to maximize the amount of traffic that can be allocated in the network. In particular, an important topic in the considered optimization problem is selection of a modulation format (MF) for a particular demand, due to the fact that each MF provides a different tradeoff between required spectrum width and transmission distance. To solve that problem, we define applicable distances for each MF (i.e., minimum and maximum length of a routing path that is supported by each MF). To find values of these distances, which provide best allocation results, we construct a regression model and then combine it with *Monte Carlo* search. It is worth noting that this work does not address dynamic problems in the context of changing the concept over time, as is often the case with processing large sets, and assumes static distribution of the concept [9].

The main novelty and contribution of the following work is an in-depth analysis of the basic regression methods stabilized by the structure of the estimator ensemble [16] and assessment of their usefulness in the task of predicting the objective function for optimization purposes. In one of the previous works [7], we confirmed the effectiveness of this type of solution using a regression algorithm of the nearest weighted neighbors, focusing, however, much more on the network aspect of the problem being analyzed. In the present work, the main emphasis is on the construction of the prediction model. Its main purpose is:

- A proposal to interpret the optimization problem in the context of pattern recognition tasks.
- Construction of the prediction model based on the introduced interpretation.
- Extensive experimental verification of the *state-of-art* regression algorithms efficiency in a problems defined by datasets being the result of computer simulations performed for the purposes of network optimization.
- Proposition and verification of the use of such models in optimization of the problem.

The rest of the paper is organized as follows. In Section 2, we introduce studied network optimization problem. In Section 3, we discuss out optimization approach for that problem. Next, in Section 4 we evaluate efficiency of the proposed approach. Eventually, Section 5 concludes the work.

## 2 Optimization of Spectrally-Spatially Flexible Optical Network

The optimization problem is known in the literature as dynamic *Routing, Space and Spectrum Allocation* (RSSA) in SS-FONS [5]. We are given with an SS-FON topology realized using MCFS. The topology consists of nodes and physical link. Each physical link comprises of a number of spatial cores. The spectrum width available on each core is divided into arrow and same-sized segments called *slices*.

The network is in its operational state – we observe it in a particular time perspective given by a number of iterations. In each iteration (i.e., a time point), a set of demands arrives. Each demand is given by a source node, destination node, duration (measured in the number of iterations) and bitrate (in *Gbps*). To realize a demand, it is required to assign it with a light-path and reserve its resources for the time of the demand duration. When a demand expires, its resources are released. A light-path consists of a routing path (a set of links connecting demand source and destination nodes) and a channel (a set of adjacent slices selected on one core) allocated on the path links. The channel width (number of slices) required for a particular demand on a particular routing path depends on the demand bitrate, path length (in kilometres) and selected modulation format. Each incoming demand has to be realized unless there is not enough free resources when it arrives. In such a case, a demand is rejected. Please note that the selected light-paths in *i*-th iteration affect network state and allocation possibilities in the next iterations. The objective function is defined here

as bandwidth blocking probability (BBP) calculated as a summed bitrate of all rejected demands divided by the summed bitrate of all offered demands. Since we aim to support as much traffic as it is possible, the objective criterion should be minimized [5, 8].

The light-paths' allocation process has to satisfy three basic RSSA constraints. First, each channel has to consist of adjacent slices. Second, the same channel (i.e., the same slices and the same core) has to be allocated on each link included in a light-path. Third, in each time point each slice on a particular physical link and a particular core can be used by at most one demand [8].

There are four modulation formats available for transmissions — 8-QAM, 16-QAM, QPSK and BPSK. Each format is described by its spectral efficiency, which determines number of slices required to realize a particular bitrate using that modulation. However, each modulation format is also characterized by the maximum transmission distance (MTD) which provides acceptable value of optical signal to noise ratio (OSNR) at the receiver side. More spectrally-efficient formats consume less spectrum, however, at the cost of shorter MTDs. Moreover, more spectrally-efficient formats are also vulnerable to XT effects which can additionally degrade  $QoT$  and lead to demands' rejection [8, 7]. Therefore, the selection of the modulation format for each demand is a compromise between spectrum efficiency and  $QoT$ .

To answer that problem, we use the procedure introduced in [7] to select a modulation format for a particular demand and routing path [7]. Let  $m = 1, 2, 3, 4$  denote modulation formats ordered in increasing MTDs (and in decreasing spectral efficiency at the same time). It means that  $m = 1$  denotes 8-QAM and  $m = 4$  denotes BPSK. Let  $MTD = [mtd_1, mtd_2, mtd_3, mtd_4]$  be a vector of MTDs for modulations 8-QAM, 16-QAM, QPSK, BPSK respectively. Moreover, let  $ATD = [atd_1, atd_2, atd_3, atd_4]$  (where  $atd_i \leq mtd_i, i = 1, 2, 3, 4$ ) be the vector of applicable transmission distances. For a particular demand and a routing path we select most spectrally-efficient modulation format  $i$  for which  $atd_i$  is greater or equal to the selected path length and the XT effect is on an acceptable level. For each candidate modulation format, we assess the XT level based on the adjacent resources' (i.e., slices and cores) availability using procedure proposed in [7]. It is important to note that we do not indicate  $atd_4$  (for BPSK) since we assume that this modulation is able to support transmission on all candidate routing paths regardless of their length. Please also note that when XT level is too high for all modulation formats, the demand is rejected regardless of the light-paths' availability.

