

ESTIMAÇÃO DA QUALIDADE DE TRANSMISSÃO EM REDES ÓPTICAS

Universidade Federal de Pernambuco
Gabriel Victor M. de O. Vital
Leonardo Didier

Introdução

A crescente complexidade das redes ópticas, projetadas para suportar uma ampla variedade de serviços, tem resultado na geração massiva de informações. A gestão eficiente desses dados torna-se crucial para otimizar o desempenho dessas redes. Nesse contexto, emerge o papel fundamental das ferramentas matemáticas avançadas, e entre elas, o aprendizado de máquina se destaca como uma abordagem promissora.

Este trabalho explora a aplicação de técnicas de datascience e machine learning para a detecção e avaliação da qualidade em redes ópticas, visando não apenas lidar com a complexidade inerente, mas também aprimorar a eficiência operacional e a entrega de serviços de alta qualidade nesse cenário dinâmico e desafiador.

Objetivo

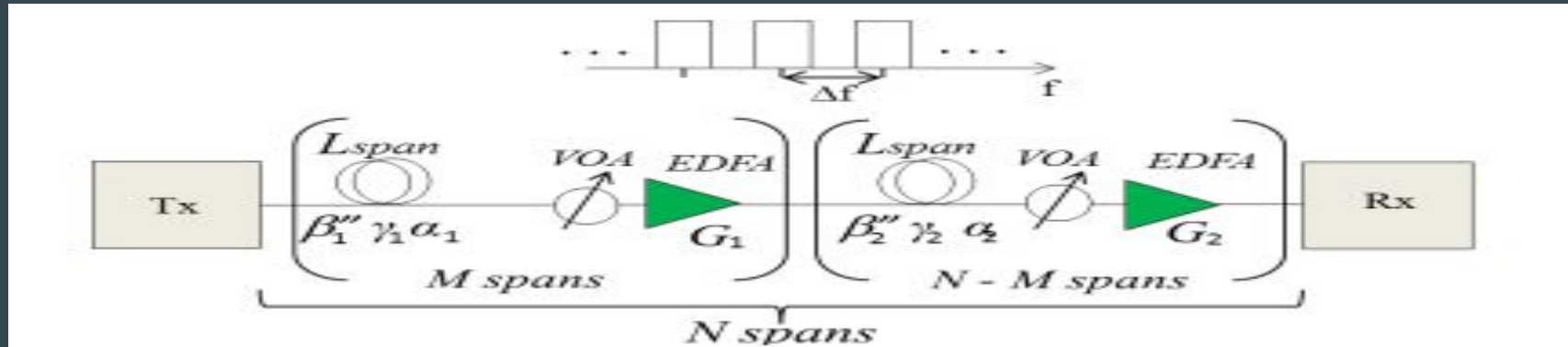
O objetivo geral deste trabalho é avaliar a eficácia de três modelos de aprendizado de máquina usados para estimar a Qualidade da Transmissão (QoT) de um caminho óptico ainda não estabelecido visando melhorar a eficiência na implementação de novos caminhos, redução de custos, otimizar o desempenho da rede e garantir a entrega de serviços com alta qualidade.

Os objetivos específicos que essa abordagem visa alcançar são:

- **Projeção de Desempenho antes da Implementação;**
- **Avaliação de Viabilidade;**
- **Otimização de Parâmetros da rede/link;**
- **Redução de Custos;**
- **Suporte à Expansão da Rede;**
- **Adaptação a Mudanças na Rede.**

Considerações sobre o escopo do trabalho

Os dados são gerados considerando links de ponta a ponta. A figura abaixo representa um link de ponta a ponta que é composta por transmissor(Tx) e receptor (Rx) nas pontas e entre ambos temos um conjunto de Fibra, amplificador.



Geração do dataset

O conjunto de dados utilizado neste trabalho foi gerado sinteticamente por meio do modelo ***Gaussian Noise (GN) de Propagação Não-Linear.***

Gaussian Noise (GN) de Propagação Não-Linear

$$SNR = \frac{P}{\sigma_{ASE}^2 + \sigma_{NLIN}^2}, \quad (1)$$

$$\sigma_{ASE}^2 = N h \nu F G B_N, \quad (2)$$

$$\sigma_{NLIN}^2 = P^3 \sum_1^{N_{ch}} \left\{ \chi_1 + \chi_2 \left(\frac{\langle |b|^4 \rangle}{\langle |b|^2 \rangle^2} - 2 \right) \right\} + P^3 \chi_3 \left(\frac{\langle |b|^6 \rangle}{\langle |b|^3 \rangle^2} - 9 \frac{\langle |b|^4 \rangle}{\langle |b|^2 \rangle^2} + 12 \right), \quad (3)$$

$$\begin{aligned} \rho_{lmn} &= \frac{\gamma}{(MT)^{3/2}} e^{-i(\omega_l + \omega_m - \omega_n)t} \\ &\times \frac{1 - e^{i\theta NL_s}}{1 - e^{i\theta L_s}} \frac{1 - e^{-\alpha L_s} e^{i\theta L_s}}{\alpha - i\theta}, \end{aligned} \quad (4)$$

$$\begin{aligned} \rho_{lmn} &= \frac{e^{-i(\omega_l + \omega_m - \omega_n)t}}{(MT)^{3/2}} \\ &\times \sum_{k=1}^N \frac{\gamma_k e^{i\theta_k L_{k-1}} (1 - e^{-\alpha_k L^k} e^{i\theta_k L^k})}{\alpha_k - i\theta_k}, \end{aligned} \quad (5)$$

Dados coletados

Os atributos listados abaixo são constantes em todo dataset e podem possivelmente servir como entrada para os algoritmos de machine learning:

- **lightspeed** - Velocidade da luz com valor de $0,299792458 \cdot 10^9$ m/s;
- **lambda** - Comprimento de onda de referência com valor de $1,55 \mu\text{m}$;
- **ChSpacing** - Espaçamento do canal com valor de 50 GHz
- **BaudRate** - Taxa de baud com valor de 28 GHz
- **Spans_DeltaPdBIntLeft** - Variação da potencia de entrada dos canais à direita do canal central, com valor 0 dBm
- **Spans_DeltaPdBIntRight** - Variação da potencia de entrada dos canais à esquerda do canal central, com valor 0 dBm
- **Spans_SpanLossdB** - Perda do Span com valor 0 dB
- **Spans_Fn** - Figura de ruído com valor de 16 dB

Dados coletados

Os atributos listados abaixo foram variados a fim de gerar uma maior quantidade e variedade de dados, para posteriormente serem utilizados como entradas para os algoritmos de machine learning.

- **NumCh** - Número de canais, com variação de 5 até 15 com passo 2;
- **NumSpans** - Número de Spans, com variação de de 5 até 15 com passo 1;
- Os parâmetros da fibra:

DispPar - Parâmetro de dispersão;

Spans_gamma - Coeficiente de não-linearidade;

Spans_alpha - Coeficiente de atenuação;

Spans_beta2 - Coeficiente de dispersão, dependente do **DisPar**.

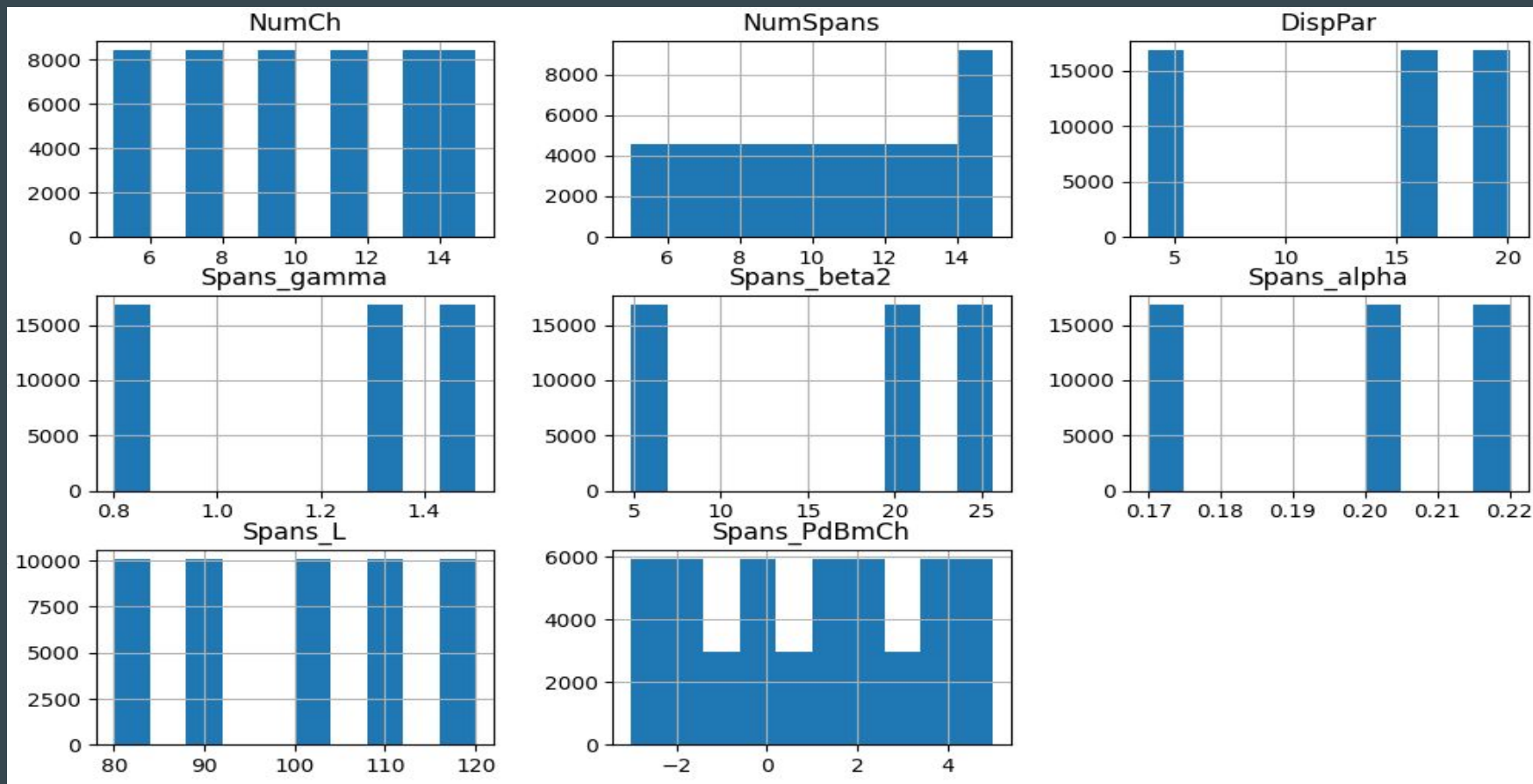
Os parâmetros listados acima tiveram uma variação em conjunto:

[(16.7, 1.3, 0.2, 21.3), (3.8, 1.5, 0.22, 4.85), (20.1, 0.8, 0.17, 25.64)] com unidades (ps/nm/km, 1/W/km, dB/km, ps²/km).

Dados coletados

- **Spans_L** - Comprimento do Span em km, com variação de 80 até 120 com passo 10 km.
- **Spans_PdBmCh** - Potência média de entrada dos canais em dBm, com variação de de -3 até 5 com passo 0,5 dBm;
- **Spans_ModFormatCh, Spans_ModFormatIntLeft e Spans_ModFormatIntRight** - Modulação do canal avaliado (central), Modulação dos canais à esquerda do canal central e Modulação dos canais à direita do canal central. As modulações consideraras variaram entre **DP-QPSK**, **DP-16QAM**, **DP-64QAM**, sem a combinação delas entre os canais.

Figura 1: Histogramas dos atributos de entrada NumCh, NumSpans, DispPar, Spans_gamma, Spans_beta2, Spans_alpha, 'Spans_L, Spans_PdBmCh



Dados coletados

Os atributos listados abaixo são resultados do modelo GN e posteriormente podem possivelmente servir como atributo target (de saída) para os algoritmos de machine learning:

- **PChOptdBm** - Potência ótima da rede em dBm.
- **NLIN_Power** - Potência do ruído com interferência não-linear em dB.
- **OSNRdB** - Relação Sinal-Ruído em dB
- **OSNRdB_NL** - Relação Sinal-Ruído em com interferência não-linear em dB.
- **BER** - Taxa de erro de bits.
- **BER_NL** - Taxa de erro de bits com interferência não-linear.

