



UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

CLASSIFICAÇÃO AUTOMÁTICA DE MODULAÇÕES EM RECEPTORES
ÓPTICOS COERENTES FLEXÍVEIS

Antonio Marcos da Costa Pereira

Manaus
Março de 2024



UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

CLASSIFICAÇÃO AUTOMÁTICA DE MODULAÇÕES EM RECEPTORES
ÓPTICOS COERENTES FLEXÍVEIS

Antonio Marcos da Costa Pereira

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, PPGEE, da Universidade Federal do Amazonas, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientadores: Waldir Sabino da Silva Júnior
André Luiz Aguiar da Costa

Manaus
Março de 2024

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

P436c Pereira, Antonio Marcos da Costa
Classificação Automática de Modulações em Receptores Ópticos
Coerentes Flexíveis / Antonio Marcos da Costa Pereira . 2024
108 f.: il. color; 31 cm.

Orientador: Waldir Sabino da Silva Júnior
Coorientador: André Luiz Aguiar da Costa
Dissertação (Mestrado em Engenharia Elétrica) - Universidade
Federal do Amazonas.

1. Classificação Automática de Modulação. 2. Redes Ópticas. 3.
Random Forest. 4. Perceptron Multicamada. I. Silva Júnior, Waldir
Sabino da. II. Universidade Federal do Amazonas III. Título



Ministério da Educação
Universidade Federal do Amazonas
Coordenação do Programa de Pós-Graduação em Engenharia Elétrica

FOLHA DE APROVAÇÃO

Poder Executivo Ministério da Educação
Universidade Federal do Amazonas
Faculdade de Tecnologia
Programa de Pós-graduação em Engenharia Elétrica

ANTONIO MARCOS DA COSTA PEREIRA

CLASSIFICAÇÃO AUTOMÁTICA DE MODULAÇÕES EM RECEPTORES ÓPTICOS COERENTES FLEXÍVEIS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Aprovada em 15 de março de 2024.

Documento assinado eletronicamente por:
Prof. Dr. Waldir Sabino da Silva Júnior Presidente
Prof. Dr. Celso Barbosa Carvalho (Membro Titular 2 - Interno)
Prof. Dr. Alexandre Coutinho Matheus (Membro Titular 1 - Externo)

Manaus, 11 de junho de 2024.



Documento assinado eletronicamente por **Waldir Sabino da Silva Júnior, Professor do Magistério Superior**, em 11/06/2024, às 09:02, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Alexandre Coutinho Mateus, Usuário Externo**, em 11/06/2024, às 09:06, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Celso Barbosa Carvalho, Professor do Magistério Superior**, em 11/06/2024, às 15:27, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufam.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2088836** e o código CRC **9C9BEB4A**.

Av. Octávio Hamilton Botelho Mourão - Bairro Coroado 1 Campus Universitário Senador Arthur Virgílio Filho,
Setor Norte - Telefone: (92) 3305-1181
CEP 69080-900 Manaus/AM - mestrado_engelettrica@ufam.edu.br

Referência: Processo nº 23105.011910/2024-21

SEI nº 2088836

Criado por [62136828215](#), versão 3 por [62136828215](#) em 11/06/2024 08:47:49.

*Gostaria de expressar minha
gratidão à minha família,
em especial à minha esposa,
Jéssica.*

*Agradeço por estar ao meu lado
em todos os momentos
e pela oportunidade de
viver esse momento único
ao lado de nossa filha,
Artemis.*

*Para mim, vocês são o verdadeiro
significado desta jornada.*

Agradecimentos

Expresso minha profunda gratidão aos meus pais, Augusto e Márcia, pela educação exemplar, por me guiarem a trilhar um caminho correto e pelo cuidado incansável que sempre dedicaram a mim.

Expresso minha gratidão aos meus orientadores, Professor Doutor Waldir Sabino e Professor Doutor André Aguiar, por todo o ciclo de orientação e pela oportunidade de trabalhar e compartilhar experiências. Sem dúvidas ambos contribuíram muito para enriquecer minha formação acadêmica, e crescimento pessoal e profissional.

Também gostaria de agradecer à Universidade Federal do Amazonas (UFAM) e ao Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) pela oportunidade de ingressar no programa, a oportunidade de participar de todas as palestras disponibilizadas e ao excelente conteúdo ministrado por todos os professores do programa.

Adicionalmente, gostaria de agradecer ao CNPq — Conselho Nacional de Desenvolvimento Científico e Tecnológico e à FAPEAM — Fundação de Amparo à Pesquisa do Estado do Amazonas pelo apoio fornecido ao longo deste trabalho.

Este trabalho não teria sido possível sem o esforço coletivo de todos aqui mencionados e por isso estou verdadeiramente grato.

Resumo da Dissertação apresentada à UFAM como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica

CLASSIFICAÇÃO AUTOMÁTICA DE MODULAÇÕES EM RECEPTORES ÓPTICOS COERENTES FLEXÍVEIS

Antonio Marcos da Costa Pereira

Orientadores: Waldir Sabino da Silva Júnior

André Luiz Aguiar da Costa

Programa: Pós-Graduação em Engenharia Elétrica

Para acompanhar o aumento na demanda por taxas de transmissão mais altas e melhorias na conectividade de internet, as pesquisas estão se concentrando em métodos eficazes para otimizar o uso de redes ópticas. Tais estudos são fundamentais para desenvolver sistemas que possam, de forma autônoma, coletar informações necessárias para ajustar os parâmetros de transmissão. Isso inclui escolher as modulações adequadas para o meio de comunicação, decidir entre uma transmissão em mono portadora ou múltiplas portadoras, e selecionar os algoritmos mais eficientes para regenerar e corrigir os sinais. Essas abordagens permitem que as redes se ajustem automaticamente para maximizar o desempenho e a eficiência sem intervenção humana direta. Neste contexto, esta dissertação investiga como as técnicas de aprendizado de máquina podem ser aplicadas para automatizar a configuração de parâmetros em sistemas de comunicação óptica. Esse estudo visa melhorar a adaptabilidade e eficiência desses sistemas, permitindo que ajustem suas configurações de forma autônoma com base nas informações adquiridas, otimizando assim a transmissão de dados. Focando em receptores ópticos coerentes flexíveis, o estudo visa aprimorar a eficiência e adaptabilidade destes sistemas ao automatizar o processo de classificação do tipo de modulação e predição do valor de OSNR, elementos cruciais para otimização do desempenho e confiabilidade na transmissão de dados ópticos. Para isso, utiliza-se um *setup back-to-back* entre um transmissor e receptor simulados, gerando 76.800 sinais com modulações DP-BPSK, DP-QPSK, DP-8-PSK,

DP-16-QAM, DP-32-QAM e DP-64-QAM em um total de 51 níveis diferentes de OSNR. Os algoritmos *ensemble*, *AdaBoost*, *Árvore de Decisão CART*, *Gradient Boosting*, *Random Forest* e a rede neural *Perceptron Multicamada* são empregados para classificar modulações e predição do valor de OSNR. Os resultados mostram acurácias superiores a 99% para classificação de modulação e predição da OSNR com faixa de erro de $\pm 0,5$ dB/0,1 nm com o modelo *Perceptron Multicamada*, demonstrando a viabilidade e eficácia da abordagem proposta.

Palavras-chave: Classificação Automática de Modulação, Redes Ópticas, *Random Forest*, *Perceptron Multicamada*.

Abstract of Dissertation presented to UFAM as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering

AUTOMATIC MODULATION CLASSIFICATION IN FLEXIBLE COHERENT OPTICAL RECEIVERS

Antonio Marcos da Costa Pereira

Advisors: Waldir Sabino da Silva Júnior

André Luiz Aguiar da Costa

Department: Postgraduate in Electrical Engineering

To keep pace with the increasing demand for higher transmission rates and improvements in internet connectivity, research is focusing on effective methods to optimize the use of optical networks. Such studies are crucial for developing systems autonomously gathering the necessary information to adjust transmission parameters. This includes choosing the most suitable modulations for the communication medium, deciding between single-carrier or multi-carrier transmission, and selecting the most efficient algorithms for signal regeneration and error correction. These approaches allow networks to automatically adjust to maximize performance and efficiency without direct human intervention. This dissertation investigates how machine learning techniques can be applied to automate parameter settings in optical communication systems, aiming to improve adaptability and efficiency by enabling autonomous adjustment of settings based on acquired information. Focusing on flexible coherent optical receivers, the study seeks to enhance system efficiency and adaptability by automating the process of modulation classification and OSNR value prediction, key elements for optimizing performance and reliability in optical data transmission. A back-to-back setup between simulated transmitters and receivers was used, generating 76,800 signals with DP-BPSK, DP-QPSK, DP-8-PSK, DP-16-QAM, DP-32-QAM, and DP-64-QAM modulations across 51 different OSNR levels. Ensemble algorithms AdaBoost, CART Decision Tree, Gradient Boosting, Random Forest, and the Perceptron Multilayer neural network algorithm were employed for

modulation classification and OSNR value prediction. The results show accuracies over 99% for modulation classification and OSNR prediction within a $\pm 0.5\text{dB}/0.1\text{nm}$ error range with the Perceptron Multilayer model, demonstrating the viability and effectiveness of the proposed approach.

Keywords: Automatic Modulation Classification, Optical Networks, Random Forest, Multilayer Perceptron.

Sumário

1	Introdução	4
1.1	Objetivos da Dissertação	6
1.1.1	Objetivo geral	6
1.1.2	Objetivos específicos	6
1.2	Justificativa	7
1.3	Organização do trabalho	8
2	Fundamentos Teóricos	9
2.1	Conceitos básicos sobre aprendizado de máquina	9
2.1.1	Algoritmos para classificação, regressão e agrupamento	11
2.1.2	Aprendizado supervisionado	12
2.1.3	Ciclo de projeto de um classificador com aprendizado super- visionado	13
2.2	Motivações para a aplicação de aprendizado de máquina no contexto de redes ópticas	17
2.2.1	Conceitos básicos sobre redes ópticas	18
2.2.2	Desafios encontrados na camada física de transporte	20
2.2.3	Desafios encontrados na camada de enlace	22
3	Metodologia	24
3.1	Visão geral sobre o <i>framework</i> de simulação	24
3.2	Comportamento dos componentes ópticos	25
3.3	Características dos sinais transmitidos	30
3.4	Modelos de aprendizado	32
3.4.1	<i>Random Forest</i>	33

3.4.2	Árvore de Decisão <i>CART</i>	34
3.4.3	<i>AdaBoost</i>	35
3.4.4	<i>Gradient Boosting</i>	36
3.4.5	<i>Perceptron</i> Multicamada	37
3.5	Obtenção da Modulação e OSNR	38
4	Simulações e Resultados	41
4.1	<i>Framework</i> de simulação proposto	41
4.2	Apresentação dos resultados	43
4.2.1	Classificação do tipo de modulação	44
4.2.2	Predição do valor de OSNR	57
4.3	Análise dos resultados	89
4.3.1	Pergunta 1: De que forma o desempenho individual de cada modelo pode ser avaliado?	89
4.3.2	Pergunta 2: Que critério pode ser utilizado para eleger o resultado de um modelo como o melhor na estratégia aplicada?	90
4.3.3	Pergunta 3: O que pode ser concluído a partir dos resultados das simulações e experimentos realizados?	97
5	Conclusão	98
5.1	Estudos futuros	99
	Referências Bibliográficas	101
A	Artigos Publicados	108

Lista de Figuras

2.1	Etapas para a realização do ciclo de aprendizado de um modelo supervisionado.	13
3.1	Diagrama em blocos do sistema óptico proposto.	25
3.2	Diagrama de blocos das funcionalidades do <i>framework</i> proposto, sendo Transmissor, Receptor e DSP.	26
3.3	Diagrama de blocos que compõem a funcionalidade do Transmissor.	26
3.4	Diagrama de blocos que compõem a funcionalidade do Receptor.	27
3.5	Diagrama de blocos que compõem a funcionalidade do DSP.	28
3.6	Relação entre BER e OSNR para os tipos de modulação utilizados no transmissor.	30
3.7	Sinais em fase e quadratura coletados para a modulação 64-QAM em 25dB/0,1nm de OSNR.	31
3.8	Sinais em fase e quadratura, e histogramas relacionados para as modulações utilizadas, em seus valores máximos de OSNR.	32
3.9	Blocos propostos para realizar a <i>Obtenção da Modulação e OSNR</i>	39
4.1	Bloco proposto para classificação do tipo de modulação.	44
4.2	Matriz de confusão representando o melhor desempenho do classificador <i>Random Forest</i>	45
4.3	Progressão dos resultados para o classificador <i>Random Forest</i> com a variação de parâmetros do modelo.	46
4.4	Matriz de confusão representando o melhor desempenho do classificador <i>AdaBoost</i>	47
4.5	Progressão dos resultados para o classificador <i>AdaBoost</i> com a variação de parâmetros do modelo.	48

4.6	Matriz de confusão representando o melhor desempenho do classificador <i>Árvore de Decisão CART</i>	49
4.7	Progressão dos resultados para o classificador <i>Árvore de Decisão CART</i> com a variação de parâmetros do modelo.	51
4.8	Matriz de confusão representando o melhor desempenho do classificador <i>Gradient Boosting</i>	52
4.9	Progressão dos resultados para o classificador <i>Gradient Boosting</i> com a variação de parâmetros do modelo.	53
4.10	Matriz de confusão representando o melhor desempenho do classificador <i>Perceptron Multicamada</i>	55
4.11	Progressão dos resultados para o classificador <i>Perceptron Multicamada</i> com a variação de parâmetros do modelo.	56
4.12	Bloco proposto para a predição do valor de <i>OSNR</i>	57
4.13	Resultados com melhor desempenho para o preditor de <i>OSNR</i> na modulação BPSK usando <i>Random Forest</i>	58
4.14	Resultados com melhor desempenho para o preditor de <i>OSNR</i> na modulação QPSK usando <i>Random Forest</i>	60
4.15	Resultados com melhor desempenho para o preditor de <i>OSNR</i> na modulação 8-PSK usando <i>Random Forest</i>	62
4.16	Resultados com melhor desempenho para o preditor de <i>OSNR</i> na modulação 16-QAM usando <i>Random Forest</i>	64
4.17	Resultados com melhor desempenho para o preditor de <i>OSNR</i> na modulação 32-QAM usando <i>Random Forest</i>	65
4.18	Resultados com melhor desempenho para o preditor de <i>OSNR</i> na modulação 64-QAM usando <i>Random Forest</i>	67
4.19	Resultados com melhor desempenho para o preditor de <i>OSNR</i> na modulação BPSK usando <i>Perceptron Multicamada</i>	69
4.20	Resultados com melhor desempenho para o preditor de <i>OSNR</i> na modulação QPSK usando <i>Perceptron Multicamada</i>	70
4.21	Resultados com melhor desempenho para o preditor de <i>OSNR</i> na modulação 8-PSK usando <i>Perceptron Multicamada</i>	72

4.22	Resultados com melhor desempenho para o preditor de OSNR na modulação 16-QAM usando <i>Perceptron Multicamada</i>	74
4.23	Resultados com melhor desempenho para o preditor de OSNR na modulação 32-QAM usando <i>Perceptron Multicamada</i>	76
4.24	Resultados com melhor desempenho para o preditor de OSNR na modulação 64-QAM usando <i>Perceptron Multicamada</i>	78
4.25	Resultados com melhor desempenho para o preditor de OSNR na modulação BPSK usando <i>Random Forest</i> como regressor.	80
4.26	Resultados com melhor desempenho para o preditor de OSNR na modulação QPSK usando <i>Random Forest</i> como regressor.	81
4.27	Resultados com melhor desempenho para o preditor de OSNR na modulação 8-PSK usando <i>Random Forest</i> como regressor.	83
4.28	Resultados com melhor desempenho para o preditor de OSNR na modulação 16-QAM usando <i>Random Forest</i> como regressor.	84
4.29	Resultados com melhor desempenho para o preditor de OSNR na modulação 32-QAM usando <i>Random Forest</i> como regressor.	86
4.30	Resultados com melhor desempenho para o preditor de OSNR na modulação 64-QAM usando <i>Random Forest</i> como regressor.	88

Lista de Tabelas

4.1	Distribuição dos sinais coletados por modulação.	42
4.2	Planejamento da exploração dos parâmetros de treinamento para cada modelo selecionado.	43
4.3	Detalhamento sobre os conjuntos de parâmetros definidos para exploração.	43
4.4	Parâmetros utilizados no treinamento do modelo <i>Random Forest</i> com melhor desempenho para a classificação da modulação.	45
4.5	Conjunto de parâmetros explorados durante os experimentos com o modelo <i>Random Forest</i>	45
4.6	Resultados obtidos a partir dos experimentos com o modelo <i>Random Forest</i>	46
4.7	Parâmetros utilizados no treinamento do modelo <i>AdaBoost</i> com melhor desempenho para a classificação da modulação.	48
4.8	Conjunto de parâmetros explorados durante os experimentos com o modelo <i>AdaBoost</i>	49
4.9	Resultados obtidos a partir dos experimentos com o modelo <i>AdaBoost</i>	49
4.10	Parâmetros utilizados no treinamento do modelo <i>Árvore de Decisão CART</i> com melhor desempenho para a classificação da modulação.	50
4.11	Conjunto de parâmetros explorados durante os experimentos com o modelo <i>Árvore de Decisão CART</i>	50
4.12	Resultados obtidos a partir dos experimentos com o modelo <i>Árvore de Decisão CART</i>	51
4.13	Parâmetros utilizados no treinamento do modelo <i>Gradient Boosting</i> com melhor desempenho para a classificação da modulação.	53

4.14	Conjunto de parâmetros explorados durante os experimentos com o modelo <i>Gradient Boosting</i>	54
4.15	Resultados obtidos a partir dos experimentos com o modelo <i>Gradient Boosting</i>	54
4.16	Parâmetros utilizados no treinamento do modelo <i>Perceptron Multicamada</i> com melhor desempenho para a classificação da modulação. . .	55
4.17	Conjunto de parâmetros explorados durante os experimentos com o modelo <i>Gradient Boosting</i>	56
4.18	Resultados obtidos a partir dos experimentos com o modelo <i>Perceptron Multicamada</i>	57
4.19	Parâmetros ótimos utilizados no treino do preditor de OSNR para modulação BPSK com <i>Random Forest</i>	59
4.20	Resultados obtidos a partir dos experimentos realizados com o classificador <i>Random Forest</i> para modulação BPSK.	59
4.21	Parâmetros ótimos utilizados no treino do preditor de OSNR para modulação QPSK com <i>Random Forest</i>	61
4.22	Resultados obtidos a partir dos experimentos realizados com o classificador <i>Random Forest</i> para modulação QPSK.	61
4.23	Parâmetros ótimos utilizados no treino do preditor de OSNR para modulação 8-PSK com <i>Random Forest</i>	62
4.24	Resultados obtidos a partir dos experimentos realizados com o classificador <i>Random Forest</i> para modulação 8-PSK.	63
4.25	Parâmetros ótimos utilizados no treino do preditor de OSNR para modulação 16-QAM com <i>Random Forest</i>	63
4.26	Resultados obtidos a partir dos experimentos realizados com o classificador <i>Random Forest</i> para modulação 16-QAM.	64
4.27	Parâmetros ótimos utilizados no treino do preditor de OSNR para modulação 32-QAM com <i>Random Forest</i>	66
4.28	Resultados obtidos a partir dos experimentos realizados com o classificador <i>Random Forest</i> para modulação 32-QAM.	66
4.29	Parâmetros ótimos utilizados no treino do preditor de OSNR para modulação 64-QAM com <i>Random Forest</i>	67

4.30	Resultados obtidos a partir dos experimentos realizados com o classificador <i>Random Forest</i> para modulação 64-QAM.	68
4.31	Parâmetros utilizados no treino do preditor de OSNR para modulação BPSK, do algoritmo <i>Perceptron Multicamada</i>	68
4.32	Resultados obtidos a partir dos experimentos com o classificador <i>Perceptron Multicamada</i> na modulação BPSK.	70
4.33	Parâmetros utilizados no treino do preditor de OSNR para modulação QPSK, do algoritmo <i>Perceptron Multicamada</i>	71
4.34	Resultados obtidos a partir dos experimentos com o classificador <i>Perceptron Multicamada</i> na modulação QPSK.	71
4.35	Parâmetros utilizados no treino do preditor de OSNR para modulação 8-PSK, do algoritmo <i>Perceptron Multicamada</i>	73
4.36	Resultados obtidos a partir dos experimentos com o classificador <i>Perceptron Multicamada</i> na modulação 8-PSK.	73
4.37	Parâmetros utilizados no treino do preditor de OSNR para modulação 16-QAM, do algoritmo <i>Perceptron Multicamada</i>	75
4.38	Resultados obtidos a partir dos experimentos com o classificador <i>Perceptron Multicamada</i> na modulação 16-QAM.	75
4.39	Resultados obtidos a partir dos experimentos com o classificador <i>Perceptron Multicamada</i> na modulação 32-QAM.	76
4.40	Parâmetros utilizados no treino do preditor de OSNR para modulação 32-QAM, do algoritmo <i>Perceptron Multicamada</i>	77
4.41	Resultados obtidos a partir dos experimentos com o classificador <i>Perceptron Multicamada</i> na modulação 64-QAM.	77
4.42	Parâmetros utilizados no treino do preditor de OSNR para modulação 64-QAM, do algoritmo <i>Perceptron Multicamada</i>	79
4.43	Parâmetros utilizados no treino do preditor de OSNR com melhor desempenho em BPSK, com <i>Random Forest</i> como regressor.	79
4.44	Resultados obtidos a partir dos experimentos realizados com <i>Random Forest</i> como regressor para modulação BPSK.	80
4.45	Parâmetros utilizados no treino do preditor de OSNR com melhor desempenho em QPSK, com <i>Random Forest</i> como regressor.	82

4.46	Resultados obtidos a partir dos experimentos realizados com <i>Random Forest</i> como regressor para modulação QPSK.	82
4.47	Parâmetros utilizados no treino do preditor de OSNR com melhor desempenho em 8-PSK, com <i>Random Forest</i> como regressor.	83
4.48	Resultados obtidos a partir dos experimentos realizados com <i>Random Forest</i> como regressor para modulação 8-PSK.	84
4.49	Parâmetros utilizados no treino do preditor de OSNR com melhor desempenho em 16-QAM, com <i>Random Forest</i> como regressor.	85
4.50	Resultados obtidos a partir dos experimentos realizados com <i>Random Forest</i> como regressor para modulação 16-QAM.	85
4.51	Parâmetros utilizados no treino do preditor de OSNR com melhor desempenho em 32-QAM, com <i>Random Forest</i> como regressor.	86
4.52	Resultados obtidos a partir dos experimentos realizados com <i>Random Forest</i> como regressor para modulação 32-QAM.	87
4.53	Parâmetros utilizados no treino do preditor de OSNR com melhor desempenho em 64-QAM, com <i>Random Forest</i> como regressor.	87
4.54	Resultados obtidos a partir dos experimentos realizados com <i>Random Forest</i> como regressor para modulação 64-QAM.	88
4.55	Resumo dos resultados para a classificação do tipo de modulação.	90
4.56	Resumo dos resultados para a predição do valor de OSNR com o classificador.	91
4.57	Resumo dos resultados para a predição do valor de OSNR com o regressor.	91
4.58	Resumo dos resultados para a predição do valor de OSNR com a rede neural.	91
4.59	Melhores resultados entre o uso do classificador e rede neural para a predição do valor de OSNR.	92
4.60	Comparativo dos resultados entre regressor e rede neural, para a predição do valor de OSNR em BPSK.	93
4.61	Comparativo dos resultados entre regressor e rede neural, para a predição do valor de OSNR em QPSK.	94

4.62	Comparativo dos resultados entre regressor e rede neural, para a predição do valor de OSNR em 8-PSK.	94
4.63	Comparativo dos resultados entre regressor e rede neural, para a predição do valor de OSNR em 16-QAM.	95
4.64	Comparativo dos resultados entre regressor e rede neural, para a predição do valor de OSNR em 32-QAM.	96
4.65	Comparativo dos resultados entre regressor e rede neural, para a predição do valor de OSNR em 64-QAM.	96

Lista de Siglas

- 16-QAM** Modulação de Amplitude em Quadratura com 16 estados. 6
- 32-QAM** Modulação de Amplitude em Quadratura com 32 estados. 6
- 5G** Quinta Geração de Redes Móveis. 4, 7
- 64-QAM** Modulação de Amplitude em Quadratura com 64 estados. 6
- 6G** Sexta Geração de Redes Móveis. 4, 7
- 8-PSK** Modulação por Deslocamento de Fase com 8 estados. 6
- ADC** Conversor Analógico-Digital. 29
- AOV** Atenuador Óptico Variável. 28
- BER** Taxa e Erro de Bit. xii, 21, 28, 30
- BPSK** Modulação por Deslocamento de Fase Binária. 6, 20
- CART** Árvore de Classificação e Regressão. vii, xi, 6, 32, 34, 35, 98
- DBSCAN** Agrupamento Espacial Baseado em Densidade de Aplicações com Ruído. 12
- DP-16-QAM** Modulação de Amplitude em Quadratura em 16 estados com Dupla Polarização. vii, 25, 30, 98
- DP-32-QAM** Modulação de Amplitude em Quadratura em 32 estados com Dupla Polarização. vii, 25, 30, 98

DP-64-QAM Modulação de Amplitude em Quadratura em 64 estados com Dupla Polarização. vii, 25, 30, 98

DP-8-PSK Modulação por Deslocamento de Fase em 8 estados com Dupla Polarização. vi, 25, 30, 98

DP-BPSK Modulação por Deslocamento de Fase Binária com Dupla Polarização. vi, 25, 30, 98

DP-m-PSK Modulação por Deslocamento de Fase em m-estados com Dupla Polarização. 6

DP-m-QAM Modulação de Amplitude em Quadratura em m-estados com Dupla Polarização. 6

DP-QPSK Modulação por Deslocamento de Fase em Quadratura com Dupla Polarização. vi, 25, 30, 98

DSP Processador Digital de Sinais. xii, 19, 25, 26, 28

EDFA Amplificador de Fibra Dopada com Érbio. 19

FPFO Filtro Passa-Faixa Óptico. 28

IA Inteligência Artificial. 5, 9, 11

LED Diodo Emissor de Luz. 18

LPF Filtro Passa-Baixa. 28

ML Aprendizado de Máquina. 5

OFDM Multiplexação por Divisão de Frequência Ortogonal. 7, 19, 20

OSNR Relação Sinal-Ruído Óptica. vi, vii, xi–xiv, xvi–xix, 5–8, 22, 24, 25, 27–32, 38–40, 42, 43, 58–99

OTN Rede de Transporte Óptico. 4

PDM Multiplexação por Divisão de Polarização. 20

PSK Modulação por Deslocamento de Fase. 18–20

QAM Modulação de Amplitude em Quadratura. 18–20

QPSK Modulação por Deslocamento de Fase em Quadratura. 6, 20

RAN Rede de Acesso por Rádio. 4

ROADM Multiplexador Óptico de Adição/Remoção Reconfigurável. 7

SDN Rede Definida por Software. 22, 23

SVM Máquina de Vetores de Suporte. 13

VoIP Tecnologia de Voz sobre Protocolo de Internet. 4

WSS Interruptor Seletivo de Comprimento de Onda. 7

Capítulo 1

Introdução

O impacto significativo do uso da internet é evidente no cotidiano da nossa sociedade. Atualmente, podemos assistir a vídeos em streaming, acessar redes sociais, fazer chamadas via VoIP e utilizar serviços de jogos em nuvem, entre outros exemplos, de qualquer lugar, graças a *smartphones* e outros dispositivos móveis. Estes dispositivos estão em constante evolução, tornado-se cada vez mais sofisticados e compactos. Dessa forma, os *smartphones* demonstram um reflexo direto sobre as transformações contínuas na tecnologia e na conectividade, assumindo um papel cada vez mais integrado e essencial em nosso cotidiano [1, 2]. Como resposta a isso, a introdução de redes 5G está em andamento [3], com um cronograma de implementação previsto no Brasil [4]. O uso do 5G promete redes com altas taxas de transmissão, latência ultra baixa e uma melhoria significativa na qualidade de serviço percebida pelos usuários [5]. A migração para a tecnologia 5G representa uma mudança significativa no uso e na estrutura das Redes de Acesso por Rádio (RAN) [6, 7]. Esta transição envolve importantes alterações arquiteturais, incluindo a descentralização das antenas e a capacitação dos dispositivos para uma distribuição mais eficiente dos sinais [6]. Mudanças também podem ser observadas na distribuição otimizada de recursos de rede e pontos de acesso, como a distribuição de fibras para cobertura de sinal [5, 7]. Em suma, a adoção da tecnologia 5G, em futuro próximo o 6G e avanço em tecnologias associadas às redes de acesso por rádio realçam a importância de um uso otimizado dos recursos das Redes de Transporte Óptico (OTN), uma vez que o emprego de tais tecnologias necessitará cada vez mais de banda passante. [5, 8]. Isso implica na necessidade de infraestruturas de rede ro-

bustas e eficientes, capazes de suportar o volume crescente de dados. [9–11], o que é essencial para garantir que as redes possam lidar com a demanda elevada, mantendo a qualidade e a velocidade do serviço. [11–14].

No cenário de telecomunicações, os avanços em inteligência artificial (IA) [12, 15] e aprendizado de máquina (ML) [16, 17] têm oferecido soluções promissoras para desafios na área de telecomunicações, em especial para comunicações ópticas conforme demonstrado em pesquisas recentes [17–19]. Paralelamente, é fundamental compreender as tendências atuais e se adaptar à crescente demanda por transmissão de dados de alta velocidade e conectividade aprimorada à internet. Nesse contexto, a eficiência das redes ópticas emergiu como um campo de pesquisa proeminente, abordando essas necessidades cruciais [7, 13, 15–23]. Os desafios primários incluem a identificação e mitigação de degradações causadas por efeitos lineares e não lineares [16–18], a avaliação da qualidade da transmissão óptica [21] e a classificação automática de modulações [22, 23]. Entre esses estudos, aqueles que envolvem técnicas de classificação automática de modulações fornecem meios para que os controladores em redes autônomas colem informações pertinentes [24]. Essas informações capacitam as redes a definir parâmetros de transmissão, como modulações ideais para o meio de transmissão, algoritmos para recuperação de frequência e fase, e a escolha entre transmissão de portadora única ou múltipla, bem como algoritmos de correção de erros [18, 25].

No escopo de comunicações ópticas [18] e da classificação automática de modulações [23], esta dissertação visa propor um método para a classificação automática de modulações e predição do valor de OSNR (Relação Sinal-Ruído Óptica) em receptores ópticos coerentes flexíveis, utilizando algoritmos de aprendizado de máquina [15, 16, 26, 27]. Essa abordagem aproveita as capacidades de aprendizado e generalização dos algoritmos de aprendizado de máquina para desenvolver um sistema capaz de identificar automaticamente a modulação usada em um sinal óptico e prever seu valor de OSNR. Ao utilizar técnicas de aprendizado de máquina, é possível treinar um modelo com base em um conjunto de dados, onde os sinais ópticos são rotulados com as modulações e valores de OSNR correspondentes. O modelo resultante pode então ser usado para classificar modulações e prever o valor de OSNR em sinais ópticos não rotulados, tornando o processo automatizado e menos depen-

dente da influência humana. Para isso, é apresentado um estudo sobre a aplicação de algoritmos de aprendizado de máquina *ensemble*, incluindo *AdaBoost*, Árvore de decisão *CART*, *Gradient Boosting* e *Random Forest*, e o algoritmo de redes neurais artificiais, *Perceptron* Multicamada para a classificação automática de modulação e predição do valor de OSNR em receptores ópticos coerentes flexíveis. O estudo utilizou um *setup back-to-back* entre um transmissor e um receptor, no contexto de comunicações ópticas. Nesse ambiente de comunicação, 76.800 sinais foram simulados com modulações DP-m-PSK e DP-m-QAM em 51 níveis de ruído distintos. Após a conclusão da fase de treinamento, os modelos demonstraram uma precisão superior a 99% na classificação das modulações. Além disso, as previsões de OSNR atingiram uma acurácia acima de 90%, com uma margem de erro de $\pm 0.5\text{dB}/0.1\text{nm}$, evidenciando o potencial promissor da abordagem investigada nesta pesquisa.

1.1 Objetivos da Dissertação

1.1.1 Objetivo geral

O principal objetivo desta dissertação é investigar o desempenho de modelos de aprendizado de máquina no contexto de um sistema de comunicação óptica. Através da aplicação de uma estratégia para a obtenção de informações sobre a modulação utilizada e o valor de OSNR, sem o conhecimento prévio de informações sobre o sinal no lado do receptor óptico, e demonstrar os resultados obtidos. Para isso, propomos um *framework* para a simulação de um sistema de comunicação óptica, capaz de permitir a configuração simplificada de parâmetros, como o tipo de modulação. Permitindo escolhas entre as modulações BPSK, QPSK, 8-PSK, 16-QAM, 32-QAM e 64-QAM em 51 níveis diferentes de OSNR, e então utilizar os sinais coletados neste *framework* para treinamento e avaliação do desempenho dos modelos de aprendizado de máquina.

1.1.2 Objetivos específicos

- Definir e parametrizar o *setup* do *framework* para cada modulação planejada, validando e armazenando os resultados obtidos.

- Implementar e analisar os resultados dos algoritmos *ensemble* e redes neurais para a classificação do tipo de modulação no contexto de comunicações ópticas do *framework*.
- Implementar e analisar os resultados dos algoritmos *ensemble* de classificação, regressão e redes neurais para predição do valor de OSNR no contexto de comunicações ópticas do *framework*.
- Obter um comparativo entre os resultados obtidos para diferentes algoritmos de aprendizado de máquina, demonstrando através das métricas selecionadas, aqueles que apresentam o melhor desempenho.