### 3 Pattern recognition model to aid optimization of SS-FONs

In Section 2 we have studied RSSA problem and emphasised the importance of efficient modulation selection task. For that task we have proposed solution method whose efficiency strongly depends on the applied ATD vector. Therefore, we aim to find  $ATD^*$  vector that provides best results. The vector elements have

to be positive and have upper bounds given by vector MTD. Moreover, the following condition have to be satisfied:  $atd_i < atd_{i+1}, i = 1, 2$ . Since solving RSSA instances is a time consuming process, it is impossible to evaluate all possible ATD vectors in a reasonable time. We therefore make use of regression methods and propose a scheme to find  $ATD^*$  depicted in Figure 1.

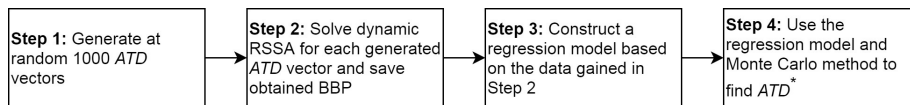


Fig. 1: Optimization procedure for dynamic SS-FON.

A representative set of 1000 different ATD vectors is generated. Then, for each of them we simulate allocation of demands in SS-FON (i.e., we solve dynamic RSSA). For the purpose of demands allocation (i.e., selection of light-paths), we use a dedicated algorithm proposed in [7]. For each considered ATD vector we save obtained BBP. Based on that data, we construct a regression model, which predicts BBP based on an ATD vector. Having that model, we use *Monte Carlo* method to find  $ATD^*$  vector, which is recommended for further experiments.

### 3.1 Solving RSSA

To solve an RSSA instance for a particular ATD vector, we use heuristic algorithm proposed in [7]. We work under the assumption that there are 30 candidate routing paths for each traffic demand (generated using Dijkstra algorithm). Since the paths are generated in advance and their lengths are known, we can use an ATD vector and preselect for these paths modulation formats based on the procedure discussed in Section 2. Therefore, RSSA is reduced to the selection of one of the candidate routing paths and a communication channel with respect to the resource availability and assessed XT levels.

### 3.2 Construction and validation of the model

From the perspective of pattern recognition methods, the abstraction of the problem is not the key element of processing. The main focus here is the representation available to construct a proper decision model. For the purposes of considerations, we assume that both input parameters and the objective function take only quantitative and not qualitative values, so we may use probabilistic pattern recognition models to process them. If we interpret the optimization task as searching for the extreme function of many input parameters, each simulation performed for their combination may also be described as a label for the training set of *supervised learning* model.

In this case, the set of parameters considered in a single simulation becomes a vector of object features  $(x_n)$ , and the value of the objective function acquired

around it may be interpreted as a continuous object label ( $y_n$ ). Repeated simulation for randomly generated parameters allows to generate a data set ( $X$ ) supplemented with a label vector ( $y$ ). A supervised machine learning algorithm can therefore gain, based on such a set, a generalization abilities that allows for precise estimation of the simulation result based on its earlier runs on the random input values.

A typical pattern recognition experiment is based on the appropriate division of the dataset into training and testing sets, in a way that guarantees their separability (most often using cross-validation), avoiding the problem of data peeking and a sufficient number of repetitions of the validation process to allow proper statistical testing of mutual model dependencies hypotheses. For the needs of the proposal contained in this paper, the usual 5-fold cross validation was adopted, which calculates the value of the  $r^2$  metric for each loop of the experiment.

### 3.3 Finding $ATD^*$

Having constructed regression model, we are able to predict BBP value for a sample  $ATD$  vector. Please note that the time required for a single prediction is significantly shorter than the time required to simulate a dynamic RSSA. The last step of our optimization procedure is to find  $ATD^*$  — vector providing lowest estimated BBP values. To this end, we use *Monte Carlo* method with a number of guesses provided by the user.

## 4 Experimental evaluation

### 4.1 Experiments set-up

The RSSA problem was solved for two network topologies — DT12 (12 nodes, 36 links) and Euro28 (28 nodes, 82 links). They model *Deutsche Telecom* (German national network) and European network, respectively. Each network physical link comprised of 7 cores wherein each of the cores offers 320 frequency slices of 12.5 GHz width. We use the same network physical assumptions and XT levels and assessments as in [7]. Traffic demands have randomly generated end nodes and birates uniformly distributed between 50 Gbps and 1 Tbps, with granularity of 50 Gbps. Their arrival follow *Poisson* process with an average arrival rate  $\lambda$  demands per time unit. The demand duration is generated according to a negative exponential distribution with an average of  $1/\mu$ . The traffic load offered is  $\lambda/\mu$  normalized traffic units (NTUS). For each testing scenario, we simulate arrival of  $10^6$  demands. Four modulations are available (8-QAM, 16-QAM, QPSK, BPSK) wherein we use the same modulation parameters as in [7].