Figura 2: Histogramas da distribuição dos dados de OSNRdB, OSNRdb_NL, NLIN_Power e PChOptdBm

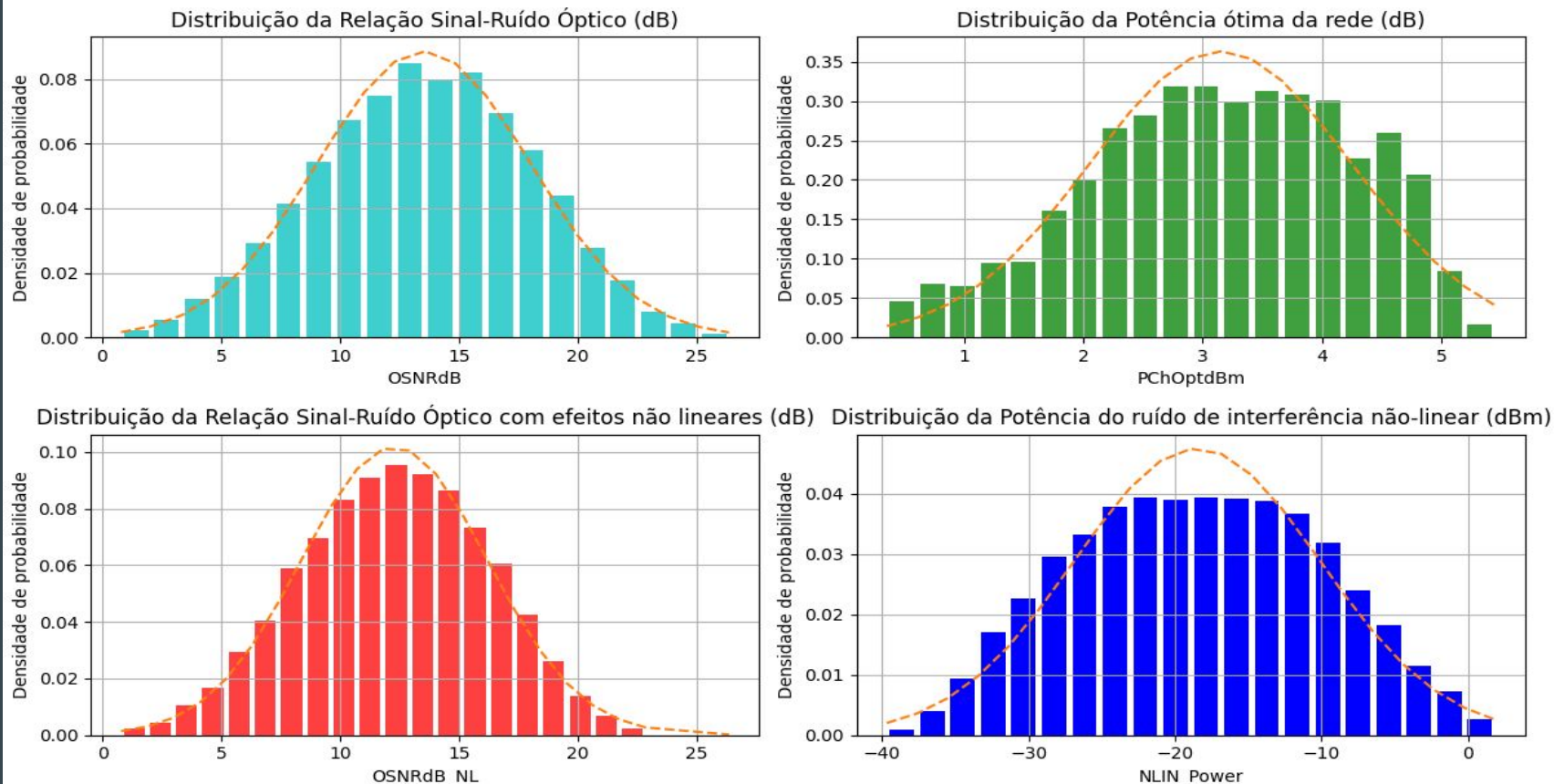


Figura 3: Curva OSNR com degradações não-lineares versus Potência de entrada dos canais

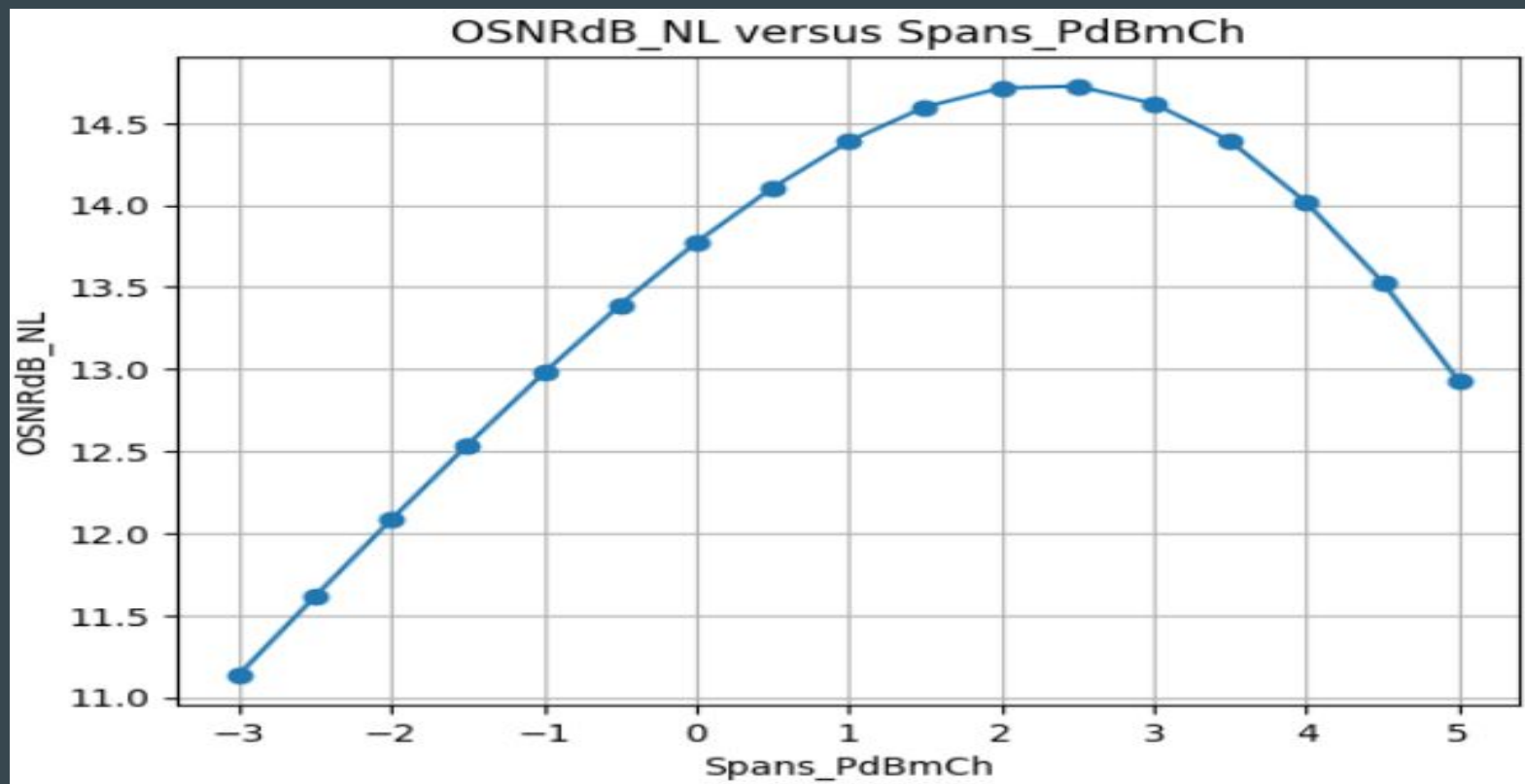
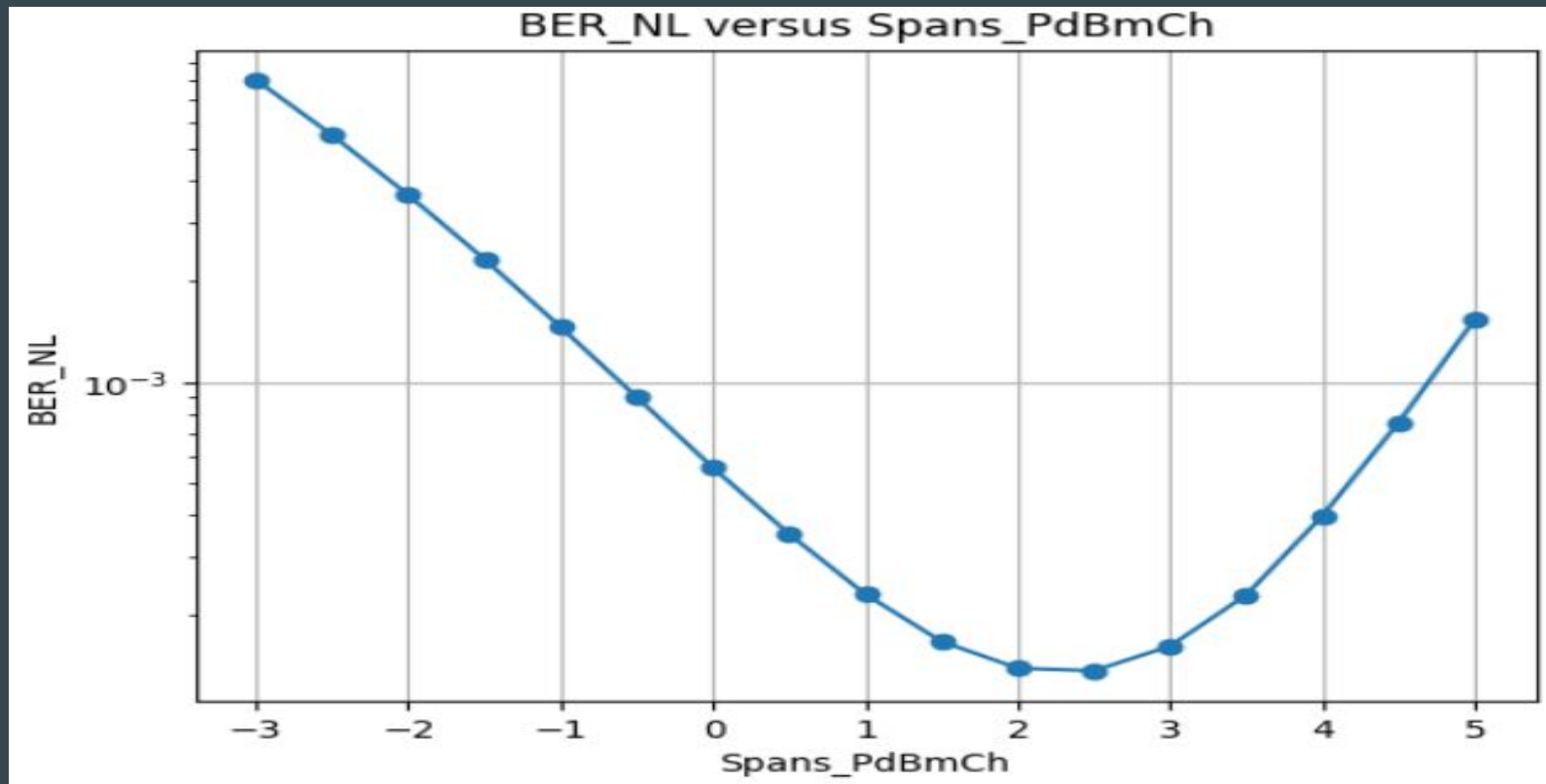


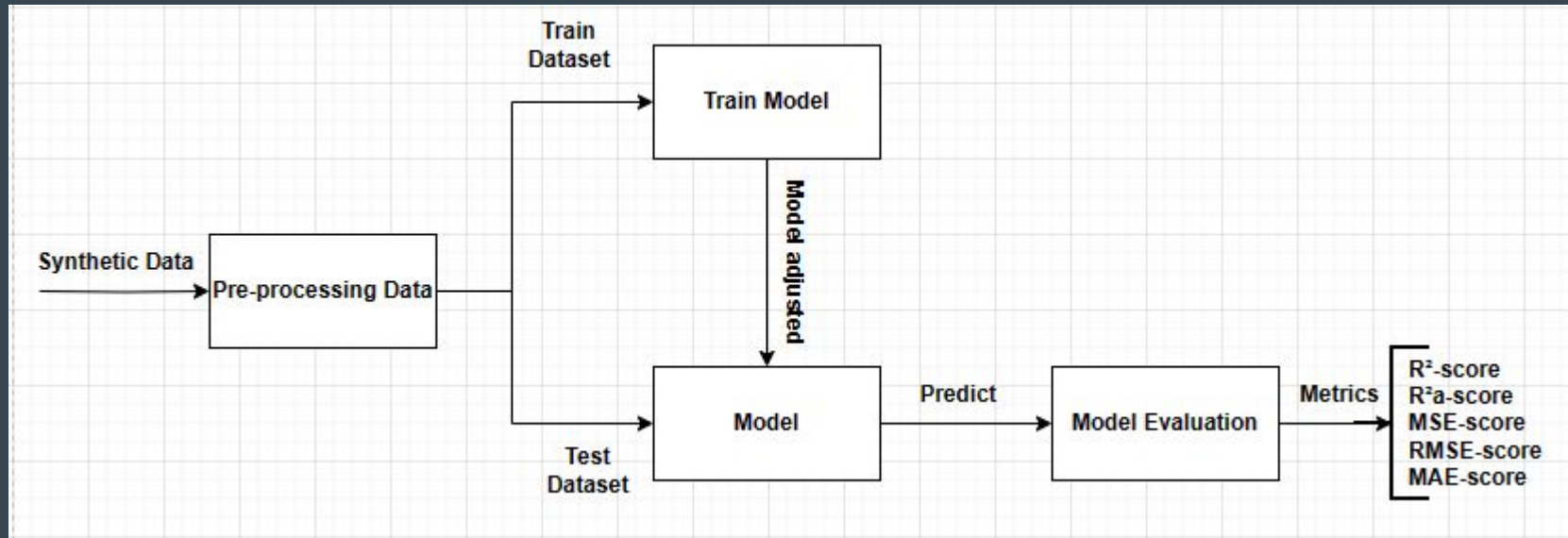
Figura 4: Curva Taxa de Erro de Bits com degradações não-lineares versus Potência de entrada dos canais



Técnicas de machine learning para estimação da qualidade de transmissão

Estruturação da implementação dos algoritmos

Figura 13: Arquitetura do processo de implementação dos algoritmos KNN e SVM



Analise dos resultados

Tabela 1: Resultados da implementação algoritmo **KNN**.