1.2 Justificativa

A utilização das redes ópticas nas telecomunicações é fundamental para atender à crescente demanda por taxas de comunicação e melhorias de conectividade com a internet. Ao empregar técnicas de otimização espectral, como o OFDM, o uso de fibras ópticas permite alcançar elevadas taxas de transmissão. Adicionalmente, a incorporação de tecnologias como ROADM e WSS, facilitam a reconfiguração flexível e eficiente de canais e caminhos ópticos. Essas tecnologias, em conjunto com outras vantagens das fibras ópticas, contribuem para uma cobertura global extensiva, representando cerca de 95% do tráfego de dados global por meio de cabos submarinos [28], ressaltando a importância das redes ópticas na infraestrutura de comunicação mundial. No contexto brasileiro, a relevância das redes ópticas se intensifica diante da necessidade de atender às exigências das tecnologias móveis emergentes, como o 5G e o futuro 6G. Essas redes de acesso móvel demandam infraestruturas de comunicação óptica avançadas para suportar altas taxas de transferência de dados, baixa latência e elevada qualidade de serviço. A relevância das redes ópticas também possui destaque em âmbito local, podendo ser comentada a competição entre provedores de serviços em busca da expansão de novas infovias [29]. Visando estender a infraestrutura de fibra óptica de alta qualidade para regiões remotas, como o interior do Amazonas, promovendo formas de melhorar significativamente o acesso e a qualidade da conectividade em áreas menos atendidas. Diante deste contexto, este trabalho visa propor métodos que permitam a configuração autônoma em redes

ópticas, através da obtenção de informações como o tipo de modulação e valor de OSNR, desconhecidas pelo sistema no lado do receptor. Dessa forma, provendo informações necessárias ao controlador de rede para ajustar configurações como tipos de modulação, definir o caminho óptico ótimo, e otimizar o ganho do amplificador, possibilitando uma rede verdadeiramente autônoma, sem necessidade de intervenção humana.

1.3 Organização do trabalho

A organização deste trabalho é observada conforme a estrutura a seguir:

- No Capítulo 2, são revisados os conceitos fundamentais utilizados nesta dissertação. Englobando conceitos sobre os algoritmos de aprendizado de máquina e treinamento de modelos. A seguir, são abordados os conceitos-chave que implicam na complexidade e motivações para o uso de aprendizado de máquina no contexto de comunicações ópticas.
- No Capítulo 3, é apresentado o detalhamento sobre a metodologia utilizada para o desenvolvimento do *framework* de simulação da comunicação óptica. Abordando a simulação da transmissão, validação dos sinais recebidos e geração de uma base de dados rotulada. A seguir, a etapa de extração de características é apresentada, e por fim temos a apresentação das estratégias em cascata para a classificação do tipo de modulação, e a partir desta informação a seleção do modelo apropriado para a predição do valor de OSNR.
- No Capítulo 4, é realizada a descrição do detalhamento dos resultados obtidos, organizados em resultados de *framework*, resultados para a estratégia de classificação do tipo de modulação e por fim, para a estratégia para a predição do valor de OSNR. A seguir, temos a análise dos resultados obtidos e a resposta a questionamentos sobre estes resultados.
- No Capítulo 5, são apresentadas as considerações finais e sugestões de trabalhos futuros.

Capítulo 2

Fundamentos Teóricos

2.1 Conceitos básicos sobre aprendizado de máquina

Aprendizado de máquina (*machine learning*), uma subárea da inteligência artificial (IA), foca em desenvolver métodos que permitem aos computadores executar tarefas não diretamente programadas [30–32]. Este campo contrasta com a programação tradicional: enquanto a programação convencional requer que um desenvolvedor escreva algoritmos específicos para resolver problemas [33, 34], o aprendizado de máquina utiliza dados e experiências anteriores de maneira que o próprio sistema identifique padrões e aprenda com eles [35]. Para compreender melhor o aprendizado de máquina, é essencial definir alguns termos-chave. Exploremos conceitos como *modelo*, *algoritmo*, *dados*, *características*, *rótulos*, *treinamento*, *teste*, *overfitting*, *underfitting* e *validação cruzada*. Cada um desses termos desempenha um papel crucial no aprendizado de máquina, desde a construção do modelo até a avaliação de seu desempenho. Em aprendizado de máquina, um *modelo* é uma representação, como um conjunto de equações, regras ou estruturas, derivada dos dados de treinamento. Sua função principal é estabelecer uma relação entre as características (dados de entrada) e os rótulos (resultados ou saídas desejadas) [36, 37]. Para desenvolver um modelo de aprendizado de máquina, é crucial definir o *algoritmo* utilizado, o qual é um conjunto de instruções específicas. Neste contexto, o *algoritmo* determina o tipo de aprendizado a ser aplicado e define o comporta-

mento esperado do modelo após o treinamento. Essa estrutura orienta o processo pelo qual o modelo aprende a realizar classificações ou fazer previsões com base nos dados fornecidos [30,36]. No contexto do aprendizado de máquina, *dados* referem-se ao conjunto completo de informações coletadas de uma fonte confiável e reproduzível, essenciais para o treinamento de modelos [30,38]. Esses *dados* são tipicamente divididos em duas categorias principais: as *características* são as medições ou propriedades observadas durante a ocorrência de um evento, e os *rótulos* representam as respostas desejadas ao evento. Essa divisão facilita o processo de treinamento, permitindo que o modelo aprenda a associar as características de um evento com seu rótulo correspondente [30,38]. O *treinamento* em aprendizado de máquina começa com a utilização do conjunto de dados de entrada, aplicado juntamente com as diretrizes estabelecidas pelo algoritmo escolhido. Nessa fase, o objetivo é criar um modelo que aprenda a correlacionar as características dos eventos com seus rótulos correspondentes. O aprendizado dos padrões subjacentes aos eventos permite que o modelo desenvolvido possa realizar previsões ou tomar decisões informadas com base em novos dados [34]. O *teste* em aprendizado de máquina é uma etapa crucial para validar a eficácia do modelo treinado. Durante o *teste*, o modelo é exposto a um novo conjunto de dados, diferente daquele utilizado no treinamento. Essa prática é essencial para avaliar se o modelo pode generalizar seu aprendizado e aplicá-lo efetivamente a situações inéditas, garantindo assim a confiabilidade e a precisão das previsões ou decisões que ele faz [36]. Em aprendizado de máquina, é importante considerar dois fenômenos comuns nos resultados do modelo: *overfitting* e *underfitting*. *Overfitting* ocorre quando um modelo aprende os dados de treinamento tão detalhadamente, incluindo ruídos e anomalias, que falha em generalizar novos dados, resultando em baixo desempenho em situações desconhecidas. Por outro lado, *underfitting* acontece quando o modelo não consegue captar adequadamente os padrões dos dados, levando a um desempenho fraco tanto nos dados de treinamento quanto em novos dados [36]. Para evitar o *overfitting* e *underfitting*, diversas metodologias podem ser aplicadas durante a fase de treinamento, visando aumentar a robustez do modelo. Estas metodologias incluem técnicas como a *validação cruzada*, que permite avaliar a generalização do modelo, e a *regulação*, que ajuda a prevenir o aprendizado excessivo dos detalhes dos dados de treino. O objetivo é garantir que

o modelo consiga realizar previsões precisas não apenas nos dados de treinamento, mas também em dados novos e desconhecidos [36]. Além dos conceitos básicos já definidos, o campo de aprendizado de máquina inclui muitas outras definições e ideias. Com esses fundamentos estabelecidos, estamos prontos para explorar tópicos mais avançados. Como uma área interdisciplinar, o aprendizado de máquina extrai conhecimentos de campos como cálculo estatístico, matemática e teoria da informação [30, 39]. Diferentemente da IA tradicional, seu foco não é replicar a cognição humana, mas sim transformar experiências e dados em conhecimento, identificando padrões complexos [30, 37]. A aplicação do aprendizado de máquina é ampla e diversificada, abrangendo, mas não se limitando a áreas como veículos autônomos, detecção de fraude, manutenção preditiva e otimização de linhas de produção e reconhecimento de padrões em comunicações ópticas [12, 16, 18, 21, 22, 40–42]. Nestes campos, o aprendizado de máquina demonstra ser uma ferramenta poderosa, oferecendo novas abordagens e soluções mais eficientes em comparação com métodos tradicionais. As tendências e benefícios do seu uso nessas áreas ilustram como o aprendizado de máquina pode superar as capacidades de técnicas convencionais na resolução de problemas complexos e na identificação de padrões sutis.

2.1.1 Algoritmos para classificação, regressão e agrupamento

A utilização de algoritmos de aprendizado de máquina é fundamental para aplicar regras e fórmulas derivadas de casos de sucesso anteriores, permitindo a automação e otimização de processos em diversas áreas. Existem diferentes tipos de algoritmos, cada um adequado a problemas com características específicas [17, 35]. Por exemplo: Os classificadores em aprendizado de máquina são modelos que categorizam dados em classes distintas [34]. Eles são amplamente utilizados para tarefas como reconhecimento de padrões e classificação de imagens. Esses modelos analisam características de entrada e atribuem um rótulo de uma das categorias pré-definidas. Exemplos comuns incluem árvores de decisão, que dividem os dados em ramificações baseadas em critérios específicos, e redes neurais, que simulam o funcionamento do cérebro humano para classificar dados complexos [30, 34, 36]. Já os regressores são modelos de aprendizado de máquina utilizados para prever valores contínuos [30]. Eles são essenciais em tarefas que envolvem a previsão de valores numéricas, como

preços de ações ou temperaturas. Diferentemente dos classificadores, que preveem categorias discretas, os regressores lidam com saídas que são contínuas e variáveis. Exemplos notáveis incluem regressão linear, que estabelece uma relação linear entre variáveis de entrada e saída, e regressão polinomial, adequada para modelar relações mais complexas [30, 34, 36]. Os agrupadores, também conhecidos como clusterizadores são algoritmos que agrupam um conjunto de objetos de modo que objetos do mesmo grupo (*cluster*) sejam mais similares entre si do que com os de outros grupos [36]. Na comunicação óptica, por exemplo, podem ser usados para identificar padrões ou anomalias nos dados de transmissão. Como exemplos deste tipo de algoritmos temos o *K-Means*, sendo um dos algoritmos de clusterização mais conhecidos e utilizados. Ele agrupa os dados em um número pré-definido de clusters, com base na proximidade das observações aos centros dos clusters, minimizando a variância dentro de cada cluster, e *DBSCAN* (*Density-Based Spatial Clustering of Applications with Noise*), que diferentemente do *K-Means*, não requer a especificação do número de clusters. Ele identifica *clusters* baseando-se na densidade espacial dos dados, sendo particularmente eficaz para identificar *clusters* de formas irregulares e lidar com *outliers*, sendo descritos como dados com diferença significativa em relação aos demais elementos com as mesmas características [30, 34, 36].

2.1.2 Aprendizado supervisionado

Dentre os tipos mais populares de aprendizado de máquina, destaca-se o aprendizado supervisionado, não supervisionado, semi-supervisionado e por reforço, cada qual com suas vantagens, desvantagens e situações ideais de uso [30, 36]. De modo geral, no aprendizado supervisionado, a influência humana é crucial durante o treinamento, com o programador fornecendo rótulos para os dados de entrada [36]. No aprendizado não supervisionado, os dados não possuem rótulos pré-definidos, e o modelo é responsável por descobrir padrões e agrupar os dados [30, 36]. No aprendizado semi-supervisionado, combinam-se técnicas de aprendizado supervisionado e não supervisionado. Nesse método, o modelo é treinado com um conjunto de dados que contém tanto exemplos rotulados quanto não rotulados. Isso permite que o modelo aprenda padrões dos dados não rotulados enquanto utiliza os dados rotulados para melhorar a precisão [36]. Já no aprendizado por reforço, o foco é no

desenvolvimento de agentes capazes de tomar decisões em um ambiente para alcançar um determinado objetivo. O agente aprende a partir das consequências de suas ações, recebendo recompensas ou penalidades, guiando o processo de aprendizado para comportamentos mais eficientes em termos de alcançar a meta definida [36]. O aprendizado supervisionado é recomendado quando existem conjuntos de dados com rótulos precisos, possibilitando a modelagem eficiente para previsões ou classificações específicas [36]. Essa abordagem é ideal para situações onde o objetivo é claro, como na classificação de imagens, previsão de valores financeiros ou diagnósticos médicos, aproveitando dados históricos rotulados para aumentar a acurácia dos modelos. Além disso, o aprendizado supervisionado se aplica a uma diversidade de tipos de dados, incluindo numéricos, categóricos, textuais e imagéticos. A metodologia abrange estratégias para enfrentar desafios como o desbalanceamento de classes e o *overfitting*, implementando técnicas de ajuste de parâmetros e validação cruzada para refinar os modelos. A avaliação do desempenho desses modelos frequentemente envolve métricas como precisão, *recall* e *F1-score*, essenciais para mensurar a efetividade das previsões ou classificações realizadas [30, 36]. Exemplos de algoritmos supervisionados incluem kNN (*k-Nearest Neighbors*), Regressão Linear, SVM (*Support Vector Machine*), Árvores de Decisão e *Random Forest*.

2.1.3 Ciclo de projeto de um classificador com aprendizado supervisionado

Durante pesquisas envolvendo o uso de algoritmos de aprendizado de máquina, há etapas fundamentais que compõem o ciclo de desenvolvimento do modelo. Estas etapas são adaptáveis ao tipo de aprendizado empregado e devem ser avaliadas em busca de oportunidades de refinamento [36, 38]. A Figura 2.1 ilustra de maneira geral as etapas implementadas no processo de aprendizado em modelos supervisionados, com maiores detalhes sobre a descrição apresentados a seguir:

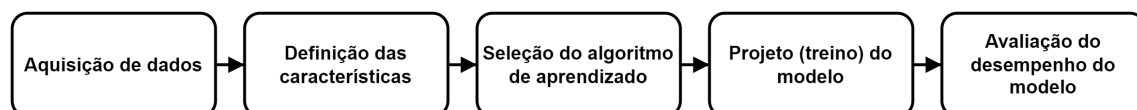


Figura 2.1: Etapas para a realização do ciclo de aprendizado de um modelo supervisionado.

- **Aquisição de dados:** A primeira etapa do ciclo de projeto é realizar a aquisição dos dados, sendo definida como a medição e persistência das informações disponíveis sobre o problema proposto, isso ocorre principalmente quando não existem dados previamente disponibilizados sobre o problema. No início da realização desta etapa, é feita uma análise do problema proposto, para direcionar a seleção dos atributos¹ base utilizados na solução do problema. Então, a partir destas definições, é iniciado o processo de aquisição, onde ocorre o registro de cada atributo selecionado, gerando uma instância na base de dados, este processo é repetido diversas vezes até o encerramento definido pelo pesquisador. Neste ponto, cabe a realização de uma segunda análise em relação ao problema proposto, para verificar a qualidade dos dados obtidos. De modo geral, uma base de dados é construída para possuir uma quantidade elevada de instâncias, diversidade de dados relativos às classes alvo do problema proposto e baixo nível de ruídos nos dados adquiridos [36].
- **Definição das características:** Foi definido que, durante a aquisição dos dados, são definidos os atributos-base, sendo estes originados nas grandezas observadas diretamente do problema proposto. Na etapa de definição das características, são criados conjuntos baseados nas relações existentes entre as características, as classes do problema, e a análise realizada pelo pesquisador. Em alguns casos, se faz necessário a geração de novas características, calculadas a partir das existentes, sendo exemplos deste processo, o uso dos histogramas e realização de transformações no domínio do tempo para frequência. Vale salientar que todas as características selecionadas são relacionadas com as instâncias, por isso, no caso de adição ou agrupamento, toda a base de dados obtida deve ser submetida ao processo e recondicionada [35].
- **Seleção do algoritmo de aprendizado:** A seleção do algoritmo de aprendizado é um passo crucial na modelagem de problemas que envolvem aprendizado de máquina. Nesta fase, após definir as características relevantes, o pesquisador avalia quais modelos têm potencial para oferecer soluções eficazes ao problema

¹*Atributos e características:* Iremos adotar o termo *atributo* como o tipo de dado, por exemplo *Cor*, enquanto o termo *característica* geralmente significa um atributo e seu valor, como *Cor = "azul"*, esta definição permite diferenciar ambos, mesmo que muitos autores utilizem os termos como sinônimos [36].

em questão. Este processo implica em uma análise detalhada das capacidades dos modelos em atender às exigências específicas do problema, permitindo uma compreensão aprofundada de suas potencialidades. A escolha do modelo não é definitiva; é um processo iterativo que pode ser ajustado conforme novas informações são adquiridas ou novos modelos são considerados, garantindo flexibilidade e adaptação ao longo do desenvolvimento do projeto [36].

- **Projeto (treino) do modelo:** Após as etapas anteriores, é realizada a etapa de projeto, onde o modelo selecionado recebe os dados disponíveis como entrada, e inicia o aprendizado. Nesta etapa dois conjuntos de dados são utilizados para o treino do modelo, sendo o conjunto de dados de treino e o conjunto de parâmetros do modelo. O conjunto de dados de treino é uma partição do conjunto de dados de entrada. Geralmente, utiliza-se uma fração de 70% do total de instâncias disponíveis para o treino, embora essa proporção não seja uma regra rígida. A forma e o tamanho da partição podem variar, cabendo ao pesquisador decidir a melhor estratégia para o seu caso específico. O objetivo é encontrar um equilíbrio adequado entre a quantidade de dados necessária para treinar o modelo e a quantidade de dados necessária para avaliar sua performance de forma eficaz. O segundo conjunto se refere à escolha dos parâmetros do modelo, e é geralmente iniciada com valores padrão, relativos a cada modelo, e conforme a avaliação dos resultados, são necessários ajustes nos valores escolhidos. Cada modelo possui um conjunto de parâmetros e a escolha destes valores influencia os resultados obtidos após a etapa de treino do modelo, cabendo ao pesquisador realizar uma avaliação crítica sobre os resultados obtidos e utilizar as ferramentas disponíveis para relacionar o conjunto de treino, modelo, conjunto de parâmetros do modelo e resultados encontrados [35].
- **Avaliação do desempenho do modelo:** Após a obtenção de um modelo treinado, cabe ao pesquisador realizar avaliações sobre o desempenho do modelo, podendo ser citada a avaliação em relação aos dados de treino utilizados e outra em relação a novos dados de entrada. Na primeira análise, o pesquisador faz a seleção de uma forma de avaliação de desempenho compatível com o cenário descrito pelo problema proposto, por exemplo, a matriz de confusão é uma tabela que relaciona as classes conhecidas com as decisões tomadas pelo

modelo treinado em relação aos dados de entrada utilizados. Neste caso, a matriz de confusão é amplamente utilizada como forma de avaliação de modelos utilizados em problemas de classificação. Então, ao inspecionar a matriz de confusão para o modelo treinado, é obtido o desempenho do modelo para este cenário, relacionando o conjunto de dados de treino e os parâmetros utilizados no treinamento, definindo assim o primeiro resultado. A partir deste momento, são feitas modificações no processo de aprendizado, por exemplo, modificar o valor de um parâmetro do modelo ou substituir características das instâncias do conjunto de treino, e após a conclusão de uma nova etapa de treino, é possível comparar os resultados obtidos na nova matriz de confusão com o primeiro resultado, o que permite encontrar as mudanças que favorecem novos resultados. Esta análise é repetida diversas vezes até que o pesquisador obtenha resultados que considere satisfatórios para as condições encontradas no problema proposto [35]. Com a obtenção de um modelo que apresenta resultados satisfatórios em relação ao conjunto de treino, se faz necessária a realização da segunda análise, onde dados diferentes dos dados utilizados para o treino são utilizados. Em geral, o conjunto de dados de entrada é particionado em conjunto de treino e teste, para a realização desta etapa existem diversas ferramentas que permitem ações úteis como realizar o sorteio das instâncias que farão parte de cada conjunto ou criar versões dos grupos, permitindo combinações distintas de conjuntos para a etapa de avaliação do modelo [30]. Uma vez obtido um modelo candidato, o conjunto de teste é utilizado na análise. Este conjunto foi obtido com os dados de treino, mas é esperado que este conjunto possua instâncias desconhecidas para o modelo, e que consequentemente não foram utilizadas no processo de treino e por isso visam testar o grau de generalização que o modelo obteve em seu treinamento. No contexto de aprendizado de máquina, o termo generalização se refere à capacidade de um modelo de atuar de maneira satisfatória ao receber dados que não estejam contidos no conjunto de treino, ou seja, ao encontrar dados nunca vistos pelo modelo. A forma de avaliação aplicada é semelhante à primeira avaliação, podendo ser aplicada a mesma avaliação, no caso a observação da matriz de confusão, porém, neste ponto o objetivo de avaliar o desempenho do modelo

é determinar qual o grau de generalização do mesmo, e em casos onde o modelo apresenta um desempenho inferior ao obtido com o conjunto de treino o mesmo pode voltar a etapa anterior, onde passa por novos ciclos de treino e avaliação. Uma das possíveis causas para esse baixo desempenho é conhecida como *overfitting*, e se trata de casos onde o treinamento faz com que o modelo se molde exclusivamente aos dados de treino, e tenha um desempenho ruim em relação a dados novos. Caso o modelo apresente resultados satisfatórios com o conjunto de treino, uma nova versão desta avaliação é executada, mas desta vez no cenário real, onde o problema proposto pode ser solucionado [43,44].

2.2 Motivações para a aplicação de aprendizado de máquina no contexto de redes ópticas

No contexto das telecomunicações, uma rede é fundamentalmente a infraestrutura necessária para o transporte de informações entre dois ou mais pontos [45,46]. Embora a definição básica de uma rede seja simples, a complexidade e o valor de uma rede surgem de suas características específicas. Estas incluem a capacidade de transporte de dados, a confiabilidade das informações transmitidas, os custos associados ao uso e a escalabilidade tecnológica. Estes aspectos são cruciais para diferentes tipos de redes e influenciam significativamente na sua eficácia e eficiência [47]. As redes ópticas se destacam principalmente por sua alta capacidade de transmissão de informações e baixa perda de potência, mesmo após longas distâncias. Essas características tornam as redes baseadas em fibra óptica ideais para comunicações de longa distância, como os *backbones*, utilizados para conexões interestaduais ou até mesmo intercontinentais. Graças a essas qualidades, as redes ópticas são fundamentais na infraestrutura de telecomunicações moderna [48]. A comunicação óptica utiliza meios para converter informações em pulsos de luz, transportados por fibras ópticas. Esses pulsos podem ser modulados para carregar dados digitais como texto, imagens ou vídeos. Dependendo das necessidades da aplicação, são escolhidos tipos específicos de fibras ópticas e fontes de luz [20]. As vantagens dos sistemas de comunicação óptica incluem a capacidade de transmitir dados a longas distâncias com pouca perda de sinal, taxas de transmissão mais altas

e maior segurança, devido à dificuldade de interceptar o sinal sem interromper a transmissão [49, 50].

2.2.1 Conceitos básicos sobre redes ópticas

Esta seção se dedica a esclarecer conceitos fundamentais e desafios associados às redes ópticas. Abordaremos elementos-chave dessas redes, incluindo as principais técnicas de modulação utilizadas e os desafios comuns enfrentados no seu uso. Esse entendimento é necessário para compreender a dinâmica e as potencialidades das redes ópticas.

2.2.1.1 Elementos de redes ópticas

Na configuração de uma rede óptica, diversos equipamentos e tecnologias são essenciais, dependendo do tipo de comunicação desejada. Em geral, as redes ópticas frequentemente incluem elementos como transmissores ópticos, fibras ópticas, amplificadores ópticos e receptores ópticos. Estes componentes são fundamentais para estabelecer uma comunicação óptica eficiente e confiável [49]. O transmissor óptico é um equipamento crucial em sistemas de comunicação óptica, responsável por converter sinais elétricos em sinais ópticos. Geralmente, utiliza diodos laser ou diodos emissores de luz (LED) como fontes de luz. LEDs são adequados para transmissões de curta distância, emitindo um feixe de luz mais amplo e menos intenso, enquanto os diodos laser, utilizados para longas distâncias, geram um feixe mais intenso e estreito, ideal para modulação precisa. Diversas técnicas de modulação, como QAM e PSK, podem ser selecionadas conforme a aplicação, variando conforme a distância de transmissão e volume de dados. [49]. Os cabos de fibra óptica são elementos-base de sistemas de comunicação óptica. Eles são transparentes, finos e flexíveis, geralmente feitos de sílica, vidro composto ou plástico de alta qualidade. As fibras ópticas são eficazes na transmissão de sinais de luz com mínima perda, mesmo a longas distâncias. Existem dois tipos principais: fibras monomodo, que transmitem um único feixe de luz, sendo ideais para longas distâncias devido à menor perda de sinal e maior largura de banda, e fibras multimodo, que transportam múltiplos feixes de luz e são mais adequadas para distâncias mais curtas [49]. Amplificadores ópticos são utilizados para amplificar sinais de luz ao longo de grandes distâncias. Um tipo

comum de amplificador óptico é o amplificador de fibra dopada com érbio (EDFA, do inglês *Erbium-Doped Fiber Amplifier*), que se destaca pela sua eficiência e capacidade de amplificar uma ampla gama de comprimentos de onda simultaneamente, minimizando a perda de sinal. Sendo fundamentais para manter a qualidade e a força do sinal em sistemas de comunicação óptica de longa distância [49]. O receptor óptico é um componente crucial em sistemas de comunicação óptica, responsável por converter sinais de luz de volta em sinais elétricos. Esses dispositivos utilizam fotodetectores para captar os sinais ópticos e os transformam em dados elétricos que podem ser processados por equipamentos eletrônicos [49]. Neste contexto, o fotodetector atua captando sinais de luz transmitidos pela fibra óptica e os converte em sinais elétricos. Os tipos mais comuns de fotodetectores incluem fotodiodos e fotodiodos de avalanche, cada um com características específicas para eficientemente transformar luz em sinais elétricos utilizáveis [49]. Além destes itens temos os multiplexadores e demultiplexadores ópticos são componentes fundamentais em redes de fibra óptica. Eles têm a função de combinar múltiplos sinais ópticos em um única fibra (multiplexação) e separá-los em sinais individuais (demultiplexação). Esses dispositivos permitem a transmissão eficiente de diferentes conjuntos de informações por meio de um único cabo de fibra óptica, otimizando o uso da infraestrutura de rede e aumentando a capacidade de transmissão de dados [49]. Por fim, temos o processador digital de sinais, (DSP, do inglês *Digital Signal Processing*), usado para processar sinais digitais após a conversão óptico-elétrica. Ele realiza funções como equalização, correção de distorção e extração de dados úteis dos sinais recebidos. O DSP é fundamental para garantir a qualidade e a integridade dos dados transmitidos [49].

2.2.1.2 Modulações utilizadas em redes ópticas

O uso das técnicas para modulação de sinais são fundamentais para a transmissão de informações em redes ópticas. Ao longo dos anos, estudos têm buscado propor melhorias nas modulações para aumentar sua eficiência. Entre os tipos de modulação mais utilizados podem ser citados a Modulação de Amplitude em Quadratura (QAM), Modulação por Deslocamento de Fase (PSK), Multiplexação por Divisão de Frequência Ortogonal (OFDM) e Multiplexação por Divisão de Polariza-

ção (PDM). Na Modulação de Amplitude em Quadratura (QAM, do inglês *Quadrature Amplitude Modulation*), tanto a amplitude quanto a fase do sinal da portadora são variadas para representar dados digitais. Esta técnica possibilita taxas de dados mais altas através do aumento no número de níveis de amplitude e fase [49]. A modulação por deslocamento de fase (PSK, do inglês *Phase-shift keying*) é um método onde a fase do sinal da portadora é alterada para representar dados digitais. Diferentes formas de PSK, como BPSK (*Binary PSK*), QPSK (*Quadrature PSK*) e DPSK (*Differential PSK*), variam a fase da portadora em diferentes graus, permitindo a transmissão eficiente de dados digitais. A PSK é valorizada por sua eficiência espectral e robustez em ambientes com ruído, sendo uma escolha popular para sistemas de comunicação óptica que requerem alta confiabilidade [49]. A Multiplexação por Divisão de Frequência Ortogonal (OFDM, do inglês *Orthogonal Frequency Division Multiplexing*) é uma técnica eficiente em comunicações ópticas, na qual os dados são divididos em várias subportadoras. Essas subportadoras são moduladas usando técnicas como QAM e PSK. O OFDM é reconhecido por permitir altas taxas de dados e uso eficiente da largura de banda, além de oferecer múltiplos canais de transmissão. Esta abordagem otimiza a transmissão em sistemas ópticos, melhorando a capacidade de transmitir grandes volumes de dados com eficiência e confiabilidade [49]. A Modulação por Divisão de Polarização (PDM, do inglês *Polarization Division Multiplexing*) é uma técnica avançada usada em comunicações ópticas. Essencialmente, o PDM aproveita duas polarizações ortogonais de luz na mesma fibra óptica para transmitir dois canais de dados independentes simultaneamente. Isso duplica efetivamente a capacidade de transmissão do sistema sem a necessidade de aumentar a largura de banda. O PDM é particularmente útil em sistemas de transmissão de alta capacidade, como links de fibra óptica de longa distância, onde é fundamental maximizar a quantidade de dados transmitidos. A eficiência e a capacidade de transmissão melhoradas fazem do PDM uma escolha importante em redes de comunicação modernas [49].

2.2.2 Desafios encontrados na camada física de transporte

A fibra óptica é amplamente adotada como canal de comunicação há décadas, valorizada principalmente pela sua excepcional capacidade de transportar grandes

volumes de informações [20]. No entanto, o uso da fibra óptica apresenta um desafio notável: os seus efeitos de não linearidade, que aumentam a complexidade da regeneração dos sinais transmitidos [45]. Devido às características desses efeitos, a obtenção das equações que descrevem os canais ópticos, não é uma tarefa trivial. Consequentemente, na prática, é necessário extrair a função de verossimilhança do sinal diretamente dos dados transmitidos pelo canal óptico, um processo que exige análises sofisticadas e precisas para garantir a qualidade e a eficiência da comunicação [18, 20, 45]. Estes efeitos não lineares têm um impacto direto sobre o sinal transmitido, afetando significativamente parâmetros de qualidade, como a Taxa de Erro de Bit (BER) e o fator de qualidade (*Q-factor*) do sinal. Essas alterações nos efeitos não lineares mencionados anteriormente podem prejudicar a demodulação do sinal e, conseqüentemente, representam um grande desafio para manter a alta qualidade e eficiência nas comunicações ópticas [18, 45]. A complexidade inerente às comunicações ópticas é amplificada pela relação independente entre transmissores e receptores, que estão muitas vezes distantes entre si. Essa distância implica que a configuração correta dos componentes depende fortemente da competência técnica dos operadores de rede. Adicionalmente, a necessidade de coordenação e troca de informações entre operadores de diferentes centros de transmissão aumenta ainda mais essa complexidade. Isso ocorre porque é essencial que tanto o transmissor quanto o receptor estejam configurados de maneira precisa para garantir uma comunicação eficiente e sem erros [18, 20, 45]. Além desses desafios, a habilidade das redes ópticas de estabelecer *lightpaths*, ou caminhos ópticos, traz consigo uma necessidade adicional: a de que todos os pontos intermediários na rede estejam não apenas alinhados, mas também corretamente configurados. Este requisito ressalta a importância crucial da precisão e do conhecimento técnico detalhado na gestão de redes ópticas. Além disso, sublinha os desafios contínuos relacionados à manutenção da alta qualidade e eficiência na transmissão de dados através desses sistemas ópticos complexos [18, 20, 45]. A eficiência e qualidade das comunicações em canais ópticos dependem crucialmente da alocação ótima de recursos. Isso requer a avaliação cuidadosa e configuração de vários fatores-chave, como a definição adequada do *lightpath*, a seleção do tipo de modulação mais eficiente, a correção precisa da dispersão cromática e da dispersão do modo de polarização, a avaliação da relação

sinal-ruído óptica (OSNR) e o entendimento da perda de sinal que depende da polarização são alguns exemplos. Todos esses elementos desempenham um papel vital na manutenção da integridade e eficácia das comunicações ópticas [18,20,45]. Complementando a discussão sobre a alocação de recursos, é essencial que esses recursos sejam estimados com exatidão e as informações sejam adequadamente transmitidas aos controladores de rede na camada de enlace. Um passo crítico nesse processo é a precisa estimativa dos parâmetros do canal óptico antes de proceder à demodulação do sinal. Esta avaliação antecipada é vital para configurar adequadamente o receptor e o regenerador óptico, garantindo assim uma comunicação óptica eficiente e segura. A complexidade envolvida na alocação desses recursos sublinha a necessidade de soluções de gerenciamento avançadas e precisas nas redes ópticas [18,45]. Nesse contexto, as técnicas de aprendizado de máquina podem fornecer soluções avançadas para os problemas abordados, especialmente na estimativa precisa dos parâmetros do canal óptico. Utilizando modelos de aprendizado de máquina, é possível prever comportamentos complexos do canal e otimizar a configuração dos dispositivos, receptores e regeneradores ópticos. Isso permite ajustes automáticos e precisos que melhoram a eficiência e a segurança da comunicação óptica, oferecendo uma gestão mais inteligente e adaptativa dos recursos da rede [16,17].

2.2.3 Desafios encontrados na camada de enlace

Ao focar na camada de enlace, desafios como a necessidade de configuração e gestão eficiente dos recursos de rede, problemas vistos na camada física, continuam sendo impactantes [18]. O projeto e gerenciamento de redes ópticas são temas constantes em pesquisas, visando aprimoramentos como a redução de latência, aumento da capacidade de transmissão e melhoria na qualidade de serviço (QoS) [18,45]. Atualmente, espera-se que as redes otimizem o uso de seus recursos, garantam a qualidade de serviço e busquem constantemente maneiras de aperfeiçoar o transporte de dados. Contudo, essas melhorias frequentemente enfrentam desafios relacionados à escalabilidade, refletindo a complexidade inerente ao gerenciamento de redes ópticas modernas. As redes definidas por software (SDN) são fundamentais na evolução das comunicações ópticas, oferecendo flexibilidade e eficiência na gestão dos recursos da rede [16,51]. A SDN possibilita um controle centralizado e programável dos

dispositivos de rede, adaptando-se dinamicamente às demandas de tráfego. Esse aspecto é especialmente importante em ambientes ópticos, onde ajustar rapidamente parâmetros como largura de banda e roteamento de forma ágil e automatizada é essencial [18, 18, 19]. A SDN, ao separar o plano de controle do plano de dados, facilita a implementação de políticas de gerenciamento de tráfego mais eficientes e oferece uma visão holística da rede óptica [52]. Essa abordagem contribui para otimizar o desempenho, a confiabilidade e a escalabilidade das infraestruturas de comunicação óptica. Neste contexto, o aprendizado de máquina oferece soluções para superar desafios na camada de enlace em redes ópticas, como a mitigação de efeitos não lineares e a melhoria do desempenho sob condições variáveis [53, 54]. Essas técnicas permitem uma análise detalhada e em tempo real de grandes volumes de dados ópticos, facilitando a detecção precoce de problemas e a implementação de ajustes automáticos [53, 54]. Esta abordagem promove uma gestão mais eficaz e confiável, adaptando-se às necessidades dinâmicas das comunicações ópticas modernas [23, 55, 56].