For each topology we have generated 9 different datasets, each consists of 1000 samples of  $ATD$  vector and corresponding BBP. The datasets differ with the XT coefficient ( $\mu = 1 \cdot 10^{-9}$  indicated as "XT1",  $\mu = 2 \cdot 10^{-9}$  indicated as "XT2", for more details we refer to [7]) and network links scaling factor (the multiplier

used to scale lengths of links in order to evaluate if different lengths of routing paths influence performance of the proposed approach). For DT12 we use following scaling factors: 0.4, 0.6, 0.8,  $\dots$ , 2.0. For Euro28 the values are as follows: 0.104, 0.156, 0.208, 0.260, 0.312, 0.364, 0.416, 0.468, 0.520. We indicate them as "Sx.xxx" where x.xxx refers to the scaling factor value. Using these datasets we can evaluate whether XT coefficient (i.e., level of the vulnerability to XT effects) and/or average link length influence optimization approach performance.

The experimental environment for the construction of predictive models, including the implementation of the proposed processing method, was implemented in Python, following the guidelines of the *state-of-art* programming interface of the *scikit-learn* library [12]. Statistical dependency assessment metrics for paired tests were calculated according to the Wilcoxon test, according to the implementation contained in *scipy* module. Each of the individual experiments was evaluated by  $r^2$  score – a typical quality assessment metric for regression problems. The full source code, supplemented with employed datasets is publicly available in a GIT repository<sup>3</sup>.

Five simple recognition models were selected as the base experimental estimators:

- KNR – *k-Nearest Neighbors* regressor with five neighbors, leaf size of 30 and euclidean metric approximated by Minkowski distance,
- dKNR – KNR regressor weighted by distance from closest patterns,
- MLP – a *Multilayer Perceptron* with one hidden layer of one hundred neurons, with the ReLU activation function and *adam* optimizer,
- DTR – CART tree with MSE split criterion,
- LIN – *Linear Regression* algorithm.

## 5 Experimental evaluation

In this section we evaluate performance of the proposed optimization approach. To this end, we conduct three experiments. Experiment 1 focuses on the number of patterns required to construct a reliable prediction model. Experiment 2 assesses the statistical dependence of built models. Eventually, experiment 3 verifies efficiency of the proposed approach as a function of number of guesses in the *Monte Carlo* search.

### 5.1 Experiment 1

The first experiment carried out as part of the approach evaluation is designed to verify how many patterns – and thus how many repetitions of simulations – must be passed to individual regression algorithms to allow the construction of a reliable prediction model. The tests were carried out on all five considered regressors in two stages. First, the range from 10 to 100 patterns was analyzed, and in the second, from 100 to 1000 patterns per processing. It is important to

<sup>3</sup> <https://github.com/w4k2/regression-aided-optimization>

note that due to the chosen approach to cross-validation, in each case the model is built on 80% of available objects.

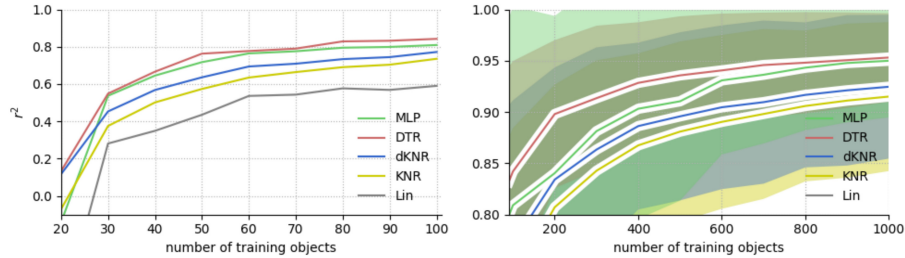


Fig. 2: Dependency between number of training objects and quality of predictions of analyzed regression models according to  $r^2$  metric for DT12 topology.

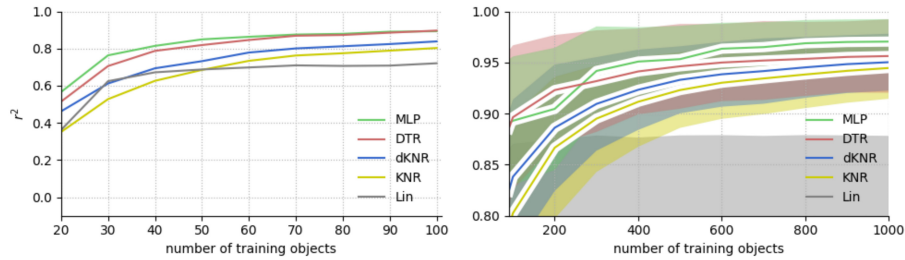


Fig. 3: Dependency between number of training objects and quality of predictions of analyzed regression models according to  $r^2$  metric for Euro28 topology.

The analysis was carried out independently on all available data sets, and due to the non-deterministic nature of sampling of available patterns, its results were additionally stabilized by repeating a choice of the objects subset five times.

In order to allow proper observations, the results were averaged for both topologies. Plots for the range from 100 to 1000 patterns were additionally supplemented by marking ranges of standard deviation of  $r^2$  metric acquired within the topology and presented in the range from the .8 value.