Entradas	Saída	Fonte de dados	Desempenho				
			R ²	R ² a	MSE (dB)	RMSE (dB)	MAE (dB)
Número de canais, Quantidade de Spans, Parâmetro de dispersão, Coeficiente de não-linearidade, Coeficiente de dispersão, Coeficiente de atenuação, Comprimento Span, Potência de entrada, coeficiente de modulação (QPSK = 4, 16QAM = 8, 64QAM = 12)	OSNR	Dados sintéticos	0.99929	0.9993	0.01418	0.11908	0.08277
Número de canais, Quantidade de Spans, Parâmetro de dispersão, Coeficiente de não-linearidade, Coeficiente de dispersão, Coeficiente de atenuação, Comprimento Span, Potência de entrada, coeficiente de modulação (QPSK = 4, 16QAM = 8, 64QAM = 12)	OSNR com interferência não-linear	Dados sintéticos	0.99699	0.99703	0.0454	0.21308	0.13422

Tabela 2: Resultados da implementação algoritmo **KNN**.

Entradas	Saída	Fonte de dados	Desempenho				
			R ²	R ² a	MSE (dB)	RMSE (dB)	MAE (dB)
Número de canais, Quantidade de Spans, Parâmetro de dispersão, Coeficiente de não-linearidade, Coeficiente de dispersão, Coeficiente de atenuação, Comprimento Span, Potência de entrada, coeficiente de modulação (QPSK = 4, 16QAM = 8, 64QAM = 12)	NLIN	Dados sintéticos	0.9933	0.99356	0.45385	0.67369	0.52719
Quantidade de canais, Quantidade de Spans, Parâmetro de dispersão, Coeficiente de não-linearidade, Coeficiente de dispersão, Coeficiente de atenuação, Comprimento do Span, Potência de entrada, coeficiente de modulação (QPSK = 4, 16QAM = 8, 64QAM = 12)	Potência ótima	Dados sintéticos	0.96979	0.97054	0.03591	0.18949	0.15667

Figura 13: Gráfico comparativo entre os valores estimados pelo algoritmo KNN e real(target gerado pelo GN) de OSNR_NL

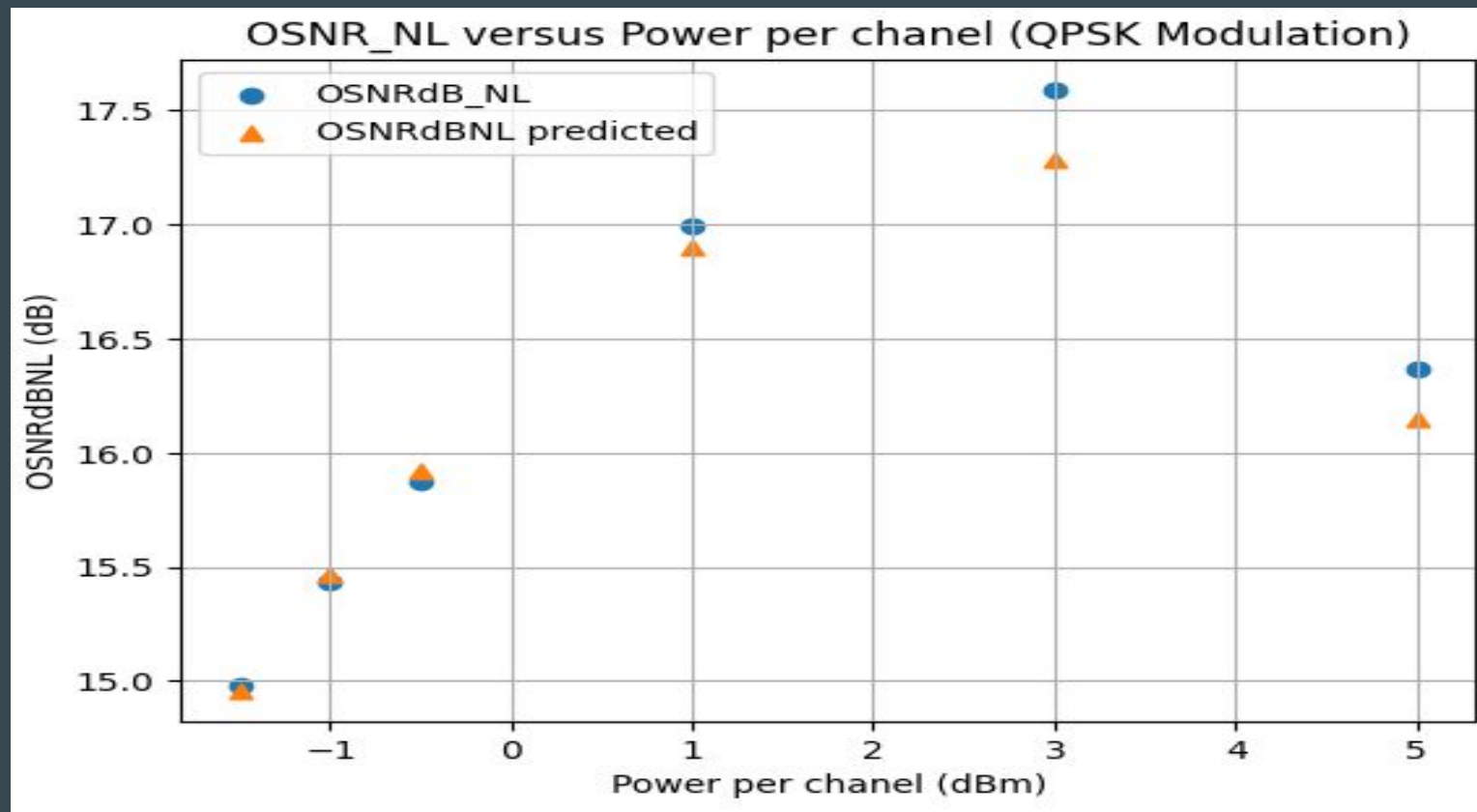


Figura 14: Gráfico comparativo entre os valores estimados pelo algoritmo KNN e real(target gerado pelo GN) de OSNR_NL para um conjunto de validação.

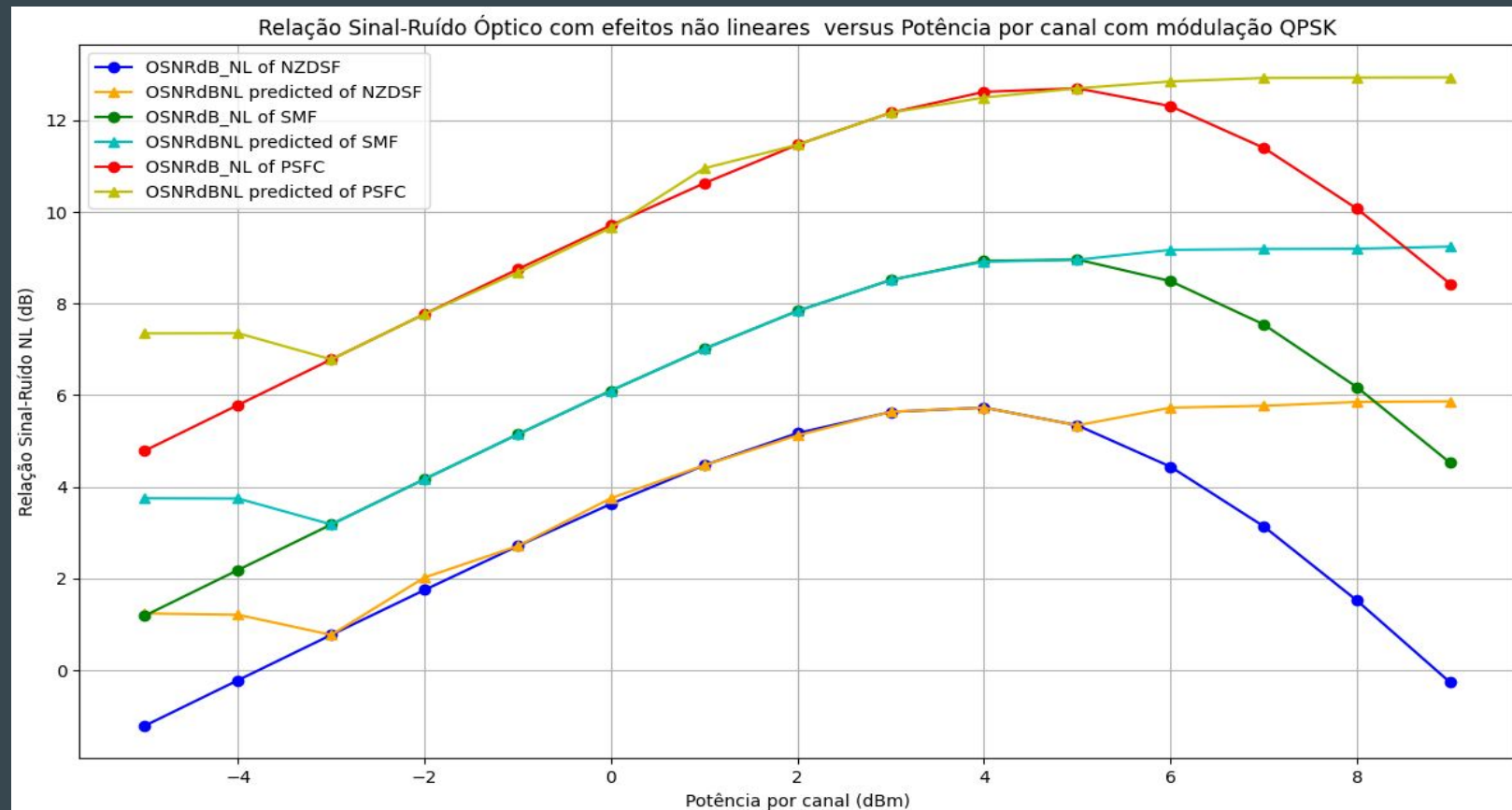


Tabela 3: Resultados da implementação algoritmo SVM.

Entradas	Saída	Fonte de dados	Desempenho				
			R ²	R ² a	MSE (dB)	RMSE (dB)	MAE (dB)
Número de canais, Quantidade de Spans, Parâmetro de dispersão, Coeficiente de não-linearidade, Coeficiente de dispersão, Coeficiente de atenuação, Comprimento Span, Potência de entrada, coeficiente de modulação (QPSK = 4, 16QAM = 8, 64QAM = 12)	OSNR	Dados sintéticos	0.99794	0.99799	0.04019	0.20047	0.15056
Quantidade de canais, Quantidade de Spans, Parâmetro de dispersão, Coeficiente de não-linearidade, Coeficiente de dispersão, Coeficiente de atenuação, Comprimento do Span, Potência de entrada, coeficiente de modulação (QPSK = 4, 16QAM = 8, 64QAM = 12)	OSNR com interferência não-linear	Dados sintéticos	0.94659	0.94882	0.78956	0.88857	0.62112

Tabela 4: Resultados da implementação algoritmo **SVM**.