Capítulo 3

Metodologia

Este trabalho visa realizar uma investigação e implementação de estratégias destinadas a aprimorar o gerenciamento de redes ópticas de transporte, por meio da aquisição de informações dos sinais transmitidos e da utilização dos algoritmos de aprendizado de máquinas, para determinar a qualidade dos sinais em tais redes. Desta forma, foi implementado um classificador autônomo, capaz de classificar modulações e estimar a relação sinal ruído óptica. O sucesso na obtenção destas informações viabilizará a configuração autônoma do sistema, otimizando sua eficiência operacional e adaptabilidade às variáveis do ambiente de comunicação. Nas seções subsequentes deste capítulo, apresentaremos a visão geral sobre o *framework* na Seção 3.1 e detalharemos o sistema óptico de comunicação *back-to-back* simulado, na Seção 3.2. A análise das características dos sinais transmitidos, e os procedimentos para coleta e processamento dos dados serão explorados na Seção 3.3. Na Seção 3.4, discutiremos a seleção dos modelos de aprendizado de máquina utilizados. Concluímos com a Seção 3.5, onde delineamos a abordagem adotada para a classificação do tipo de modulação e a estimativa do valor de $OSNR$ no sistema proposto.

3.1 Visão geral sobre o *framework* de simulação

O *framework* de simulação tem o objetivo de possibilitar a coleta de informações sobre os sinais transmitidos no contexto óptico, permitindo assim que modelos de aprendizado de máquina possam ser treinados para classificar o tipo de modulação e valor de $OSNR$, presentes nestes sinais. Para isso, o *framework* de

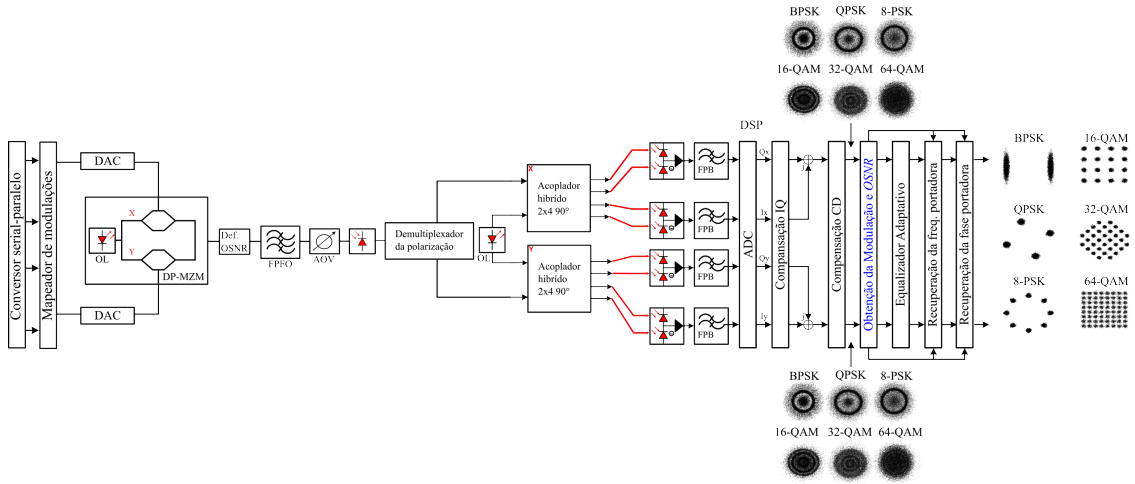


Figura 3.1: Diagrama em blocos do sistema óptico proposto, ilustrando o percurso do sinal desde a geração no transmissor até o processamento no receptor coerente.

simulação trata um sistema de comunicação *back-to-back*, executa a regeneração e coleta dos dados necessários para treinar os modelos de aprendizado de máquina, na identificação do tipo de modulação e previsão do valor de OSNR. Este sistema permite os para validar a seleção e configuração automática do processador digital de sinais (DSP, digital signal processor), selecionando os algoritmos e parâmetros apropriados, com base nas informações disponibilizadas pelos modelos treinados. As funções do *framework* são representadas na Figura 3.1, incluindo os blocos de transmissão, recepção, DSP, e as representações gráficas, com as saídas de dados para o treino dos modelos e do resultado obtido com o processo de regeneração bem-sucedido. O *framework* de simulação foi desenvolvido utilizando o simulador *VPIphotonics Design Suite*, versão 11.0. Sendo idealizado para suportar várias modulações, incluindo DP-BPSK, DP-QPSK, DP-8-PSK, DP-16-QAM, DP-32-QAM e DP-64-QAM (do inglês, respectivamente, dual-polarization quadrature phase-shift keying e dual-polarization quadrature amplitude modulation), com taxas de 28 Gbauds/s. Essas modulações, variando de 1 a 6 bits por símbolo, permitiram uma simulação abrangente e detalhada do meio de comunicação óptico.

3.2 Comportamento dos componentes ópticos

A seguir, apresentaremos detalhes abrangentes relacionados ao meio de comunicação óptica no *framework* de simulação proposto. O sistema de comunicação

proposto neste trabalho se baseia na arquitetura *back-to-back*, onde temos o transmissor ligado diretamente ao receptor sem a presença de elementos intermediários significativos, como processamentos ou amplificação. A Figura 3.2 ilustra uma visão geral do *framework* proposto e os respectivos diagramas de blocos que compõem as funcionalidades necessárias para a comunicação óptica, sendo Transmissor, Receptor e Processador Digital de Sinais (DSP).

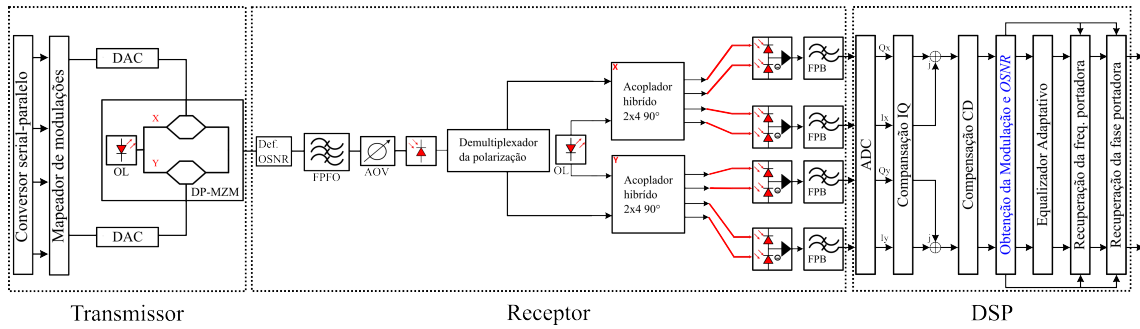


Figura 3.2: Diagrama de blocos das funcionalidades do *framework* proposto, sendo Transmissor, Receptor e DSP.

Iniciando pelo grupo que compõe o Transmissor, este é responsável por converter sinais lógicos para o meio óptico, possibilitando a transmissão através da fibra óptica até o Receptor. Na Figura 3.3, podemos observar os blocos que compõem o Transmissor, iniciando pelo bloco conversor serial-paralelo, que é responsável por receber um conjunto definido de bits e os paralelizar nas N saídas do bloco.

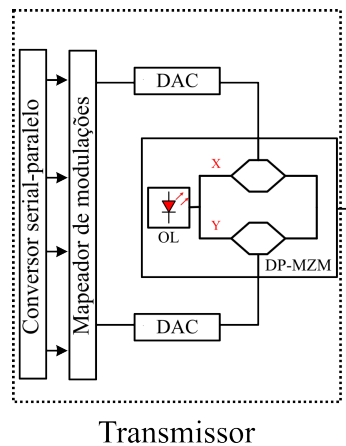


Figura 3.3: Diagrama de blocos que compõem a funcionalidade do Transmissor.

A seguir temos o bloco mapeador de modulações, cuja função é auxiliar na etapa de conversão do sinal elétrico de entrada para o sinal óptico modulado a ser transmitido. Neste contexto, sua função é a realização do condicionamento dos sinais elétricos disponíveis nas N portas paralelas para sinais modulantes relacionados

com a modulação que se planeja utilizar na transmissão. As definições sobre os tipos de modulação utilizados podem ser configuradas neste bloco, como o número de bits por símbolo. O próximo bloco é o modulador Mach-Zehnder, este bloco é responsável por realizar efetivamente a conversão do sinal transmitido do meio elétrico para o meio óptico, permitindo assim o início do uso da fibra óptica. A estrutura interna deste bloco normalmente consiste em dois acopladores direcionais de 3 dB interconectados por dois caminhos ópticos com comprimentos diferenciados, possuindo óptica integrada com as regiões de guia de onda e revestimento feitos de sílica (SiO_2) ou outro material sensível ao campo elétrico [57]. Durante o uso deste bloco, temos na entrada dos caminhos ópticos um diodo laser que gera o sinal da portadora na frequência $f_o = 192,1$ THz, sendo utilizado para gerar dois estados de polarização sendo \vec{E}_x e \vec{E}_y . Os sinais gerados pelo bloco de mapeamento da modulação são convertidos por um conversor digital-analógico e então são utilizados como estímulo diretamente ao revestimento dos caminhos ópticos para modular cada estado de polarização com informações independentes e assim transmitir as informações por meio da fibra óptica.

Devido às características da arquitetura *back-to-back* escolhida, o canal óptico não é representado, porém, algumas funcionalidades foram adicionadas ao Receptor para atuar em comportamentos esperados do sistema, conforme observado na Figura 3.4, temos o diagrama de blocos utilizados na recepção do sinal, compondo a funcionalidade do Receptor. Iniciando pelo bloco Def. OSNR, que realiza a adição

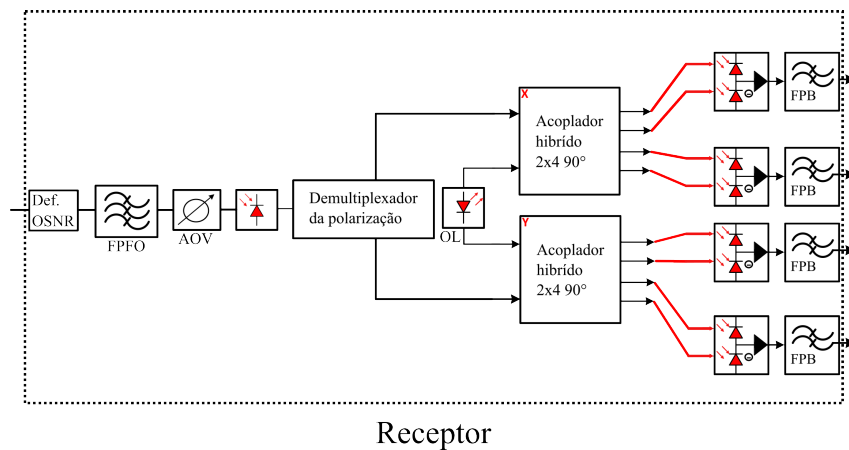


Figura 3.4: Diagrama de blocos que compõem a funcionalidade do Receptor.

de ruído no sinal transmitido em meio óptico. O nível de ruído pode ser definido arbitrariamente na unidade de medida dB/0,1nm. Cada modulação teve o intervalo

de OSNR testado e definido com base nos resultados obtidos para a taxa de erro de bit, do inglês BER. A seguir temos o filtro passa-faixa óptico (FPFO), utilizado para limitar a largura de banda do sinal, possibilitando que componentes de alta frequência sejam filtrados, melhorando a qualidade do sinal e permitindo a redução de erros e auxiliando no aumento da distância transmitida e o atenuador óptico variável (AOV) utilizado para manter constante o nível de potência que chega no fotodetector, igual a -2 dBm. Após o atenuador, temos o fotodetector, onde o sinal é captado e então condicionado para passar pelo bloco demultiplexador de modulações, onde os estados de polarizações \vec{E}_x e \vec{E}_y são utilizados como entrada dos acopladores híbridos 2×4 de 90° , assim como o sinal do laser do oscilador local centrado em $f_{LO} = 192,1$ THz, para separar as componentes em fase e quadratura de cada polarização de igual potência e defasados 90° entre si, bem como a componente relativa ao sinal do oscilador local para posterior amplificação. Na sequência, os sinais em fase e quadratura de ambos estados de polarização seguem para os fotodetectores, amplificadores de corrente e filtro passa baixa (bloco LPF), condicionando os sinais do meio óptico para o meio elétrico para passar a próxima etapa do processo.

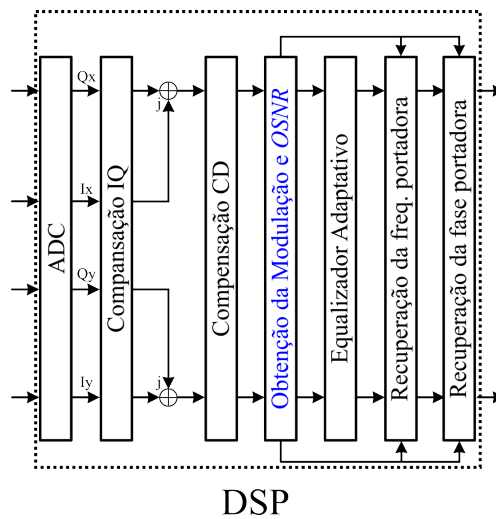


Figura 3.5: Diagrama de blocos que compõem a funcionalidade do DSP.

Por fim, temos o diagrama de blocos que compõem o processador digital de sinais (DSP), conforme observado na Figura 3.5. A função principal deste bloco é a compensação dos efeitos e distorções sofridas pelo sinal durante a transmissão, permitindo assim melhoras na legibilidade das informações recebidas. O DSP é

composto por oito blocos. O primeiro trata-se do bloco ADC, onde os sinais são convertidos de analógico para digital. Em seguida, as componentes em fase e quadratura são condicionadas e utilizadas em conjunto pelo bloco Compensação IQ, então a dispersão cromática é compensada pelo bloco Compensação CD. Neste ponto, os sinais foram processados por blocos comuns às modulações em uso no esquema de transmissão e chegam à entrada do bloco de *Obtenção da Modulação e OSNR*. Este bloco tem por objetivo indicar aos blocos de recuperação da frequência e da fase das portadoras de transmissão, o tipo de modulação e valor de OSNR da transmissão atual, permitindo que os respectivos blocos realizem a parametrização de algoritmos de estimativa de frequência e fase corretamente. Resultando na capacidade de regeneração dos sinais de ambos os estados de polarização sem a necessidade de conhecimento prévio da modulação. Para a conclusão do processo de compensação dos sinais transmitidos temos na sequência o bloco de Equalização adaptativa, que entre algumas das funções disponíveis pode ser utilizado para compensar as distorções causadas pela interferência entre símbolos, do inglês *inter-symbol interference (ISI)*. Na sequência temos o primeiro bloco reconfigurável sendo o bloco de Recuperação da frequência da portadora e na sequência o último bloco reconfigurável e também do sistema, sendo o bloco de Recuperação da fase da portadora do sinal. Ambos os blocos atuam para compensar os efeitos da fibra, porém de forma específica para cada modulação, aplicando, algoritmos e coeficientes adequados a cada esquema de transmissão. Para uma eficaz compensação de sinais e minimização de erros na recepção de dados, é necessária a pré-configuração dos dispositivos com informações detalhadas sobre o sinal transmitido. A ausência desta preparação ou erros nesse processo podem levar a falhas de comunicação. Este trabalho propõe um sistema capaz de ajustar autonomamente suas configurações durante a transmissão, adaptando-se conforme necessário, sem depender de intervenção manual ou conhecimento prévio específico sobre o sinal, visando melhorar a eficiência e a confiabilidade da transmissão de dados.

3.3 Características dos sinais transmitidos

Para a coleta dos sinais utilizados no treino dos modelos de aprendizado de máquina, foi necessário primeiro a realização da configuração ótima do *framework* de simulação, incluindo o transmissor, receptor e DSP. Para isso, cada bloco do *framework* de simulação foi configurado adequadamente conforme o tipo de modulação e os valores de OSNR utilizados pelo sistema de comunicação óptica. Identificamos seis modulações principais: DP-BPSK, DP-QPSK, DP-8-PSK, DP-16-QAM, DP-32-QAM e DP-64-QAM. Cada modulação escolhida opera idealmente em um intervalo específico de OSNR, determinado neste *setup* através de testes de transmissão e posterior análise da taxa de erro de bits (BER), conforme demonstrado na Figura 3.6. Para isso, foram realizadas transmissões utilizando o *framework* de simulação para cada modulação alvo, variando o valor de OSNR de 8dB até 33dB, com incrementos de 0,2dB/0,1nm. Após a análise dos resultados, os intervalos fo-

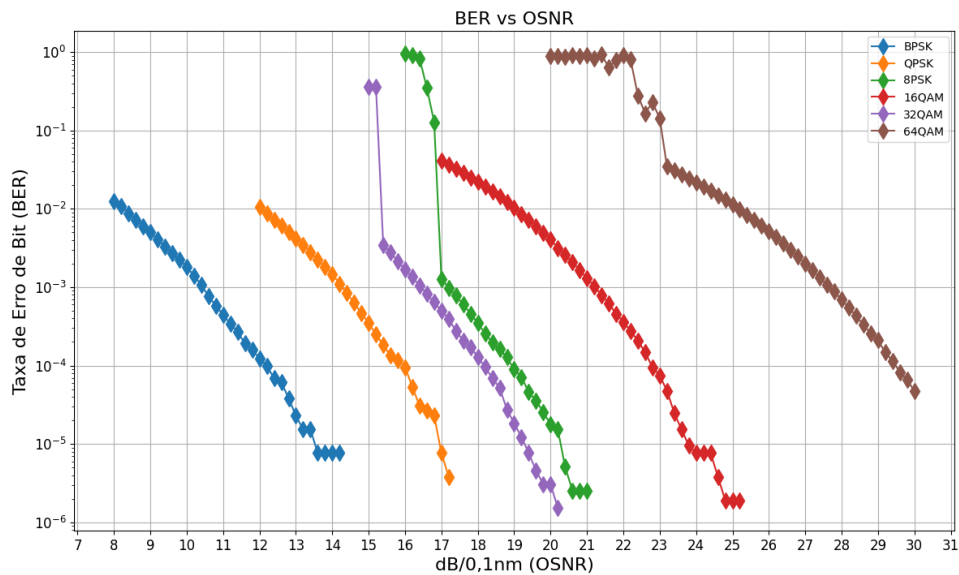


Figura 3.6: Relação entre BER e OSNR para os tipos de modulação utilizados no transmissor.

ram definidos para: DP-BPSK de 8,0dB a 15,0dB, DP-QPSK de 12,0dB a 20,0dB, DP-8-PSK de 17,0dB a 21,0dB, DP-16-QAM de 17,0dB a 26,0dB, DP-32-QAM de 15,0dB a 21,0dB, e DP-64-QAM de 22,0dB a 33,0dB, e a variação definida foi de 0,5dB/0,1nm. Essas definições, armazenadas em macros, orientaram a configuração dos demais blocos do sistema.

A simulação do sistema óptico de comunicação segue um processo estruturado: inicialmente, selecionamos um tipo de modulação via macro, configurando o ambiente para a transmissão, transporte e recepção do sinal modulado. Ao iniciar a simulação, uma segunda macro ajusta os valores de OSNR no intervalo predefinido. Em cada valor de OSNR, realizamos um ciclo de 200 aquisições, onde cada uma representa uma transmissão completa entre transmissor e receptor. Após cada ciclo, incrementamos o valor da OSNR em 0,5dB e iniciamos um novo ciclo, até cobrir todo o intervalo. Destacamos que os efeitos de transmissão no simulador são dinâmicos, garantindo diversidade nas repetições. Durante cada ciclo de simulação, os sinais coletados passam pelo bloco DSP, sendo o responsável pela regeneração do sinal. O *framework* possui quatro pontos-chave de coleta de dados: na entrada do bloco DSP (sinal sem correção dos efeitos ópticos), após a correção da dispersão cromática (etapa comum na regeneração), após a correção da fase da portadora (depende do tipo de modulação e é sensível a ruídos), e finalmente, após a regeneração completa do sinal (válida a regeneração realizada). A Figura 3.7 ilustra estas etapas com sinais da modulação 64-QAM a 25dB de OSNR. Após várias rodadas de aquisição e análise dos sinais, algumas decisões importantes foram tomadas para otimizar os resultados e simplificar a coleta de dados. Optamos por coletar dados experi-

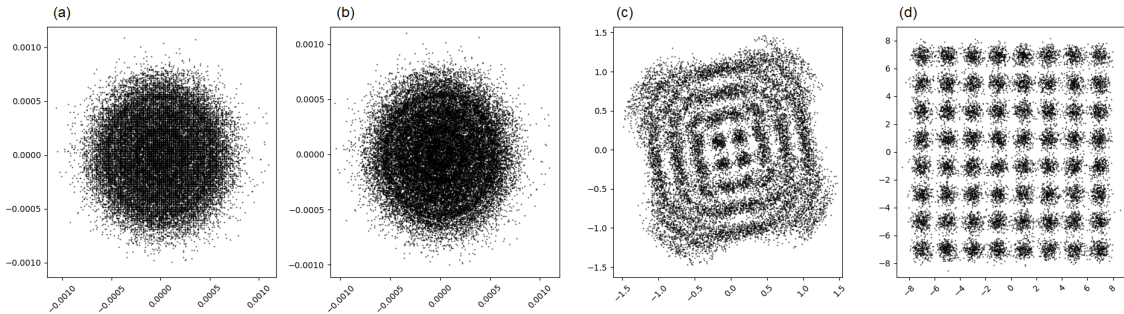


Figura 3.7: Sinais em fase e quadratura coletados para a modulação 64-QAM em 25dB/0,1nm de OSNR, sendo (a) na entrada do DSP, (b) após a correção da dispersão cromática, (c) após a correção da frequência da portadora e (d) na saída do DSP.

mentais após a compensação da dispersão cromática, uma etapa comum a todas as modulações. Além disso, tratamos cada estado de polarização das modulações como sinais independentes, simplificando o tipo de modulação nos experimentos. A Figura 3.8 ilustra os dados extraídos a partir de cada modulação, sendo utilizados para a construção de duas bases de histogramas, com 200 e 400 intervalos, destacando as

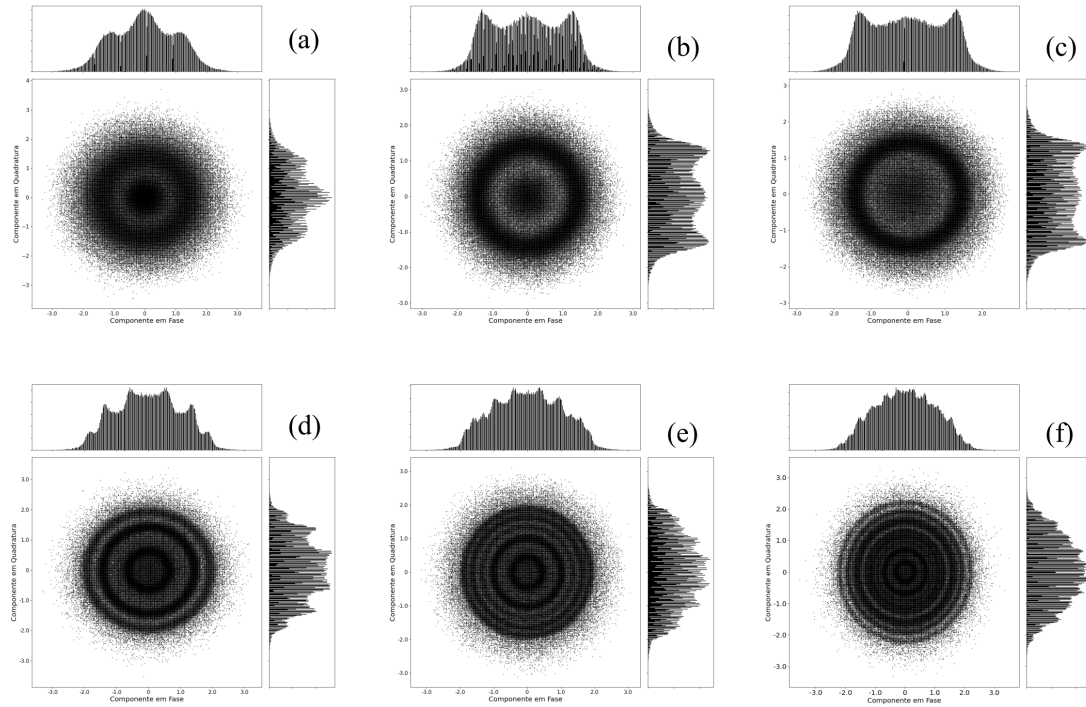


Figura 3.8: Sinais em fase e quadratura, e respectivos histogramas para as modulações (a) BPSK, (b) QPSK, (c) 8-PSK, (d) 16-QAM, (e) 32-QAM e (f) 64-QAM, nos valores de OSNR 15dB, 20dB, 21dB, 26dB, 21dB e 33dB, respectivamente.

variações entre as modulações em seus valores máximos de OSNR.

3.4 Modelos de aprendizado

A necessidade de classificar o tipo de modulação e prever valores de OSNR exige o uso de modelos de aprendizado de máquina, devido à sua capacidade de identificar padrões e extrair informações não explicitamente visíveis. Exploramos modelos reconhecidos na literatura [16, 17, 19], como Árvore de Decisão pela simplicidade, algoritmos *ensembles* pela robustez, *boosting* para aprimorar resultados e redes neurais devido à sua complexidade e versatilidade. Neste contexto, detalharemos modelos específicos: *Random Forest* (*ensemble*), Árvore de Decisão *CART*, *AdaBoost* e *Gradient Boosting* (ambos de *boosting*), e a rede neural *Perceptron* Multicamada.

3.4.1 *Random Forest*

O algoritmo *Random Forest* é um método de aprendizado de máquina versátil e amplamente utilizado, que pertence ao grupo dos algoritmos *ensemble* ou comitê. Opera construindo um conjunto com diversas Árvores de Decisão durante o treinamento, e produz o modo das classes (para classificação) ou a previsão média (para regressão) das árvores individuais. O *Random Forest* provou ser eficaz em uma variedade de aplicações devido à sua robustez contra o superajuste e sua capacidade de lidar com grandes conjuntos de dados com alta dimensionalidade [58].

3.4.1.1 Equação do Modelo

O algoritmo *Random Forest* pode ser representado da seguinte forma:

$$Y = \frac{1}{B} \sum_{b=1}^B T_b(X; \Theta_b)$$

sendo Y , B , T_b , X e Θ_b respectivamente a previsão feita pelo modelo *Random Forest*, o número de árvores na floresta, a previsão feita pela b -ésima árvore, as características de entrada e os parâmetros amostrados aleatoriamente usados para crescer a b -ésima árvore.

3.4.1.2 Mecanismo Operacional

1. **Agregação de *Bootstrap* (*Bagging*):** Cada árvore de um *Random Forest* é construída a partir de uma amostra retirada com substituição (ou seja, uma amostra de *bootstrap*) do conjunto de treinamento.
2. **Aleatoriedade de Características:** Ao dividir um nó durante a construção da árvore, a divisão escolhida não é mais a melhor divisão entre todas as características. Em vez disso, a divisão escolhida é a melhor divisão entre um subconjunto aleatório das características.
3. **Previsão:** Para regressão, a previsão é a média das previsões de todas as árvores. Para classificação, é o voto majoritário (modo) das classes previstas pelas árvores individuais.

3.4.1.3 Vantagens e Aplicações

O *Random Forest* é particularmente adequado para situações onde temos grandes conjuntos de dados com alta dimensionalidade e a necessidade de um modelo que possa capturar interações complexas e não linearidades sem sofrer de superajuste. Sua capacidade de lidar com conjuntos de dados complexos enquanto permanece relativamente interpretável é uma vantagem significativa em muitos cenários práticos [59].

3.4.2 Árvore de Decisão *CART*

A Árvore de Decisão *CART* (Classification and Regression Trees) é um algoritmo de aprendizado de máquina utilizado tanto para tarefas de classificação quanto de regressão. Este método envolve a divisão do conjunto de dados em subconjuntos baseados em decisões, estruturadas em forma de árvore [60].

3.4.2.1 Equação do Modelo

Embora a Árvore de Decisão *CART* não seja representada por uma equação tradicional como outros modelos, sua construção pode ser descrita pelo processo de divisão. Em cada nó da árvore, um teste é aplicado a um atributo para dividir os dados em subconjuntos mais homogêneos. Este processo é descrito pela função:

$$\text{Dividir}(D, \text{atributo}) \rightarrow (D_{\text{esquerda}}, D_{\text{direita}})$$

sendo D , atributo , D_{esquerda} e D_{direita} respectivamente o conjunto de dados no nó atual, o atributo escolhido para a divisão e os subconjuntos de dados resultantes da divisão.

3.4.2.2 Mecanismo Operacional

1. O algoritmo começa com o conjunto de dados inteiro e seleciona o melhor atributo para dividir os dados com base em um critério de pureza (como coeficiente de *Gini* ou Entropia).
2. O conjunto de dados é dividido em subconjuntos conforme o valor do atributo

escolhido.

3. Este processo é repetido recursivamente para cada subconjunto até que um critério de parada seja alcançado (por exemplo, profundidade máxima da árvore, número mínimo de amostras em um nó).

3.4.2.3 Vantagens e Aplicações

As Árvores de Decisão *CART* são populares devido à sua simplicidade, facilidade de interpretação e capacidade de lidar com dados não lineares. Elas são amplamente utilizadas em diversas áreas, como medicina, finanças e engenharia. As Árvores de Decisão *CART* são úteis por sua habilidade de modelar complexidades em dados de forma intuitiva e acessível [60].

3.4.3 *AdaBoost*

O *AdaBoost* (*Adaptive Boosting*) é um método de aprendizado de máquina que faz parte do grupo de algoritmos de *boosting*. Sua principal característica é combinar vários modelos “fracos” de aprendizado de máquina, tipicamente Árvores de Decisão, para criar um modelo “forte” e robusto. O *AdaBoost* adapta-se continuamente, focando nas instâncias mais difíceis ao ajustar os pesos das instâncias de treinamento [61].

3.4.3.1 Equação do Modelo

A ideia central do AdaBoost pode ser expressa pela seguinte equação:

$$f(x) = \sum_{t=1}^T \alpha_t h_t(x)$$

sendo $f(x)$, T , α_t e $h_t(x)$ respectivamente a previsão final do modelo AdaBoost, o número total de modelos “fracos” utilizados, o peso do t -ésimo modelo, que reflete sua precisão e a previsão feita pelo t -ésimo modelo.

3.4.3.2 Mecanismo Operacional

1. Inicialmente, o *AdaBoost* atribui pesos iguais a todas as instâncias de treinamento.
2. Em cada etapa de iteração, um modelo é treinado e os pesos das instâncias são ajustados conforme os erros do modelo atual; as instâncias classificadas incorretamente ganham mais peso.
3. O processo se repete, com cada modelo subsequente focando mais nas instâncias difíceis.
4. Os modelos são combinados com base em seus pesos de precisão para formar o modelo final.

3.4.3.3 Vantagens e Aplicações

O *AdaBoost* é eficaz em aumentar a precisão de modelos simples e é especialmente poderoso em problemas de classificação. No entanto, é sensível a dados ruidosos e *outliers*. Uma das forças do *AdaBoost* é sua capacidade de incrementar a eficiência de modelos simples, tornando-os altamente efetivos [61].

3.4.4 *Gradient Boosting*

O *Gradient Boosting* é uma técnica poderosa de aprendizado de máquina que faz parte dos algoritmos de *boosting*. Ele constrói um modelo preditivo iterativamente, adicionando modelos “fracos”, geralmente Árvore de Decisão, de maneira sequencial, onde cada modelo subsequente corrige os erros cometidos pelos modelos anteriores. O Gradient Boosting utiliza o gradiente do erro para direcionar a construção dos modelos subsequentes [62].

3.4.4.1 Equação do Modelo

A abordagem do Gradient Boosting pode ser descrita pela seguinte equação:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

sendo $F_m(x)$, $F_{m-1}(x)$, γ_m e $h_m(x)$ respectivamente o modelo aprimorado após a adição do m -ésimo modelo “fraco”, o modelo composto pelos $m - 1$ modelos “fracos” anteriores, a taxa de aprendizagem, determinando o peso do m -ésimo modelo, e o m -ésimo modelo “fraco”.

3.4.4.2 Mecanismo Operacional

1. Inicialmente, o modelo é uma constante que minimiza a função de perda.
2. Em cada etapa, um novo modelo “fraco” é ajustado aos resíduos do modelo atual.
3. O modelo é então atualizado para incluir o novo modelo “fraco”.
4. Esse processo é repetido, minimizando progressivamente a função de perda.

3.4.4.3 Vantagens e Aplicações

O Gradient Boosting é eficaz em uma variedade de problemas de aprendizado de máquina, particularmente em tarefas de regressão e classificação. Ele consegue capturar interações complexas entre as características e é altamente personalizável. O Gradient Boosting é notável pela sua capacidade de otimização e ajuste fino, tornando-o uma escolha excelente para problemas complexos [63].

3.4.5 *Perceptron* Multicamada

O *Perceptron* Multicamada (MLP, do inglês *Multilayer Perceptron*) é um tipo de rede neural artificial básica, composta por múltiplas camadas de neurônios em uma estrutura de alimentação direta. Cada neurônio em uma camada está conectado a todos os neurônios da camada subsequente, permitindo que a MLP capture relações complexas nos dados [64].

3.4.5.1 Equação do Modelo

A função de um neurônio individual em uma MLP pode ser descrita pela equação:

$$y = f \left(\sum_{i=1}^n w_i x_i + b \right)$$

sendo y , f , w_i , x_i , b e n respectivamente a saída do neurônio, a função de ativação (por exemplo, ReLU, *sigmoid*), os pesos, as entradas, o termo de viés, e o número de entradas. Para uma rede com múltiplas camadas, a saída de uma camada se torna a entrada para a próxima, criando uma cadeia de processamento de informações.

3.4.5.2 Mecanismo Operacional

1. A entrada é processada sequencialmente por camadas ocultas, onde cada neurônio aplica sua função de ativação à soma ponderada das entradas.
2. Os pesos e vieses da rede são ajustados durante o treinamento usando um algoritmo de retropropagação, que minimiza a diferença entre a saída da rede e o resultado esperado.
3. O treinamento continua até que a rede alcance um nível aceitável de precisão ou um número definido de iterações.

