# Self-Adaptive Erbium-Doped Fiber Amplifiers Using Machine Learning

Erick de A. Barboza
and Carmelo J. A. Bastos-Filho
Polytechnic School of Pernambuco
University of Pernambuco
Recife, Brazil
Email: carmelofillho@ieee.org

Joaquim F. Martins-Filho
Department of Electronics and Systems
Federal University of Pernambuco
Recife, Brazil
Email: jfmf@ufpe.br

Uiara C. de Moura
and Juliano R. F. de Oliveira
CPqD Telecom R&D Center
Campinas, Brasil
Email: {umoura,jrfo}@cpqd.com.br

***Abstract*—This paper presents a method to autonomously adjust the operating point of amplifiers in a cascade using an approach based on machine learning. The goal is to smoothly adjust the gain of each amplifier in the cascade in order to reach predefined input and output power levels for the entire link, aiming to minimize both the noise figure and the gain flatness of the transmission system. The proposal uses an iterative method and performs feedforward and backward error adjustments based on local information. The experimental results indicate that our proposal can optimize the performance of the link ensuring predefined input and output power levels, which is important in a network scenario. As an example, our proposal was capable to define the gain of $6$ amplifiers returning a link with a noise figure equal to $30.06$ dB and a gain flatness equal to $5.26$ dB, while maintaing the input and output powers around $3$ dBm with an error lower than $0.1$ dB.**

***Index Terms*—Optical Amplifiers, Noise Figure, Machine Learning, Self-adaptation, Backpropagation.**

## I. Introduction

The growth of the traffic demand generated by recent released Internet video-on-demand and cloud computing services has driven optical networks to evolve by incorporating advanced modulation formats and reconfigurable devices. In this new scenario, it is desirable to have devices that can self-adapt their operating points considering different situations aiming to achieve an acceptable Quality of Transmission (*QoT*) for every active lightpath of the network.

Some adaptive and cognitive approaches were recently proposed to simplify and automate the configuration of devices and networks [1]–[4]. These approaches comprise on the fly adjustments of the operating point regarding the past experiences. In general, these configurations are performed based on transmission metrics, such as Noise Figure (*NF*), Optical Signal to Noise Ratio (*OSNR*), Frequency Response Flatness (*FRF*) or Bit Error Rate (*BER*). These metrics are important since they are directly related to the *QoT* of the lightpaths.

Optical amplifiers are often deployed in optical systems and have a crucial impact on the overall *QoT* of the lightpath. It occurs mainly because the Average Gain, *NF* and Gain Flatness (*GF*) of the amplifier depend on the total input power of the amplifier. The Erbium-Doped Fiber Amplifier (*EDFA*) is the most used type of amplifier in optical communications systems. *EDFA*s add noise and have a non-flat gain spectrum. The *NF* and the *GF* of the amplifier depend on the operating point, which also depends on internal parameters, such as the pump power. Moreover, the input power can vary during the network operation, specially in dynamic optical networks, in which lightpaths can be added or dropped along the time. Therefore, the proper adjustment of the operating point of the amplifiers can lead to better *QoT* of the lightpaths.

The adjustments of the operating point of the devices can be done in two different manners: using a single step, in which the device adjusts its parameters toward its own best performance point, or using an iterative approach, which allows a smooth variation of the operating point of the devices aiming to improve the *QoT* of the entire lightpath. It is easier to adjust the operating point of the amplifiers using a single step based on static information, such as *NF* and *GF*. In this line, one approach was recently introduced in [5]. This proposal presented an scheme to change the *EDFA* operating point in a cascade using a single step method based on power masks in a feed-forward manner. This concept was evaluated in terms of *BER* measurements for a dynamic optical link with four C-band channels constrained by some power impairments. Although this approach presented good preliminary results, it can not guarantee the best link performance and can not maintain the input and output powers of the link in a predetermined level, which is important to apply it in network scenarios.

In this paper, we present a novel approach for self-adaptive amplifiers based on a machine learning process, that varies the operating point according to the power masks of the amplifiers through an iterative update process. The operating point of each amplifier is updated locally with a controlled step using the power mask of the amplifier and the current input and output powers. The step decreases along the iterations to guarantee convergence. The update process is performed in a loop through the cascade of amplifiers in a recursive way, using a loop of feedforward and backpropagation adjustments, aiming to obtain the global optimization of the entire lightpath. Our proposal aims to attend to the input and output powers restriction.