Entradas	Saída	Fonte de dados	Desempenho				
			R ²	R ² a	MSE (dB)	RMSE (dB)	MAE (dB)
Número de canais, Quantidade de Spans, Parâmetro de dispersão, Coeficiente de não-linearidade, Coeficiente de dispersão, Coeficiente de atenuação, Comprimento Span, Potência de entrada, coeficiente de modulação (QPSK = 4, 16QAM = 8, 64QAM = 12)	PChOptdBm	Dados sintéticos	0.99862	0.99864	0.09528	0.30867	0.22949
Número de canais, Quantidade de Spans, Parâmetro de dispersão, Coeficiente de não-linearidade, Coeficiente de dispersão, Coeficiente de atenuação, Comprimento Span, Potência de entrada, coeficiente de modulação (QPSK = 4, 16QAM = 8, 64QAM = 12)	NLIN_Power	Dados sintéticos	0.99863	0.99866	0.09552	0.30907	0.23069

Figura 15: Gráfico comparativo entre os valores estimados pelo algoritmo SVM e real (target gerado pelo GN) de OSNR_NL

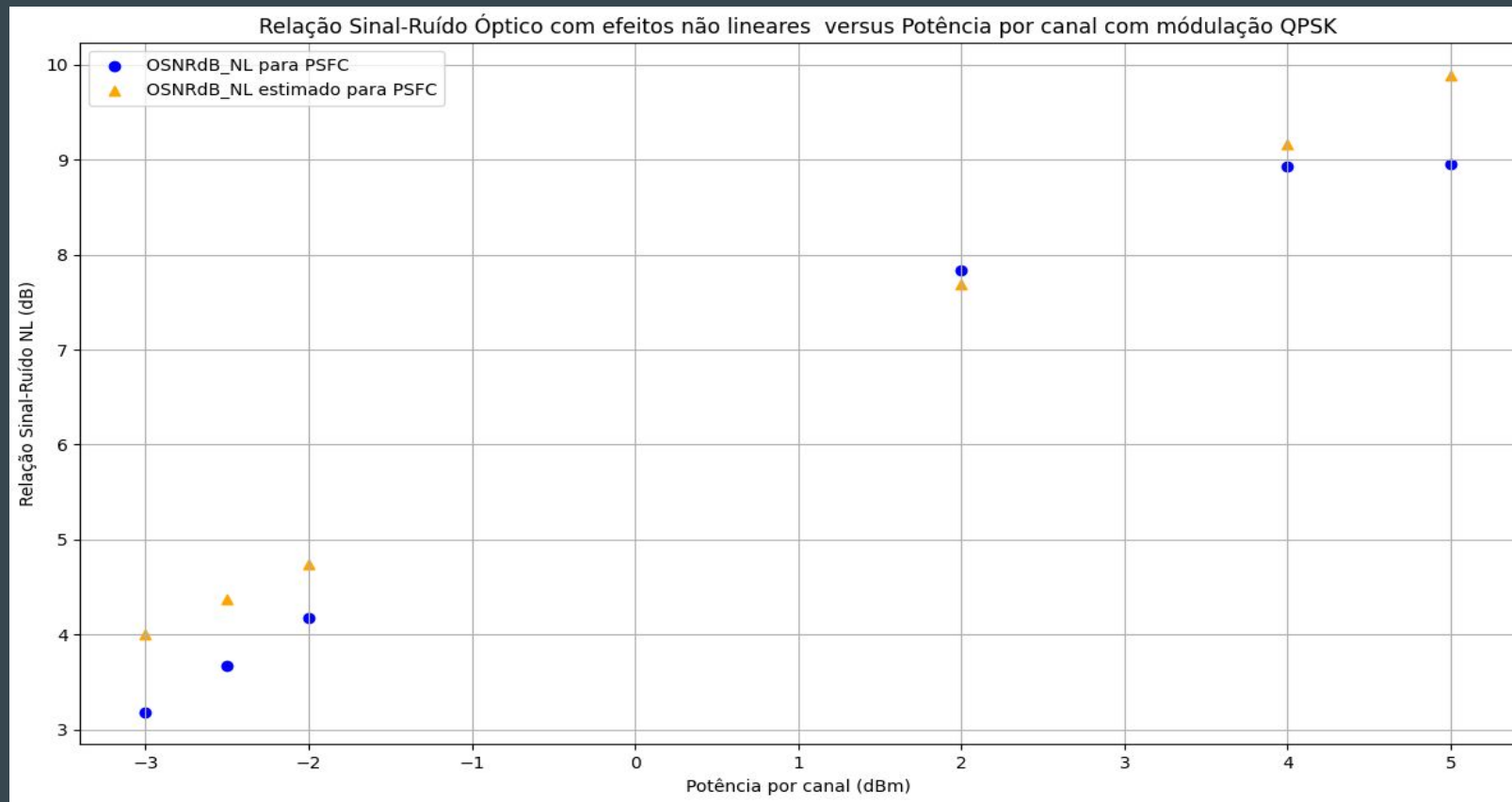


Figura 16: Gráfico comparativo entre os valores estimados pelo algoritmo SVM e real(target gerado pelo GN) de OSNR_NL para um conjunto de validação.

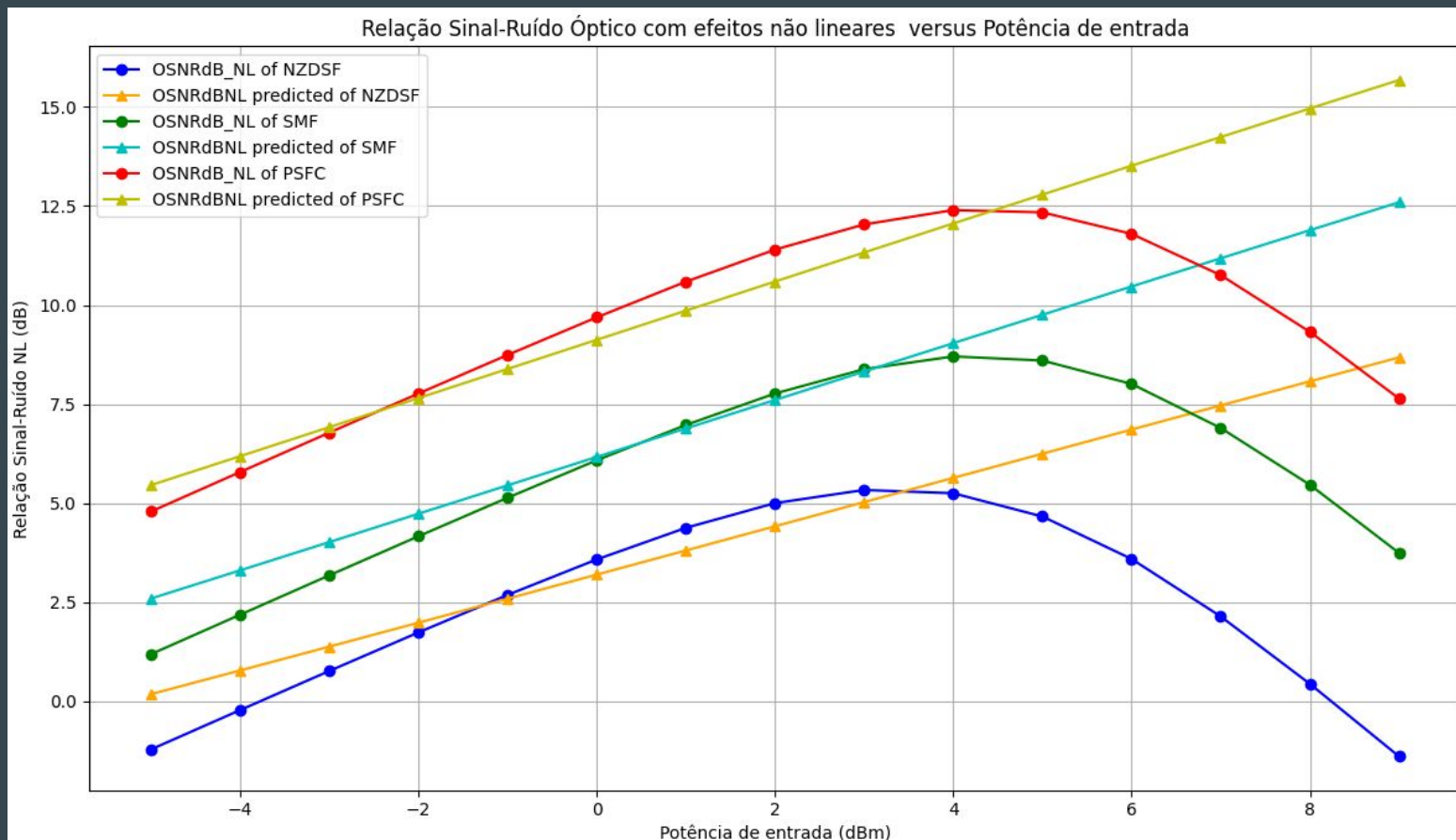


Tabela 3: Resultados da implementação algoritmo ANN.

Entradas	Saída	Fonte de dados	Desempenho				
			R ²	R ² a	MSE (dB)	RMSE (dB)	MAE (dB)
Número de canais, Quantidade de Spans, Parâmetro de dispersão, Coeficiente de não-linearidade, Coeficiente de dispersão, Coeficiente de atenuação, Comprimento Span, Potência de entrada, coeficiente de modulação (QPSK = 4, 16QAM = 8, 64QAM = 12)	OSNR	Dados sintéticos	0.97592	0.97647	0.47739	0.69093	0.65754
Quantidade de canais, Quantidade de Spans, Parâmetro de dispersão, Coeficiente de não-linearidade, Coeficiente de dispersão, Coeficiente de atenuação, Comprimento do Span, Potência de entrada, coeficiente de modulação (QPSK = 4, 16QAM = 8, 64QAM = 12)	OSNR com interferência não-linear	Dados sintéticos	0.89309	0.90706	1.42254	1.1927	1.09091

Tabela 4: Resultados da implementação algoritmo ANN.

Entradas	Saída	Fonte de dados	Desempenho				
			R ²	R ² a	MSE (dB)	RMSE (dB)	MAE (dB)
Número de canais, Quantidade de Spans, Parâmetro de dispersão, Coeficiente de não-linearidade, Coeficiente de dispersão, Coeficiente de atenuação, Comprimento Span, Potência de entrada, coeficiente de modulação (QPSK = 4, 16QAM = 8, 64QAM = 12)	PChOptdBm	Dados sintéticos	0.99808	0.99805	0.00233	0.04827	0.04064
Número de canais, Quantidade de Spans, Parâmetro de dispersão, Coeficiente de não-linearidade, Coeficiente de dispersão, Coeficiente de atenuação, Comprimento Span, Potência de entrada, coeficiente de modulação (QPSK = 4, 16QAM = 8, 64QAM = 12)	NLIN_Power	Dados sintéticos	0.99058	0.99048	0.67282	0.82026	0.73063