3.4.5.3 Vantagens e Aplicações

As MLPs são fundamentais no campo do aprendizado profundo e são amplamente utilizadas para tarefas de classificação e regressão, processamento de linguagem natural e reconhecimento de padrões. O Perceptron Multicamada é um modelo clássico e poderoso em aprendizado de máquina, servindo como base para muitas arquiteturas de redes neurais mais avançadas. Sua flexibilidade e capacidade de aprendizado profundo o tornam extremamente valioso em uma variedade de aplicações complexas [64].

3.5 Obtenção da Modulação e OSNR

No conjunto DSP do framework de simulação proposto, o bloco de *Obtenção da Modulação e OSNR*, detalhado na Figura 3.9, desempenha um papel vital. Suas principais funções incluem processar o sinal de entrada para geração dos histogramas utilizados pelos blocos seguintes, realizar a classificação da modulação em uso e a predição do valor de OSNR. Essas tarefas são realizadas utilizando modelos de aprendizado de máquina previamente treinados com dados coletados. A seguir,

exploraremos as estratégias e definições aplicadas a cada função desempenhada por este bloco, enfatizando sua interação e processamento de informações.

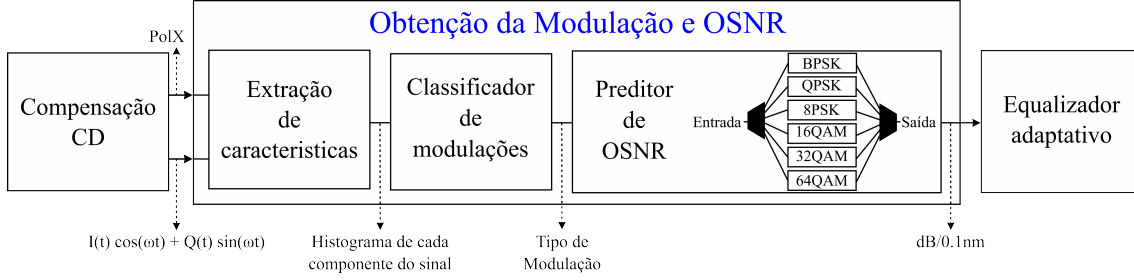


Figura 3.9: Blocos propostos para realizar a *Obtenção da Modulação e OSNR*.

O primeiro bloco do sistema é *Extração de características*, cuja principal tarefa é gerar histogramas conforme as especificações do sistema proposto. A fim de otimizar a eficácia deste bloco na obtenção da modulação e OSNR, optamos por processar uma única componente por polarização. É interessante observar que, mesmo em um sinal singular, cada componente é tratada individualmente. Por exemplo, um sinal com modulação DP-QPSK resultará em quatro instâncias distintas: $\vec{E}_x I(t) \cos(\omega t)$, $\vec{E}_x Q(t) \sin(\omega t)$, $\vec{E}_y I(t) \cos(\omega t)$ e $\vec{E}_y Q(t) \sin(\omega t)$, todas categorizadas como QPSK. Essa abordagem aumenta o volume de dados disponibilizados para o treinamento, visando uma maior robustez do modelo resultante.

O próximo componente do sistema é o bloco *Classificador de Modulações*. Este bloco tem a função de classificar o tipo de modulação utilizando os histogramas fornecidos pelo bloco *Extração de características*, empregando algoritmos de aprendizado de máquina para isso. Os modelos são previamente treinados com dados disponíveis e, uma vez identificado o modelo com o melhor desempenho, ele é implementado no bloco. No contexto do *framework* de simulação, a classificação ocorre de forma autônoma. Inicialmente, o DSP começa com configurações padrão, insuficientes para a regeneração completa do sinal. Após a identificação correta da modulação, o sistema consegue ajustar os algoritmos e parâmetros específicos do DSP que dependem da modulação, permitindo uma regeneração eficaz do sinal.

O último componente do sistema é o *Preditor de OSNR*, que realiza a predição do valor de OSNR. Para isso, foram adotadas duas escolhas estratégicas: primeiro, cada modelo é treinado para reconhecer a OSNR associada a uma modulação específica devido à intersecção dos intervalos de OSNR entre diferentes modulações. Isso implica que cada modelo é otimizado para situações únicas, influenciando tanto as

características de entrada quanto o número de classes envolvidas. Essa estratégia, embora simplifique a complexidade em comparação com a análise do conjunto completo de dados, garante uma abordagem mais eficiente para a identificação de OSNR. A segunda escolha diz respeito à seleção do modelo para a predição de OSNR, sendo definido a partir da saída do *Classificador de Modulações*, tornando-o assim condicionado à acurácia da etapa inicial. Ao identificar corretamente o valor de OSNR, o sistema pode ajustar os demais algoritmos e parâmetros do DSP, permitindo a regeneração completa do sinal, otimizando o desempenho da comunicação óptica.

Capítulo 4

Simulações e Resultados

Neste capítulo, exploramos os resultados obtidos pela aplicação de algoritmos de aprendizado de máquina na classificação de modulações e na predição dos valores de *OSNR* em receptores ópticos coerentes flexíveis. Detalhamos o *framework* de simulação utilizado, essencial para representar o comportamento dos receptores ópticos coerentes flexíveis em meio óptico e fundamental para a compreensão dos resultados. A apresentação dos resultados é realizada mediante tabelas e gráficos, facilitando a visualização e compreensão dos dados. Uma análise comparativa é conduzida, enfocando nas métricas de desempenho dos algoritmos *AdaBoost*, *Árvore de Decisão CART*, *Gradient Boosting*, *Random Forest* e *Perceptron Multicamada*. O capítulo está organizado da seguinte maneira: primeiramente, apresentamos detalhes sobre o *framework* de simulação. Posteriormente, exibimos os resultados para a classificação de modulação e predição do *OSNR*, salientando as diferenças entre os algoritmos. Concluimos com uma discussão dos resultados obtidos, oferecendo uma análise comparativa detalhada do desempenho dos modelos treinados.

4.1 *Framework* de simulação proposto

O *framework* de simulação foi desenvolvido utilizando o simulador *VPIphotonics Design Suite*, e a configuração dos componentes ópticos foi realizada utilizando as funcionalidades integradas do simulador. Foi utilizada a linguagem *Python*, versão 3.10, para programar os componentes de aprendizado de máquina e funcionalidades externas ao *framework* de simulação. Essa versão do *Python* já está integrada ao si-

mulador *VPIphotonics*, facilitando a interação direta com o *framework* de simulação. Esta integração permite uma programação eficiente e alinhada com as funcionalidades do simulador. O *framework* de simulação foi desenvolvido com três metas principais: simular e facilitar a coleta de dados sobre o comportamento de sinais em meios ópticos, oferecer um ambiente adequado para a validação de resultados experimentais, e proporcionar uma plataforma flexível para a realização contínua de novos experimentos. Com essas metas estabelecidas, tomamos decisões específicas para alcançá-las. Uma delas é a quantidade de sinais coletados, detalhada na Tabela 4.1. Esta tabela, ilustra a quantidade de sinais coletados conforme cada tipo de modulação e os respectivos intervalos de OSNR. Este detalhamento permite uma compreensão clara da abrangência dos dados coletados, essenciais para a análise e validação dos resultados experimentais. Outra decisão tomada foi em relação aos

Base de dados	
Classes	Nº de Instâncias
BPSK	12.000
QPSK	13.600
8-PSK	7.200
16-QAM	15.200
32-QAM	10.400
64-QAM	18.400

Tabela 4.1: Distribuição dos sinais coletados por modulação.

experimentos realizados, configurando o *framework* de simulação para utilizar 6 tipos de modulação em um intervalo total de 51 valores de OSNR, distribuídos entre as modulações conforme o seguinte esquema, BPSK de 8,0dB a 15,0dB, QPSK de 12,0dB a 20,0dB, 8-PSK de 17,0dB a 21,0dB, 16-QAM de 17,0dB a 26,0dB, 32-QAM de 15,0dB a 21,0dB e 64-QAM de 22,0dB a 33,0dB. Os sinais da simulação foram convertidos duas bases de histogramas com 200 e 400 intervalos. Então, foram utilizados em dois experimentos distintos, o primeiro experimento visando classificar corretamente o tipo de modulação do sinal e para isso utiliza 5 modelos de classificação sendo, *árvore de decisão CART*, *Random Forest*, *AdaBoost*, *Gradient Boosting* e *Perceptron Multicamada*. O segundo experimento visa prever o valor de OSNR, e para isso utiliza um conjunto de 3 modelos, sendo *Random Forest* como classificador e regressor, e *Perceptron Multicamada*.

Em relação à quantidade de experimentos planejado para esta etapa, duas tabelas demonstrativas foram preparadas para esta fase: a Tabela 4.2 ilustra os

Planejamento dos treinos		
Modelo	Classificação da modulação	Predição da OSNR
Árvore de Decisão <i>CART</i>	Conjunto 1	—
<i>AdaBoost</i>	Conjunto 2	—
<i>Gradient Boosting</i>	Conjunto 2	—
<i>Random Forest</i> Classificador	Conjunto 2	Conjunto 4
<i>Perceptron</i> Multicamada	Conjunto 3	Conjunto 5
<i>Random Forest</i> Regressor	—	Conjunto 6

Tabela 4.2: Planejamento da exploração dos parâmetros de treinamento para cada modelo selecionado.

conjuntos de parâmetros utilizados no treino conforme a estratégia utilizada, e a Tabela 4.3 que aborda a exploração dos parâmetros por conjunto. Essas tabelas facilitam a execução e a comparação dos resultados de cada modelo em diferentes cenários.

Conjuntos planejados para os treinos						
Parâmetros	Estimadores	Histograma	Épocas	Critério	Profundidade	Saída
Conjunto 1	—	200, 400	—	Gini, Entropia	10, inf	Tipo de Modulação
Conjunto 2	100, 200, 600	200, 400	—	Gini	—	Tipo de Modulação
Conjunto 3	200, 600	200, 400	100, 700	—	—	Tipo de Modulação
Conjunto 4	200, 600	200, 400	—	Entropia	—	osnr por Modulação
Conjunto 5	200, 600	200, 400	100, 700	—	—	osnr por Modulação
Conjunto 6	200, 600	200, 400	—	Erro quadrático	—	osnr por Modulação

Tabela 4.3: Detalhamento sobre os conjuntos de parâmetros definidos para exploração.

4.2 Apresentação dos resultados

O bloco de *Obtenção da Modulação e OSNR*, que faz parte do conjunto *DSP* do *framework* de simulação, tem um papel crucial na classificação dos tipos de modulação e na predição dos valores de *OSNR*, elementos essenciais para o funcionamento autônomo do *framework*. Durante a nossa metodologia, adotamos estratégias específicas para cada situação, as quais direcionaram o processo de treinamento dos algoritmos de aprendizado de máquina. Os resultados alcançados por cada algoritmo foram cuidadosamente coletados e analisados. Na sequência, apresentemos

estes resultados, seguindo a estratégia empregada.

4.2.1 Classificação do tipo de modulação

Nesta seção, apresentamos os resultados da classificação do tipo de modulação, organizados conforme o modelo empregado no bloco *Classificador de Modulações*, como demonstrado na Figura 4.1. Para esta análise, selecionamos cinco algoritmos de aprendizado de máquina: *AdaBoost*, *Árvore de Decisão CART*, *Gradient Boosting*, *Random Forest* e *Perceptron Multicamada*. No início, cada algoritmo foi treinado utilizando duas bases de dados distintas: a primeira contendo histogramas com 200 intervalos e a segunda com histogramas de 400 intervalos. Ambas as bases foram derivadas dos sinais coletados através do *framework* de simulação. Os algoritmos foram treinados com parâmetros de configuração padrão. Com os resultados iniciais em mãos, conduzimos experimentos exploratórios, ajustando os parâmetros de cada algoritmo na busca por um desempenho aprimorado. Em relação à utilização das bases de dados, ambas foram divididas de forma que 30% dos dados foram utilizados para o treinamento, enquanto os 70% restantes foram empregados na etapa de testes.

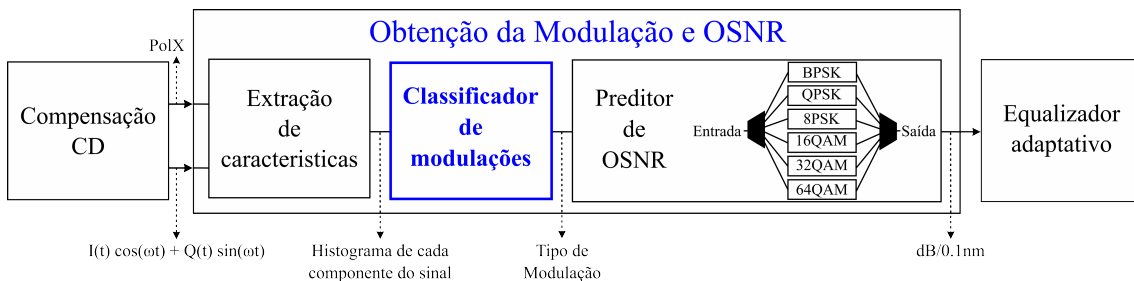


Figura 4.1: Bloco proposto para classificação do tipo de modulação.

4.2.1.1 Resultados para o classificador de modulações com *Random Forest*.

Discutiremos agora os resultados alcançados com *Random Forest* como classificador dos tipos de modulação. A Figura 4.2 mostra o desempenho ótimo, com os parâmetros detalhados na Tabela 4.4. Este modelo atingiu uma acurácia de 99,31%, com detalhes de desempenho como MAE de 0,0089, MSE de 0,0160 e RMSE de 0,1264. A Figura 4.3 exibe a evolução dos resultados ao utilizar os parâ-

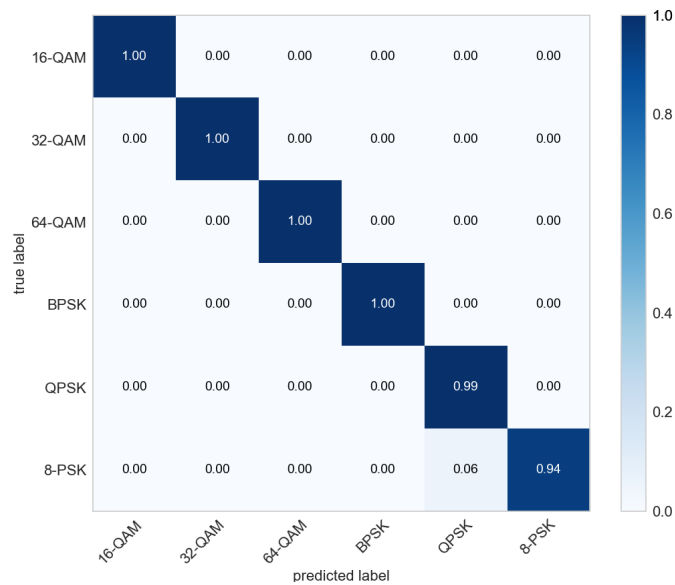


Figura 4.2: Matriz de confusão representando o melhor desempenho do classificador *Random Forest* a partir dos parâmetros ótimos encontrados na Tabela 4.4.

Parâmetros utilizados	
Nome	Valor
Número de estimadores	600
Critério de avaliação da qualidade da divisão	Gini
Profundidade máxima da árvore	∞
Núm. min. de amostras para realizar uma divisão	2
Núm. máx. de características para realizar uma divisão	$\sqrt{F} \approx 14$
Quantidade máxima de folhas	∞
Valor mín. de impureza para dividir um ramo	≥ 0
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	200

Tabela 4.4: Parâmetros utilizados no treinamento do modelo *Random Forest* com melhor desempenho para a classificação da modulação.

Parâmetros explorados durante o treino	
Nome	Valores
Número de estimadores	100, 200, 600
Intervalos do histograma	200, 400

Tabela 4.5: Conjunto de parâmetros explorados durante os experimentos com o modelo *Random Forest*.

metros descritos na Tabela 4.5. Observamos a influência do número de estimadores, explorando valores de 100, 200 e 600, bem como o impacto do número de intervalos do histograma, com bases de dados construídas com 200 e 400 intervalos. Essas alterações são evidenciadas na Figura 4.3, onde a sub figura (a) mostra o resultado com menor acurácia, e as subsequentes demonstram melhorias incrementais até a sub

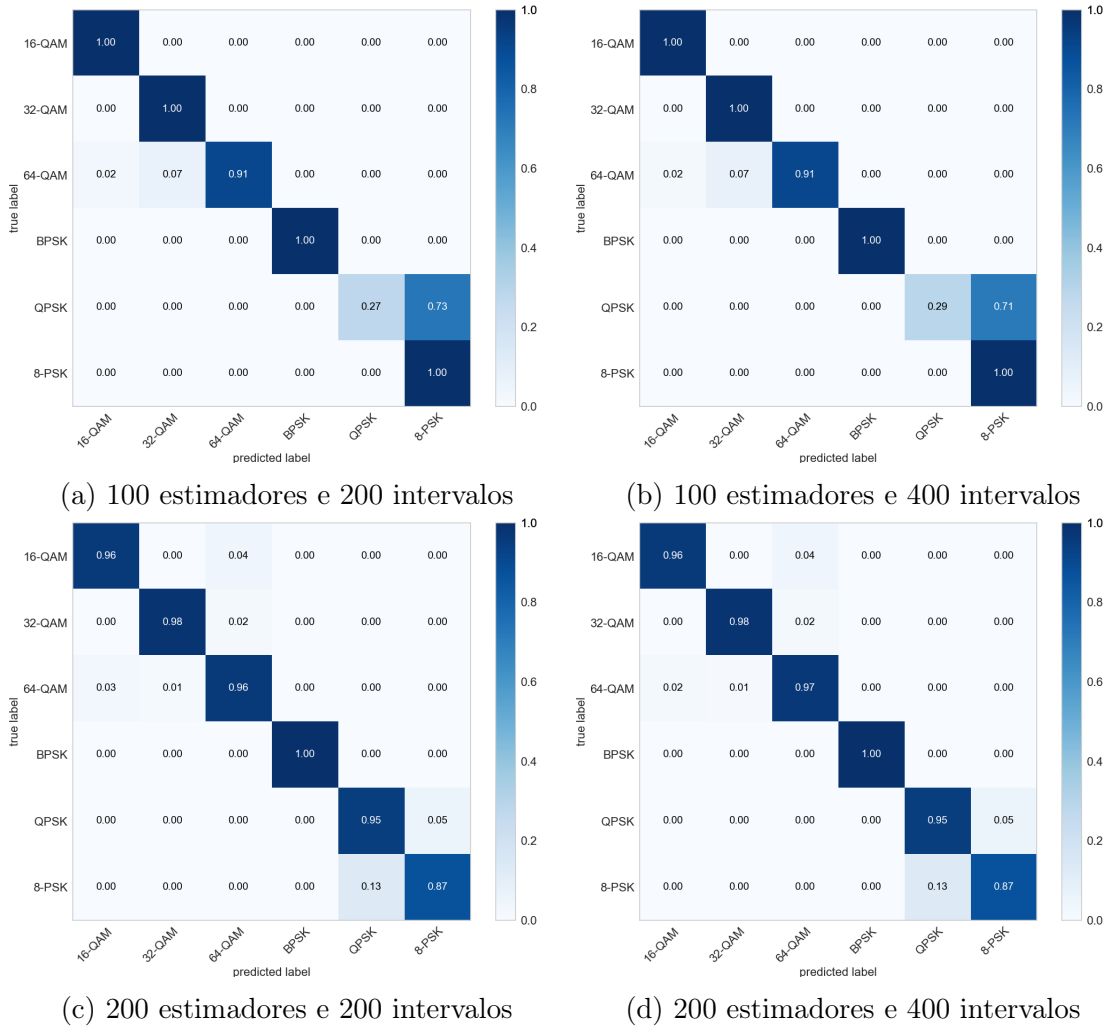


Figura 4.3: Progressão dos resultados para o classificador *Random Forest* com a variação de parâmetros do modelo.

figura (d). Os resultados são apresentados em ordem de acurácia obtida, refletindo a eficácia das configurações testadas.

Resultados consolidados para <i>Random Forest</i>						
Nome	Valores					
Intervalos do histograma	200	400	200	400	400	200
Número de estimadores	100	100	200	200	600	600
MAE	0,1564	0,1563	0,0563	0,0536	0,0122	0,0089
MSE	0,1677	0,1735	0,0855	0,0813	0,0196	0,0160
RMSE	0,4095	0,4165	0,2923	0,2852	0,1401	0,1264
Acurácia	84,93%	85,03%	95,83%	96,01%	99,00%	99,31%

Tabela 4.6: Resultados obtidos a partir dos experimentos com o modelo *Random Forest*.

Por fim, temos a Tabela 4.6 que exhibe todos os resultados conforme a configuração utilizada. Observamos que para este classificador, a influência do número de estimadores apresentou resultados interessantes, demonstrando que uma quantidade

maior de estimadores não necessariamente representa melhoria nos resultados.

4.2.1.2 Resultados para o classificador de modulações com *AdaBoost*.

A seguir, são apresentados os resultados obtidos com o uso do algoritmo *AdaBoost* para a classificação do tipo de modulação. A Figura 4.4 exibe o melhor desempenho do modelo, com detalhes dos parâmetros utilizados na Tabela 4.7.

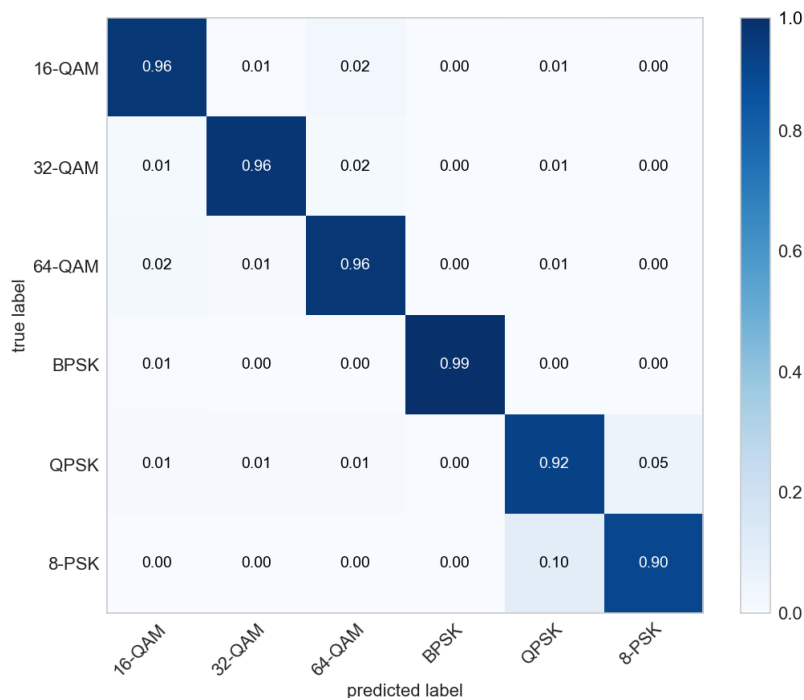


Figura 4.4: Matriz de confusão representando o melhor desempenho do classificador *AdaBoost* a partir dos parâmetros ótimos encontrados na Tabela 4.7.

Nota-se que este resultado apresenta uma acurácia de 95,06%. Além disso, observamos uma acurácia de treino de 100%, MAE de 0,0842, MSE de 0,1894 e RMSE de 0,4352. Esses valores indicam um alto nível de precisão preditiva do modelo na classificação das instâncias.

A Figura 4.5 exibe a evolução dos resultados ao utilizar os parâmetros descritos na Tabela 4.8. Observamos a influência do número de estimadores, explorando valores de 100, 200 e 600, bem como o impacto do número de intervalos do histograma, com bases de dados construídas com 200 e 400 intervalos. Essas alterações são evidenciadas na Figura 4.5, onde a sub figura (a) mostra o resultado com menor acurácia, e as subsequentes demonstram melhorias incrementais até a sub figura (d). Os resultados são apresentados em ordem de acurácia obtida, refletindo a eficácia das configurações testadas.

Parâmetros	
Nome	Valor
Estimador base	Árvore de decisão
Número de estimadores	600
Taxa de aprendizado	1,0
Algoritmo	<i>SAMME.R</i>
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Amostras do sinal original	262.144
Intervalos do histograma (F)	200

Tabela 4.7: Parâmetros utilizados no treinamento do modelo *AdaBoost* com melhor desempenho para a classificação da modulação.

Por fim, temos a Tabela 4.9 que exibe todos os resultados conforme a configuração utilizada. Observamos que para este classificador, a influência do número de estimadores apresentou resultados interessantes, demonstrando que uma quantidade

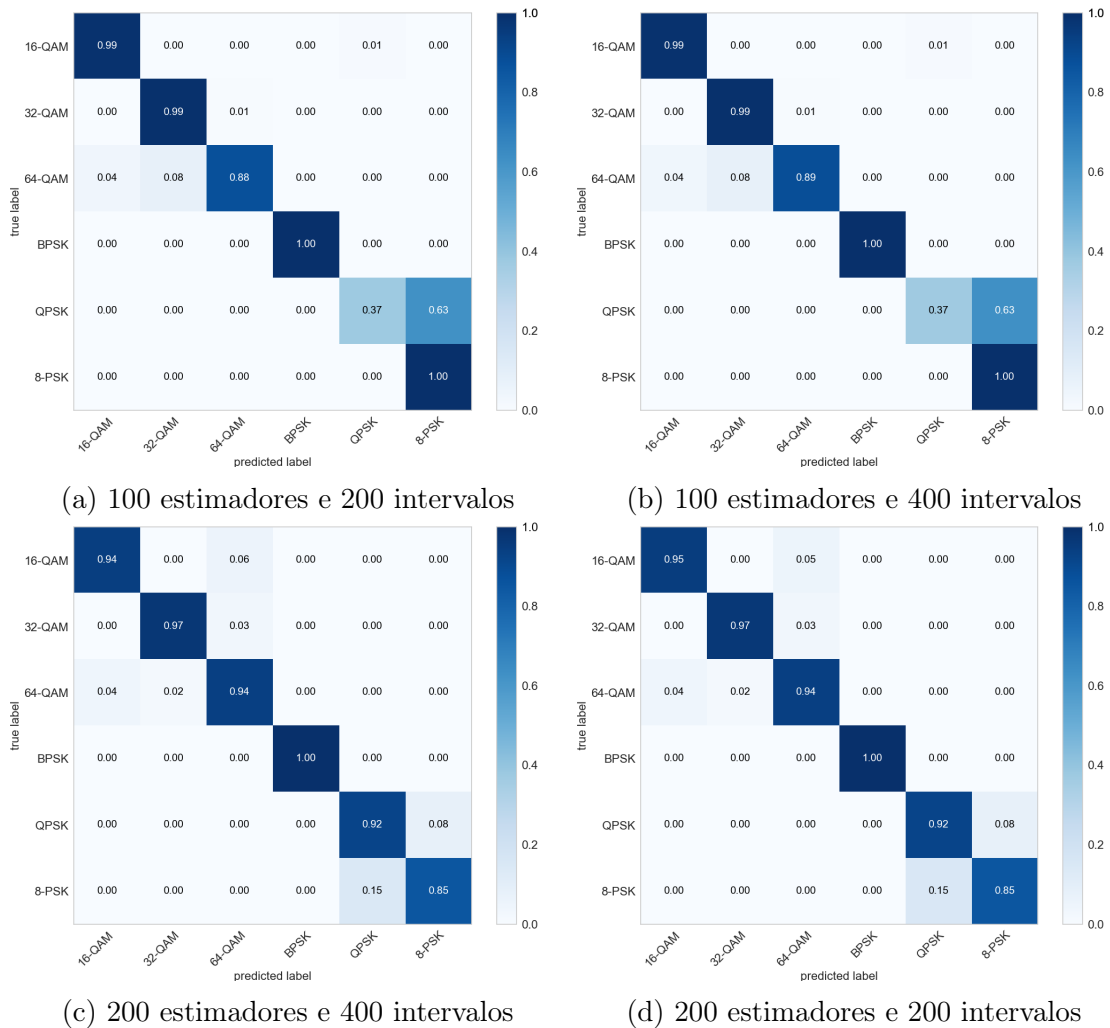


Figura 4.5: Progressão dos resultados para o classificador *AdaBoost* com a variação de parâmetros do modelo.

Parâmetros explorados durante o treino	
Nome	Valores
Número de estimadores	100, 200, 600
Intervalos do histograma	200, 400

Tabela 4.8: Conjunto de parâmetros explorados durante os experimentos com o modelo *AdaBoost*.

maior de estimadores não representa melhoria nos resultados.

Resultados consolidados para <i>AdaBoost</i>						
Nome	Valores					
Intervalos do histograma	200	400	400	200	400	200
Número de estimadores	100	100	200	200	600	600
MAE	0,1579	0,1548	0,0804	0,0797	0,0921	0,0842
MSE	0,1974	0,1916	0,1254	0,1234	0,2050	0,1894
RMSE	0,4443	0,4377	0,3542	0,3512	0,4528	0,4352
Acurácia	85,70%	85,86%	94,18%	94,18%	94,45%	95,06%

Tabela 4.9: Resultados obtidos a partir dos experimentos com o modelo *AdaBoost*.

4.2.1.3 Resultados para o classificador de modulações com *Árvore de decisão CART*

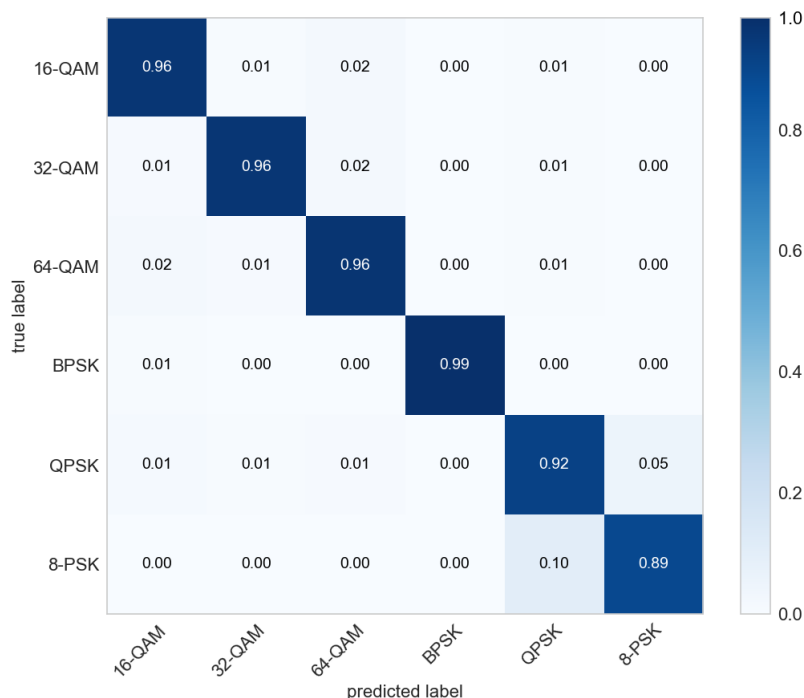


Figura 4.6: Matriz de confusão representando o melhor desempenho do classificador *Árvore de Decisão CART* a partir dos parâmetros ótimos encontrados na Tabela 4.10.

A seguir, apresentam-se os resultados da aplicação do algoritmo de *Árvore*

de *Decisão CART* para a classificação do tipo de modulação. A Figura 4.6 ilustra o melhor resultado alcançado após a conclusão do treinamento do modelo, utilizando os parâmetros indicados na Tabela 4.10.

Parâmetros	
Nome	Valor
Critério de avaliação da qualidade da divisão	<i>Gini</i>
Profundidade máxima da árvore	∞
Núm. min. de amostras para realizar uma divisão	2
Núm. máx. de características para realizar uma divisão	$\sqrt{F} \approx 14$
Quantidade máxima de folhas	∞
Valor mín. de impureza para dividir um ramo	≥ 0
Dicionário com os pesos de cada classe	∞
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	200

Tabela 4.10: Parâmetros utilizados no treinamento do modelo *Árvore de Decisão CART* com melhor desempenho para a classificação da modulação.

Nota-se que este resultado apresenta uma acurácia de 95,10%. Além disso, observamos uma acurácia de treino de 100%, MAE de 0,0842, MSE de 0,1931 e RMSE de 0,4394. Esses valores indicam um alto nível de precisão preditiva do modelo na classificação das instâncias.

A Figura 4.7 exibe a evolução dos resultados ao utilizar os parâmetros descritos na Tabela 4.11. Observamos a influência da profundidade da árvore, explorando valores de 10 e ∞ , bem como o impacto do número de intervalos do histograma, com bases de dados construídas com 200 e 400 intervalos. Essas alterações são evidenciadas na Figura 4.7, onde a sub figura (a) mostra o resultado com menor acurácia, e as subseqüentes demonstram melhorias incrementais até a sub figura (d). Os resultados são apresentados em ordem de acurácia obtida, refletindo a eficácia das configurações testadas.