The remainder of the paper is organized as follows. In

978-1-4799-1397-8/13/$31.00 ©2013 IEEE

Section II and III, we present some basic concepts regarding adaptive amplifiers and machine learning, respectively. Section IV presents our proposal for iterative self-adaptive optical amplifiers. Section V and VI present the simulation setup using data from real *EDFA*s and some results, respectively. Some conclusions and future works are presented in Section VII.

## II. Adaptive Amplifier

The adaptive amplifier concept is applied to the EDFA in order to adjust its operating point when some input power change occurs. This adjustment aims to achieve the best trade-off between amplifier *NF* and *GF*. The alteration of the operating point of adaptive amplifier is based on a previous characterization process, in which *NF* and *GF* are measured for some operating points inside the power mask [6]. Fig. 1 presents an example in which the objective space with the operating points measured previously are depicted as a function of the *NF* and the *GF* of the amplifier. The objective function values are coded in the right color bar.

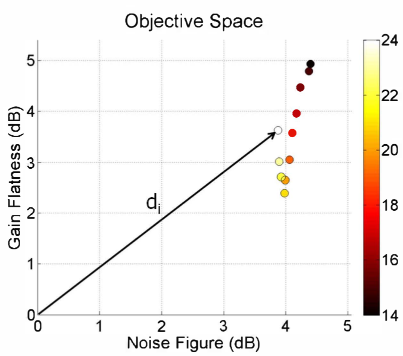



Fig. 1: Objective space illustration with some operating points for a specific input power.

Fig. 1 also illustrates a metrics used in this paper, the distance $d_i$ of the operating point to the origin in the objective space. It gives a general quality measurement and for this paper it is given by (1). $d_i$ is deployed in order to simultaneously optimize *NF* and *GF*. In this case, the adaptive amplifier selects the operating point that has the lowest distance $d_i$ to be the target amplifier operating point.

$$d_i = \sqrt{NF^2 + GF^2}. \tag{1}$$

## III. Machine Learning

Machine Learning is a computational intelligence subfield that aims to provide self-adaptation capabilities to computational processes through learning methods [7]. In this context, Learning regards to complex pattern recognition and intelligent decision making. Machine Learning has been widely applied in different areas [8]–[10].

One of the most traditional learning techniques is the delta rule. This rule was developed by Widrow and Hoff in 1969 [11]. It is an example of a supervised learning that uses the method of gradient descent to minimize the error between the actual output of the system and the desired output. The most known application of this technique is to train ADALINE neural networks [12]. The delta rule is given by the formula depicted in (2).

$$\Delta w_{ij} = \alpha(d_i - y_i)x_j f'(net_i), \tag{2}$$

in which the $\alpha$ is the learning rate, $w_{ij}$ is the weight between the neuron $i$ and the neuron $j$, $d_i$ is the desired output, $y_i$ is the current output, $x_j$ is the input variable value and $f'(net_i)$ is the derivative of the output function.

If the output function is linear, its derivative is equal to 1. In this case, equation (2) can be simplified and one can calculate the new value for the weight between the neuron $i$ and the neuron $j$ using (3).

$$w_{ij}(new) = w_{ij}(old) + \alpha(d_i - y_i)x_j. \tag{3}$$

Backpropagation (*BP*) is another algorithm that is used to train neural networks. This algorithm is a generalization of the delta rule, which is used to train a Multi-Layer Perceptron networks (MLP). This generalization was made because it is necessary to propagate the error recursively through the layers of the MLP. *BP* is basically divided in two steps. In the first step, the values are propagated from the input to the output (feedforward process), and then the error between the desired output and the current output is calculated. In the second step, the errors are propagated recursively from the output to the input (backward process) and the weights of the network are updated.

## IV. Proposed Approach

Our approach is based on the backpropagation algorithm, but we use the propagation of the error in two directions (feedforward and backward) to update the operating point of the amplifiers in the cascade aiming to reach predefined power levels in the input and output of the link. We defined a maximum step to adjust the operating point of the amplifiers in order to find the configuration for every amplifier that returns the best performance for the entire link. Our hypothesis is that the best operating point of the amplifier is not necessarily the best option for the entire link. Therefore, an iterative method may allow a compromise between the best amplifier operating point and the best operating point for the neighbor amplifiers.