Figura 19: Curva do MAE versus Epoca

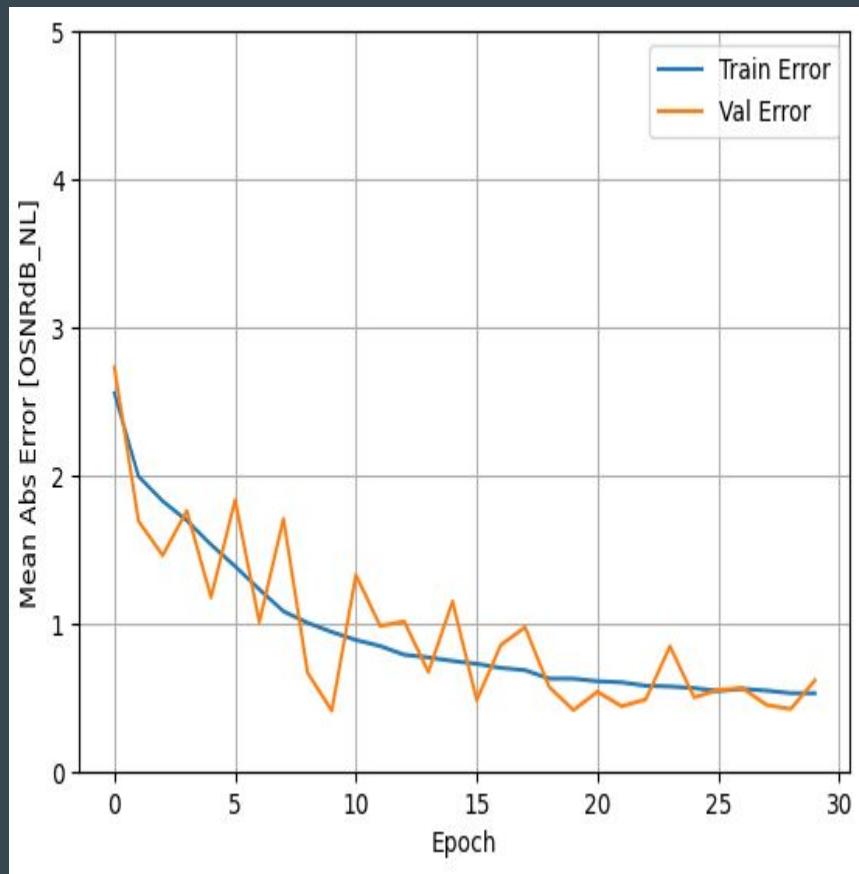


Figura 19: Curva do MSE versus Epoca

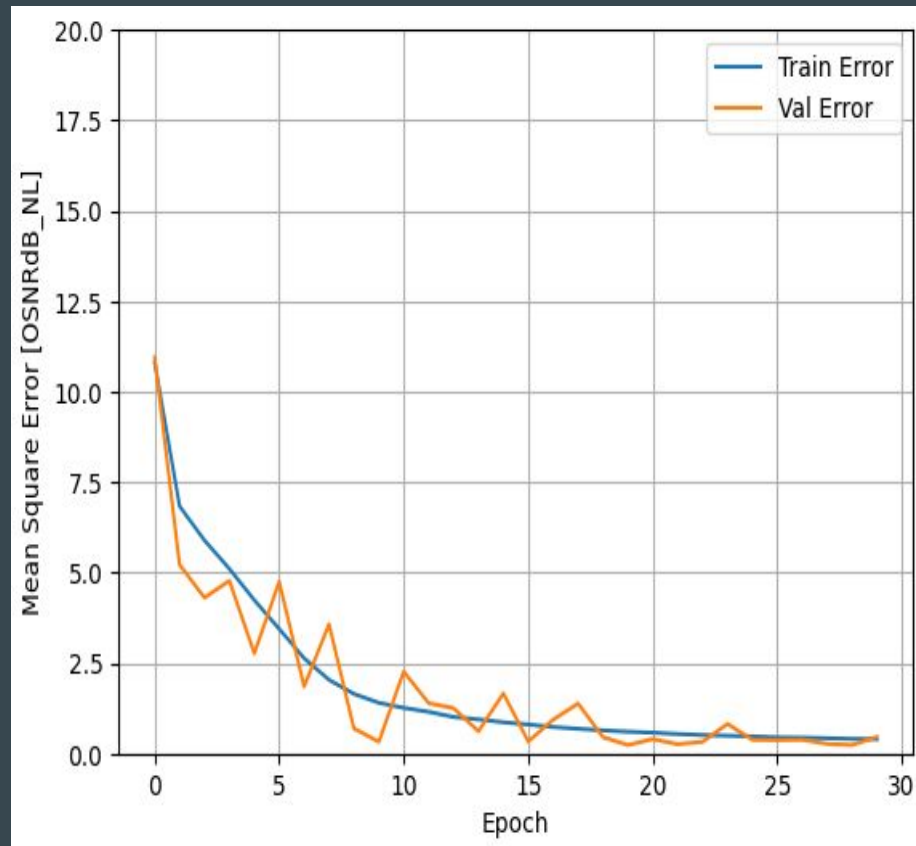


Figura 19: Gráfico comparativo entre os valores estimados pelo algoritmo ANN e real (target gerado pelo GN) de OSNR_NL para conjunto de testes

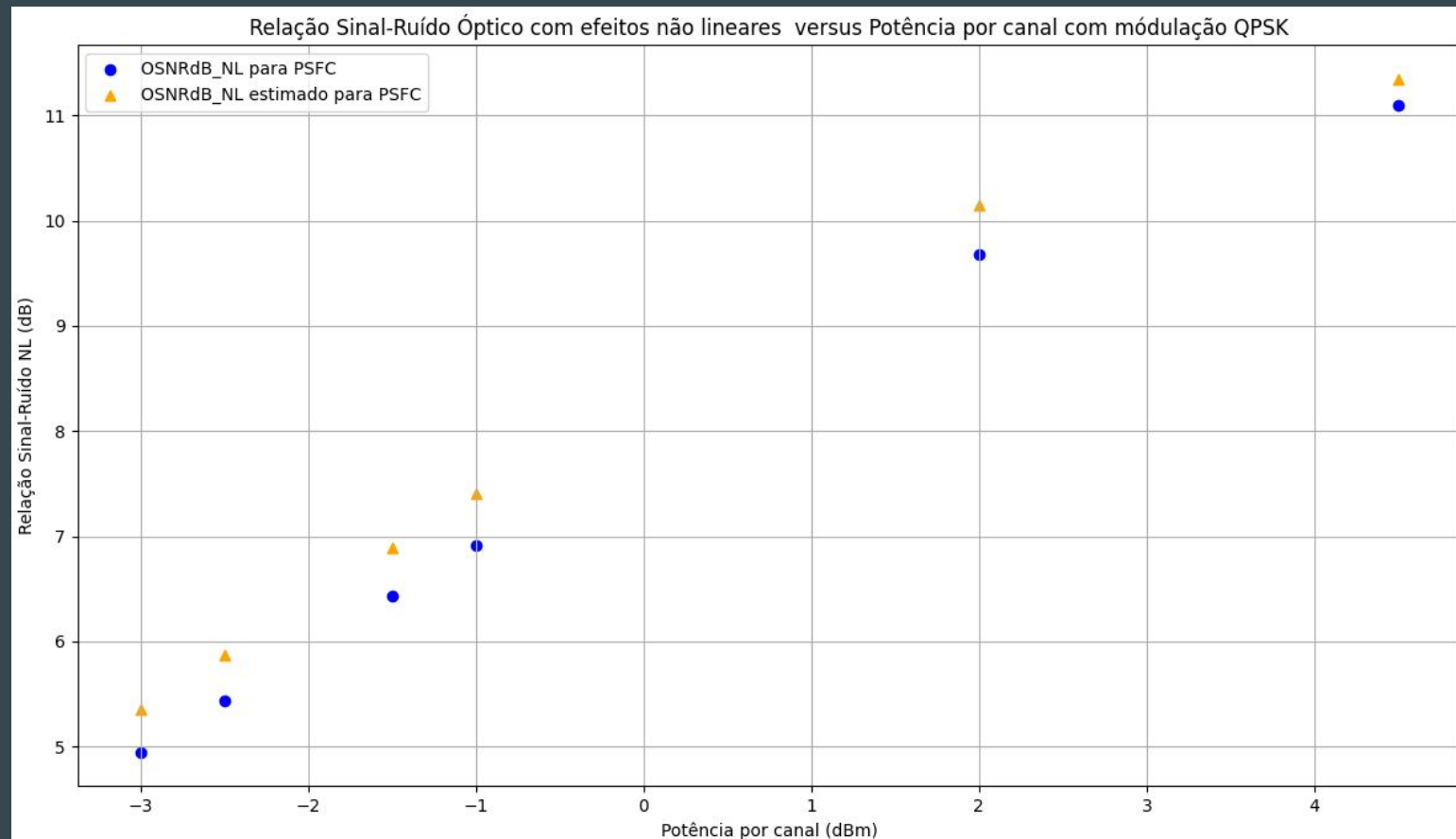


Figura 20: Gráfico comparativo entre os valores estimados pelo algoritmo ANNe real(target gerado pelo GN) de OSNR_NL para um conjunto de validação.

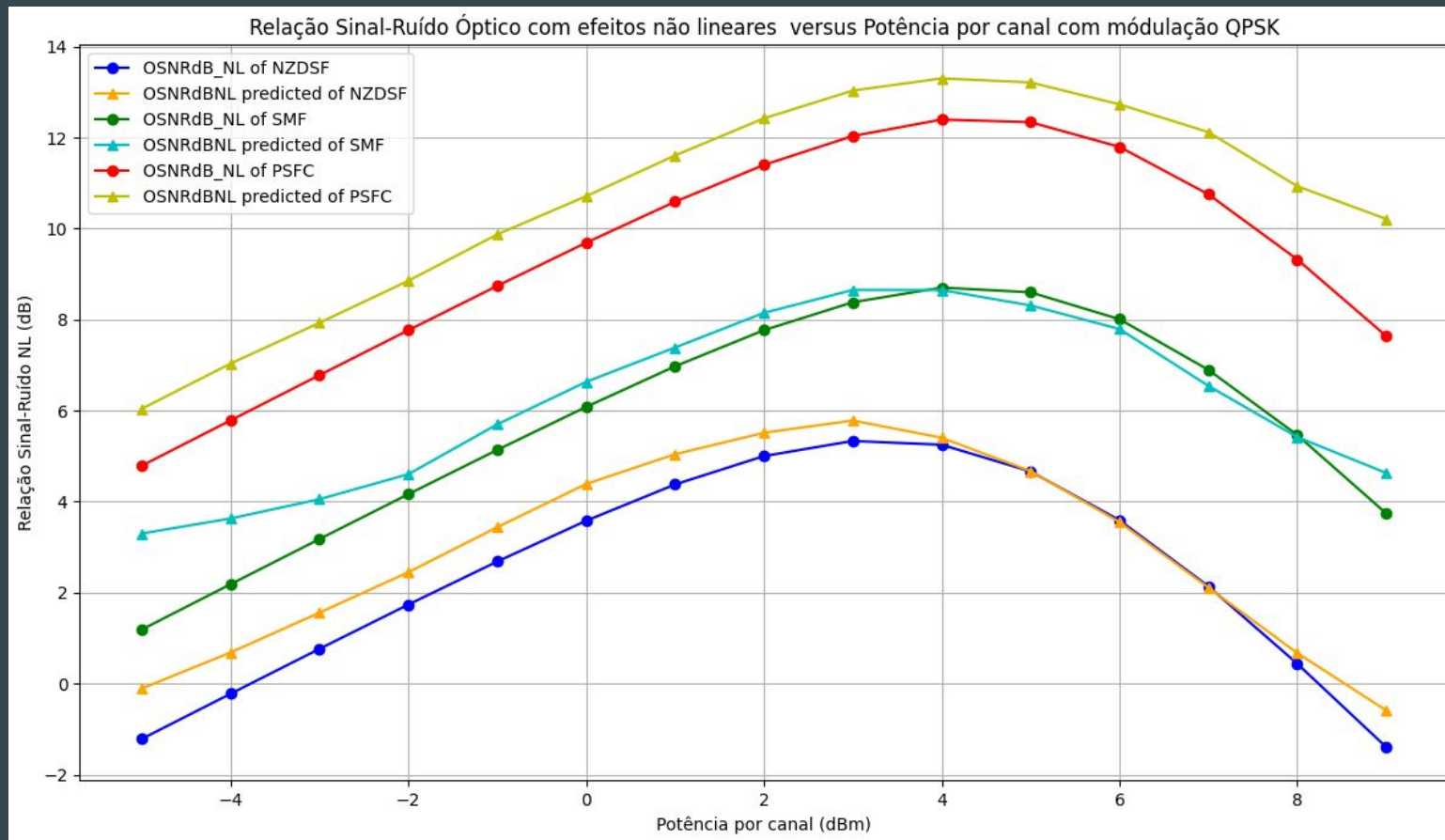


Tabela 8: Comparação entre os tempos de treinamento dos algoritmos, utilizando um conjunto de treinamento com 35341 amostras.

Algoritmo	Quantidade de atributos	
	9	14
	Tempo médio (s)	
KNN	0.17	0.22
SVM	94	105
ANN	42	57

Tabela 8: Comparação entre os tempos de predição dos algoritmos, utilizando um conjunto de testes com 15147 amostras.

Algoritmo	Quantidade de atributos	
	9	14
	Tempo médio (s)	
KNN	0.37	0.96
SVM	33	36
ANN	47	56

- Para gerar uma amostra utilizando o modelo Gaussian Noise (GN) no MatLab, levando em consideração um processador AMD Ryzen™ 5 5600G, com 6 núcleos de CPU, 12 de threads e Clock base 3.9GHz, foi de 4.12 segundos. (Vale ressaltar que o tempo para gerar essa nova amostra é proporcional ao número de canais do link que queremos testar)

Estratégias futuras

- Melhorar os modelos já criados visando a diminuição do MSE e MAE.
- Geração de novos conjunto de dados.
- DNNs.