Parâmetros explorados durante o treino	
Nome	Valores
Intervalos do histograma	200, 400
Profundidade da árvore	10, ∞

Tabela 4.11: Conjunto de parâmetros explorados durante os experimentos com o modelo *Árvore de Decisão CART*

Por fim, temos a Tabela 4.12 que exibe todos os resultados conforme a con-

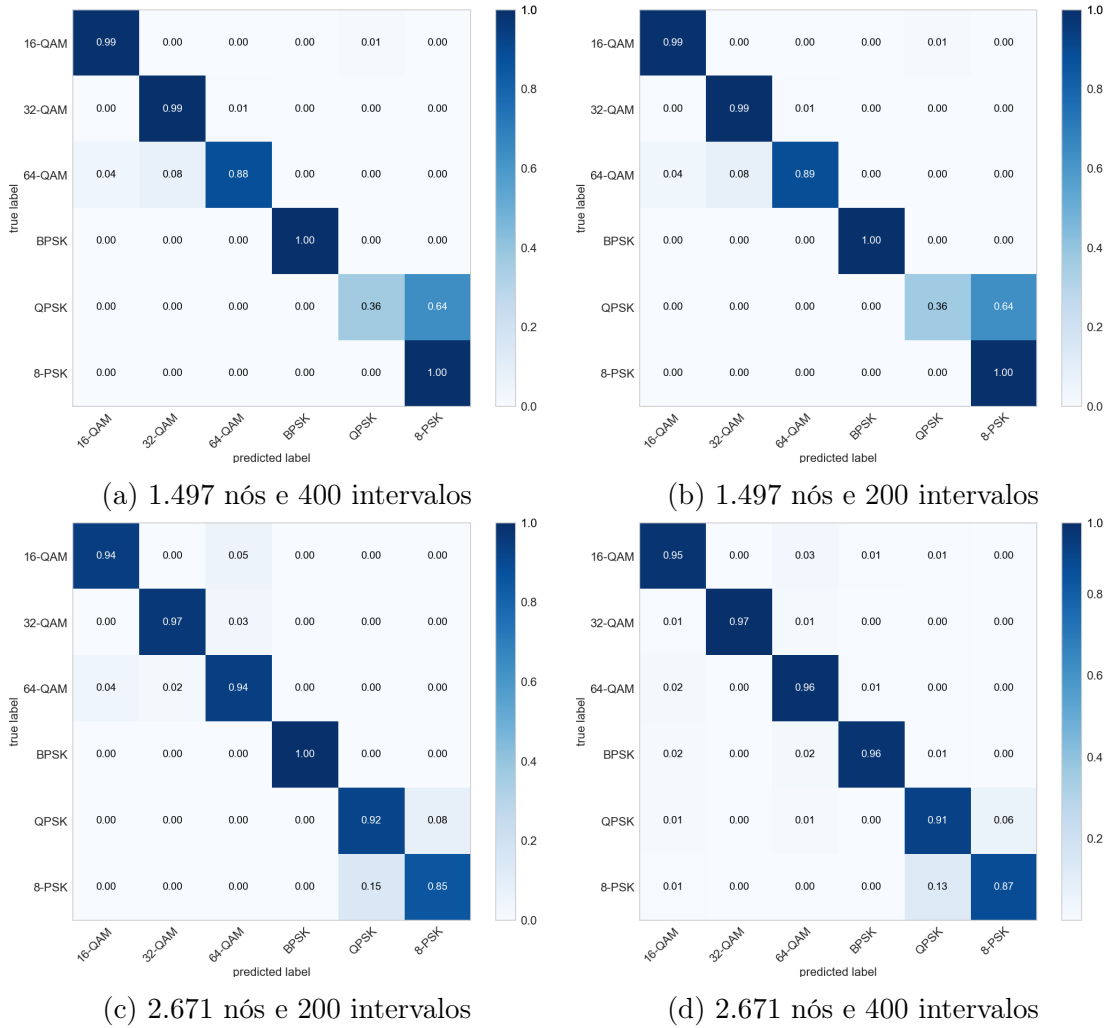


Figura 4.7: Progressão dos resultados para o classificador *Árvore de Decisão CART* com a variação de parâmetros do modelo.

figuração utilizada. Observamos que para este classificador, a influência dos nós de profundidade apresentou resultados interessantes, onde todos os melhores resultados foram para uma quantidade significativa de nós, uma vez que o parâmetro ∞ não limita a quantidade de nós da árvore final.

Resultados consolidados para <i>Árvore de Decisão CART</i>					
Nome	Valores				
Intervalos do histograma	400	200	200	400	200
Nós de profundidade	1.497	1.497	2.671	2.671, 2.995	2.995
MAE	0,1596	0,1563	0,0801	0,0955	0,0842
MSE	0,1999	0,1935	0,1244	0,2137	0,1931
RMSE	0,4471	0,4399	0,3528	0,4623	0,4394
Acurácia	85,56%	85,74%	94,18%	94,28%	95,10%

Tabela 4.12: Resultados obtidos a partir dos experimentos com o modelo *Árvore de Decisão CART*

4.2.1.4 Resultados para o classificador de modulações com *Gradient Boosting*

A seguir, apresentam-se os resultados da aplicação do algoritmo de *Gradient Boosting* para a classificação do tipo de modulação. A Figura 4.8 ilustra o melhor resultado alcançado após a conclusão do treinamento do modelo, utilizando os parâmetros indicados na Tabela 4.13.

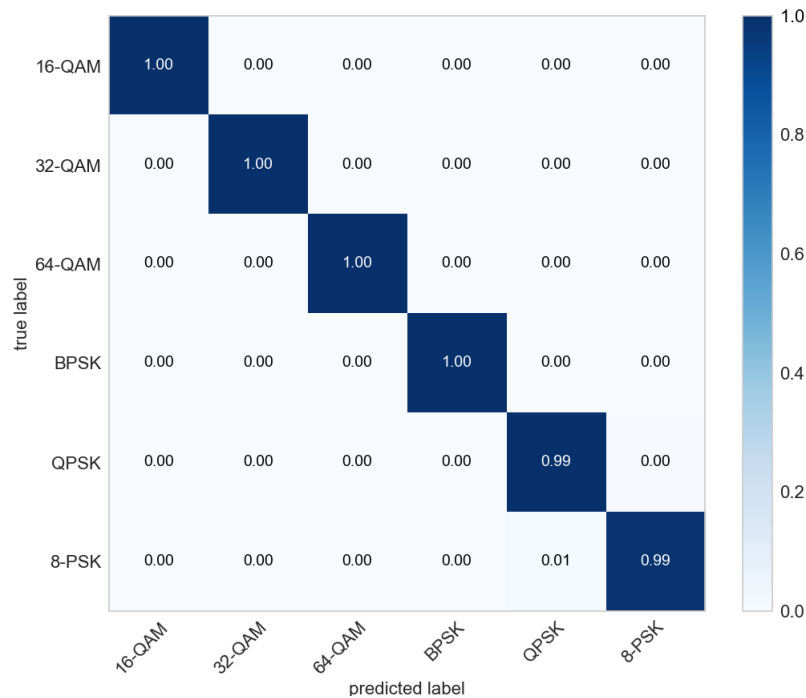


Figura 4.8: Matriz de confusão representando o melhor desempenho do classificador *Gradient Boosting* a partir dos parâmetros ótimos encontrados na Tabela 4.13.

Nota-se que este resultado apresenta uma acurácia de 99,69%. Além disso, observamos uma acurácia de treino de 100%, MAE de 0,0036, MSE de 0,0045 e RMSE de 0,0672. Esses valores indicam um alto nível de precisão preditiva do modelo na classificação das instâncias. A Figura 4.9 exibe a evolução dos resultados ao utilizar os parâmetros descritos na Tabela 4.14. Observamos a influência do número de estimadores, explorando valores de 100, 200 e 600, bem como o impacto do número de intervalos do histograma, com bases de dados construídas com 200 e 400 intervalos. Essas alterações são evidenciadas na Figura 4.9, onde a sub figura (a) mostra o resultado com menor acurácia, e as subsequentes demonstram melhorias incrementais até a sub figura (d). Os resultados são apresentados em ordem de acurácia obtida, refletindo a eficácia das configurações testadas.

Parâmetros	
Nome	Valor
Função de perda (<i>loss</i>)	<i>log_loss</i>
Taxa de aprendizado	0,1
Número de estimadores	600
Subamostragem	Sim
Critério de avaliação da qualidade da divisão	<i>friedman_mse</i>
Núm. min. de amostras para realizar uma divisão	2
Profundidade máxima da árvore	∞
Valor mín. de impureza para dividir um ramo	≥ 0
Núm. máx. de características para realizar uma divisão	∞
Quantidade máxima de folhas	∞
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Amostras do sinal original	262.144
Intervalos do histograma (F)	200

Tabela 4.13: Parâmetros utilizados no treinamento do modelo *Gradient Boosting* com melhor desempenho para a classificação da modulação.

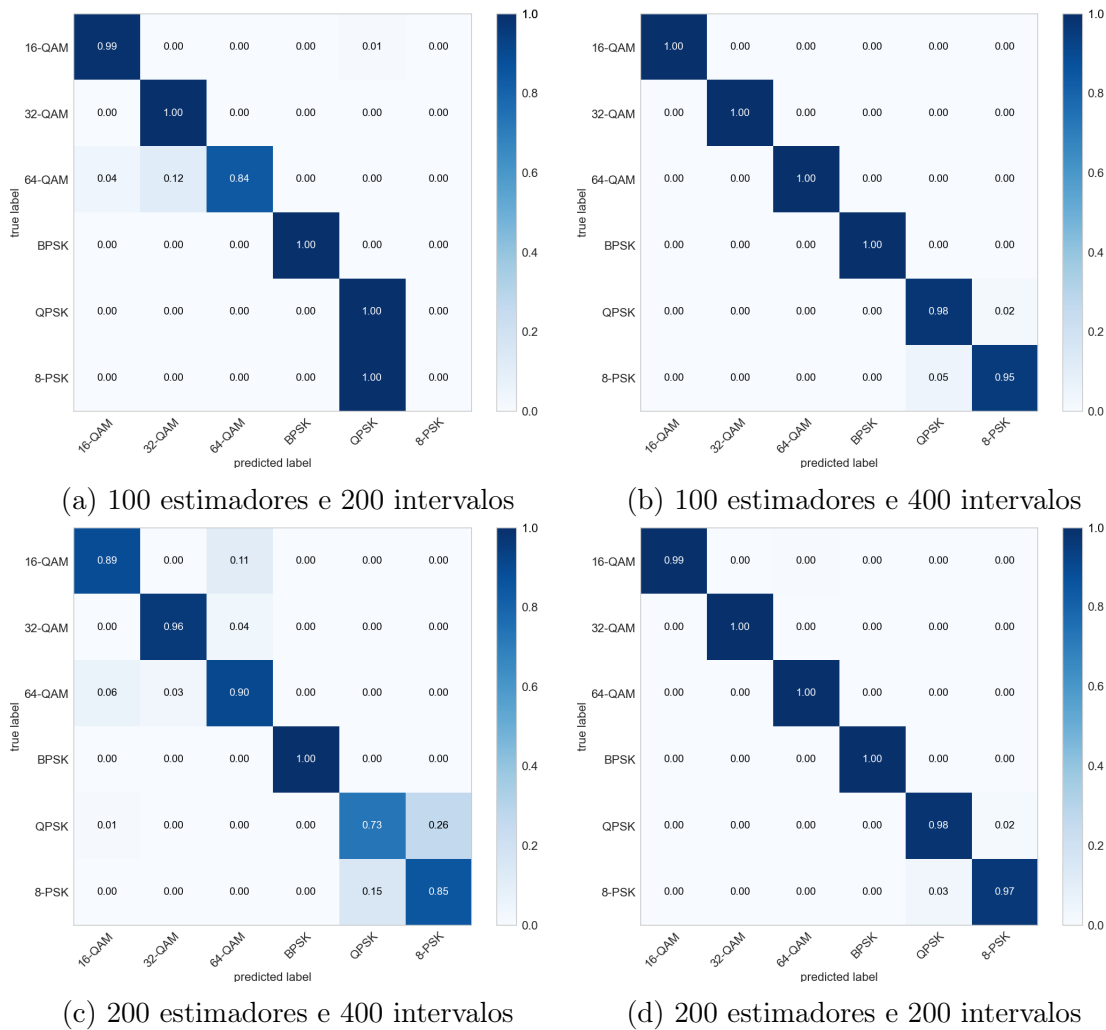


Figura 4.9: Progressão dos resultados para o classificador *Gradient Boosting* com a variação de parâmetros do modelo.

Parâmetros explorados durante o treino	
Nome	Valores
Número de estimadores	100, 200, 600
Intervalos do histograma	200, 400

Tabela 4.14: Conjunto de parâmetros explorados durante os experimentos com o modelo *Gradient Boosting*

Por fim, temos a Tabela 4.15 que exhibe todos os resultados conforme a configuração utilizada. Observamos que para este classificador, a influência do número de estimadores apresentou resultados interessantes, demonstrando que uma quantidade maior de estimadores não representa melhoria nos resultados.

Resultados consolidados para <i>Gradient Boosting</i>						
Nome	Valores					
Intervalos do histograma	400	200	200	200	200	200
Número de estimadores	100	100	100	100	200	600
MAE	0,1474	0,1558	0,0142	0,0109	0,0081	0,0036
MSE	0,1865	0,2555	0,0254	0,0148	0,0132	0,0045
RMSE	0,4319	0,5055	0,1593	0,1215	0,1149	0,0672
Acurácia	86,72%	88,78%	98,99%	99,04%	99,37%	99,69%

Tabela 4.15: Resultados obtidos a partir dos experimentos com o modelo *Gradient Boosting*

4.2.1.5 Resultados para o classificador de modulações com *Perceptron Multicamada*

A seguir, apresentam-se os resultados da aplicação do algoritmo de *Perceptron Multicamada* para a classificação do tipo de modulação. A Figura 4.10 ilustra o melhor resultado alcançado após a conclusão do treinamento do modelo, utilizando os parâmetros indicados na Tabela 4.16.

Nota-se que este resultado apresenta uma acurácia de 99,85%. Além disso, observamos uma acurácia de treino de 100%, MAE de 0,0024, MSE de 0,0051 e RMSE de 0,0713. Esses valores indicam um alto nível de precisão preditiva do modelo na classificação das instâncias.

A Figura 4.11 exhibe a evolução dos resultados ao utilizar os parâmetros descritos na Tabela 4.17. Observamos a influência do número de estimadores, explorando valores de 100, 200 e 600, bem como o impacto do número de intervalos do histograma, com bases de dados construídas com 200 e 400 intervalos. Essas alterações são evidenciadas na Figura 4.9, onde a sub figura (a) mostra o resultado com menor

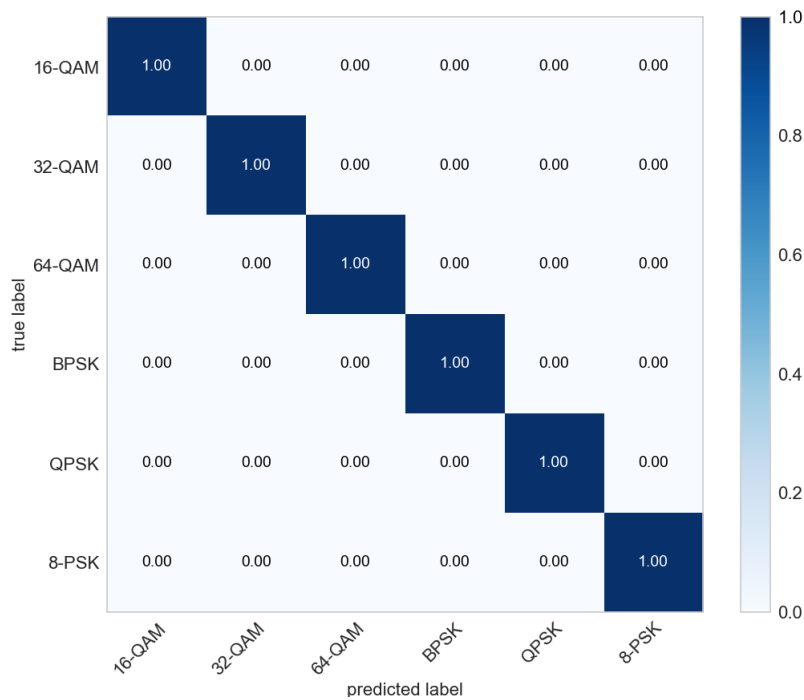


Figura 4.10: Matriz de confusão representando o melhor desempenho do classificador *Perceptron Multicamada* a partir dos parâmetros ótimos encontrados na Tabela 4.16.

Parâmetros	
Nome	Valor
Camadas ocultas	100
Função de ativação	<i>relu</i>
Método de otimização	<i>adam</i>
Tamanho do <i>batch</i>	auto
Taxa de aprendizado	adaptativa
Taxa inicial de aprendizado	0,001
Expoente de decaimento da taxa de aprendizado	0,5
Número máximo de iterações	700
<i>Momentum</i>	0,9
<i>Momentum</i> de Nesterov	Sim
Beta 1	0,9
Beta 2	0,999
Epsilon	$1e^{-08}$
Número máximo de funções de perda	15.000
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Amostras do sinal original	262.144
Intervalos do histograma (F)	200

Tabela 4.16: Parâmetros utilizados no treinamento do modelo *Perceptron Multicamada* com melhor desempenho para a classificação da modulação.

acurácia, e as subsequentes demonstram melhorias incrementais até a sub figura (d). Os resultados são apresentados em ordem de acurácia obtida, refletindo a eficácia das configurações testadas.

Por fim, temos a Tabela 4.18 que exhibe todos os resultados conforme a confi-

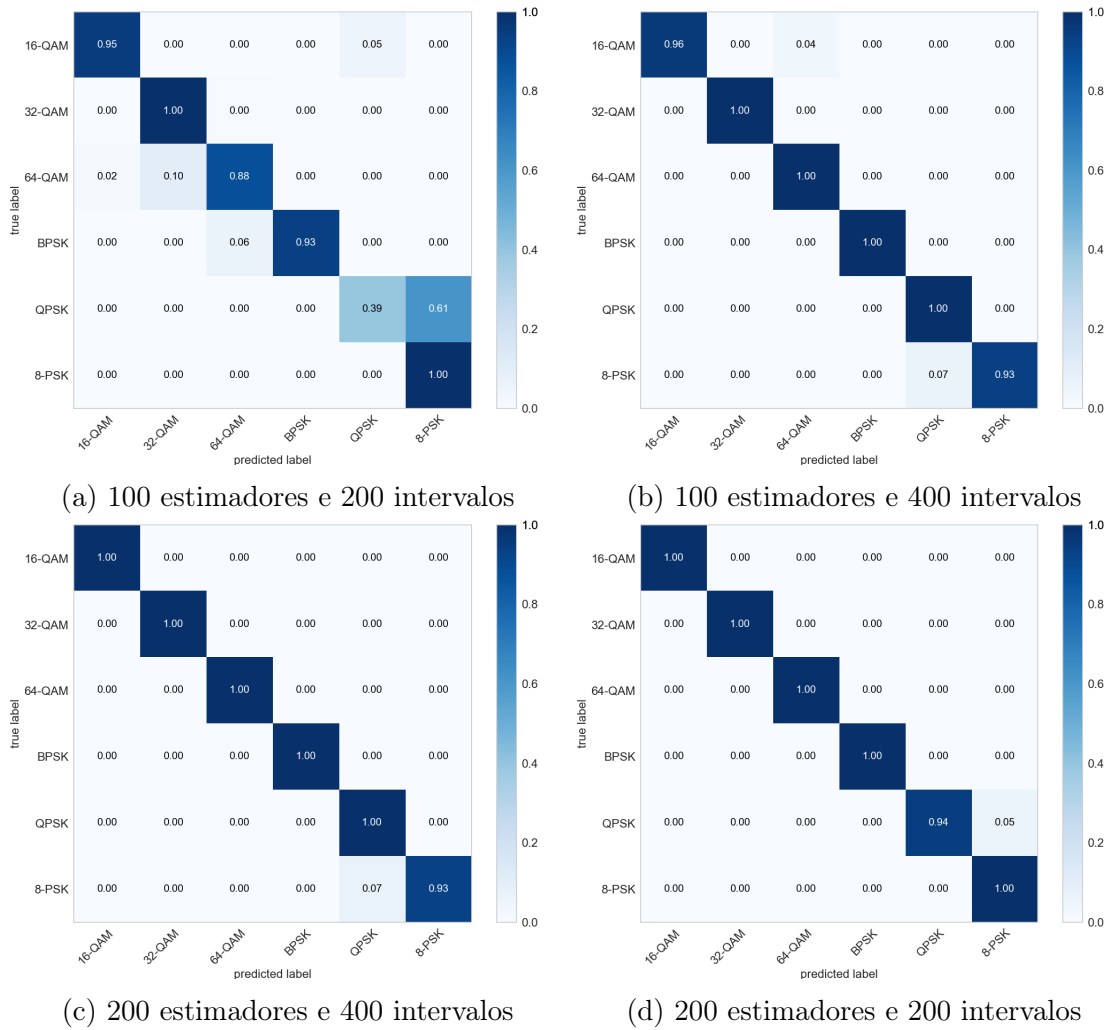


Figura 4.11: Progressão dos resultados para o classificador *Perceptron Multicamada* com a variação de parâmetros do modelo.

Parâmetros explorados durante o treino	
Nome	Valores
Número de estimadores	200, 600
Intervalos do histograma	200, 400
Quantidade de Épocas	100, 700

Tabela 4.17: Conjunto de parâmetros explorados durante os experimentos com o modelo *Gradient Boosting*

guração utilizada. Observamos que para este classificador, a influência do número de estimadores apresentou resultados interessantes, demonstrando que uma quantidade maior de estimadores não representa melhoria nos resultados.

Resultados consolidados para <i>Perceptron Multicamada</i>						
Nome	Valores					
Intervalos do histograma	200, 400	200	200	400	200	200
Número de estimadores	100	400	200	600	600	100
Épocas	100, 700	700	700	100, 700	100	100, 700
MAE	1,3150	0,0217	0,0084	0,0059	0,0029	0,0024
MSE	2,6156	0,0375	0,0100	0,0112	0,0059	0,0051
RMSE	1,6173	0,1938	0,1001	0,1056	0,0765	0,0713
Acurácia	24,22%	98,59%	99,23%	99,62%	99,80%	99,85%

Tabela 4.18: Resultados obtidos a partir dos experimentos com o modelo *Perceptron Multicamada*

4.2.2 Predição do valor de OSNR

Nesta seção, detalhamos os resultados para a predição do valor de OSNR. A Figura 4.12 ilustra o componente interno do *Preditor de OSNR*, essencial para nossos experimentos. Selecionamos três algoritmos de aprendizado de máquina: *Random Forest* e *Perceptron Multicamada* para classificação, e *Random Forest* para regressão. Utilizamos uma estratégia sequencial, onde a modulação identificada pelo *Classificador de Modulações* direciona a escolha do *Preditor de OSNR* correspondente. Os sinais são caracterizados por valores de OSNR distribuídos em intervalos de 0,5dB/0,1nm, o que nos permite usar esses intervalos como categorias para os classificadores ou para definir uma relação numérica para os regressores. Os algoritmos são treinados com bases de dados de histogramas com 200 e 400 intervalos, seguindo os parâmetros padrão. A partir dos resultados iniciais, ajustamos os parâmetros para aprimorar o desempenho. A distribuição dos dados consiste em 30% para treinamento e 70% para testes.

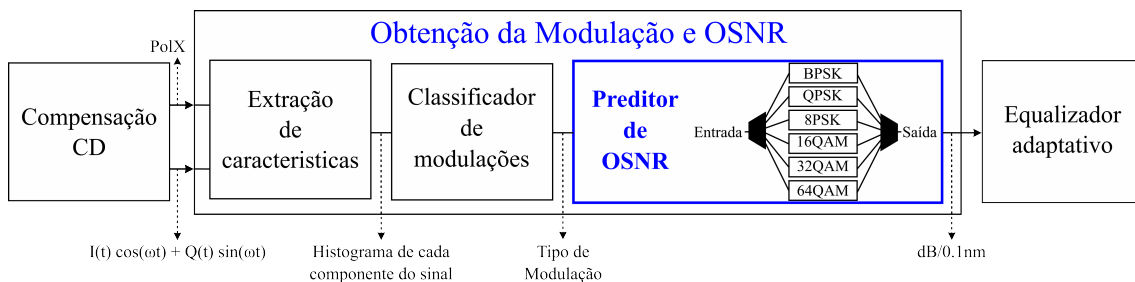


Figura 4.12: Bloco proposto para a predição do valor de *OSNR*.

4.2.2.1 Predição da OSNR para modulação BPSK usando *Random Forest* como classificador

Os resultados do uso do algoritmo *Random Forest* para prever o valor de OSNR em modulação BPSK são detalhados a seguir. A eficácia do modelo é evidenciada na Figura 4.13, cujos parâmetros são especificados na Tabela 4.19. A análise da matriz de confusão revelou uma acurácia inicial de 40,81%. No entanto, ao considerar uma margem de erro de $\pm 0,5dB/0,1nm$, a acurácia melhora significativamente para 82,97%, ilustrando a importância de uma abordagem flexível na avaliação dos resultados de predição de OSNR.

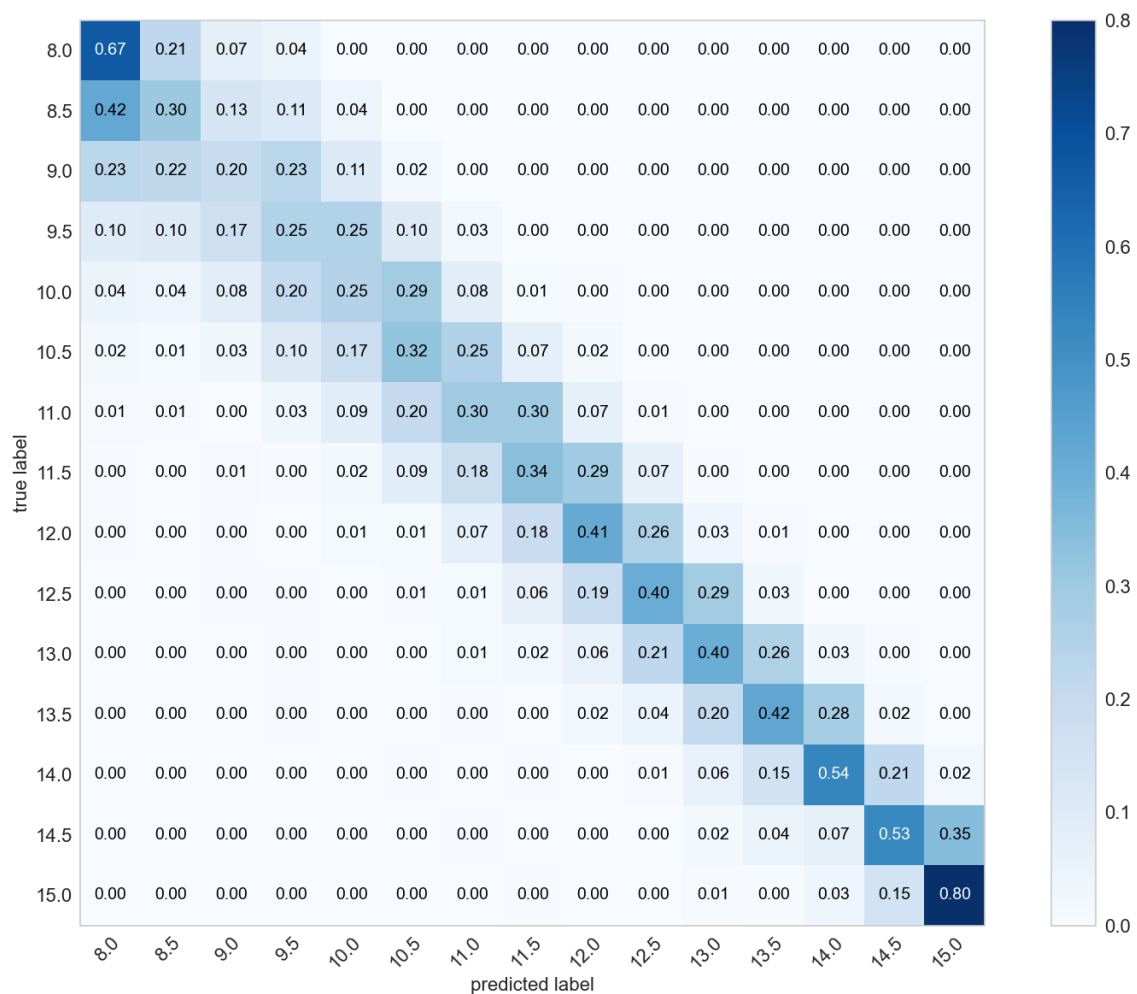


Figura 4.13: Resultados com melhor desempenho para o preditor de OSNR na modulação BPSK usando *Random Forest*, a partir dos parâmetros encontrados na Tabela 4.19.

A análise da Tabela 4.20 para a modulação BPSK mostra que o aumento do número de estimadores de 200 para 600 melhora levemente a acurácia, mas não significativamente, indicando que mais estimadores não levam necessariamente a

Parâmetros	
Nome	Valor
Número de estimadores	600
Critério de avaliação da qualidade da divisão	<i>entropy</i>
Profundidade máxima da árvore	∞
Núm. min. de amostras para realizar uma divisão	2
Núm. máx. de características para realizar uma divisão	$\sqrt{F} \approx 14$
Quantidade máxima de folhas	∞
Valor mín. de impureza para dividir um ramo	Não
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	200

Tabela 4.19: Parâmetros ótimos utilizados no treino do preditor de OSNR para modulação BPSK com *Random Forest*

melhores resultados. A variação dos intervalos do histograma de 200 para 400 não apresentou uma melhoria proporcional na acurácia, sugerindo um ponto ótimo de configuração. A acurácia composta, considerando uma variação aceitável de erro, demonstra uma eficiência considerável do modelo em classificar corretamente os sinais.

Resultados consolidados para <i>Random Forest</i> em BPSK				
Nome	Valores			
Intervalos do histograma	400	400	200	200
Número de estimadores	200	600	200	600
MAE	1,1322	1,1178	0,8556	0,8328
MSE	2,8217	2,7983	1,6022	1,5494
RMSE	1,6798	1,6728	1,2658	1,2448
Acurácia	33,97%	34,75%	39,92%	40,81%
Acurácia composta	72,69%	73,22%	82,11%	82,97%

Tabela 4.20: Resultados obtidos a partir dos experimentos realizados com o classificador *Random Forest* para modulação BPSK.

4.2.2.2 Predição da OSNR para modulação QPSK usando *Random Forest* como classificador

Prosseguindo com a apresentação do desempenho dos modelos, detalhamos os resultados do algoritmo *Random Forest* na predição de OSNR para modulação QPSK. A precisão do modelo é destacada na Figura 4.14, com os parâmetros de treinamento disponíveis na Tabela 4.21. Similarmente ao caso anterior, este resultado também é dividido em duas partes: uma acurácia direta de 45,42% e uma acurácia ajustada, considerando uma variação aceitável, de 89,80%. Esta flexibilidade na

avaliação evidencia a adaptabilidade e eficiência do modelo em diferentes contextos de modulação. A Tabela 4.22 revela que o número de estimadores, seja 200 ou 600,

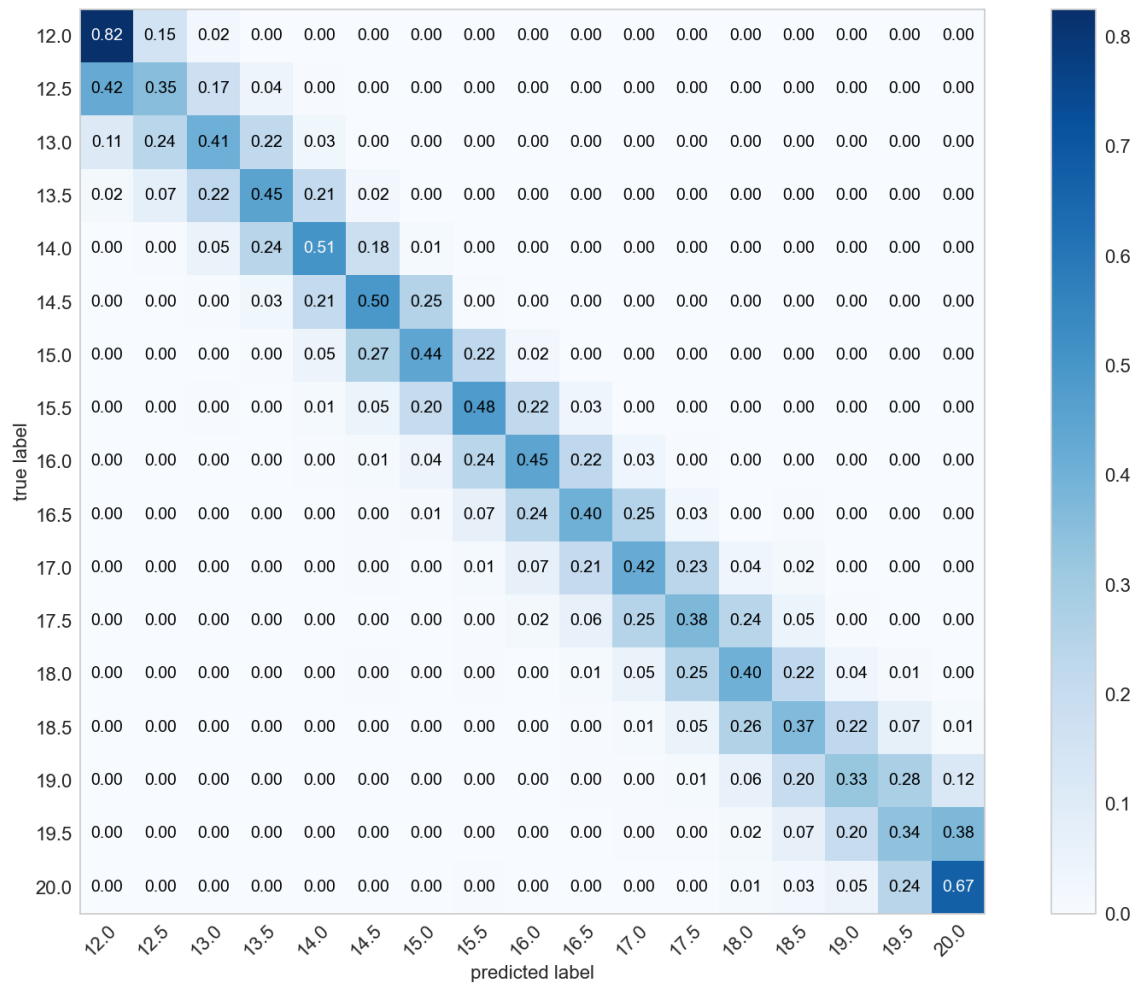


Figura 4.14: Resultados com melhor desempenho para o preditor de OSNR na modulação QPSK usando *Random Forest*, a partir dos parâmetros encontrados na Tabela 4.21.

tem um impacto marginal na acurácia para a modulação QPSK. As mudanças nos intervalos do histograma também não resultaram em melhorias significativas na acurácia, sugerindo que o modelo atingiu uma estabilidade em sua capacidade preditiva com as configurações testadas. A acurácia composta, que considera uma margem de erro, demonstra uma melhoria substancial, indicando a eficácia do modelo em prever com precisão em uma faixa de tolerância aceitável.

Parâmetros	
Nome	Valor
Número de estimadores	600
Critério de avaliação da qualidade da divisão	<i>entropy</i>
Profundidade máxima da árvore	∞
Núm. min. de amostras para realizar uma divisão	2
Núm. máx. de características para realizar uma divisão	$\sqrt{F} \approx 14$
Quantidade máxima de folhas	∞
Valor mín. de impureza para dividir um ramo	Não
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	200

Tabela 4.21: Parâmetros ótimos utilizados no treino do preditor de OSNR para modulação QPSK com *Random Forest*.