The pseudocode of the proposed approach is depicted in Algorithm 1. Some specific parts of our proposals are detailed in the subsections IV-A, IV-B, IV-C and IV-D. $P_{in}(i)$ and $P_{out}(i)$ are the input and output powers of the amplifier $i$, respectively. $P_{in}^{desired}$ and $P_{out}^{desired}$ are the predefined input and output power levels for the link. One can observe that we have a main loop that contains two inner loops, one for the feedforward correction and another one for the backward correction. This main loop is executed a predefined number of times and each execution is performed with a step. In Machine Learning, it is quite common to use annealing processes to refine learning and increase its exploitation capability along the time. Therefore, we decrease the step along the iterations to guarantee convergence to a good configuration for the cascade.

**Algorithm 1**: Pseudocode of the Proposed Approach for the Self-Adaptive EDFA.

```
Initialize the amplifiers (detailed in Subsection IV-A);
Error_input = Error_output = 0;
while The total number of iterations was not reached do
    foreach Amplifier 1 to n − 1 do
        if First amplifier then
            P_out^new(1) = P_out^old(1) − Error_input;
        else
            Select a set of amplifier operating points with P_in
            near to the P_in of the current amplifier (detailed in
            Subsection IV-B);
            Select the target operating point, within this set,
            that has the lowest d_i as shown in equation (1);
            Calculate the step (detailed in Subsection IV-C);
            P_out^new(i) = P_out^old(i) + step;
        Define the new Gain, NF and GF of this
        amplifier (detailed in Subsection IV-D);
        Calculate the input power of the next amplifier
        considering the losses using
        P_in(i + 1) = P_out^new(i) − Loss(i);
    Error_output = P_out(n) − P_out^desired;
    foreach Amplifier n to 2 do
        if Last Amplifier then
            P_in^new(n) = P_in^old(n) − Error_output;
        else
            Select a set of operating points with P_out near to
            the P_out of the current amplifier (detailed in
            Subsection IV-B);
            Select the target operating point, within this set,
            that has the lowest d_i as shown in equation (1);
            Calculate the step (detailed in Subsection IV-C);
            Calculate the output power of the next amplifier
            considering the losses using
            P_in^new(i) = P_in^old(i) + step;
        Define the new Gain, NF and GF of this
        amplifier (detailed in Subsection IV-D);
        P_out(i − 1) = P_in^N(i) + Loss(i);
    Error_input = P_in(0) − P_in^desired;
Return the gain of each amplifier.
```

### *A. Initialization Strategy*

All amplifiers are initialized with the average gain. This average gain is calculated according to (4)

$$G_a = \frac{\sum_i^{n+1} Loss(i)}{n}, \tag{4}$$

where $Loss(i)$ is the link loss between the amplifier $i$ and the amplifier $i+1$, and $n$ is the number of amplifiers.

### *B. Selecting the Reference Amplifier Operating points*

To select the reference amplifier operating points, the input power $P_{in}$ (or the output power $P_{out}$ in the backward process), of the current amplifier is used to select points in the power mask that has values of $P_{in}$ (or $P_{out}$ in the backward process) near to the current operating point. Suppose a power mask that contains information about the amplifier with a resolution of 0.5. Therefore, an amplifier operating with $P_{in} = 0$ dB can only select the following reference points $P_{in} = -0.5$ dB and $P_{in} = 0.5$ dB. In our case, we use an interpolator to define the power mask values. One must observe that the reference operating points must be within the objective space in Fig. 1.

### *C. Calculating the Step Size*

The calculation of the step size is the most important procedure in the optimization process. Our approach uses an annealing concept [13] to change the size of the step along the iterations. The mainly idea is to decrease the step in order mitigate a selfish behavior of the amplifier. The operating point of the amplifier is adjusted aiming at reaching the target amplifier operating point (*i.e.* the point with the lowest $d_i$ in the set of reference operating points). The step decreases along the iteration according to (5).

$$step = (e^{-T} - e^{-1}) \cdot [P_{in}(i) - P_{in}(ref)], \tag{5}$$

where the $P_{in}(i)$ is the input power of the amplifier $i$ and $P_{in}(ref)$ is the input power of the reference operating point. The same equation is used to update the output power by substituting $P_{in}$ by $P_{out}$.

The parameter $T$ is the temperature factor of the annealing process. $T$ is evaluated according to

$$T = \frac{iteration_{current}}{iteration_{total}}, \tag{6}$$

in which $iteration_{current}$ is the number of the current iteration and $iteration_{total}$ is the total number of iterations. Therefore, $T$ will increase along the iterations, reaching $T = 1$ in the last iteration. Thus, the step depicted in (5) will decrease along the iterations, reaching $step = 0$ in the last iteration.