The results achieved for averaging individual topologies are presented in Figures 2 and 3. For DT12 topology, MLP and DTR algorithms are competitively the best models, both in terms of the dynamics of the relationship between the number of patterns and the overall regression quality. The *Linear Regression* clearly stands out from the rate. A clear observation is also the *saturation* of the models, understood by approaching the maximum predictive ability, as soon as around 100 patterns in the data set. The best algorithms already achieve quality within .8, and with 600 patterns they stabilize around .95. The relationship between each of the recognition algorithms and the number of patterns takes the form of a logarithmic curve in which, after fast initial growth, each subsequent object gives less and less potential for improving the quality of prediction. This



suggests that it is not necessary to carry out further simulations to extend the training set, because it will not significantly affect the predictive quality of the developed model.

Very similar observations may be made for *Euro28* topology, however, noting that it seems to be a simpler problem, allowing faster achievement of the maximum model predictive capacity. It is also worth noting here the fact that the *standard deviation* of results obtained by MLP is smaller, which may be equated with the potentially greater stability of the model achieved by such a solution.

### 5.2 Experiment 2

The second experiment extends the research contained in Experiment 1 by assessing the statistical dependence of models built on a full datasets consisting of a thousand samples for each case. The results achieved are summarized in Tables 1a and b.

Table 1: Mean  $r^2$  scores achieved by particular regression algorithms for all cases of both analyzed topologies.

| (a) DT12 topology. |       |         |         |         |       | (b) <i>Euro28</i> topology. |       |         |         |         |       |
|--------------------|-------|---------|---------|---------|-------|-----------------------------|-------|---------|---------|---------|-------|
| DATASET            | KNR   | dKNR    | MLP     | DTR     | LIN   | DATASET                     | KNR   | dKNR    | MLP     | DTR     | LIN   |
| <i>XT1-S0.4</i>    | 0.621 | 0.631   | 0.725   | 0.789   | 0.000 | <i>XT1-S0.104</i>           | 0.971 | 0.974   | 0.977   | 0.967   | 0.875 |
|                    | 5     | 1, 5    | 1, 2, 5 | all     | —     |                             | 5     | 1, 5    | 1, 5    | 5       | —     |
| <i>XT1-S0.6</i>    | 0.902 | 0.928   | 0.927   | 0.928   | 0.585 | <i>XT1-S0.156</i>           | 0.966 | 0.972   | 0.990   | 0.991   | 0.746 |
|                    | 5     | 1, 5    | 1, 5    | 5       | —     |                             | 5     | 1, 5    | 1, 2, 5 | all     | —     |
| <i>XT1-S0.8</i>    | 0.934 | 0.946   | 0.967   | 0.964   | 0.516 | <i>XT1-S0.208</i>           | 0.969 | 0.970   | 0.969   | 0.986   | 0.751 |
|                    | 5     | 1, 5    | 1, 2, 5 | 1, 2, 5 | —     |                             | 5     | 5       | 5       | all     | —     |
| <i>XT1-S1.0</i>    | 0.928 | 0.941   | 0.959   | 0.962   | 0.654 | <i>XT1-S0.260</i>           | 0.852 | 0.858   | 0.889   | 0.830   | 0.534 |
|                    | 5     | 1, 5    | 1, 5    | 1, 2, 5 | —     |                             | 4, 5  | 1, 4, 5 | all     | 5       | —     |
| <i>XT1-S1.2</i>    | 0.833 | 0.852   | 0.891   | 0.915   | 0.582 | <i>XT1-S0.312</i>           | 0.960 | 0.963   | 0.971   | 0.963   | 0.752 |
|                    | 5     | 1, 5    | 1, 2, 5 | all     | —     |                             | 5     | 5       | 5       | 5       | —     |