Novos resultados

- Utilização da técnica de estratificação dos dados para garantir o balanceamento entre treino e teste.

```
{'R²': 0.9962,  
'R²a': 0.99625,  
'MSE': 0.0576,  
'RMSE': 0.24001,  
'MAE': 0.16386,  
'MAPE': 1.561283}
```

KNN

```
{'R²': 0.94908,  
'R²a': 0.95097,  
'MSE': 0.77016,  
'RMSE': 0.87759,  
'MAE': 0.60295,  
'MAPE': 5.745908}
```

SVM kernel RBF

```
{'R²': 0.95172,  
'R²a': 0.95597,  
'MSE': 0.67484,  
'RMSE': 0.82149,  
'MAE': 0.52376,  
'MAPE': 4.481094}
```

SVM kernel
polynomial
de grau 6

```
{'R²': 0.98113,  
'R²a': 0.98105,  
'MSE': 0.28706,  
'RMSE': 0.53578,  
'MAE': 0.3169,  
'MAPE': 2.942796}
```

SVM kernel
polynomial
de grau 8

```
{'R²': 0.99949,  
'R²a': 0.99949,  
'MSE': 0.00781,  
'RMSE': 0.0884,  
'MAE': 0.0611,  
'MAPE': 0.658211}
```

ANN

Figura 21: Gráfico comparativo entre os valores estimados pelo algoritmo KNN, real(target gerado pelo GN) de OSNR_NL para um conjunto de validação.

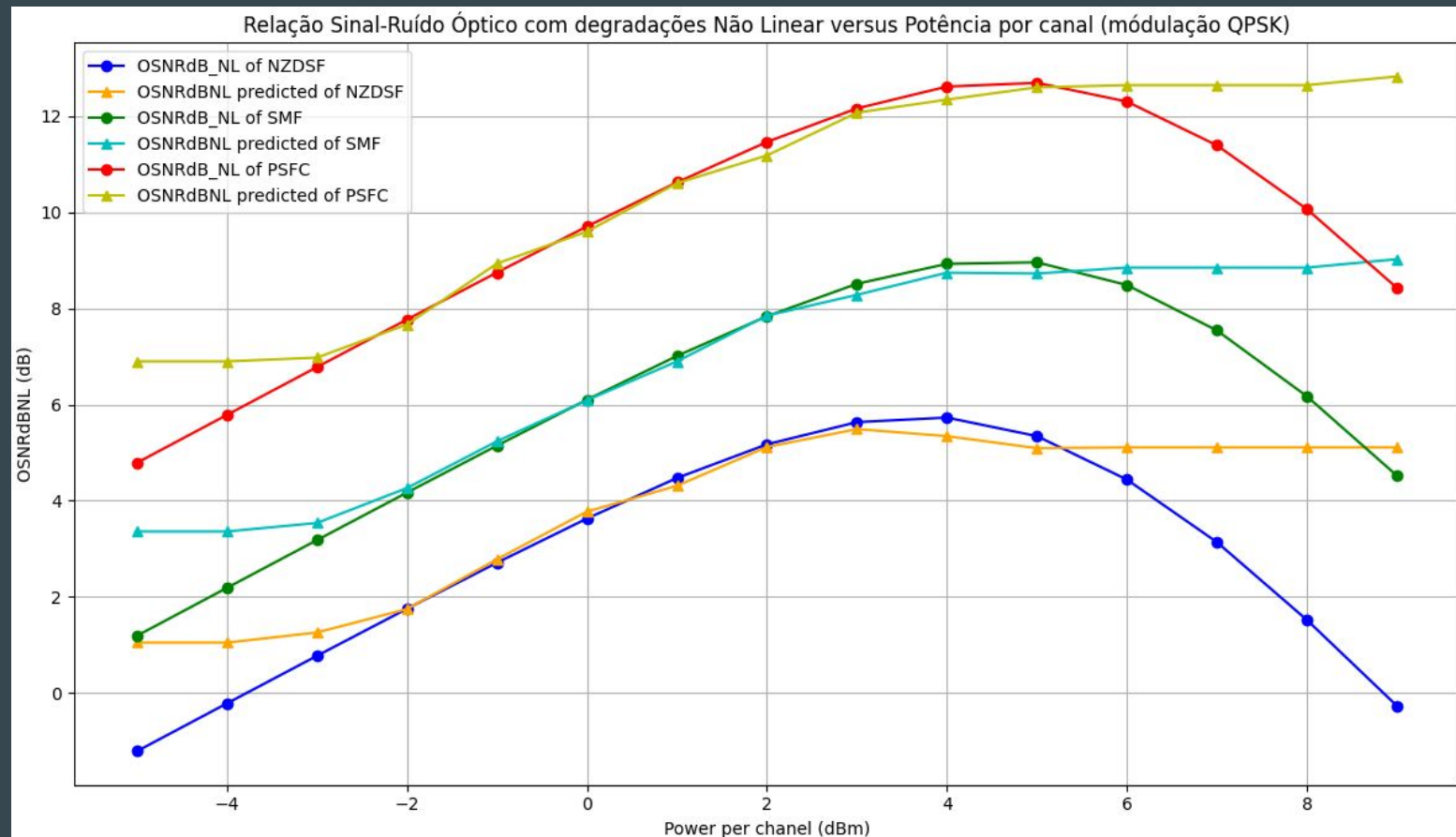


Figura 22: Gráfico comparativo entre os valores estimados pelo algoritmo SVM com kernel RBF, real(target gerado pelo GN) de OSNR_NL para um conjunto de validação.

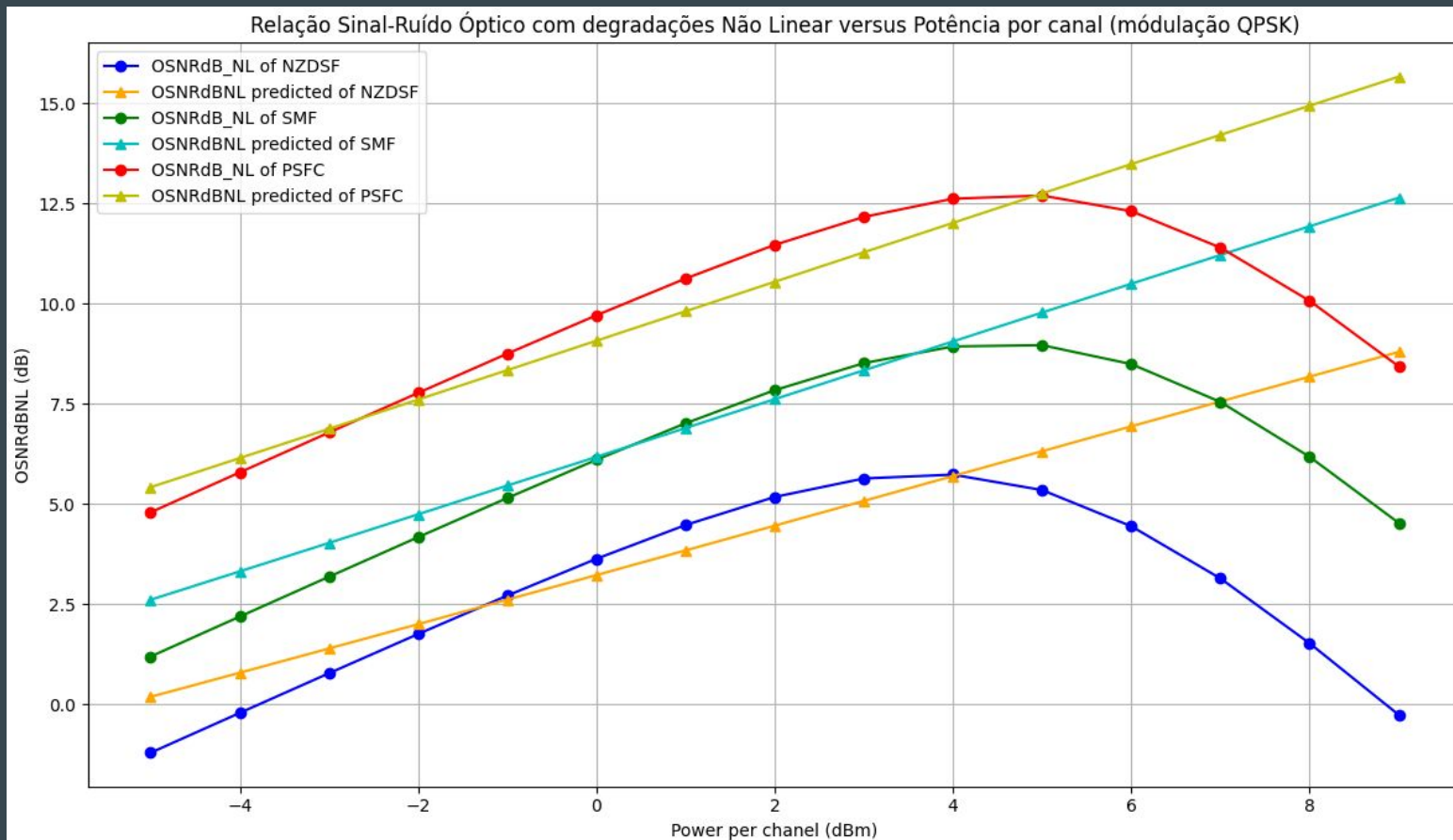


Figura 23: Gráfico comparativo entre os valores estimados pelo algoritmo SVM com kernel polinomial de grau 8, real(target gerado pelo GN) de OSNR_NL para um conjunto de validação.

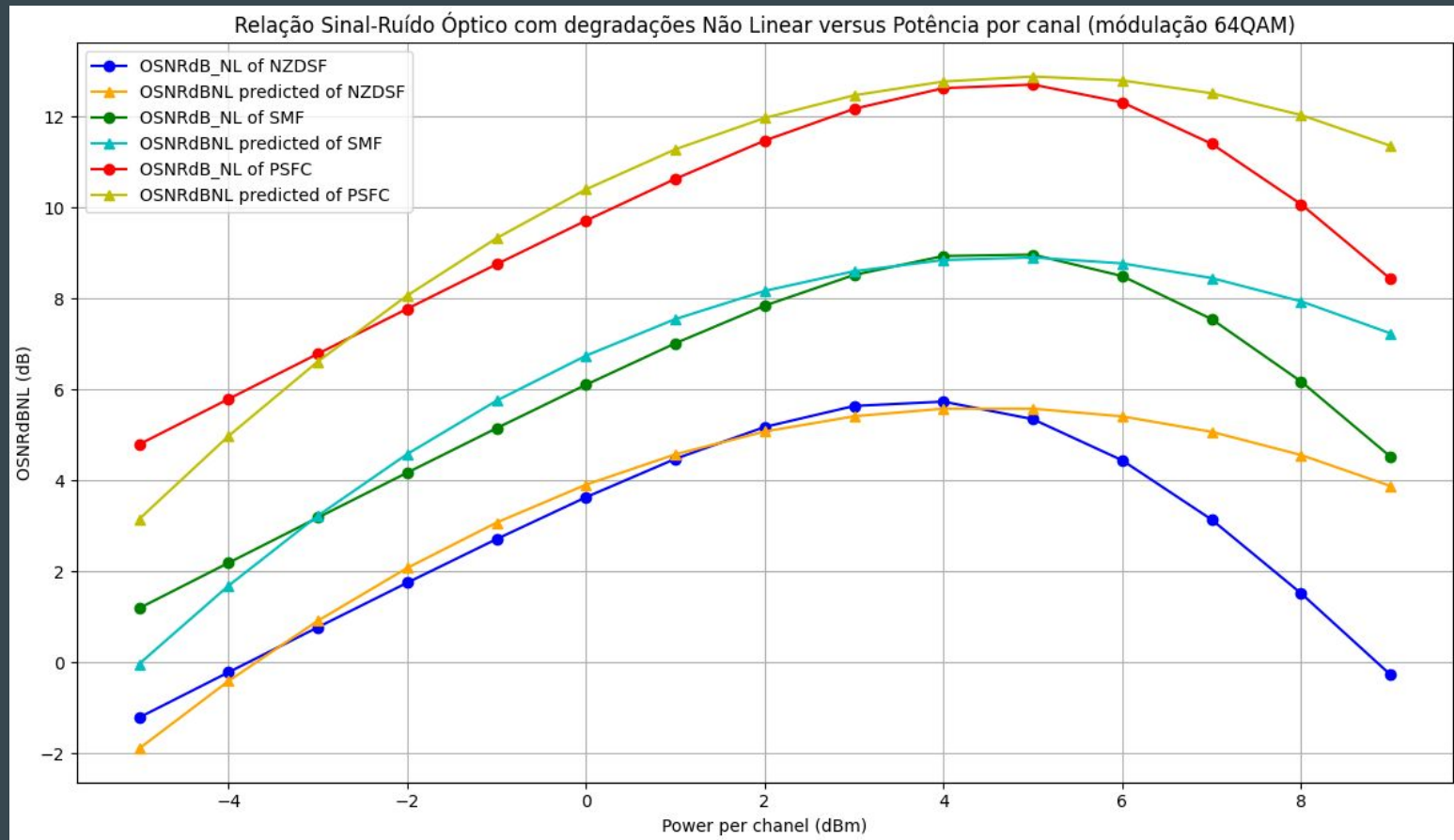


Figura 25: Gráfico comparativo entre os valores estimados pelo algoritmo SVM com kernel polinomial de grau 6, real(target gerado pelo GN) de OSNR_NL para um conjunto de validação.