### *D. Defining the Metrics*

The proposed approach is guided by a fitness function that depends on the *NF* and the *GF*, and must be defined for any pair of input and output powers within the operating range. We can calculate this fitness function using the Power Mask. In general, the Power Mask is build with samples from a characterization process [6]. The characterization process generates a discrete space and sometimes the resolution is not suitable to tackle the selected target operating points.

In order to overcome this limitation, we developed an interpolation method to generate a continuous Power Mask from the discrete data. We used a multilayer perceptron (*MLP*) neural network to accomplish it [14]. Fig. 2 represents the *MLP* architecture used in this work. The *MLP* has two inputs, the amplifier input power and amplifier output power, and returns two values according to the inputs, the *NF* and *GF*. This *MLP* has 4 neurons in the hidden layer, and was trained with the data of the power mask using the back propagation algorithm during $5,000$ epochs. We observed from the study that the error in the interpolation process is lower than 0.1 dB. We just used 4 neurons in the hidden layer because we observed that more neurons do not lead to a better performance.

As a consequence of the novel interpolation method, we can use any value for the amplifier input power and the amplifier output power. Thus, we have continuous values for *NF* and *GF* to evaluate $d_i$ according to equation (1). This new scenario in less restrictive when compared to the approach presented in [5].

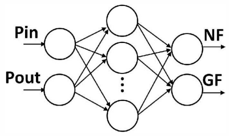



Fig. 2: *MLP* used to interpolate the points of the power mask.

## V. Simulation Setup

We used the same scenario presented in [5] to perform our simulations. This scenario considers 6 amplifiers. Amplifiers 1 and 6 are boosters with maximum output power equal to 24 dBm. Amplifiers 2 and 4 are line amplifiers, which work as boosters and have maximum output power equal to 21 dBm. The amplifier 5 are similar to amplifiers 2 and 4, but its maximum output power is 24 dBm. The amplifier 3 is a line amplifier that works as a pre-amplifier and has maximum output power equal to 21 dBm. The losses between the amplifiers are: 17.22 dB (between amplifiers 1 and 2), 28.19 dB (between amplifiers 2 and 3), 13 dB (between amplifiers 3 and 4), 22.58 dB (between amplifiers 4 and 5) and 28.74 dB (between amplifiers 5 and 6).

The $NF$ of the link is calculated according to the equation defined in [15]. This equation defines the noise factor of an amplifier cascade as:

$$F = F_1 + \frac{F_2}{G_1 L_1} + ... + \frac{F_N}{G_1 L_1 G_2 L_2 ... L_{N-1}}, \quad (7)$$

in which $F_i$ is the noise factor of the amplifier $i$, $G_i$ is the gain of this amplifier and $L_i$ is the loss of the span between the amplifiers $i$ and $i+1$. The noise figure is calculated from the noise factor according to (8):

$$NF = 10log(F). \quad (8)$$

The *GF* of the link is obtained by calculating the arithmetic average of the *GF* of every amplifier in the cascade.

We performed simulations varying the number of iterations of the optimization process in order to evaluate the best number of iteration to be used as the stop criterion. We also analyzed the influence of the input/output power restriction by performing simulations for different values of $P_{in}^{desired}$. In all cases, we set $P_{out}^{desired}$ equal to $P_{in}^{desired}$. The rationale for this is to maintain the power levels when cascading links along and hypothetic network.

## VI. Results

In this section we present the simulations results. The results are depicted in terms of the error regarding the predefined desired link input power ($Error_{input}$), the error regarding the predefined desired link output power ($Error_{output}$), the *NF* of the link and the *GF* of the link.

### A. Parametrical Analysis

In this Subsection we perform a parametrical analysis considering the influence of the deployed number of iterations and the influence of the algorithm performance regarding the predefined desired link input and output powers.

Table I shows the values for the errors and the metrics (*NF* and *GF*) for different numbers of iterations and $P_{in}^{desired} = P_{out}^{desired} = 3$ dBm. One can observe that 10 iterations is not enough to obtain low errors, specially for the input. On the other hand, it is not worth to use more than 50 iterations because the results are similar. Therefore, we will use 50 iterations in the rest of the paper.