| <i>XT1-S1.4</i>    | 0.883 | 0.903   | 0.962   | 0.954   | 0.479 | <i>XT1-S0.364</i>           | 0.957 | 0.962   | 0.967   | 0.951   | 0.861 |
|                    | 5     | 1, 5    | 1, 2, 5 | 1, 5    | —     |                             | 5     | 1, 4, 5 | 1, 4, 5 | 5       | —     |
| <i>XT1-S1.6</i>    | 0.896 | 0.912   | 0.976   | 0.975   | 0.481 | <i>XT1-S0.416</i>           | 0.907 | 0.918   | 0.970   | 0.945   | 0.677 |
|                    | 5     | 1, 5    | 1, 2, 5 | 1, 2, 5 | —     |                             | 5     | 1, 5    | all     | 1, 2, 5 | —     |
| <i>XT1-S1.8</i>    | 0.961 | 0.968   | 0.987   | 0.988   | 0.623 | <i>XT1-S0.468</i>           | 0.933 | 0.942   | 0.983   | 0.971   | 0.607 |
|                    | 5     | 1, 5    | 1, 2, 5 | 1, 2, 5 | —     |                             | 5     | 1, 5    | all     | 1, 2, 5 | —     |
| <i>XT1-S2.0</i>    | 0.964 | 0.969   | 0.989   | 0.992   | 0.725 | <i>XT1-S0.520</i>           | 0.960 | 0.965   | 0.988   | 0.986   | 0.644 |
|                    | 5     | 1, 5    | 1, 2, 5 | 1, 2, 5 | —     |                             | 5     | 1, 5    | 1, 2, 5 | 1, 2, 5 | —     |
| <i>XT2-S0.4</i>    | 0.966 | 0.969   | 0.954   | 0.987   | 0.776 | <i>XT2-S0.104</i>           | 0.956 | 0.959   | 0.972   | 0.947   | 0.810 |
|                    | 3, 5  | 1, 3, 5 | 5       | all     | —     |                             | 5     | 1, 5    | all     | 5       | —     |
| <i>XT2-S0.6</i>    | 0.952 | 0.958   | 0.963   | 0.978   | 0.681 | <i>XT2-S0.156</i>           | 0.975 | 0.978   | 0.993   | 0.992   | 0.777 |
|                    | 5     | 1, 5    | 1, 5    | all     | —     |                             | 5     | 1, 5    | all     | 1, 2, 5 | —     |
| <i>XT2-S0.8</i>    | 0.965 | 0.966   | 0.970   | 0.968   | 0.752 | <i>XT2-S0.208</i>           | 0.950 | 0.951   | 0.956   | 0.959   | 0.893 |
|                    | 5     | 1, 5    | 5       | 5       | —     |                             | 5     | 5       | 5       | 5       | —     |
| <i>XT2-S1.0</i>    | 0.951 | 0.953   | 0.953   | 0.944   | 0.857 | <i>XT2-S0.260</i>           | 0.952 | 0.958   | 0.965   | 0.964   | 0.696 |
|                    | 4, 5  | 1, 4, 5 | 4, 5    | 5       | —     |                             | 5     | 1, 5    | 1, 5    | 5       | —     |
| <i>XT2-S1.2</i>    | 0.888 | 0.901   | 0.908   | 0.907   | 0.744 | <i>XT2-S0.312</i>           | 0.957 | 0.962   | 0.978   | 0.963   | 0.670 |
|                    | 5     | 1, 5    | 5       | 5       | —     |                             | 5     | 1, 5    | 1, 2, 5 | 5       | —     |
| <i>XT2-S1.4</i>    | 0.879 | 0.897   | 0.965   | 0.941   | 0.500 | <i>XT2-S0.364</i>           | 0.911 | 0.921   | 0.951   | 0.911   | 0.698 |
|                    | 5     | 1, 5    | all     | 1, 2, 5 | —     |                             | 5     | 1, 5    | all     | 5       | —     |
| <i>XT2-S1.6</i>    | 0.933 | 0.945   | 0.983   | 0.971   | 0.539 | <i>XT2-S0.616</i>           | 0.897 | 0.911   | 0.972   | 0.961   | 0.483 |
|                    | 5     | 1, 5    | all     | 1, 2, 5 | —     |                             | 5     | 1, 5    | 1, 2, 5 | 1, 2, 5 | —     |
| <i>XT2-S1.8</i>    | 0.967 | 0.971   | 0.989   | 0.980   | 0.700 | <i>XT2-S0.468</i>           | 0.938 | 0.949   | 0.986   | 0.979   | 0.551 |
|                    | 5     | 1, 5    | all     | 1, 2, 5 | —     |                             | 5     | 1, 5    | all     | 1, 2, 5 | —     |
| <i>XT2-S2.0</i>    | 0.976 | 0.979   | 0.989   | 0.984   | 0.809 | <i>XT2-S0.520</i>           | 0.955 | 0.962   | 0.990   | 0.989   | 0.660 |
|                    | 5     | 1, 5    | all     | 1, 2, 5 | —     |                             | 5     | 1, 5    | 1, 2, 5 | 1, 2, 5 | —     |