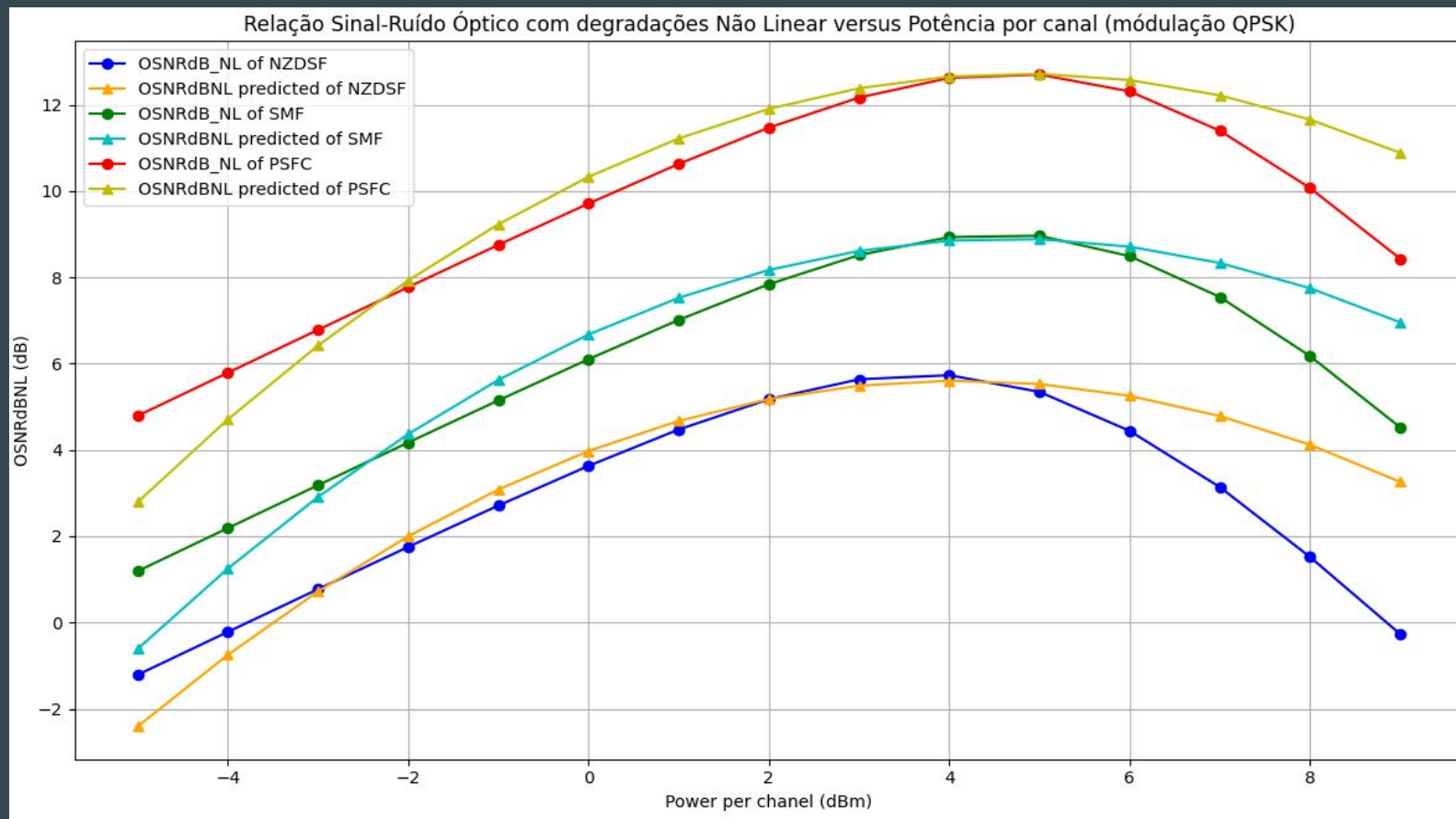
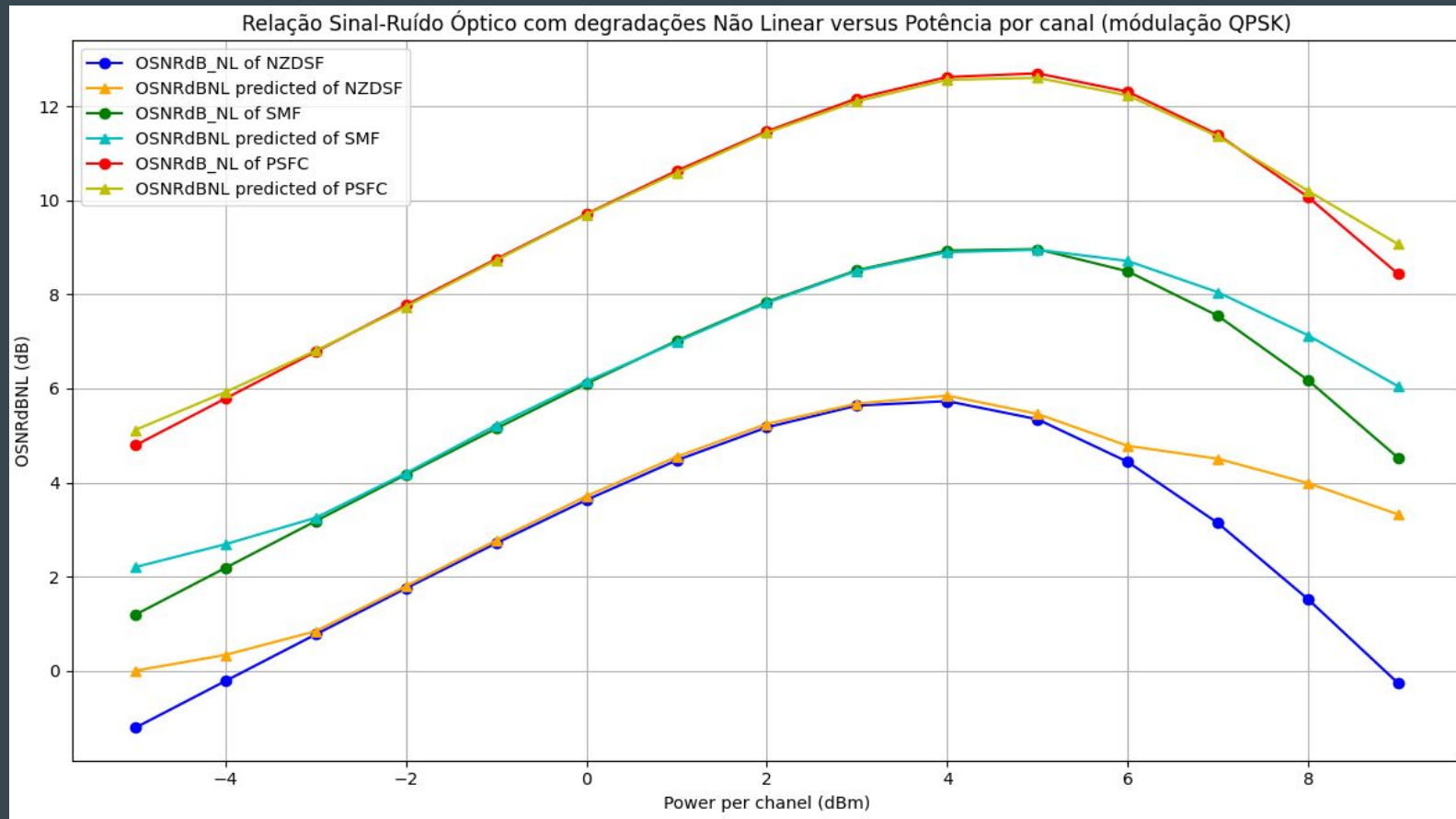


Figura 28: Gráfico comparativo entre os valores estimados pelo algoritmo ANN, real(target gerado pelo GN) de OSNR_{NL} para um conjunto de validação.



Comparativo entre os resultados dos algoritmos (KNN, SVM e ANN) na extrapolação por tipo de fibra.

Novos resultados

Realizar avaliação dos modelos utilizando novos tipos de fibra.

- 1. LEAF, com parâmetros :

$\text{DisPar} = 4.05 \text{ (ps/nm)};$

$\text{Alpha} = 0.24 \text{ (dB/km)};$

$\text{Gamma} = 1.3 \text{ (1/[W km])}$

- 2. TWC, com parâmetros:

$\text{DisPar} = 2.08 \text{ (ps/nm)};$

$\text{Alpha} = 0.24 \text{ (dB/km)};$

$\text{Gamma} = 2.0 \text{ (1/[W km])}$

Figura 34: Gráfico comparativo entre os resultados do algoritmo KNN, considerando 5 vizinhos

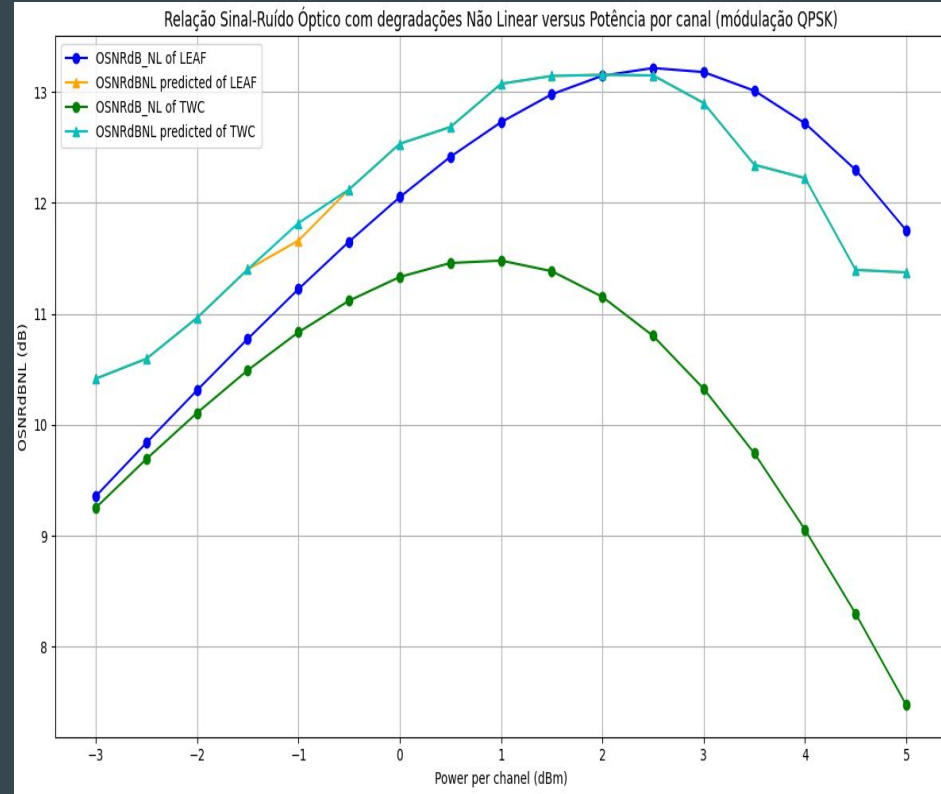
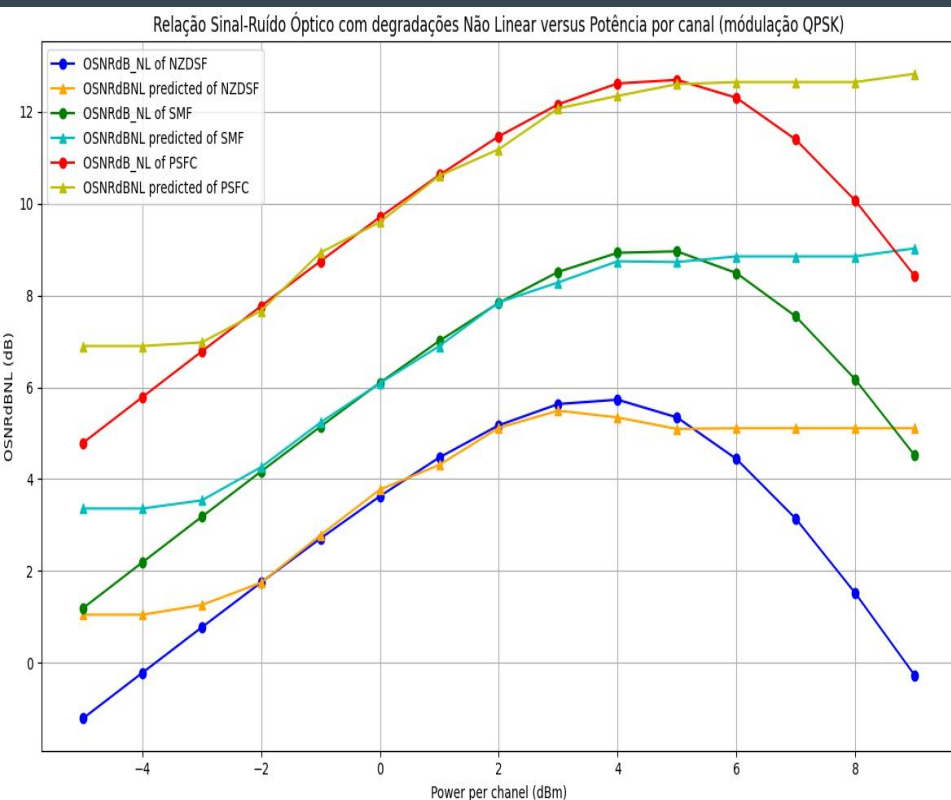


Figura 35: Gráfico comparativo entre os resultados do algoritmo SVM com kernel polinomial de grau 8.