Resultados consolidados para <i>Random Forest</i> em QPSK				
Nome	Valores			
Intervalos do histograma	400	400	200	200
Número de estimadores	200	600	200	600
MAE	0,7267	0,7123	0,6811	0,6723
MSE	1,1547	1,1078	1,0179	1,0017
RMSE	1,0745	1,0525	1,0089	1,0009
Acurácia	44,04%	44,26%	45,44%	45,42%
Acurácia composta	86,78%	87,67%	88,82%	89,80%

Tabela 4.22: Resultados obtidos a partir dos experimentos realizados com o classificador *Random Forest* para modulação QPSK.

4.2.2.3 Predição da OSNR para modulação 8-PSK usando *Random Forest* como classificador

Nesta seção, exploramos os resultados alcançados com o algoritmo *Random Forest* na previsão de OSNR para modulação 8-PSK. A Figura 4.15 destaca o desempenho ótimo do modelo, conforme detalhado na Tabela 4.23. As acurácias observadas foram de 46,11% para a previsão exata e de 88,33% ao considerar uma variação aceitável, ressaltando a capacidade do modelo de adaptar-se às variações inerentes ao processo de modulação.

A análise dos resultados para a modulação 8-PSK utilizando o classificador *Random Forest*, conforme exibido na Tabela 4.24, indica que variações no número de estimadores e nos intervalos do histograma têm impacto limitado na melhoria da acurácia. Interessantemente, a acurácia composta, que considera uma tolerância de erro, mostra uma eficácia considerável, sugerindo que o modelo tem capacidade de prever em uma faixa de erro aceitável, reforçando a utilidade do classificador em

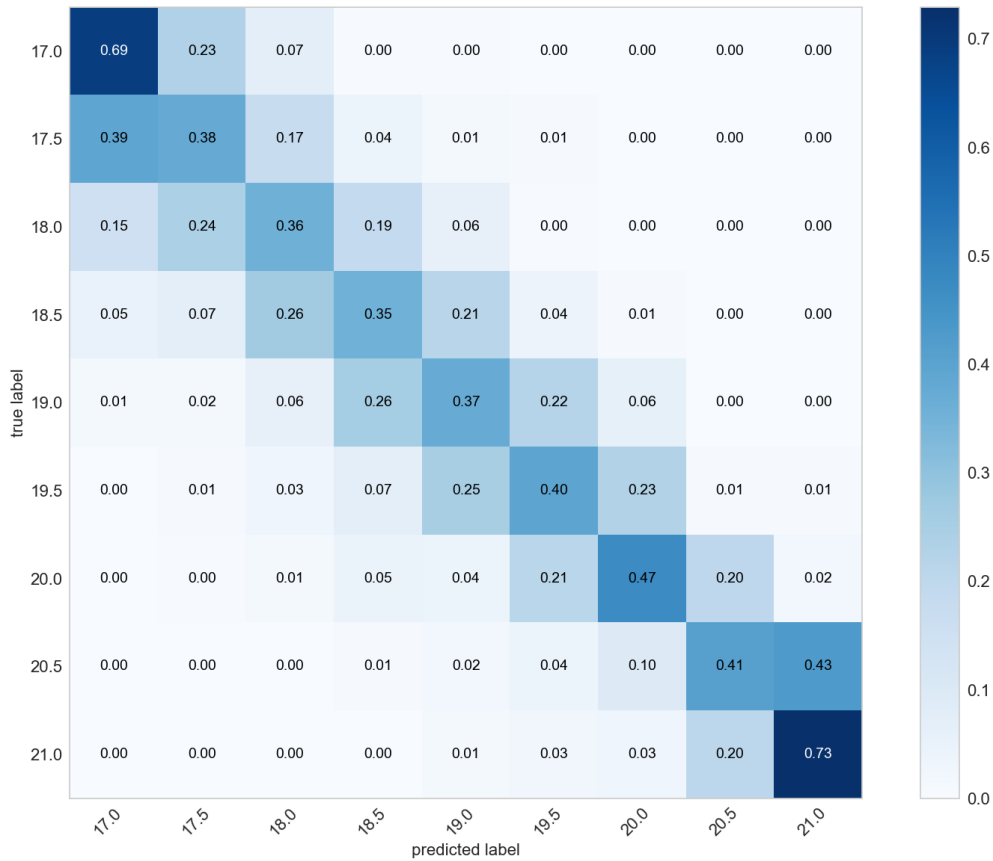


Figura 4.15: Resultados com melhor desempenho para o preditor de OSNR na modulação 8-PSK usando *Random Forest*, a partir dos parâmetros encontrados na Tabela 4.23.

Parâmetros	
Nome	Valor
Número de estimadores	600
Critério de avaliação da qualidade da divisão	<i>entropy</i>
Profundidade máxima da árvore	∞
Núm. min. de amostras para realizar uma divisão	2
Núm. máx. de características para realizar uma divisão	$\sqrt{F} \approx 14$
Quantidade máxima de folhas	∞
Valor mín. de impureza para dividir um ramo	Não
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	400

Tabela 4.23: Parâmetros ótimos utilizados no treino do preditor de OSNR para modulação 8-PSK com *Random Forest*

contextos práticos.

Resultados consolidados para <i>Random Forest</i> em 8-PSK				
Nome	Valores			
Intervalos do histograma	400	200	200	400
Número de estimadores	200	600	200	600
MAE	0,7023	0,6866	0,6972	0,6968
MSE	1,0931	1,0292	1,0685	1,1125
RMSE	1,0455	1,0145	1,0337	1,0548
Acurácia	45,05%	45,69%	45,88%	46,11%
Acurácia composta	88,28%	88,15%	86,94%	88,33%

Tabela 4.24: Resultados obtidos a partir dos experimentos realizados com o classificador *Random Forest* para modulação 8-PSK.

4.2.2.4 Predição da OSNR para modulação 16-QAM usando *Random Forest* como classificador

Proseguimos com a análise do desempenho do *Random Forest* na previsão de OSNR para a modulação 16-QAM. A Figura 4.16 ilustra o desempenho mais eficaz alcançado com os parâmetros definidos na Tabela 4.25. O modelo registrou uma acurácia direta de 33,16% e uma acurácia ajustada, considerando uma margem de erro, de 74,61%, ilustrando a habilidade do modelo de prever a OSNR em um espectro de tolerância estabelecido. A análise dos resultados para a modulação 16-QAM

Parâmetros	
Nome	Valor
Número de estimadores	600
Critério de avaliação da qualidade da divisão	<i>entropy</i>
Profundidade máxima da árvore	∞
Núm. min. de amostras para realizar uma divisão	2
Núm. máx. de características para realizar uma divisão	$\sqrt{F} \approx 14$
Quantidade máxima de folhas	∞
Valor mín. de impureza para dividir um ramo	Não
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	200

Tabela 4.25: Parâmetros ótimos utilizados no treino do preditor de OSNR para modulação 16-QAM com *Random Forest*

com o classificador *Random Forest*, detalhada na Tabela 4.26, revela uma relação interessante entre o número de estimadores e o desempenho do modelo. Aumentar o número de estimadores não necessariamente traduz em melhorias significativas na acurácia, sugerindo um ponto de saturação na eficácia do modelo. Notavelmente, a acurácia composta, ajustada para considerar uma margem de erro, indica uma

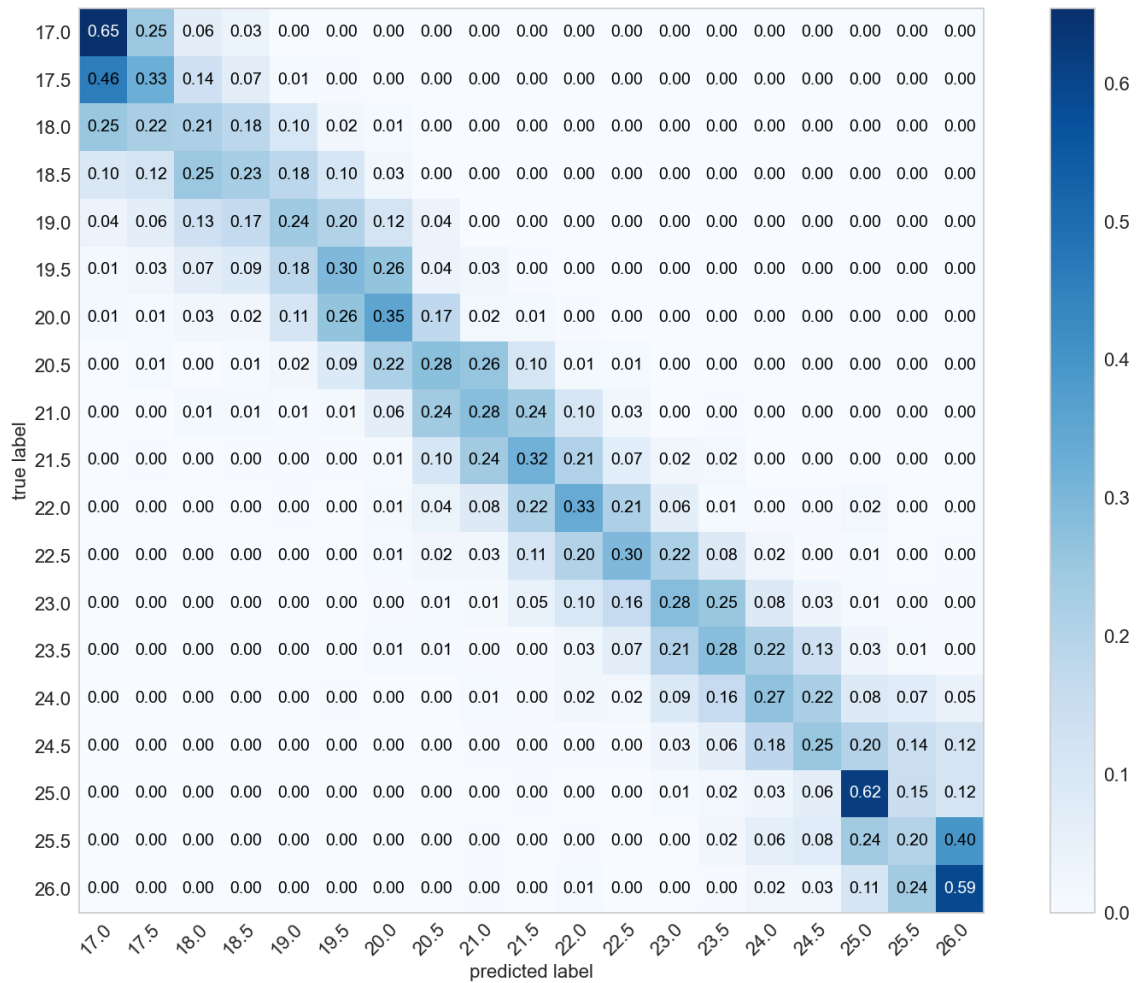


Figura 4.16: Resultados com melhor desempenho para o preditor de OSNR na modulação 16-QAM usando *Random Forest*, a partir dos parâmetros encontrados na Tabela 4.25.

aplicabilidade prática mais ampla, destacando a capacidade do modelo de operar eficientemente em uma variação aceitável.

Resultados consolidados para <i>Random Forest</i> em 16-QAM				
Nome	Valores			
Intervalos do histograma	400	400	200	200
Número de estimadores	200	600	200	600
MAE	1,1989	1,1601	1,1274	1,0645
MSE	3,2371	3,0263	2,5875	2,3399
RMSE	1,7992	1,7396	1,6086	1,5297
Acurácia	31,51%	31,89%	31,91%	33,16%
Acurácia composta	71,93%	73,33%	72,19%	74,61%

Tabela 4.26: Resultados obtidos a partir dos experimentos realizados com o classificador *Random Forest* para modulação 16-QAM.

4.2.2.5 Predição da OSNR para modulação 32-QAM usando *Random Forest* como classificador

Avançamos para a análise da previsão de OSNR na modulação 32-QAM utilizando o *Random Forest*. A Figura 4.17 destaca o ótimo desempenho obtido, cujos detalhes estão especificados na Tabela 4.27. Observamos uma acurácia inicial de 22,31%, que se eleva para 57,88% ao considerarmos uma margem de erro permitida, evidenciando a capacidade do modelo de adaptar-se dentro de uma faixa de tolerância pré-definida. A análise detalhada da Tabela 4.28 revela nuances im-

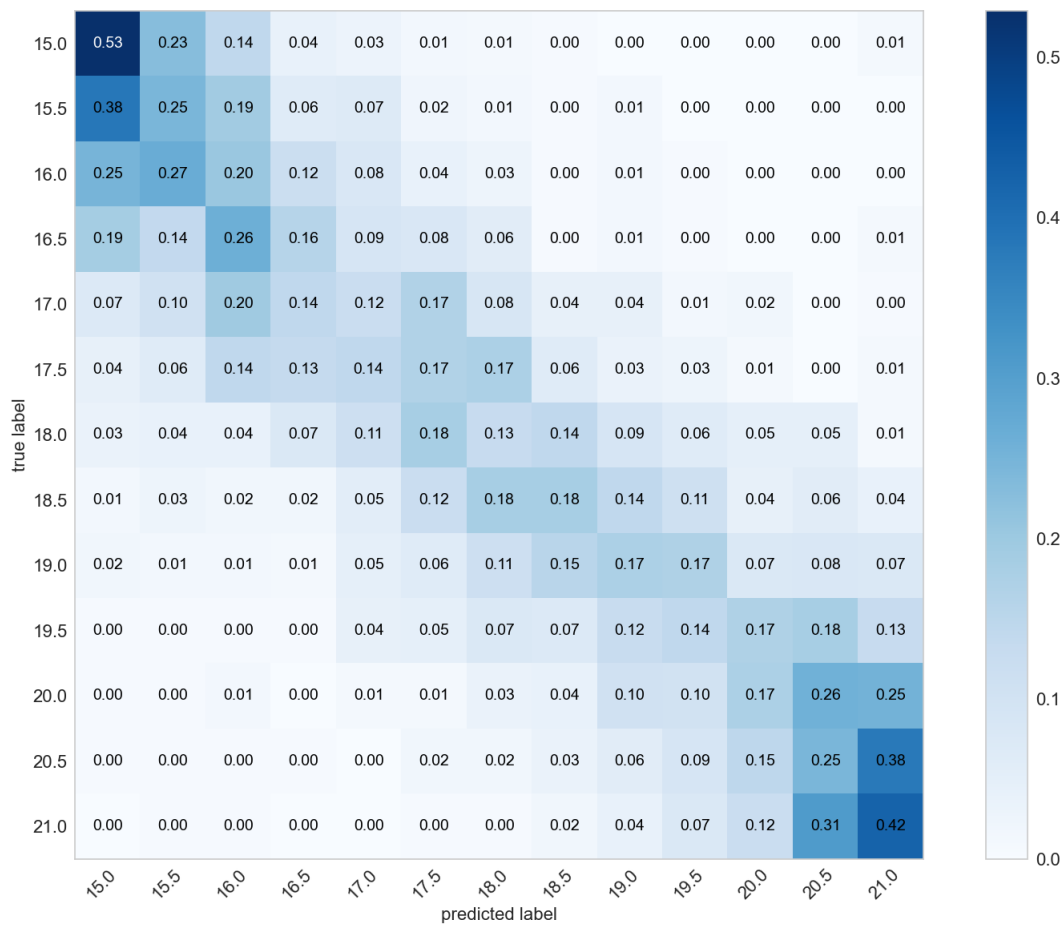


Figura 4.17: Resultados com melhor desempenho para o preditor de OSNR na modulação 32-QAM usando *Random Forest*, a partir dos parâmetros encontrados na Tabela 4.27.

portantes sobre a eficácia do classificador *Random Forest* em modulação 32-QAM. A variação no número de estimadores, de 200 para 600, não corresponde a um aumento proporcional na acurácia, indicando um ponto de diminuição dos retornos. Interessantemente, o aumento dos intervalos do histograma de 200 para 400 melhora a precisão do modelo, como evidenciado pelo aumento na acurácia composta. Esses

Parâmetros	
Nome	Valor
Número de estimadores	600
Critério de avaliação da qualidade da divisão	<i>entropy</i>
Profundidade máxima da árvore	∞
Núm. min. de amostras para realizar uma divisão	2
Núm. máx. de características para realizar uma divisão	$\sqrt{F} \approx 14$
Quantidade máxima de folhas	∞
Valor mín. de impureza para dividir um ramo	Não
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	400

Tabela 4.27: Parâmetros ótimos utilizados no treino do preditor de OSNR para modulação 32-QAM com *Random Forest*

resultados sublinham a importância de uma escolha criteriosa dos parâmetros na otimização do desempenho do classificador.

Resultados consolidados para <i>Random Forest</i> em 32-QAM				
Nome	Valores			
Intervalos do histograma	400	200	200	400
Número de estimadores	200	200	600	600
MAE	1,7500	1,7442	1,6904	1,6192
MSE	5,5397	5,5231	5,1878	4,9724
RMSE	2,3537	2,3501	2,2777	2,2299
Acurácia	20,38%	20,77%	21,54%	22,31%
Acurácia composta	52,88%	53,30%	53,59%	57,88%

Tabela 4.28: Resultados obtidos a partir dos experimentos realizados com o classificador *Random Forest* para modulação 32-QAM.

4.2.2.6 Predição da OSNR para modulação 64-QAM usando *Random Forest* como classificador

Concluimos a análise dos resultados utilizando *Random Forest*, com a modulação 64-QAM. A Figura 4.18 mostra os resultados superiores, cujos detalhes técnicos estão na Tabela 4.29. A precisão do modelo foi de 14,33%, que aumenta para 36,99% ao aceitarmos um intervalo de erro. Este achado realça a importância de considerar tolerâncias de erro na avaliação da precisão, especialmente em previsões em contextos complexos como a modulação 64-QAM. A avaliação dos resultados para a modulação 64-QAM com o classificador *Random Forest*, como mostrado na Tabela 4.30, indica que um aumento no número de estimadores não leva necessariamente a melhorias na acurácia do modelo. Isso sugere que a eficácia do modelo pode

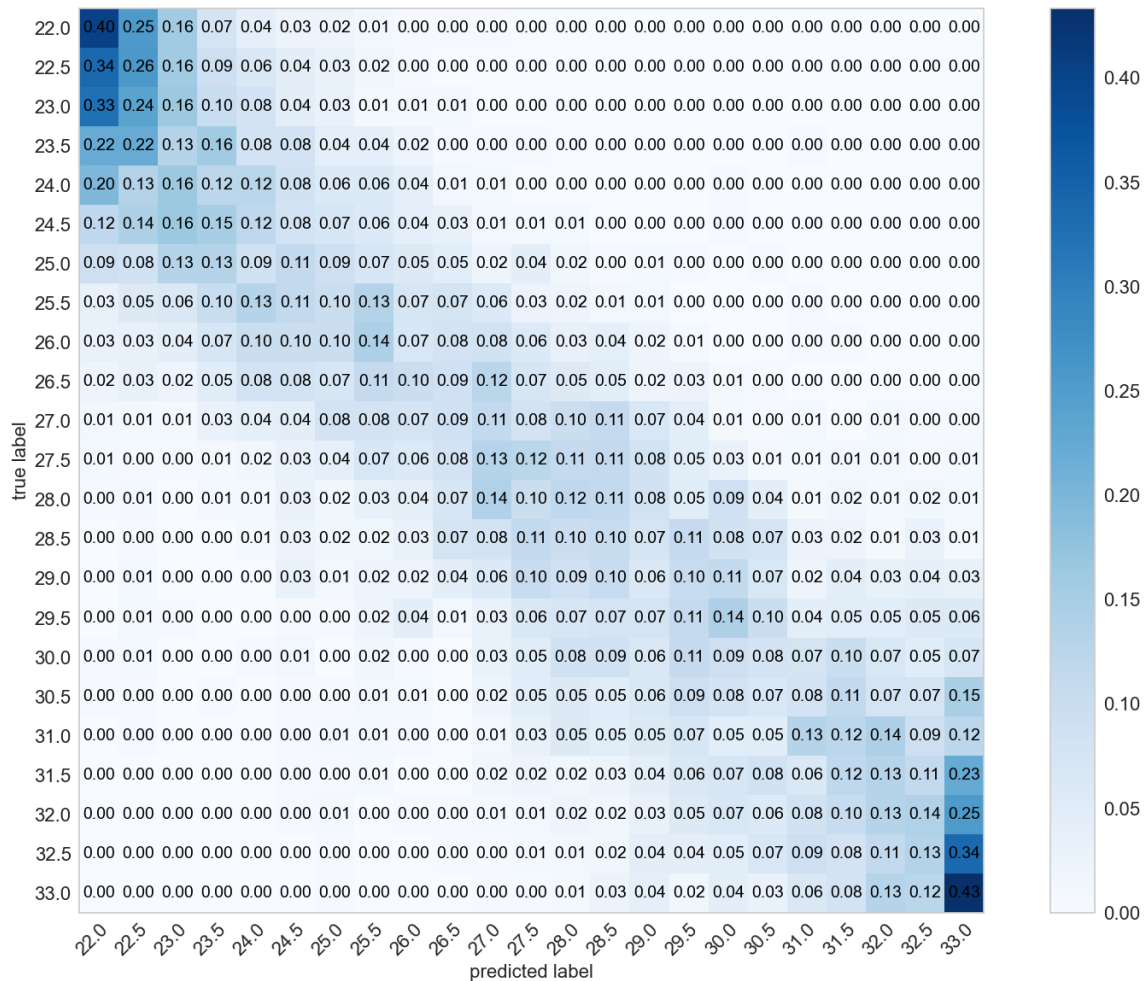


Figura 4.18: Resultados com melhor desempenho para o preditor de OSNR na modulação 64-QAM usando *Random Forest*, a partir dos parâmetros encontrados na Tabela 4.29.

Parâmetros	
Nome	Valor
Número de estimadores	600
Critério de avaliação da qualidade da divisão	<i>entropy</i>
Profundidade máxima da árvore	∞
Núm. min. de amostras para realizar uma divisão	2
Núm. máx. de características para realizar uma divisão	$\sqrt{F} \approx 14$
Quantidade máxima de folhas	∞
Valor mín. de impureza para dividir um ramo	Não
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	200

Tabela 4.29: Parâmetros ótimos utilizados no treino do preditor de OSNR para modulação 64-QAM com *Random Forest*

atingir um ponto de saturação mesmo com o aumento dos recursos computacionais. Além disso, a diferença entre a acurácia simples e a acurácia composta ressalta a

importância de considerar uma margem de erro aceitável na predição da OSNR , ampliando a aplicabilidade prática do modelo em cenários reais.

Resultados consolidados para <i>Random Forest</i> em 64-QAM				
Nome	Valores			
Intervalos do histograma	200	400	400	200
Número de estimadores	200	200	600	600
MAE	2,7846	2,7694	2,6353	2,6605
MSE	13,5371	13,9212	12,3498	12,4225
RMSE	3,6793	3,7311	3,5142	3,5246
Acurácia	13,64%	13,75%	13,97%	14,33%
Acurácia composta	36,32%	36,67%	37,73%	36,99%

Tabela 4.30: Resultados obtidos a partir dos experimentos realizados com o classificador *Random Forest* para modulação 64-QAM.

4.2.2.7 Predição da OSNR para modulação BPSK usando *Perceptron Multicamada* como classificador

Parâmetros	
Nome	Valor
Camadas ocultas	600
Função de ativação	<i>relu</i>
Método de otimização	<i>adam</i>
Tamanho do <i>batch</i>	auto
Taxa de aprendizado	adaptativa
Taxa inicial de aprendizado	0,001
Expoente de decaimento da taxa de aprendizado	0,5
Número máximo de iterações	700
<i>Momentum</i>	0,9
<i>Momentum</i> de Nesterov	Sim
Beta 1	0,9
Beta 2	0,999
Epsilon	$1e^{-08}$
Número máximo de funções de perda	15.000
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	400

Tabela 4.31: Parâmetros utilizados no treino do preditor de OSNR para modulação BPSK, do algoritmo *Perceptron Multicamada*

Nesta seção, exploramos o desempenho do modelo *Perceptron Multicamada* na tarefa de prever a OSNR para a modulação BPSK. Diferenciando-se dos resultados do *Random Forest*, este modelo apresentou uma acurácia inicial de 58,97%, que, com a aplicação de uma margem de erro de $\pm 0,5dB/0,1nm$, elevou-se para 95,55%.

Esse aumento substancial na acurácia ajustada destaca a capacidade do *Perceptron Multicamada* de fornecer previsões precisas, mesmo diante de variações na OSNR, conforme detalhado na Figura 4.19 que ilustra a matriz de confusão obtida a partir dos parâmetros encontrados na Tabela 4.31.



Figura 4.19: Resultados com melhor desempenho para o preditor de OSNR na modulação BPSK usando *Perceptron Multicamada* com os parâmetros encontrados na Tabela 4.31.

Ao examinar os resultados consolidados na Tabela 4.32 para a modulação BPSK usando o *Perceptron Multicamada*, notamos que o ajuste de parâmetros, especialmente o número de épocas, tem um impacto significativo na precisão do modelo. O aumento no número de épocas levou a uma melhora na acurácia, destacando a importância de um treinamento mais extenso para alcançar resultados ótimos. Em contraste, variações no número de camadas ocultas não mostrou uma tendência clara de melhoria, sugerindo que outros fatores do modelo podem ter um papel mais crítico na otimização do desempenho do classificador.

atingindo uma precisão inicial de 70,93%. Com a inclusão de uma tolerância de $\pm 0,5dB/0,1nm$ para o erro, essa acurácia foi significativamente elevada para 98,63%. Esse resultado enfatiza a eficiência do *Perceptron Multicamada* em ajustar-se às variações de OSNR, detalhado na matriz de confusão ilustrada pela Figura 4.20, obtida a partir dos parâmetros encontrados na Tabela 4.33, reforçando sua aplicabilidade em contextos de modulação complexa.

Parâmetros	
Nome	Valor
Camadas ocultas	600
Função de ativação	<i>relu</i>
Método de otimização	<i>adam</i>
Tamanho do <i>batch</i>	auto
Taxa de aprendizado	adaptativa
Taxa inicial de aprendizado	0,001
Expoente de decaimento da taxa de aprendizado	0,5
Número máximo de iterações	700
<i>Momentum</i>	0,9
<i>Momentum</i> de Nesterov	Sim
Beta 1	0.9
Beta 2	0.999
Epsilon	$1e^{-08}$
Número máximo de funções de perda	15.000
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	400

Tabela 4.33: Parâmetros utilizados no treino do preditor de OSNR para modulação QPSK, do algoritmo *Perceptron Multicamada*

Resultados consolidados para <i>Perceptron Multicamada</i> em QPSK						
Nome	Valores					
Intervalos do histograma	200	200	200	200	400	400
Camadas ocultas	200	200	600	600	600	600
Épocas	100	700	100	700	100	700
MAE	0,6833	0,5532	0,4880	0,3877	0,3613	0,3047
MSE	0,9417	0,7066	0,5566	0,4387	0,4162	0,3331
RMSE	0,9704	0,8406	0,7461	0,6624	0,6451	0,5771
Acurácia	43,48%	51,57%	54,51%	63,68%	66,25%	70,93%
Acurácia composta	89,19%	93,85%	96,81%	97,65%	97,89%	98,63%

Tabela 4.34: Resultados obtidos a partir dos experimentos com o classificador *Perceptron Multicamada* na modulação QPSK.

A análise dos resultados na Tabela 4.34 revela que o ajuste fino dos parâmetros, como número de épocas, tem um papel crucial na melhoria da acurácia do modelo *Perceptron Multicamada* para a modulação QPSK. Um aumento nas épocas

resulta em melhor desempenho, evidenciando a importância de um treinamento adequado. Por outro lado, alterações no número de camadas ocultas não mostram uma melhoria linear na acurácia, sugerindo que o foco em outros aspectos do treinamento pode ser mais benéfico para otimizar o desempenho do classificador.

4.2.2.9 Predição da OSNR para modulação 8-PSK usando *Perceptron Multicamada* como classificador

Na análise da modulação 8-PSK com o *Perceptron Multicamada*, os resultados destacam a robustez do modelo na predição de OSNR, alcançando uma acurácia direta de 46,54%. Este desempenho salta para 90,48% ao considerar uma margem de erro de $\pm 0,5dB/0,1nm$. Essa melhoria substancial sublinha a capacidade do modelo de adaptar-se e prever com precisão em cenários de modulação complexa, conforme detalhado na Figura 4.21 com parâmetros usados a partir da Tabela 4.35, evidenciando sua aplicabilidade prática.

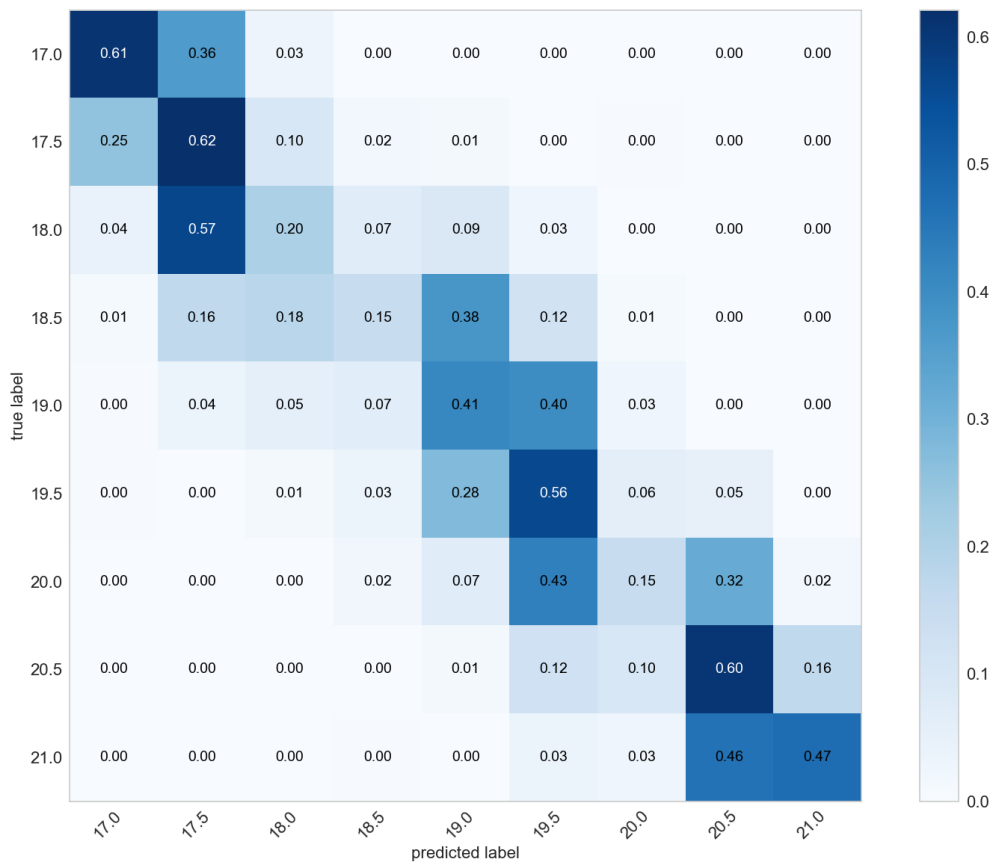


Figura 4.21: Resultados com melhor desempenho para o preditor de OSNR na modulação 8-PSK usando *Perceptron Multicamada* com os parâmetros encontrados na Tabela 4.35.

Parâmetros	
Nome	Valor
Camadas ocultas	600
Função de ativação	<i>relu</i>
Método de otimização	<i>adam</i>
Tamanho do <i>batch</i>	auto
Taxa de aprendizado	adaptativa
Taxa inicial de aprendizado	0,001
Expoente de decaimento da taxa de aprendizado	0,5
Número máximo de iterações	700
<i>Momentum</i>	0,9
<i>Momentum</i> de Nesterov	Sim
Beta 1	0,9
Beta 2	0,999
Epsilon	$1e^{-08}$
Número máximo de funções de perda	15.000
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	400

Tabela 4.35: Parâmetros utilizados no treino do preditor de OSNR para modulação 8-PSK, do algoritmo *Perceptron Multicamada*

A análise detalhada dos resultados na Tabela 4.36 revela que, apesar de o número de camadas ocultas ser um parâmetro ajustável, seu aumento não necessariamente traduz em melhorias significativas de desempenho para o classificador *Perceptron Multicamada* na modulação 8-PSK. A variação das épocas, por outro lado, demonstra um impacto mais expressivo na acurácia, especialmente ao se considerar um intervalo maior de treinamento. Isso sublinha a importância de um equilíbrio entre os parâmetros para alcançar um desempenho ótimo.

Resultados consolidados para <i>Perceptron Multicamada</i> em 8-PSK						
Nome	Valores					
Intervalos do histograma	200	200	400	200	400	400
Camadas ocultas	600	600	200	200	600	600
Épocas	100	700	100	100	100	700
MAE	0,8514	0,8282	0,7727	0,4079	0,7366	0,6474
MSE	1,3579	1,3319	1,1884	1,2555	1,0866	0,9219
RMSE	1,1653	1,1541	1,0901	1,1070	1,0424	0,9602
Acurácia	36,30%	37,96%	40,05%	40,79%	41,62%	46,54%
Acurácia composta	81,94%	83,01%	85,69%	84,95%	86,90%	90,48%

Tabela 4.36: Resultados obtidos a partir dos experimentos com o classificador *Perceptron Multicamada* na modulação 8-PSK.

4.2.2.10 Predição da OSNR para modulação 16-QAM usando *Perceptron Multicamada* como classificador

Para a modulação 16-QAM utilizando o *Perceptron Multicamada*, os resultados evidenciam um desempenho notável do modelo na predição de OSNR, atingindo uma acurácia base de 46,54%. Este valor aumenta expressivamente para 90,48% quando se considera uma tolerância de erro de $\pm 0,5dB/0,1nm$. Tal desempenho destaca a capacidade adaptativa e a precisão do modelo em contextos desafiadores de modulação, conforme detalhado na Figura 4.22 com parâmetros usados a partir da Tabela 4.37, reforçando sua eficácia e potencial aplicabilidade em cenários reais.