As may be seen, for the DT12 topology, the LIN algorithm clearly deviates negatively from the other methods, in absolutely every case being a worse solution

than any of the others, which leads to the conclusion that we should completely reject it from considering as a base for a stable recognition model. Algorithms based on neighborhood (KNN and dKNN) are in the middle of the rate, in most cases statistically giving way to MLP and DTR, which would also suggest departing from them in the construction of the final model. The statistically best solutions, almost equally, in this case are MLP and DTR.

For *Euro28* topology, the results are similar when it comes to LIN, KNN and dKNN approaches. A significant difference, however, may be seen for the achievements of DTR, which in one case turns out to be the worst in the rate, and in many is significantly worse than MLP. These observations suggest that in the final model for the purposes of optimization lean towards the application of *neural networks*.

What is important, the highest quality prediction does not exactly mean the best optimization. It is one of the very important factors, but not the only one. It is also necessary to be aware of the shape of the decision function. For this purpose, the research was supplemented with visualizations contained in Figure 4.

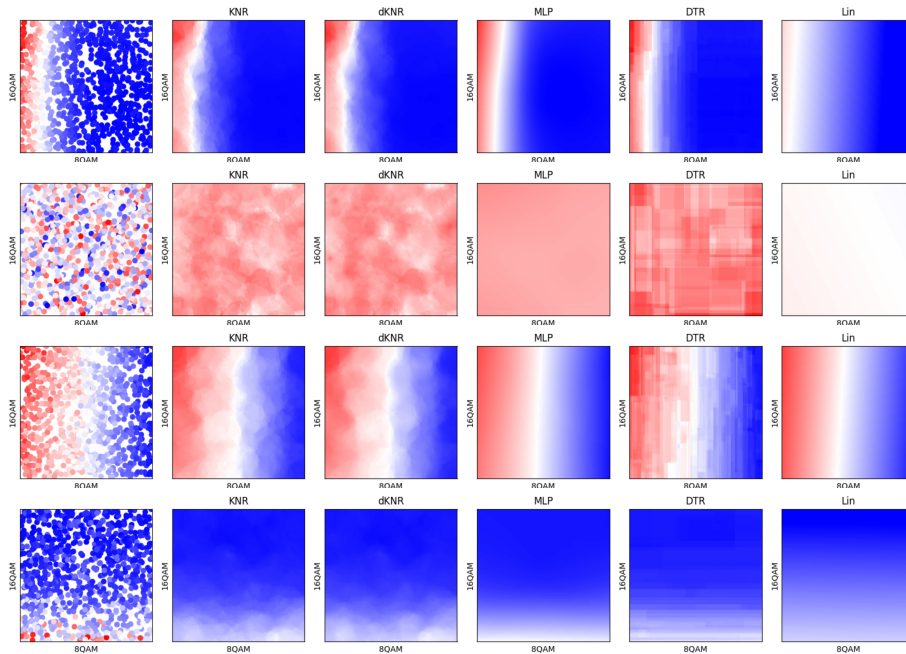


Fig. 4: Exemplary subspace visualization of decision function modeled by different regression algorithms supplemented by scatter plot of original objects inside subspace.

Algorithms based on neighborhood (KNN, dKNN) and decision trees (DTR) are characterized by a discrete decision boundary, which in the case of visu-

alization resembles a picture with a low level of quantization. In the case of an ensemble model, stabilized by cross-validation, actions are taken to reduce this property in order to develop as continuous a border as possible. As may be seen in the illustrations, compensation occurs, although in the case of KNN and dKNN leads to some disturbances in the decision boundary (interpreted as thresholding the predicted label value), and for the DTR case, despite the general correctness of the performed decisions, it generates image artifacts. Such a model may still retain high predictive ability, but it has too much tendency to overfit and leads to insufficient continuity of the optimized function to perform effective optimization.

Clear decision boundaries are implemented by both the LIN and MLP approaches. However, it is necessary to reject LIN from processing due to the linear nature of the prediction, which (i) in each optimization will lead to the selection of the extreme value of the analyzed range and (ii) is not compatible with the distribution of the explained variable and must have the largest error in each of the optimas.

Summing up the observations of Experiments 1 and 2, the MLP algorithm was chosen as the base model for the optimization task. It is characterized by (i) statistically best predictive ability among the methods analyzed and (ii) the clearest decision function from the perspective of the optimization task.

Table 2: Results of Experiment 3 for DT12 (*min* Y-R is minimum observed BBP (in the dataset), *min* Y-E is minimum estimated BBP, Y-MC(*g*) is BBP estimation for ATD found using regression model and *Monte Carlo* with *g* guesses.

| DATASET  | MIN Y-R | MIN Y-E | Y-MC(G=...) |         |         |         |         |         | $r^2$ |
|----------|---------|---------|-------------|---------|---------|---------|---------|---------|-------|
|          |         |         | 10          | 100     | 1k      | 10k     | 100k    | 1M      |       |
| XT1-S0.4 | 0.0062  | 0.0075  | 0.0085      | 0.0076  | 0.0075  | 0.0074  | 0.0074  | 0.0074  | 0.731 |
| XT1-S0.6 | 0.0001  | 0.0003  | 0.0003      | 0.0001  | 0.0001  | 0.0001  | 0.0001  | 0.0001  | 0.933 |
| XT1-S0.8 | 0.0000  | 0.0001  | 0.0001      | -0.0000 | -0.0002 | -0.0002 | -0.0002 | -0.0002 | 0.971 |
| XT1-S1.0 | 0.0011  | 0.0010  | 0.0024      | 0.0012  | 0.0008  | 0.0008  | 0.0007  | 0.0007  | 0.965 |
| XT1-S1.2 | 0.0030  | 0.0039  | 0.0042      | 0.0038  | 0.0035  | 0.0034  | 0.0034  | 0.0034  | 0.899 |
| XT1-S1.4 | 0.0042  | 0.0050  | 0.0053      | 0.0050  | 0.0048  | 0.0047  | 0.0047  | 0.0047  | 0.966 |
| XT1-S1.6 | 0.0024  | 0.0026  | 0.0036      | 0.0025  | 0.0024  | 0.0024  | 0.0024  | 0.0024  | 0.976 |
| XT1-S1.8 | 0.0034  | 0.0044  | 0.0046      | 0.0029  | 0.0027  | 0.0027  | 0.0026  | 0.0026  | 0.987 |
| XT1-S2.0 | 0.0035  | 0.0047  | 0.0049      | 0.0040  | 0.0033  | 0.0032  | 0.0032  | 0.0032  | 0.989 |
| XT2-S0.4 | 0.0000  | -0.0002 | -0.0000     | -0.0002 | -0.0003 | -0.0003 | -0.0003 | -0.0004 | 0.951 |
| XT2-S0.6 | 0.0000  | -0.0001 | -0.0002     | -0.0003 | -0.0003 | -0.0004 | -0.0004 | -0.0004 | 0.963 |
| XT2-S0.8 | 0.0006  | 0.0009  | 0.0010      | 0.0009  | 0.0008  | 0.0008  | 0.0008  | 0.0008  | 0.971 |
| XT2-S1.0 | 0.0037  | 0.0041  | 0.0068      | 0.0046  | 0.0039  | 0.0038  | 0.0038  | 0.0038  | 0.954 |
| XT2-S1.2 | 0.0047  | 0.0050  | 0.0078      | 0.0063  | 0.0042  | 0.0042  | 0.0042  | 0.0041  | 0.922 |
| XT2-S1.4 | 0.0048  | 0.0059  | 0.0083      | 0.0068  | 0.0055  | 0.0054  | 0.0054  | 0.0054  | 0.965 |
| XT2-S1.6 | 0.0059  | 0.0065  | 0.0091      | 0.0074  | 0.0063  | 0.0061  | 0.0061  | 0.0061  | 0.982 |
| XT2-S1.8 | 0.0073  | 0.0080  | 0.0102      | 0.0085  | 0.0074  | 0.0072  | 0.0071  | 0.0071  | 0.989 |
| XT2-S2.0 | 0.0060  | 0.0066  | 0.0086      | 0.0066  | 0.0061  | 0.0057  | 0.0057  | 0.0056  | 0.988 |