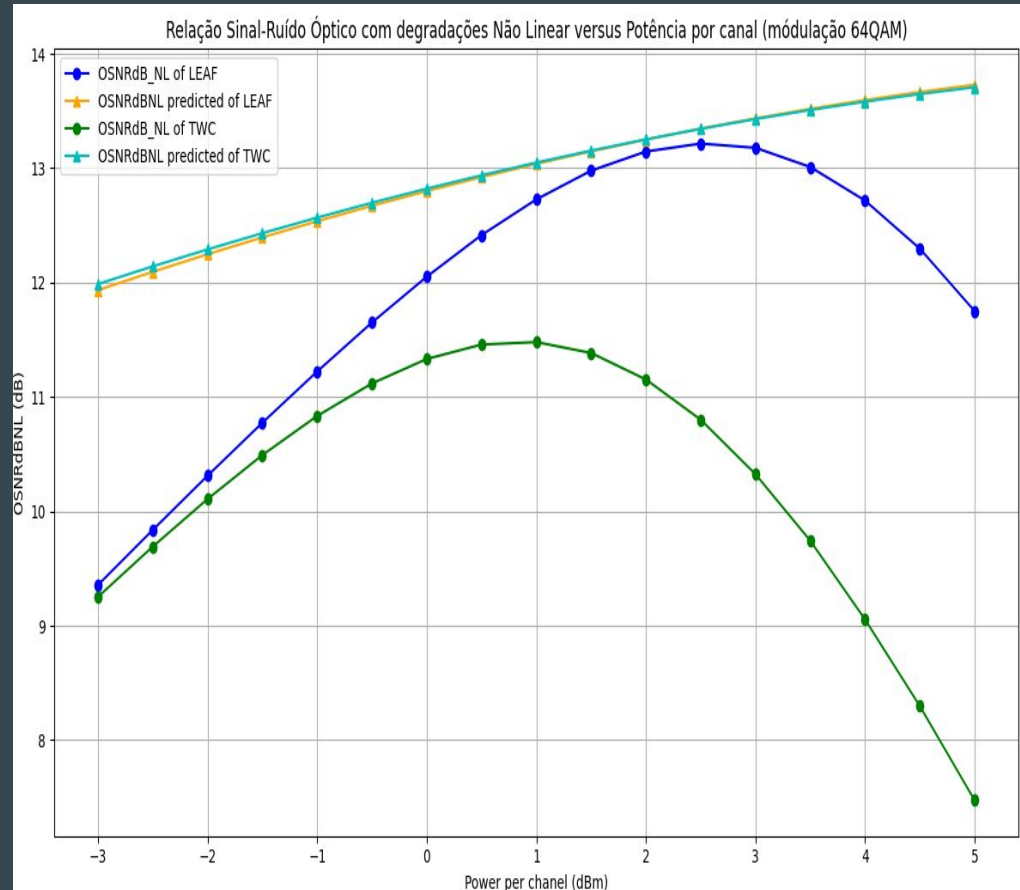
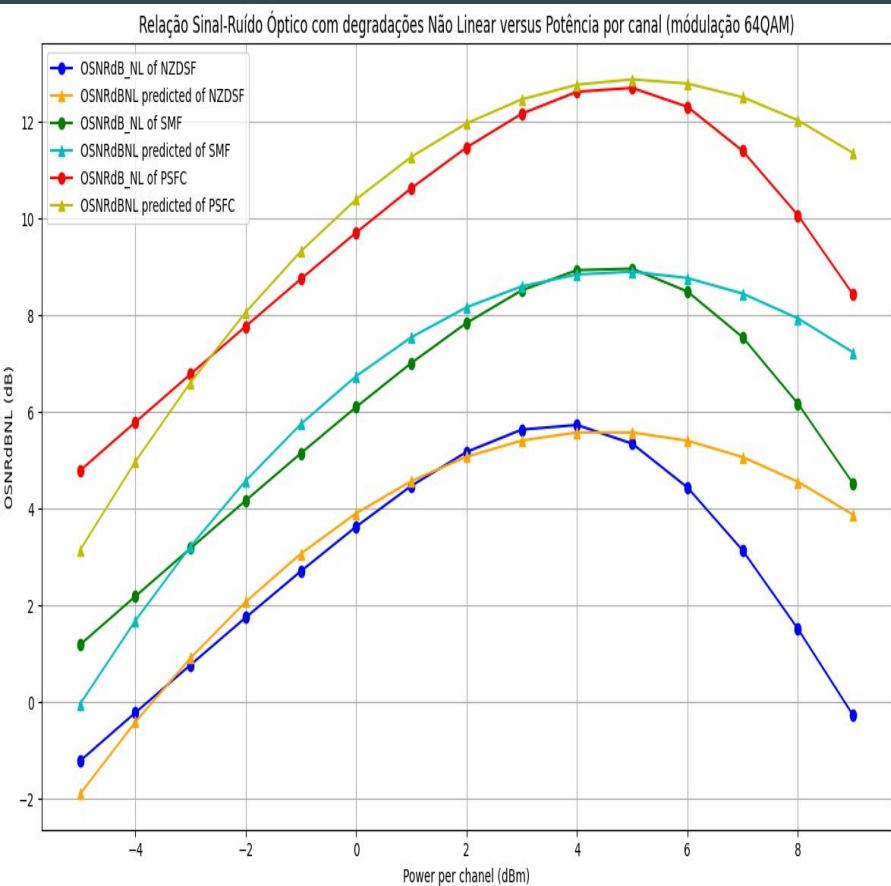
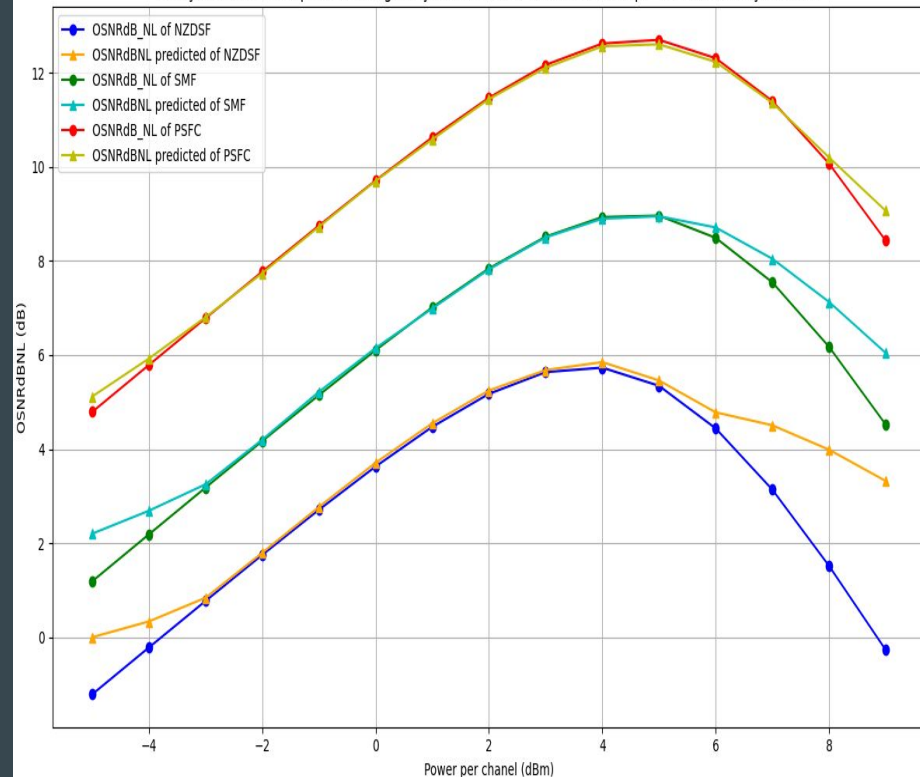
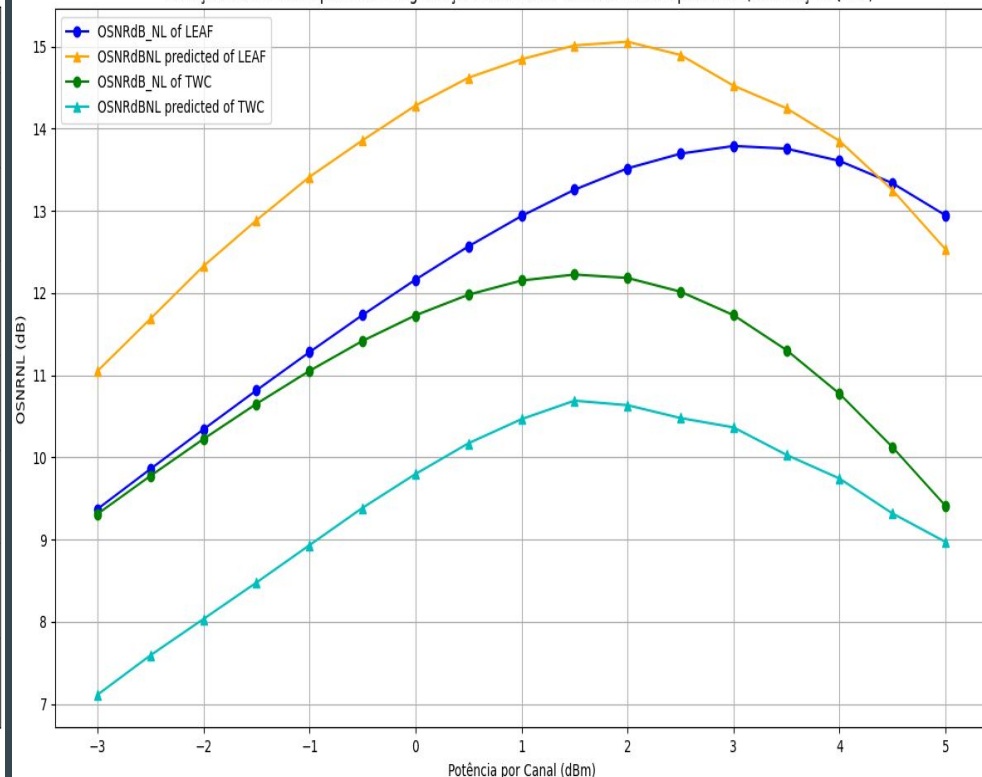


Figura 36: Gráfico comparativo entre os resultados do algoritmo ANN.

Relação Sinal-Ruído Óptico com degradações Não Linear versus Potência por canal (modulação QPSK)



Relação Sinal-Ruído Óptico com degradações Não Linear versus Potência por canal (modulação QPSK)



Feature scaling

Tabela 9: Resultados comparativos para um modelo de ANN com e sem feature scaling

Rede Neural					
Métricas					
	R ²	R ² a	MSE	MAE	RMSE
raw_data	0.9915	0.99149	0.12938	0.3597	0.31533
feature scaling	0.99646	0.99661	0.05153	0.227	0.20099

Observa-se que quanto utiliza-se feature scaling as métricas ficam melhores são melhores quando utilizamos, porém quando tentamos extrapolar as previsões para dados que estão muito distante do universo de treinamento o erro do modelo aumenta significativamente, ver figura 37.

Figura 37: Gráfico comparativo entre os resultados do algoritmo ANN. No gráfico do lado esquerdo não houve feature scaling. No gráfico do lado direito foi realizado feature scaling.