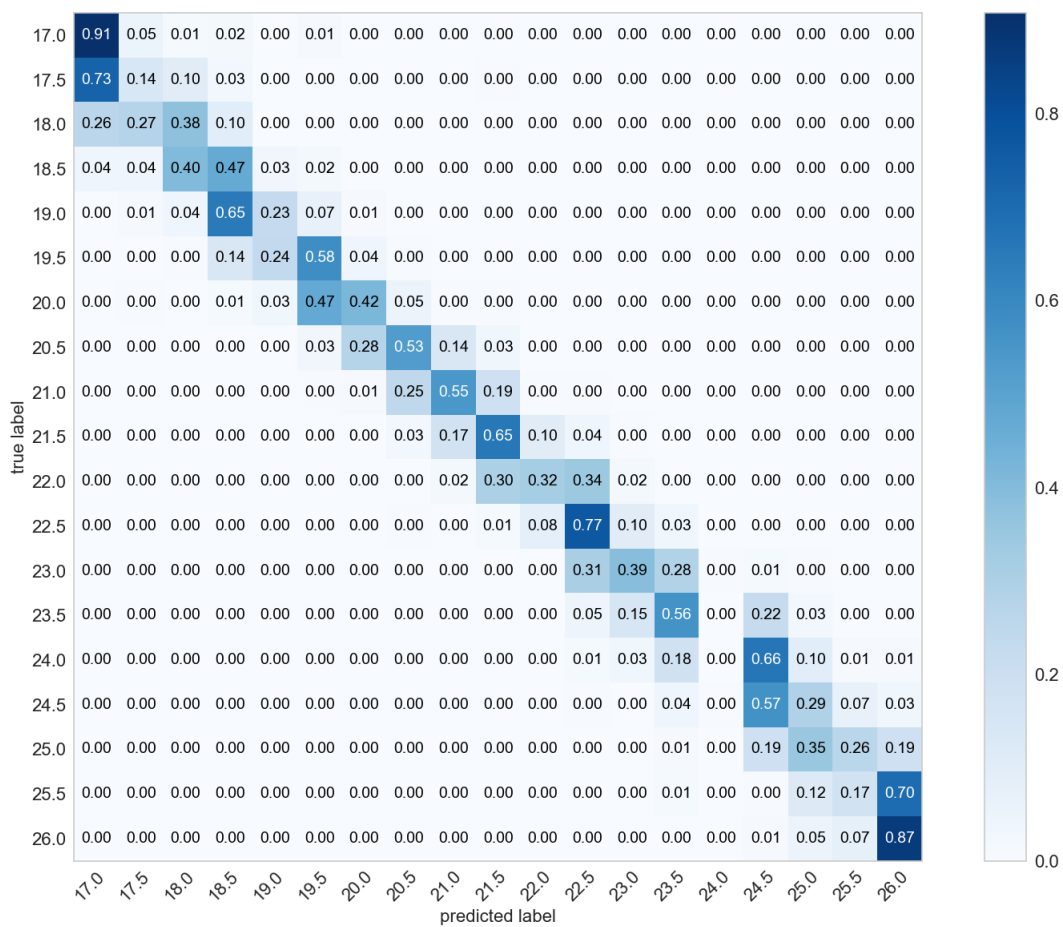


Figura 4.22: Resultados com melhor desempenho para o preditor de OSNR na modulação 16-QAM usando *Perceptron Multicamada* com os parâmetros encontrados na Tabela 4.37.

Por fim, temos a Tabela 4.38 que exhibe todos os resultados conforme a configuração utilizada. Observamos que para este classificador, a influência do número de camadas ocultas apresentou resultados interessantes, demonstrando que uma quantidade maior de camadas ocultas não representa melhoria nos resultados.

Parâmetros	
Nome	Valor
Camadas ocultas	600
Função de ativação	<i>relu</i>
Método de otimização	<i>adam</i>
Tamanho do <i>batch</i>	auto
Taxa de aprendizado	adaptativa
Taxa inicial de aprendizado	0,001
Expoente de decaimento da taxa de aprendizado	0,5
Número máximo de iterações	700
<i>Momentum</i>	0,9
<i>Momentum</i> de Nesterov	Sim
Beta 1	0,9
Beta 2	0,999
Epsilon	$1e^{-08}$
Número máximo de funções de perda	15.000
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	400

Tabela 4.37: Parâmetros utilizados no treino do preditor de OSNR para modulação 16-QAM, do algoritmo *Perceptron Multicamada*

Resultados consolidados para <i>Perceptron Multicamada</i> em 16-QAM						
Nome	Valores					
Intervalos do histograma	200	200	200	400	200	400
Camadas ocultas	200	200	600	600	600	600
Épocas	100	700	100	100	700	700
MAE	1,3632	1,2816	1,1390	1,0175	0,7053	0,6474
MSE	3,4333	3,0127	2,3991	2,0588	1,0439	0,9219
RMSE	1,8529	1,7357	1,5489	1,4348	1,0217	0,9602
Acurácia	24,80%	24,85%	28,73%	33,42%	43,31%	46,54%
Acurácia composta	63,68%	66,36%	72,23%	76,18%	88,42%	90,48%

Tabela 4.38: Resultados obtidos a partir dos experimentos com o classificador *Perceptron Multicamada* na modulação 16-QAM.

4.2.2.11 Predição da OSNR para modulação 32-QAM usando *Perceptron Multicamada* como classificador

Esta seção apresenta os resultados alcançados com o uso do algoritmo *Perceptron Multicamada* para a classificação da modulação 32-QAM e a previsão do valor de OSNR. A Figura 4.23 com parâmetros usados a partir da Tabela 4.40, mostra o desempenho ótimo do modelo após o treinamento, com detalhes fornecidos na tabela associada. Destaca-se que a precisão do modelo atingiu 25,83% e 64,39%, indicando capacidade significativa de previsão ao considerar uma margem de erro, demonstrando a eficácia do *Perceptron Multicamada* na previsão de OSNR para a

modulação 32-QAM.

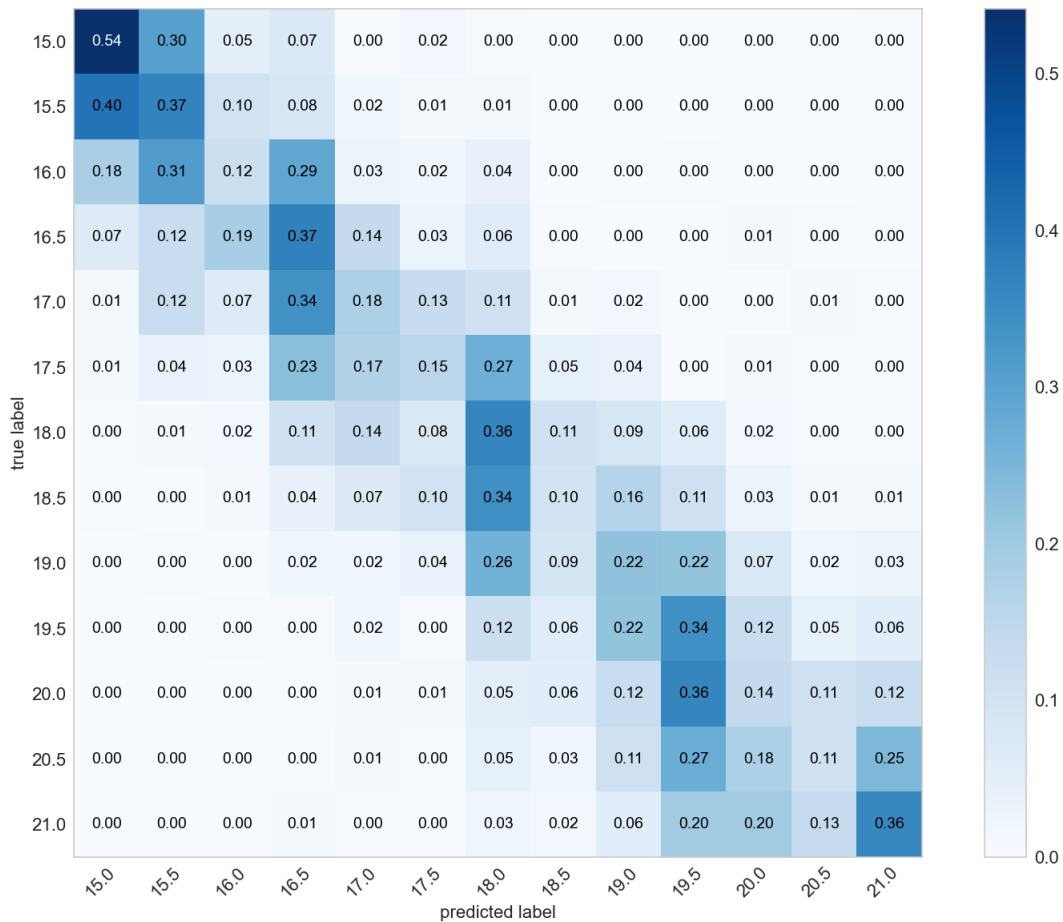


Figura 4.23: Resultados com melhor desempenho para o preditor de OSNR na modulação 32-QAM usando *Perceptron Multicamada* com os parâmetros encontrados na Tabela 4.40.

Resultados consolidados para <i>Perceptron Multicamada</i> em 32-QAM						
Nome	Valores					
Intervalos do histograma	400	200	200	400	200	400
Camadas ocultas	200	200	600	600	600	600
Épocas	100	100	100	100	700	700
MAE	2,7817	2,4000	1,9686	2,2311	1,4808	1,3638
MSE	12,8247	9,8333	6,9096	9,0407	3,9397	3,5465
RMSE	3,5812	3,1358	2,6286	3,0068	1,9849	1,8832
Acurácia	12,05%	13,59%	17,12%	17,15%	24,20%	25,83%
Acurácia composta	34,56%	41,47%	49,77%	45,54%	59,13%	64,39%

Tabela 4.39: Resultados obtidos a partir dos experimentos com o classificador *Perceptron Multicamada* na modulação 32-QAM.

A análise da Tabela 4.39 revela que o aumento no número de camadas ocultas não leva necessariamente a uma melhoria significativa no desempenho do classificador *Perceptron Multicamada* para a modulação 32-QAM. Este resultado sugere que

Parâmetros	
Nome	Valor
Camadas ocultas	600
Função de ativação	<i>relu</i>
Método de otimização	<i>adam</i>
Tamanho do <i>batch</i>	auto
Taxa de aprendizado	adaptativa
Taxa inicial de aprendizado	0,001
Expoente de decaimento da taxa de aprendizado	0,5
Número máximo de iterações	700
<i>Momentum</i>	0,9
<i>Momentum</i> de Nesterov	Sim
Beta 1	0,9
Beta 2	0,999
Epsilon	$1e^{-08}$
Número máximo de funções de perda	15.000
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	400

Tabela 4.40: Parâmetros utilizados no treino do preditor de OSNR para modulação 32-QAM, do algoritmo *Perceptron Multicamada*.

outros fatores, possivelmente relacionados à estrutura do modelo ou à seleção de características, podem ter um impacto mais substancial na acurácia da classificação. Assim, a escolha cuidadosa dos parâmetros e uma análise detalhada de sua influência são essenciais para otimizar o desempenho do modelo.

4.2.2.12 Predição da OSNR para modulação 64-QAM usando *Perceptron Multicamada* como classificador

Resultados consolidados para <i>Perceptron Multicamada</i> em 64-QAM				
Nome	Valores			
Intervalos do histograma	200	200	200	200
Camadas ocultas	200	600	200	600
Épocas	100	100	700	700
MAE	5,8795	3,6616	2,6014	2,4442
MSE	53,2078	23,3620	12,2500	10,7094
RMSE	7,2944	4,8334	3,5000	3,2725
Acurácia	5,72%	11,30%	13,97%	15,31%
Acurácia composta	16,11%	28,86%	3,50%	41,39%

Tabela 4.41: Resultados obtidos a partir dos experimentos com o classificador *Perceptron Multicamada* na modulação 64-QAM.

Nesta seção, os resultados alcançados com o algoritmo *Perceptron Multicamada* para a previsão do valor de OSNR na modulação 64-QAM são discutidos. A

Figura 4.24 com parâmetros usados a partir da Tabela 4.42, destaca o desempenho mais eficaz do modelo, baseado nos parâmetros detalhados na tabela mencionada. A precisão do modelo foi de 15,31% e, ao considerar uma margem de erro, a precisão ajustada aumentou para 41,39%, ressaltando a aplicabilidade do *Perceptron Multicamada* em prever OSNR para a modulação 64-QAM de maneira eficaz.

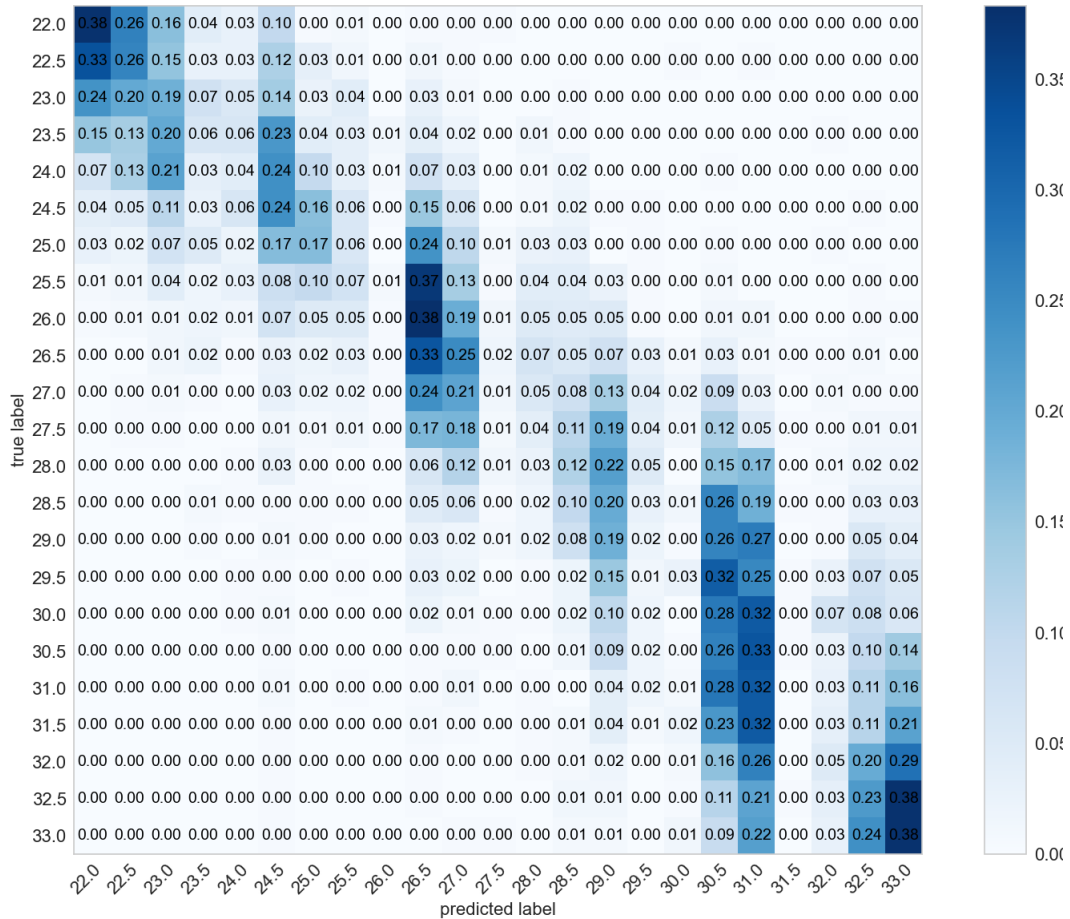


Figura 4.24: Resultados com melhor desempenho para o preditor de OSNR na modulação 64-QAM usando *Perceptron Multicamada* com os parâmetros encontrados na Tabela 4.42.

Por fim, temos a análise da Tabela 4.41 revela um padrão consistente observado nas modulações anteriores: aumentar o número de camadas ocultas no classificador *Perceptron Multicamada* para a modulação 64-QAM não se traduz em melhorias de desempenho significativas. Este resultado reforça a importância de explorar outras variáveis e técnicas de otimização para aprimorar a eficácia do modelo, especialmente em cenários de alta complexidade de modulação onde os desafios são amplificados.

Parâmetros	
Nome	Valor
Camadas ocultas	600
Função de ativação	<i>relu</i>
Método de otimização	<i>adam</i>
Tamanho do <i>batch</i>	auto
Taxa de aprendizado	adaptativa
Taxa inicial de aprendizado	0,001
Expoente de decaimento da taxa de aprendizado	0,5
Número máximo de iterações	700
<i>Momentum</i>	0,9
<i>Momentum</i> de Nesterov	Sim
Beta 1	0,9
Beta 2	0,999
Epsilon	$1e^{-08}$
Número máximo de funções de perda	15.000
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	200

Tabela 4.42: Parâmetros utilizados no treino do preditor de OSNR para modulação 64-QAM, do algoritmo *Perceptron Multicamada*

4.2.2.13 Predição da OSNR para modulação BPSK usando o algoritmo *Random Forest* como regressor

A seguir, apresentam-se os resultados obtidos por meio da aplicação do algoritmo *Random Forest* como regressor para a predição do valor de OSNR na modulação BPSK. A Figura 4.25 ilustra o melhor resultado alcançado após a conclusão do treinamento do modelo, utilizando os parâmetros indicados na Tabela 4.43.

Parâmetros	
Nome	Valor
Número de estimadores	600
Critério de avaliação da qualidade da divisão	Erro quadrático
Profundidade máxima da árvore	∞
Núm. min. de amostras para realizar uma divisão	2
Núm. máx. de características para realizar uma divisão	Sim
Quantidade máxima de folhas	∞
Valor mín. de impureza para dividir um ramo	Não
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	400

Tabela 4.43: Parâmetros utilizados no treino do preditor de OSNR com melhor desempenho em BPSK, com *Random Forest* como regressor.

Para uma análise detalhada dos regressores, a Tabela 4.44 sumariza os resul-

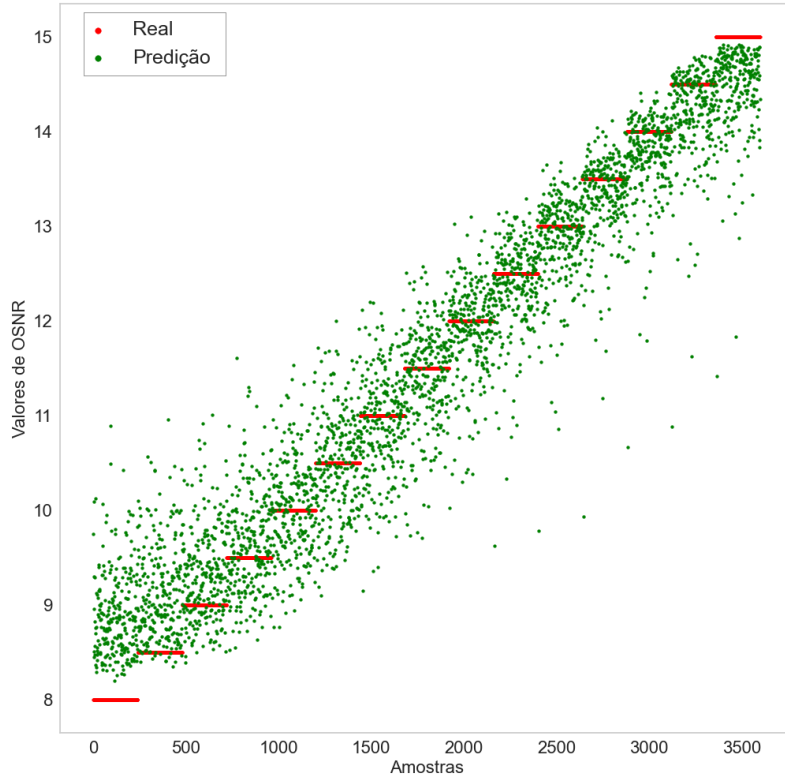


Figura 4.25: Resultados com melhor desempenho para o preditor de OSNR na modulação BPSK usando *Random Forest* como regressor, com os parâmetros encontrados na Tabela 4.43

Resultados consolidados para <i>Random Forest</i> em BPSK				
Nome	Valores			
Intervalos do histograma	200	400	200	400
Número de estimadores	200	200	400	600
MAE	0,3313	0,2947	0,3596	0,4498
MSE	0,1735	0,1505	0,2451	0,3801
RMSE	0,4165	0,3879	0,4951	0,6165
Coef. de Determinação R^2	0,8790	0,9580	0,9660	0,9720
Coef. de Determinação ajustado R^2	0,8670	0,9460	0,9640	0,9680
F -statistic	71,52	416,60	489,90	274,60

Tabela 4.44: Resultados obtidos a partir dos experimentos realizados com *Random Forest* como regressor para modulação BPSK.

tados conforme diferentes configurações. Interessantemente, a variação no número de estimadores mostra que incrementos nesta métrica não necessariamente traduzem-se em melhorias de desempenho, indicando um ponto de otimização específico para o modelo em questão. A análise do MAE, MSE, e RMSE, juntamente com os coeficientes de determinação R^2 e ajustado R^2 , fornecem uma visão completa da eficácia do modelo, destacando a precisão na predição dos valores de OSNR para a modulação BPSK. Estes resultados reforçam a importância de uma escolha criteriosa na

configuração dos parâmetros para alcançar a máxima eficiência do modelo.

4.2.2.14 Predição da OSNR para modulação QPSK usando o algoritmo *Random Forest* como regressor

A seguir, apresentam-se os resultados obtidos por meio da aplicação do algoritmo *Random Forest* como regressor para a predição do valor de OSNR na modulação QPSK. A Figura 4.26 ilustra o melhor resultado alcançado após a conclusão do treinamento do modelo, utilizando os parâmetros indicados na Tabela 4.45.

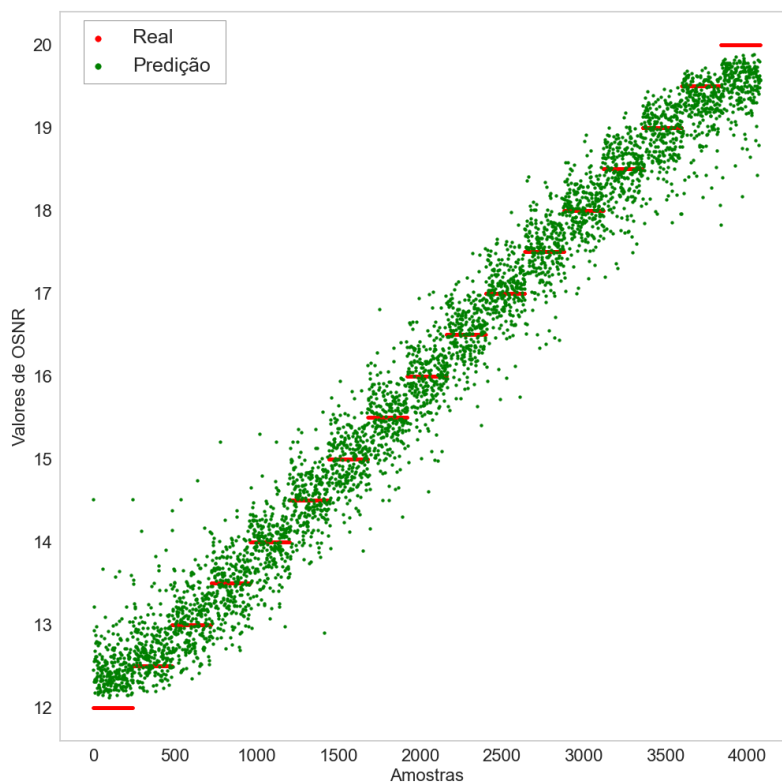


Figura 4.26: Resultados com melhor desempenho para o preditor de OSNR na modulação QPSK usando *Random Forest* como regressor, com os parâmetros encontrados na Tabela 4.45

A análise dos resultados para o regressor *Random Forest* aplicado à modulação QPSK revela que aumentos no número de estimadores não melhoram necessariamente o desempenho do modelo. Esta observação é corroborada pela análise dos dados consolidados na Tabela 4.46, que mostra o desempenho do modelo sob diferentes configurações. Métricas como MAE, MSE, RMSE, e os coeficientes de determinação R^2 e ajustado R^2 oferecem uma visão completa sobre a eficácia do modelo, sublinhando a precisão nas predições dos valores de OSNR para a modula-

Parâmetros	
Nome	Valor
Número de estimadores	200
Critério de avaliação da qualidade da divisão	Erro quadrático
Profundidade máxima da árvore	∞
Núm. min. de amostras para realizar uma divisão	2
Núm. máx. de características para realizar uma divisão	Sim
Quantidade máxima de folhas	∞
Valor mín. de impureza para dividir um ramo	Não
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	400

Tabela 4.45: Parâmetros utilizados no treino do preditor de OSNR com melhor desempenho em QPSK, com *Random Forest* como regressor.

ção QPSK.

Resultados consolidados para <i>Random Forest</i> em QPSK				
Nome	Valores			
Intervalos do histograma	200	200	400	400
Número de estimadores	200	400	600	200
MAE	0,3313	0,2988	0,2947	0,2902
MSE	0,1735	0,1566	0,1405	0,1473
RMSE	0,4165	0,3957	0,3879	0,3839
Coef. de Determinação R^2	0,8790	0,9730	0,9780	0,9780
Coef. de Determinação ajustado R^2	0,8670	0,9720	0,9760	0,9760
<i>F-statistic</i>	71,52	714,30	416,60	416,60

Tabela 4.46: Resultados obtidos a partir dos experimentos realizados com *Random Forest* como regressor para modulação QPSK.

4.2.2.15 Predição da OSNR para modulação 8-PSK usando o algoritmo *Random Forest* como regressor

A seguir, apresentam-se os resultados obtidos por meio da aplicação do algoritmo *Random Forest* como regressor para a predição do valor de OSNR para a modulação 8-PSK. A Figura 4.27 ilustra o melhor resultado alcançado após a conclusão do treinamento do modelo, utilizando os parâmetros indicados na Tabela 4.47.

A análise do desempenho do regressor *Random Forest* para modulação 8-PSK, conforme apresentada na Tabela 4.48, evidencia que variações no número de estimadores não implicam necessariamente em melhorias significativas de desempenho. Este fenômeno sugere um ponto de saturação na capacidade do modelo de

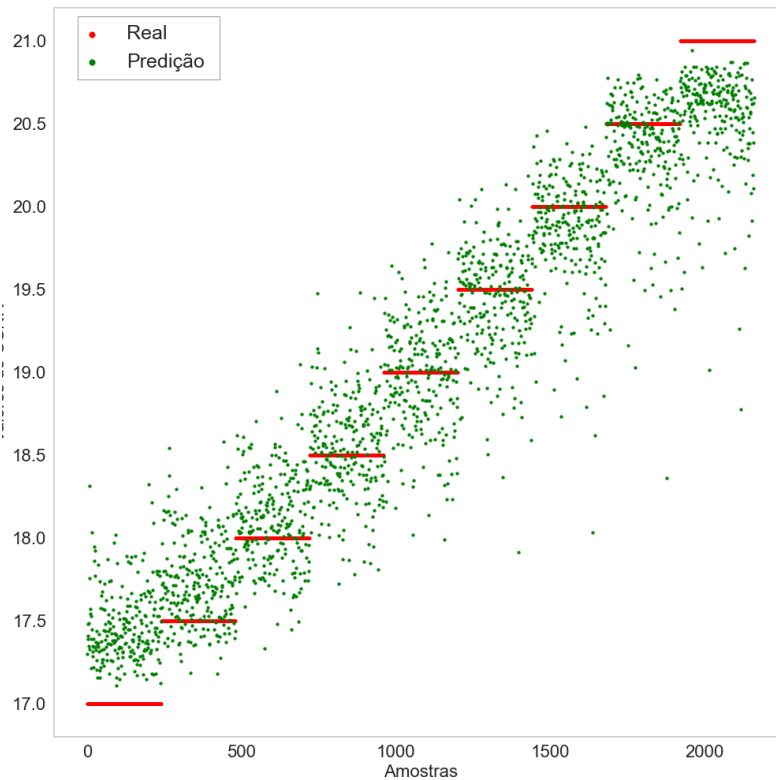


Figura 4.27: Resultados com melhor desempenho para o preditor de OSNR na modulação 8-PSK usando *Random Forest* como regressor, com os parâmetros encontrados na Tabela 4.47

Parâmetros	
Nome	Valor
Número de estimadores	200
Critério de avaliação da qualidade da divisão	Erro quadrático
Profundidade máxima da árvore	∞
Núm. mín. de amostras para realizar uma divisão	2
Núm. máx. de características para realizar uma divisão	Sim
Quantidade máxima de folhas	∞
Valor mín. de impureza para dividir um ramo	Não
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	400

Tabela 4.47: Parâmetros utilizados no treino do preditor de OSNR com melhor desempenho em 8-PSK, com *Random Forest* como regressor.

capturar nuances adicionais com o aumento dos estimadores. A consistência dos coeficientes de determinação R^2 e ajustado R^2 , juntamente com as métricas MAE, MSE e RMSE, reforça a noção de uma relação complexa entre a quantidade de estimadores e a precisão do modelo.

Resultados consolidados para <i>Random Forest</i> em 8-PSK				
Nome	Valores			
Intervalos do histograma	200	200	400	400
Número de estimadores	200	400	600	200
MAE	0,3313	0,3274	0,2978	0,2972
MSE	0,1735	0,1695	0,1523	0,1525
RMSE	0,4165	0,4117	0,3903	0,3905
Coef. de Determinação R^2	0,8790	0,8790	0,9160	0,9160
Coef. de Determinação ajustado R^2	0,8670	0,8670	0,8960	0,8960
F -statistic	71,52	71,52	47,82	47,82

Tabela 4.48: Resultados obtidos a partir dos experimentos realizados com *Random Forest* como regressor para modulação 8-PSK.

4.2.2.16 Predição da OSNR para modulação 16-QAM usando o algoritmo *Random Forest* como regressor

A seguir, apresentam-se os resultados obtidos por meio da aplicação do algoritmo *Random Forest* como regressor para a predição do valor de OSNR na modulação 16-QAM. A Figura 4.28 ilustra o melhor resultado alcançado após a conclusão do treinamento do modelo, utilizando os parâmetros indicados na Tabela 4.49.

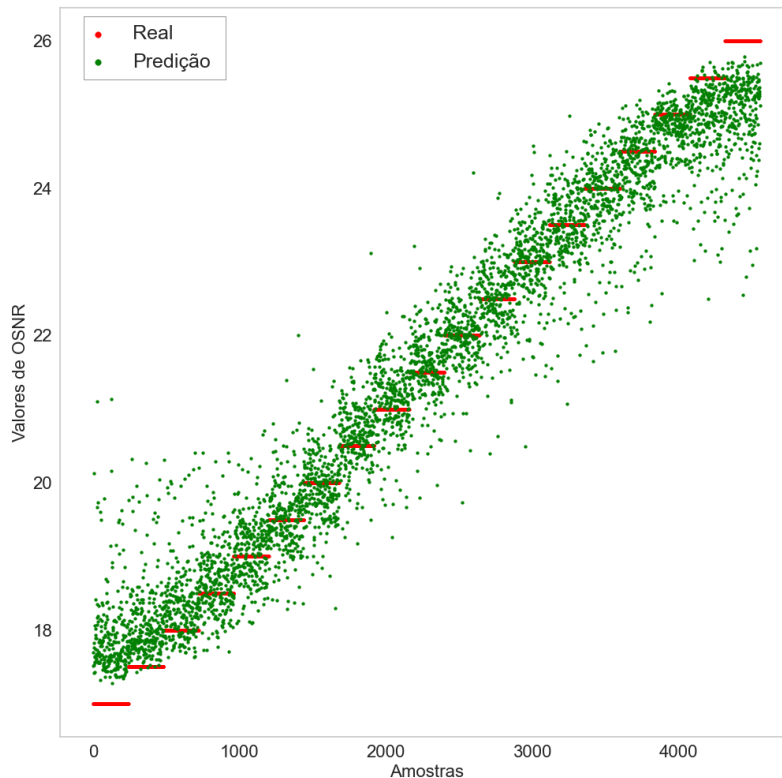


Figura 4.28: Resultados com melhor desempenho para o preditor de OSNR na modulação 16-QAM usando *Random Forest* como regressor, com os parâmetros encontrados na Tabela 4.49

Parâmetros	
Nome	Valor
Número de estimadores	600
Critério de avaliação da qualidade da divisão	Erro quadrático
Profundidade máxima da árvore	∞
Núm. min. de amostras para realizar uma divisão	2
Núm. máx. de características para realizar uma divisão	Sim
Quantidade máxima de folhas	∞
Valor mín. de impureza para dividir um ramo	Não
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	400

Tabela 4.49: Parâmetros utilizados no treino do preditor de OSNR com melhor desempenho em 16-QAM, com *Random Forest* como regressor.

A análise dos resultados para o regressor *Random Forest* aplicado à modulação 16-QAM, conforme demonstrado na Tabela 4.50, revela que um aumento no número de estimadores não conduz necessariamente a uma melhoria na precisão das previsões. Este comportamento é evidenciado pela estabilidade nas métricas MAE, MSE, e RMSE, e pelos coeficientes de determinação R^2 e ajustado R^2 , que sugerem uma eficácia limitada na melhoria dos resultados com o incremento de estimadores. Este padrão sugere uma reflexão sobre o equilíbrio entre complexidade do modelo e ganhos marginais em desempenho.

Resultados consolidados para <i>Random Forest</i> em 16-QAM				
Nome	Valores			
Intervalos do histograma	200	400	200	400
Número de estimadores	200	200	400	600
MAE	0,8313	0,7745	0,4960	0,4580
MSE	0,8735	0,9505	0,4423	0,4112
RMSE	0,7165	0,8879	0,6650	0,6413
Coef. de Determinação R^2	0,7790	0,7780	0,8240	0,8760
Coef. de Determinação ajustado R^2	0,7670	0,7760	0,8160	0,8640
<i>F-statistic</i>	71,52	116,60	102,50	73,80

Tabela 4.50: Resultados obtidos a partir dos experimentos realizados com *Random Forest* como regressor para modulação 16-QAM.

4.2.2.17 Predição da OSNR para modulação 32-QAM usando o algoritmo *Random Forest* como regressor

A seguir, apresentam-se os resultados obtidos por meio da aplicação do algoritmo *Random Forest* como regressor para a predição do valor de OSNR na modu-

lação 32-QAM. A Figura 4.29 ilustra o melhor resultado alcançado após a conclusão do treinamento do modelo, utilizando os parâmetros indicados na Tabela 4.51.