Table 3: Results of experiment 3 for *Euro28* (*min* Y-R is minimum observed BBP (in the dataset), *min* Y-E is minimum estimated BBP, Y-MC(*g*) is BBP estimation for ATD found using regression model and *Monte Carlo* with *g* guesses.

| DATASET           | MIN Y-R | MIN Y-E | Y-MC(G=...) |         |         |         |         |         | $r^2$ |
|-------------------|---------|---------|-------------|---------|---------|---------|---------|---------|-------|
|                   |         |         | 10          | 100     | 1K      | 10K     | 100K    | 1M      |       |
| <i>XT1-S0.104</i> | 0.0023  | 0.0030  | 0.0056      | 0.0032  | 0.0024  | 0.0023  | 0.0022  | 0.0022  | 0.979 |
| <i>XT1-S0.156</i> | 0.0055  | 0.0068  | 0.0076      | 0.0068  | 0.0066  | 0.0065  | 0.0064  | 0.0064  | 0.989 |
| <i>XT1-S0.208</i> | 0.0075  | 0.0091  | 0.0093      | 0.0090  | 0.0088  | 0.0088  | 0.0087  | 0.0087  | 0.970 |
| <i>XT1-S0.260</i> | 0.0033  | 0.0048  | 0.0048      | 0.0043  | 0.0043  | 0.0042  | 0.0042  | 0.0042  | 0.891 |
| <i>XT1-S0.312</i> | 0.0016  | 0.0027  | 0.0027      | 0.0024  | 0.0024  | 0.0023  | 0.0023  | 0.0023  | 0.971 |
| <i>XT1-S0.364</i> | 0.0052  | 0.0063  | 0.0100      | 0.0063  | 0.0055  | 0.0053  | 0.0053  | 0.0053  | 0.968 |
| <i>XT1-S0.416</i> | 0.0053  | 0.0057  | 0.0094      | 0.0063  | 0.0054  | 0.0054  | 0.0053  | 0.0053  | 0.969 |
| <i>XT1-S0.468</i> | 0.0055  | 0.0068  | 0.0095      | 0.0070  | 0.0064  | 0.0061  | 0.0061  | 0.0061  | 0.983 |
| <i>XT1-S0.520</i> | 0.0050  | 0.0058  | 0.0080      | 0.0060  | 0.0057  | 0.0056  | 0.0056  | 0.0056  | 0.988 |
| <i>XT2-S0.104</i> | 0.0015  | 0.0023  | 0.0033      | 0.0023  | 0.0016  | 0.0015  | 0.0015  | 0.0014  | 0.971 |
| <i>XT2-S0.156</i> | 0.0078  | 0.0092  | 0.0106      | 0.0090  | 0.0090  | 0.0084  | 0.0084  | 0.0084  | 0.993 |
| <i>XT2-S0.208</i> | 0.0006  | 0.0013  | 0.0005      | 0.0003  | 0.0004  | 0.0003  | 0.0002  | 0.0002  | 0.951 |
| <i>XT2-S0.260</i> | 0.0000  | 0.0000  | 0.0001      | -0.0000 | -0.0001 | -0.0001 | -0.0001 | -0.0001 | 0.964 |
| <i>XT2-S0.312</i> | 0.0002  | 0.0005  | 0.0005      | 0.0003  | 0.0003  | 0.0003  | 0.0003  | 0.0003  | 0.977 |
| <i>XT2-S0.364</i> | 0.0030  | 0.0034  | 0.0054      | 0.0041  | 0.0033  | 0.0033  | 0.0033  | 0.0033  | 0.957 |
| <i>XT2-S0.416</i> | 0.0038  | 0.0043  | 0.0060      | 0.0048  | 0.0040  | 0.0040  | 0.0039  | 0.0039  | 0.975 |
| <i>XT2-S0.468</i> | 0.0050  | 0.0052  | 0.0078      | 0.0063  | 0.0053  | 0.0052  | 0.0051  | 0.0051  | 0.986 |
| <i>XT2-S0.520</i> | 0.0064  | 0.0079  | 0.0087      | 0.0073  | 0.0069  | 0.0064  | 0.0063  | 0.0063  | 0.990 |

### 5.3 Experiment 3

The last experiment focuses on the finding of best ATD vector based on the constructed regression model. To this end, we use *Monte Carlo* method with different number of guesses. Tables 2 and 3 present the obtained results as a function of number of guesses, which changes from  $10^1$  up to  $10^9$ . The results quality increases with the number of guesses up to some threshold value. Then, the results do not change at all or change only a little bit. According to the presented values, *Monte Carlo* method applied with  $10^3$  guesses provides satisfactory results. We therefore recommend that value for further experiments.