TABLE I: Comparison of the Errors and the metrics for different number of iterations.

| Iteration | $Error_{input}$ | $Error_{output}$ | NF | GF |
|---|---|---|---|---|
| 10 | 2.03 dB | 0.37 dB | 22.09 dB | 4.08 dB |
| 20 | 0.19 dB | 1.00 dB | 27.93 dB | 5.67 dB |
| **50** | **0.08 dB** | **0.06 dB** | **30.06** dB | **5.97** dB |
| 100 | 0.03 dB | 0.04 dB | 30.06 dB | 6.01 dB |
| 200 | 0.02 dB | 0.01 dB | 30.12 dB | 5.98 dB |

One can observe from Table I that the *NF* of the link is lower for a lower number of iteration. In order to analyze the reason for this, we show in Figures 3a and 3b the values of *NF* and *GF*, and $Error_{input}$ and $Error_{output}$, along the iterations for 50 iterations and $P_{in}^{desired} = 3$ dBm. As one can observe during the first iterations, the *NF* of the link is lower, but the link is not satisfying the input/output restriction. To obtain errors as low as 0.08 dB and 0.06 dB, the process drives the *NF* and *GF* to 30.06 dB and 5.97 dB, respectively. However, if the restriction is not required, one can achieve *NF* and *GF* around 18.8 dB and 3.2 dB, respectively (see iteration number 7).

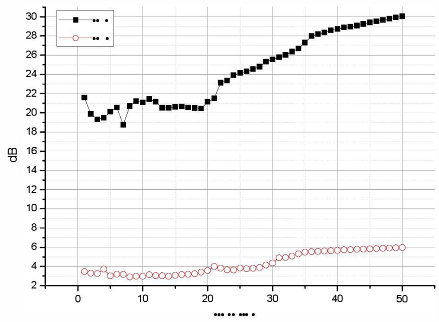
It is also important to analyze if the proposal works for different values of $P_{in}^{desired}$. Table II shows the results for $P_{in}^{desired} = P_{out}^{desired}$ equal to $-3$ dBm, 0 dBm and 3 dBm. One can observe that low errors were obtained in all cases.

TABLE II: Comparison of the Errors and the metrics for different Input/output power levels.

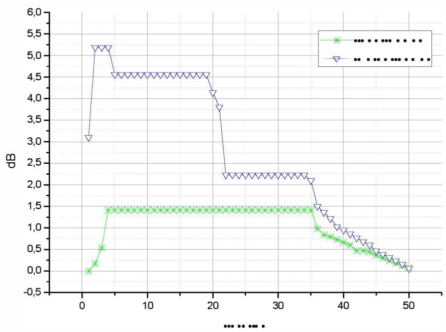
| $P_{in}^{desired}$ | $Error_{output}$ | $Error_{input}$ | NF | GF |
|---|---|---|---|---|
| - 3 dBm | 0.06 | 0.08 | 30.22 dB | 6.11 dB |
| 0 dBm | 0.03 | 0.08 | 27.69 dB | 5.22 dB |
| 3 dBm | 0.06 | 0.08 | 30.06 dB | 5.94 dB |

### B. Comparison to the Previous Approach

Table III shows the characteristics of each amplifier in the cascade returned by our approach, whereas Table IV shows the same information returned by the approach used in [5] for the same simulation setup. The link returned by the previous




(a) Metrics




(b) Errors

Fig. 3: Evolution of (a) *NF* and *GF*, and (b) $Error_{input}$ and $Error_{output}$.

approach presents *NF* and *GF* equal to 22.3 dB and 3.55 dB, but this approach does not take in consideration the input/output power restriction. The previous approach returned $Error_{input} = 0.19$ dB and $Error_{output} = 8.07$ dB (regarding 3 dBm), whereas our approach returned $Error_{input} = 0.08$ dB and $Error_{output} = 0$ dB. It is important to observe that if the input/output power restriction is relaxed, we obtained $NF = 18.8$ dB and $GF = 3.2$ dB, $Error_{input} = 1.4$ dB and $Error_{output} = 4.6$ dB, which is better than the previous approach.