## 6 Conclusions

The following work has considered the topic of employing pattern recognition methods to support SS-FON optimization process. For a wide pool of generated cases, analyzing two real network topologies, the effectiveness of solutions implemented by five different, typical regression methods was analyzed, starting from *Logistic Regression* and ending with *neural networks*.

Conducted experimental analysis shows, with high probability obtained by conducting proper statistical validation, that MLP is characterized by the greatest potential in this type of solutions. Even with a relatively small pool of input simulations, constructing a data set for learning purposes, interpretable in both

the space of optimization and machine learning problems, simple networks of this type achieve both high quality prediction measured by the  $r^2$  metric, and continuous decision space creating the potential for conducting optimization.

Basing the model on the stabilization realized by using ensemble of estimators additionally allows to reduce the influence of noise on optimization, which – in a *state-of-art* optimization methods – could show a tendency to select invalid optimas, burdened by the nondeterministic character of the simulator. Further research, developing ideas presented in this article, will focus on the generalization of the presented model for a wider pool of network optimization problems.

## Acknowledgements

The work of Paweł Ksieniewicz, Róża Goścień, and K. Walkowiak was supported by National Science Centre, Poland under Grant 2017/27/B/ST7/00888. The work of Mirosław Klinkowski was supported by National Science Centre, Poland under Grant No. 2018/31/B/ST7/03456.

## References

1. Awaji, Y., Sakaguchi, J., Puttnam, B.J., Luis, R.S., Mendinueta, J.M.D., Klaus, W., Wada, N.: High-capacity transmission over multi-core fibers. *Opt. Fiber Technol.* **35**, 100–107 (2017)
2. Boutaba, R., Salahuddin, M.A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., Caicedo, O.M.: A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities. *J. of Int. Serv. and Appl.* **9**(1), 16 (Jun 2018). <https://doi.org/10.1186/s13174-018-0087-2>
3. CISCO: Cisco Visual Networking Index: Forecast and Trends, 2017–2022 . Tech. rep. (2019)
4. Gerstel, O., Jinno, M., Lord, A., Yoo, S.J.B.: Elastic optical networking: A new dawn for the optical layer? *IEEE Comm. Mag.* **50**(2), 12–20 (Feb 2012)
5. Goścień, R.: On the efficient dynamic routing in spectrally-spatially flexible optical networks. In: 2019 11th International Workshop on Resilient Networks Design and Modeling (RNDM). pp. 1–8 (Oct 2019)
6. Goścień, R., Lechowicz, P.: On the complexity of RSSA of anycast demands in spectrally-spatially flexible optical networks. In: 9th International Network Optimization Conference Avignon, France, June 12–14, 2019. p. 7 (2019)
7. Klinkowski, M., Ksieniewicz, P., Jaworski, M., Zalewski, G., Walkowiak, K.: Machine learning assisted optimization of dynamic crosstalk-aware spectrally-spatially flexible optical networks. *Journal of Lightwave Technology* pp. 1–1 (2020). <https://doi.org/10.1109/JLT.2020.2967087>
8. Klinkowski, M., Lechowicz, P., Walkowiak, K.: Survey of resource allocation schemes and algorithms in spectrally-spatially flexible optical networking. *Opt. Switch. and Netw.* **27**, 58–78 (2018)
9. Ksieniewicz, P., Woźniak, M., Cyganek, B., Kasprzak, A., Walkowiak, K.: Data stream classification using active learned neural networks. *Neurocomputing* **353**, 74–82 (2019)

10. Mata, J., de Miguel, I., Durán, R.J., Merayo, N., Singh, S.K., Jukan, A., Chamania, M.: Artificial intelligence (AI) methods in optical networks: A comprehensive survey. *Opt. Switch. and Netw.* **28**, 43 – 57 (2018). <https://doi.org/https://doi.org/10.1016/j.osn.2017.12.006>
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. Surv. and Tutorials* (2018)
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011)
13. Rafique, D., Velasco, L.: Machine learning for network automation: Overview, architecture, and applications [invited tutorial]. *IEEE/OSA J. of Opt. Commun. and Netw.* **10**(10), D126–D143 (Oct 2018). <https://doi.org/10.1364/JOCN.10.00D126>
14. Saridis, G.M., Alexandropoulos, D., Zervas, G., Simeonidou, D.: Survey and evaluation of space division multiplexing: From technologies to optical networks. *IEEE Commun. Surv. and Tutorials* **17**(4), 2136–2156 (2015)
15. Walkowiak, K.: *Modeling and Optimization of Cloud-Ready and Content-Oriented Networks*, Studies in Systems, Decision and Control, vol. 56. Springer (2016)
16. Zyblewski, P., Ksieniewicz, P., Woźniak, M.: Classifier selection for highly imbalanced data streams with minority driven ensemble. In: *International Conference on Artificial Intelligence and Soft Computing*. pp. 626–635. Springer (2019)