TABLE III: Amplifiers characteristics returned by our approach.

| | $P_{in}$ (dBm) | Gain (dB) | $P_{out}$ (dBm) | NF (dB) | GF (dB) |
|---|---|---|---|---|---|
| Amp1 | **2.92** | 24.00 | 26.92 | 5.15 | 3.78 |
| Amp2 | 9.70 | 14.00 | 23.70 | 15.90 | 8.15 |
| Amp3 | -4.49 | 14.00 | 9.51 | 6.61 | 4.81 |
| Amp4 | -3.49 | 17.41 | 13.92 | 7.29 | 5.24 |
| Amp5 | -8.66 | 16.40 | 7.74 | 5.41 | 9.64 |
| Amp6 | -21.00 | 24.00 | **3.00** | 5.64 | 4.18 |

## VII. Conclusion and Future Works

We proposed in this paper a novel approach for self-adaptive *EDFA*s based on machine learning, that varies the operating point according to the power masks of the amplifiers through an iterative update process. The operating point is updated locally with a controlled step in order to avoid a selfish behavior of the amplifier and minimize the Noise Figure and Ripple of the Frequency Response of the entire link.

TABLE IV: Amplifiers characteristics returned by the approach proposed in [5].

| | $P_{in}$ (dBm) | Gain (dB) | $P_{out}$ (dBm) | NF (dB) | GF (dB) |
|---|---|---|---|---|---|
| Amp1 | **2.81** | 21.00 | 23.98 | 5.69 | 1.53 |
| Amp2 | 7.21 | 14.00 | 20.96 | 15.49 | 8.37 |
| Amp3 | -6.82 | 15.00 | 8.12 | 5.70 | 4.71 |
| Amp4 | -4.95 | 24.00 | 19.14 | 6.30 | 2.42 |
| Amp5 | -2.97 | 23.00 | 20.06 | 4.64 | 3.75 |
| Amp6 | -9.09 | 20.00 | **11.07** | 5.49 | 0.51 |

The results show that our approach is capable to define the gain of the amplifiers, considering the restriction in which the input power and the output power of the link can be predefined. This restriction leads to an increasement in the Noise Figure and Ripple of the link, but can be obeyed if our approach is deployed. If this restriction is relaxed, we can obtain a better result when compared to the previous approach. However, this restriction is essential to network scenarios.

For future works, we intend to add the energetic consumption of the amplifier as another performance metrics and use a variable optical attenuator (VOA) in the output of the link in order to simplify the restriction regarding the desired output power level.

## References


[1] Q. H. Mahmoud, *Cognitive networks*. Wiley Online Library, 2007.

[2] G. S. Zervas and D. Simeonidou, "Cognitive optical networks: Need, requirements and architecture," in *Transparent Optical Networks (ICTON), 2010 12th International Conference on*. IEEE, 2010, pp. 1–4.

[3] I. de Miguel et al., "Cognitive dynamic optical networks," in *Optical Fiber Communication Conference*. Optical Society of America, 2013.

[4] P. V. N. C. et al., "Some schemes for controlling gain dynamics of edfa chains in wavelength-routed optical networks," in *Advanced Networks and Telecommunication Systems (ANTS), 2009 IEEE 3rd International Symposium on*. IEEE, 2009, pp. 1–3.

[5] J. R. O. et al., "Demonstration of edfa cognitive gain control via gmpls for mixed modulation formats in heterogeneous optical networks," in *Optical Fiber Communication Conference*. Optical Society of America, 2013.

[6] U. C. M. et al., "Caracterizador automatizado de máscara de potência de amplificadores ópticos para redes wdm reconfiguráveis," in *XXX Brazilian Symposium on Telecommunication*, Brasília, Brazil, 2012.

[7] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 44, no. 1.2, pp. 206–226, 2000.

[8] A. Maedche and S. Staab, "Ontology learning for the semantic web," *Intelligent Systems, IEEE*, vol. 16, no. 2, pp. 72–79, 2001.

[9] Z. H. et al., "Credit rating analysis with support vector machines and neural networks: a market comparative study," *Decision support systems*, vol. 37, no. 4, pp. 543–558, 2004.

[10] I. Kononenko, "Machine learning for medical diagnosis: history, state of the art and perspective," *Artificial Intelligence in medicine*, vol. 23, no. 1, pp. 89–109, 2001.

[11] B. W. et al., "Adaptive switching circuits." 1960.

[12] M. Valena, *Fundamentos das Redes Neurais*. Livro Rpido, 2010, pp. 107–156.

[13] S. K. et al., "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[14] K. H. et al., "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[15] P. M. B. et al., *Erbium-doped fiber amplifiers: fundamentals and technology*. Academic press, 1999, ch. 7, pp. 228–234.