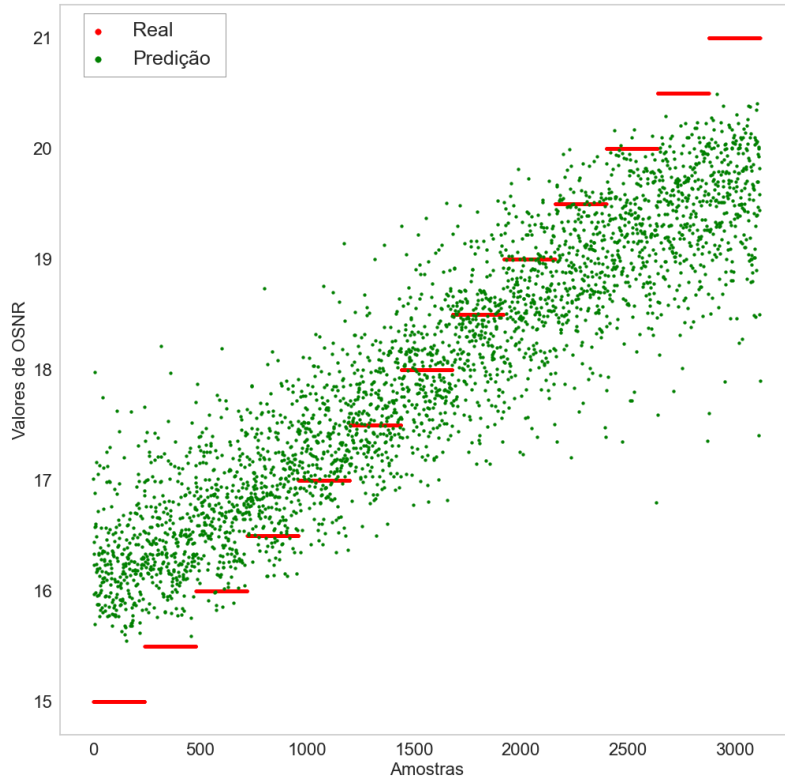


Figura 4.29: Resultados com melhor desempenho para o preditor de OSNR na modulação 32-QAM usando *Random Forest* como regressor, com os parâmetros encontrados na Tabela 4.51

Parâmetros	
Nome	Valor
Número de estimadores	600
Critério de avaliação da qualidade da divisão	Erro quadrático
Profundidade máxima da árvore	∞
Núm. min. de amostras para realizar uma divisão	2
Núm. máx. de características para realizar uma divisão	Sim
Quantidade máxima de folhas	∞
Valor mín. de impureza para dividir um ramo	Não
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	400

Tabela 4.51: Parâmetros utilizados no treino do preditor de OSNR com melhor desempenho em 32-QAM, com *Random Forest* como regressor.

A análise dos resultados para o regressor *Random Forest* na modulação 32-QAM revela uma observação interessante sobre o impacto do número de estimadores, conforme visto na Tabela 4.52. Apesar da expectativa de que mais estimadores

possam traduzir-se em melhor desempenho, os dados apresentados não corroboram essa suposição. As métricas de MAE, MSE, e RMSE melhoram ligeiramente com o aumento dos estimadores, mas os coeficientes de determinação R^2 e R^2 ajustado, juntamente com o F -*statistic*, sugerem que os ganhos são marginais. Essa tendência indica que outros fatores, além do número de estimadores, são cruciais para aprimorar a precisão das previsões nesse contexto.

Resultados consolidados para <i>Random Forest</i> em 32-QAM				
Nome	Valores			
Intervalos do histograma	200	400	200	400
Número de estimadores	400	200	400	600
MAE	0,9974	0,8550	0,7585	0,7380
MSE	1,1735	0,9954	0,8698	0,8387
RMSE	1,1145	1,0017	0,9326	0,9158
Coef. de Determinação R^2	0,3870	0,4030	0,4580	0,6150
Coef. de Determinação ajustado R^2	0,3720	0,3920	0,4210	0,5580
F - <i>statistic</i>	51,12	15,30	12,38	10,89

Tabela 4.52: Resultados obtidos a partir dos experimentos realizados com *Random Forest* como regressor para modulação 32-QAM.

4.2.2.18 Predição da OSNR para modulação 64-QAM usando o algoritmo *Random Forest* como regressor

A seguir, apresentam-se os resultados obtidos por meio da aplicação do algoritmo *Random Forest* como regressor para a predição do valor de OSNR na modulação 64-QAM. A Figura 4.30 ilustra o melhor resultado alcançado após a conclusão do treinamento do modelo, utilizando os parâmetros indicados na Tabela 4.53.

Parâmetros	
Nome	Valor
Número de estimadores	600
Critério de avaliação da qualidade da divisão	Erro quadrático
Profundidade máxima da árvore	∞
Núm. mín. de amostras para realizar uma divisão	2
Núm. máx. de características para realizar uma divisão	Sim
Quantidade máxima de folhas	∞
Valor mín. de impureza para dividir um ramo	Não
Método de aleatorização das amostras	<i>bootstrap</i>
Quantidade de amostras para o <i>bootstrap</i>	10%
Dicionário com os pesos de cada classe	[1 1 ... 1]
Amostras do sinal original	262.144
Intervalos do histograma (F)	400

Tabela 4.53: Parâmetros utilizados no treino do preditor de OSNR com melhor desempenho em 64-QAM, com *Random Forest* como regressor.

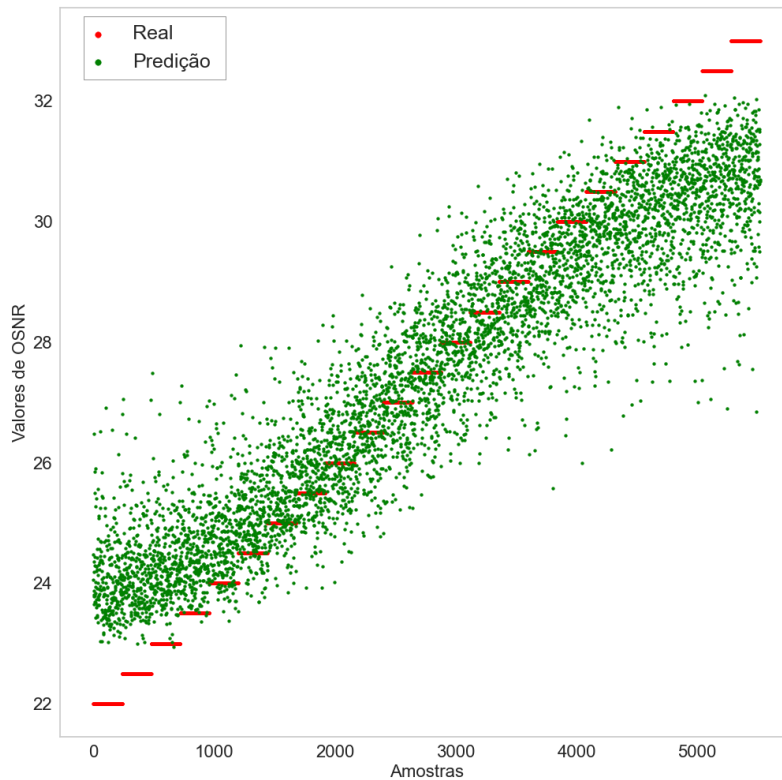


Figura 4.30: Resultados com melhor desempenho para o preditor de OSNR na modulação 64-QAM usando *Random Forest* como regressor, com os parâmetros encontrados na Tabela 4.53

Resultados consolidados para <i>Random Forest</i> em 64-QAM				
Nome	Valores			
Intervalos do histograma	200	400	200	400
Número de estimadores	200	200	400	600
MAE	1,3151	1,2778	1,1974	1,0594
MSE	2,1735	2,1242	2,2783	1,8494
RMSE	1,6805	1,6421	1,5094	1,3599
Coef. de Determinação R^2	0,3670	0,3720	0,4250	0,5540
Coef. de Determinação ajustado R^2	0,3590	0,3590	0,4030	0,5190
F -statistic	102,14	55,30	19,72	15,92

Tabela 4.54: Resultados obtidos a partir dos experimentos realizados com *Random Forest* como regressor para modulação 64-QAM.

Nesta seção, destacamos os resultados alcançados com o uso do regressor *Random Forest* em diversas configurações de modulação. A análise detalhada dos resultados, apresentada na Tabela 4.54, revela que o aumento no número de estimadores não leva necessariamente a uma melhora significativa nos resultados. Este padrão sugere que, além da quantidade de estimadores, outros fatores devem ser considerados para otimizar o desempenho do modelo. Os indicadores como MAE, MSE, e RMSE fornecem uma visão ampla da precisão do modelo, enquanto os coe-

coeficientes de determinação R^2 e ajustado R^2 , juntamente com o F -*statistic*, oferecem uma visão geral sobre a adequação do modelo aos dados.

4.3 Análise dos resultados

Nesta seção, discutiremos detalhadamente os resultados apresentados anteriormente, explorando suas implicações práticas e teóricas no contexto de receptores ópticos coerentes flexíveis. A análise é estruturada em torno de três perguntas principais que visam desvendar os aspectos cruciais dos dados obtidos, permitindo uma compreensão mais aprofundada da eficácia das estratégias adotadas para classificação de modulação e previsão de OSNR. Essa abordagem nos ajuda a contextualizar os resultados no panorama atual de pesquisa e prática na área, evidenciando sua relevância e aplicabilidade.

4.3.1 Pergunta 1: De que forma o desempenho individual de cada modelo pode ser avaliado?

Para avaliar o desempenho de cada modelo, é importante adotar critérios específicos conforme o algoritmo utilizado. Para os classificadores e redes neurais, a análise da matriz de confusão e da acurácia fornece uma visão clara da eficácia do modelo, complementada pela avaliação dos índices MAE (Erro Absoluto Médio), MSE (Erro Quadrático Médio), e RMSE (Raiz do Erro Quadrático Médio) para detalhar a precisão. Para os regressores, a análise do coeficiente de determinação (R^2 e R^2 ajustado), indica quão bem o modelo se ajusta aos dados observados. Esses coeficientes, em conjunto com métricas de erro como MAE, MSE, e RMSE, fornecem uma compreensão detalhada da precisão do modelo. Estas métricas nos ajudam a quantificar a diferença entre os valores previstos pelo modelo e os valores reais. Essa metodologia permite uma comparação justa entre modelos destinados à classificação da modulação e à previsão de OSNR, agrupando-os em classificadores, regressores e redes neurais para uma análise detalhada de desempenho.

4.3.2 Pergunta 2: Que critério pode ser utilizado para eleger o resultado de um modelo como o melhor na estratégia aplicada?

Para determinar o melhor resultado entre os modelos, considera-se não só a acurácia, mas também métricas como MAE, MSE, e RMSE, permitindo uma avaliação detalhada do desempenho em diferentes cenários. Esta abordagem assegura uma compreensão abrangente da eficácia dos modelos para a classificação da modulação e predição do valor de OSNR, destacando a importância de uma análise criteriosa além da acurácia para identificar o modelo mais adequado para cada estratégia aplicada.

Na estratégia de classificação da modulação, é possível fazer uma comparação direta entre modelos *ensemble* e redes neurais, considerando que ambos são avaliados sob os mesmos critérios de desempenho. Isso facilita uma análise objetiva da eficácia de cada modelo em classificar corretamente a modulação, permitindo identificar qual abordagem oferece os melhores resultados com base nas métricas estabelecidas.

Resultados consolidados para classificação do tipo de modulação					
Nome	<i>AdaBoost</i>	<i>CART</i>	<i>Random F.</i>	<i>Gradient B.</i>	<i>Perceptron M.</i>
MAE	0,0842	0,0842	0,0089	0,0036	0,0024
MSE	0,1894	0,1931	0,0160	0,0045	0,0051
RMSE	0,4352	0,4394	0,1264	0,0672	0,0713
Acurácia	95,06%	95,10%	99,31%	99,69%	99,85%

Tabela 4.55: Resumo dos resultados para a classificação do tipo de modulação.

A Tabela 4.55 destaca os resultados excepcionais alcançados na classificação do tipo de modulação, com acurácias notavelmente altas, acima de 95%. *Random Forest*, *Gradient Boosting* e *Perceptron Multicamada* são particularmente impressionantes, excedendo 99% de acurácia, demonstrando eficiência na identificação de modulações. Esses resultados facilitam a seleção do modelo mais adequado para a aplicação em questão, considerando tanto a precisão quanto a complexidade computacional.

Na estratégia de predição do valor de OSNR, é crucial diferenciar entre os modelos utilizados: o *Random Forest* como classificador e regressor, e o *Perceptron Multicamada* como rede neural. Essa diferenciação permite uma avaliação detalhada e específica dos resultados, conforme a modulação. A análise é dividida em duas etapas principais: avaliação individual dos modelos e comparação entre eles para

diferentes modulações.

Resultados consolidados para a predição do valor de OSNR com o classificador						
Nome	BPSK	QPSK	8-PSK	16-QAM	32-QAM	64-QAM
MAE	0,8328	0,6723	0,6968	1,0645	1,6192	2,6605
MSE	1,5494	1,0017	1,1125	2,3399	4,9724	12,4225
RMSE	1,2448	1,0009	1,0548	1,5297	2,2299	3,5246
Acurácia	40,81%	45,42%	46,11%	33,13%	22,31%	14,33%
Acurácia flexível	82,97%	89,80%	88,33%	74,61%	57,88%	36,99%

Tabela 4.56: Resumo dos resultados para a predição do valor de OSNR com o classificador.

Resultados consolidados para a predição do valor de OSNR com o regressor						
Nome	BPSK	QPSK	8-PSK	16-QAM	32-QAM	64-QAM
MAE	0,4498	0,2902	0,2972	0,4580	0,7380	1,0594
MSE	0,3801	0,1473	0,1525	0,4112	0,8387	1,8494
RMSE	0,6165	0,3839	0,3905	0,6413	0,9158	1,3599
R^2	0,9720	0,9780	0,9160	0,8760	0,6150	0,5540
R^2 ajustado	0,9720	0,9760	0,8960	0,8640	0,5580	0,5190
F -statistic	274,60	416,60	47,82	73,80	10,89	15,92

Tabela 4.57: Resumo dos resultados para a predição do valor de OSNR com o regressor.

Resultados consolidados para a predição do valor de OSNR com a rede neural						
Nome	BPSK	QPSK	8-PSK	16-QAM	32-QAM	64-QAM
MAE	0,4717	0,3047	0,6474	0,6474	1,3638	2,4442
MSE	0,6478	0,3331	0,9219	0,9219	3,5465	10,7094
RMSE	0,8048	0,5771	0,9602	0,9602	1,8832	3,2725
Acurácia	58,97%	70,93%	45,54%	46,54%	25,83%	15,31%
Acurácia flexível	95,55%	98,63%	90,48%	90,48%	64,39%	41,39%

Tabela 4.58: Resumo dos resultados para a predição do valor de OSNR com a rede neural.

Primeiramente, analisa-se o desempenho de cada modelo por tipo de modulação, permitindo identificar os melhores resultados em classificação, regressão e redes neurais para cada modulação. Esta etapa fornece as Tabelas 4.56, 4.57 e 4.58, oferecendo uma visão comparativa clara do desempenho dos modelos. O objetivo é discernir qual modelo é mais eficaz na predição do valor de OSNR para as modulações previstas.

Com os resultados dos modelos por tipo de modulação em mãos, procedemos à segunda fase de análise, onde é realizada a comparação entre os resultados dos modelos. Iniciando pela comparação entre os modelos compatíveis: classificador e rede neural. A análise concentra-se em destacar os desempenhos superiores por

modulação entre o classificador *Random Forest* (RF) e o *Perceptron Multicamada* (MLP). Essa comparação é resumida na Tabela 4.59, facilitando a visualização de qual dos modelos apresenta maior eficácia na predição do valor de OSNR para cada tipo de modulação analisada. A partir destes dados é possível observar que o *Perceptron Multicamada* dominou todas as melhores colocações, em suma devido às acurácias normal e flexíveis superiores e valores de erro (MAE, MSE e RMSE) menores. Mesmo em casos onde a acurácia do *Random Forest* é melhor, como na modulação 8-PSK, os demais parâmetros acabam contribuindo para a escolha do *Perceptron Multicamada*.

Comparação para a predição do valor de OSNR entre classificador e rede neural						
Nome	BPSK	QPSK	8-PSK	16-QAM	32-QAM	64-QAM
Modelo	MLP	MLP	MLP	MLP	MLP	MLP
MAE	0,4717	0,3047	0,6474	0,6474	1,3638	2,4442
MSE	0,6478	0,3331	0,9219	0,9219	3,5465	10,7094
RMSE	0,8048	0,5771	0,9602	0,9602	1,8832	3,2725
Acurácia	58,97%	70,93%	45,54%	46,54%	25,83%	15,31%
Acurácia flexível	95,55%	98,63%	90,48%	90,48%	64,39%	41,39%

Tabela 4.59: Melhores resultados entre o uso do classificador e rede neural para a predição do valor de OSNR.

Finalizaremos comparando os melhores resultados de classificadores e redes neurais, com aqueles obtidos pelos regressores. Para tornar essa comparação viável, transformaremos os dados dos regressores em categorias, possibilitando uma análise comparativa direta. Esse método simplifica a avaliação do desempenho de cada abordagem, destacando suas características únicas. Para transformar os valores contínuos preditos pelo regressor em categorias discretas, adotou-se uma estratégia de agrupamento baseada na proximidade dos valores a marcos definidos. Isso envolve arredondar os valores contínuos para a classe mais próxima, de maneira que valores intermediários, como 16.8dB e 17.2dB, fossem categorizados uniformemente como pertencentes à classe de 17.0dB. Esse processo de categorização permitiu uma análise direta e simplificada entre os resultados do regressor e as classes definidas previamente, facilitando a interpretação dos dados e a comparação de desempenho entre diferentes abordagens de modelagem.

A comparação dos resultados apresentados na Tabela 4.60 para a predição de OSNR na modulação BPSK entre regressor e rede neural revela diferenças marcantes. A rede neural mostra um desempenho consistente e superior em todas as classes,

demonstrando sua habilidade em adaptar-se e generalizar bem ao longo de diferentes níveis de OSNR. O regressor, embora comece forte em classes de baixa OSNR, vê uma queda acentuada em eficácia nas classes mais altas, indicando limitações na captura de complexidades em dados com maior variação. Este contraste destaca a robustez das redes neurais em tarefas de classificação complexas, fornecendo indicativos valiosos para a escolha do modelo para aplicação prática.

Comparação entre a predição do valor de OSNR com regressor e rede neural em BPSK								
Classes	Regressor				Rede neural			
OSNR	Precisão	<i>Recall</i>	<i>F1-score</i>	Suporte	Precisão	<i>Recall</i>	<i>F1-score</i>	Suporte
8, 0dB	1,00	0,25	0,40	556	0,91	0,68	0,78	556
8, 5dB	0,49	0,55	0,52	569	0,62	0,82	0,71	569
9, 0dB	0,55	0,69	0,62	565	0,72	0,62	0,67	565
9, 5dB	0,58	0,70	0,63	553	0,69	0,59	0,63	553
10, 0dB	0,61	0,69	0,65	558	0,59	0,84	0,69	558
10, 5dB	0,65	0,68	0,67	545	0,82	0,38	0,52	545
11, 0dB	0,69	0,71	0,70	561	0,67	0,72	0,69	561
11, 5dB	0,73	0,74	0,73	547	0,72	0,90	0,80	547
12, 0dB	0,73	0,77	0,75	547	0,85	0,78	0,81	547
12, 5dB	0,77	0,77	0,77	560	0,80	0,54	0,64	560
13, 0dB	0,77	0,81	0,79	585	0,54	0,96	0,69	585
13, 5dB	0,72	0,79	0,75	563	0,90	0,43	0,58	563
14, 0dB	0,72	0,74	0,73	555	0,83	0,88	0,85	555
14, 5dB	0,70	0,75	0,72	589	0,79	0,93	0,86	589
15, 0dB	0,99	0,60	0,74	547	0,95	0,82	0,88	547

Tabela 4.60: Comparativo dos resultados entre regressor e rede neural, para a predição do valor de OSNR em BPSK.

A comparação do desempenho apresentado na Tabela 4.61 para a predição de OSNR em QPSK, entre o regressor e a rede neural destaca a superior adaptabilidade e precisão da rede neural em todo o espectro de classes de OSNR. Notavelmente, a rede neural demonstra precisão e *recall* consistentemente superiores, especialmente em faixas médias a altas de OSNR, onde supera significativamente o regressor. Isso sugere a capacidade aprimorada da rede neural em capturar e prever as nuances das variações de OSNR de forma mais eficaz do que o regressor, que, embora comece forte, não mantém seu desempenho de maneira uniforme em toda a gama de valores de OSNR.

A análise dos resultados apresentados na Tabela 4.62 para a predição do valor de OSNR em 8-PSK, entre regressor e rede neural revela diferenças significativas em desempenho. O regressor mostrou-se altamente preciso em algumas classes, alcançando até 99% de precisão, mas com variações no *recall*, indicando uma certa

Comparação entre a predição do valor de OSNR com regressor e rede neural em QPSK								
Classes	Regressor				Rede neural			
OSNR	Precisão	<i>Recall</i>	<i>F1-score</i>	Suporte	Precisão	<i>Recall</i>	<i>F1-score</i>	Suporte
12, 0dB	0,99	0,61	0,76	558	0,95	0,89	0,92	558
12, 5dB	0,67	0,82	0,74	571	0,80	0,88	0,84	571
13, 0dB	0,73	0,81	0,77	568	0,86	0,81	0,83	568
13, 5dB	0,78	0,82	0,80	555	0,77	0,92	0,84	555
14, 0dB	0,82	0,83	0,82	560	0,87	0,77	0,82	560
14, 5dB	0,82	0,82	0,82	546	0,79	0,88	0,83	546
15, 0dB	0,81	0,82	0,82	561	0,90	0,66	0,76	561
15, 5dB	0,82	0,83	0,83	547	0,76	0,92	0,83	547
16, 0dB	0,82	0,84	0,83	547	0,80	0,81	0,81	547
16, 5dB	0,85	0,81	0,83	560	0,92	0,62	0,74	560
17, 0dB	0,83	0,83	0,83	585	0,73	0,93	0,82	585
17, 5dB	0,81	0,82	0,82	555	0,88	0,72	0,79	555
18, 0dB	0,80	0,83	0,81	558	0,80	0,88	0,84	558
18, 5dB	0,80	0,81	0,81	563	0,85	0,77	0,81	563
19, 0dB	0,75	0,84	0,79	563	0,74	0,89	0,81	563
19, 5dB	0,68	0,82	0,74	568	0,87	0,61	0,72	568
20, 0dB	1,00	0,60	0,75	555	0,80	0,95	0,87	555

Tabela 4.61: Comparativo dos resultados entre regressor e rede neural, para a predição do valor de OSNR em QPSK.

Comparação entre a predição do valor de OSNR com regressor e rede neural em 8-PSK								
Classes	Regressor				Rede neural			
OSNR	Precisão	<i>Recall</i>	<i>F1-score</i>	Suporte	Precisão	<i>Recall</i>	<i>F1-score</i>	Suporte
17, 0dB	1,00	0,59	0,74	576	0,82	0,80	0,81	576
17, 5dB	0,68	0,86	0,76	565	0,51	0,81	0,63	565
18, 0dB	0,78	0,85	0,82	554	0,66	0,38	0,48	554
18, 5dB	0,79	0,88	0,83	551	0,77	0,35	0,48	551
19, 0dB	0,83	0,86	0,84	566	0,51	0,64	0,57	566
19, 5dB	0,81	0,86	0,83	555	0,47	0,77	0,59	555
20, 0dB	0,83	0,84	0,84	559	0,77	0,29	0,42	559
20, 5dB	0,73	0,87	0,79	554	0,57	0,83	0,67	554
21, 0dB	0,99	0,67	0,80	560	0,90	0,62	0,73	560

Tabela 4.62: Comparativo dos resultados entre regressor e rede neural, para a predição do valor de OSNR em 8-PSK.

especialização na identificação de classes específicas. Em contraste, a rede neural apresentou um desempenho mais equilibrado entre precisão e *recall*, destacando-se na capacidade de generalização. Notavelmente, a rede neural manteve um desempenho superior, ilustrando sua eficácia em capturar a variabilidade de OSNR com maior consistência.

A análise dos resultados apresentados na Tabela 4.63 para a predição do valor de OSNR em 16-QAM destaca a superioridade do modelo de rede neural em comparação ao regressor, especialmente em termos de precisão e *recall*. O regressor demonstra desempenho consistente em classes com valores de OSNR mais baixos,

mas sua eficácia diminui significativamente para classes com valores mais altos, indicando uma limitação na capacidade de generalização para toda a gama de OSNR. Em contraste, a rede neural exibe desempenho robusto e equilibrado, com destaque para sua habilidade em manter altos níveis de precisão e *recall* mesmo nas classes de maior valor de OSNR, evidenciando sua versatilidade e eficácia na predição de OSNR em cenários variados de 16-QAM.

Comparação entre a predição do valor de OSNR com regressor e rede neural em 16-QAM								
Classes	Regressor				Rede neural			
OSNR	Precisão	<i>Recall</i>	<i>F1-score</i>	Suporte	Precisão	<i>Recall</i>	<i>F1-score</i>	Suporte
17, 0dB	1,00	0,29	0,45	561	0,49	0,96	0,65	561
17, 5dB	0,52	0,57	0,54	572	0,29	0,15	0,20	572
18, 0dB	0,58	0,71	0,64	569	0,47	0,41	0,44	569
18, 5dB	0,64	0,76	0,69	557	0,37	0,55	0,44	557
19, 0dB	0,70	0,73	0,71	564	0,44	0,22	0,29	564
19, 5dB	0,67	0,76	0,71	547	0,51	0,60	0,55	547
20, 0dB	0,78	0,77	0,77	564	0,61	0,47	0,53	564
20, 5dB	0,80	0,76	0,78	547	0,65	0,56	0,60	547
21, 0dB	0,75	0,77	0,76	549	0,62	0,56	0,59	549
21, 5dB	0,72	0,75	0,73	561	0,59	0,67	0,62	561
22, 0dB	0,73	0,79	0,76	586	0,71	0,34	0,46	586
22, 5dB	0,75	0,75	0,75	559	0,50	0,81	0,62	559
23, 0dB	0,70	0,73	0,71	552	0,61	0,40	0,48	552
23, 5dB	0,69	0,72	0,70	559	0,58	0,64	0,61	559
24, 0dB	0,67	0,74	0,70	551	0,27	0,01	0,01	551
24, 5dB	0,69	0,71	0,70	562	0,37	0,58	0,45	562
25, 0dB	0,62	0,84	0,72	571	0,42	0,41	0,41	571
25, 5dB	0,51	0,57	0,54	552	0,30	0,18	0,23	552
26, 0dB	1,00	0,27	0,42	557	0,50	0,91	0,65	557

Tabela 4.63: Comparativo dos resultados entre regressor e rede neural, para a predição do valor de OSNR em 16-QAM.

A comparação entre os resultados do regressor e da rede neural apresentados na Tabela 4.64 para a predição de OSNR em 32-QAM revela diferenças significativas de desempenho. O regressor mostra precisão máxima em classes específicas, mas com *recall* extremamente baixo, indicando uma limitada capacidade de identificação correta dos valores de OSNR nestas faixas. Por outro lado, a rede neural apresenta uma distribuição mais equilibrada de precisão e *recall* através das classes, sugerindo uma melhor generalização e capacidade de identificar corretamente uma gama ampla de valores de OSNR. Esse desempenho superior da rede neural destaca sua aplicabilidade mais eficaz para a predição de OSNR em cenários de 32-QAM.

Por fim, a comparação entre os resultados do regressor e da rede neural apresentados na Tabela 4.65, para a predição do valor de OSNR em 64-QAM mostra um

Comparação entre a predição do valor de OSNR com regressor e rede neural em 32-QAM								
Classes	Regressor				Rede neural			
OSNR	Precisão	<i>Recall</i>	<i>F1-score</i>	Suporte	Precisão	<i>Recall</i>	<i>F1-score</i>	Suporte
15, 0dB	1,00	0,03	0,06	556	0,57	0,72	0,64	556
15, 5dB	0,32	0,29	0,30	575	0,42	0,57	0,48	575
16, 0dB	0,42	0,51	0,46	568	0,42	0,25	0,31	568
16, 5dB	0,50	0,64	0,56	552	0,38	0,54	0,44	552
17, 0dB	0,57	0,73	0,64	563	0,43	0,35	0,38	563
17, 5dB	0,60	0,76	0,67	536	0,39	0,25	0,30	536
18, 0dB	0,66	0,78	0,72	575	0,31	0,52	0,39	575
18, 5dB	0,60	0,76	0,67	552	0,38	0,20	0,27	552
19, 0dB	0,55	0,68	0,61	547	0,35	0,35	0,35	547
19, 5dB	0,50	0,63	0,56	559	0,28	0,48	0,36	559
20, 0dB	0,46	0,51	0,48	574	0,33	0,26	0,29	574
20, 5dB	0,32	0,30	0,31	569	0,46	0,15	0,23	569
21, 0dB	1,00	0,04	0,08	554	0,54	0,48	0,51	554

Tabela 4.64: Comparativo dos resultados entre regressor e rede neural, para a predição do valor de OSNR em 32-QAM.

Comparação entre a predição do valor de OSNR com regressor e rede neural em 64-QAM								
Classes	Regressor				Rede neural			
OSNR	Precisão	<i>Recall</i>	<i>F1-score</i>	Suporte	Precisão	<i>Recall</i>	<i>F1-score</i>	Suporte
22, 0dB	0,00	0,00	0,00	564	0,57	0,01	0,01	564
22, 5dB	0,06	0,03	0,04	571	0,50	0,00	0,00	571
23, 0dB	0,14	0,12	0,13	572	0,75	0,01	0,01	572
23, 5dB	0,26	0,31	0,28	559	0,57	0,01	0,01	559
24, 0dB	0,33	0,44	0,38	564	0,40	0,00	0,01	564
24, 5dB	0,40	0,54	0,46	562	0,00	0,00	0,00	562
25, 0dB	0,49	0,60	0,54	568	0,22	0,00	0,01	568
25, 5dB	0,51	0,62	0,56	532	0,00	0,00	0,00	532
26, 0dB	0,51	0,58	0,54	548	0,10	0,00	0,00	548
26, 5dB	0,55	0,58	0,57	563	0,00	0,00	0,00	563
27, 0dB	0,56	0,62	0,59	582	0,12	0,00	0,00	582
27, 5dB	0,56	0,62	0,59	563	0,04	1,00	0,08	563
28, 0dB	0,55	0,63	0,58	555	0,00	0,00	0,00	555
28, 5dB	0,53	0,59	0,56	554	0,00	0,00	0,00	554
29, 0dB	0,50	0,58	0,54	565	0,00	0,00	0,00	565
29, 5dB	0,48	0,56	0,52	544	0,17	0,00	0,01	544
30, 0dB	0,45	0,60	0,51	549	0,00	0,00	0,00	549
30, 5dB	0,44	0,54	0,48	562	0,00	0,00	0,00	562
31, 0dB	0,42	0,49	0,45	566	0,29	0,00	0,01	566
31, 5dB	0,35	0,35	0,35	562	0,00	0,00	0,00	562
32, 0dB	0,23	0,23	0,23	559	0,00	0,00	0,00	559
32, 5dB	0,16	0,09	0,11	570	0,00	0,00	0,00	570
33, 0dB	0,00	0,00	0,00	546	0,14	0,00	0,00	546

Tabela 4.65: Comparativo dos resultados entre regressor e rede neural, para a predição do valor de OSNR em 64-QAM.

contraste marcante entre as duas abordagens. O regressor apresenta um desempenho modesto, com precisão e *recall* baixos em várias classes, indicando dificuldade em capturar a variação nos valores de OSNR. Notavelmente, a rede neural demonstra limitações significativas, com baixa eficácia em quase todas as classes, refletido

por *scores* de precisão e *recall* próximos a zero. Esta análise sugere que, para cenários complexos como 64-QAM, ambos os modelos enfrentam desafios substanciais, necessitando de ajustes ou abordagens alternativas para melhorar a predição de OSNR.

Com a conclusão da análise entre o regressor e a rede neural para diferentes modulações, observa-se que, em certas modulações e faixas específicas de OSNR, o regressor demonstrou desempenho pontualmente superior. No entanto, a rede neural mostrou-se mais eficaz e consistente na maioria dos cenários analisados, oferecendo um desempenho geralmente superior. Apesar dessa vantagem, há espaço para aprimoramento em modulações complexas, como 32-QAM e 64-QAM, sugerindo a necessidade de futuras investigações para otimizar a predição do OSNR.

4.3.3 Pergunta 3: O que pode ser concluído a partir dos resultados das simulações e experimentos realizados?

A definição das estratégias de classificação da modulação e predição dos valores de OSNR guiou a configuração dos experimentos. Diversos modelos foram submetidos a treinamentos extensivos, ajustando uma variedade de parâmetros para gerar uma ampla gama de resultados. Através da análise comparativa no conjunto de resultados para cada modelo, identificamos os desempenhos mais destacados. Posteriormente, esses resultados permitiram comparações entre diferentes modelos sob a mesma estratégia. Identificamos os modelos *Random Forest*, *Gradient Boosting* e *Perceptron Multicamada* como os mais eficazes para a estratégia de classificação das modulações, diferenciando-se principalmente pela complexidade computacional. Na estratégia de predição de OSNR, a diversidade de classes representou um desafio significativo. A complexidade do teste aumentou ao adaptar os modelos para diferentes tipos de modulação, resultando em uma ampla gama de modelos analisados. Observou-se que o regressor *Random Forest* alcançou desempenho notável em configurações específicas, enquanto o *Perceptron Multicamada* se sobressaiu pela sua habilidade de generalizar os resultados. Embora progressos tenham sido feitos, há oportunidades para melhorias e investigações futuras para aprimorar as técnicas de regeneração de sinais em sistemas de comunicação óptica coerente flexíveis.

Capítulo 5

Conclusão

Nesta dissertação, investigamos o uso de algoritmos de aprendizado de máquina na identificação do tipo de modulação e na predição de valores de OSNR para uma gama de modulações, incluindo DP-BPSK, DP-QPSK, DP-8-PSK, DP-16-QAM, DP-32-QAM e DP-64-QAM. Abrangemos valores de OSNR que variam de 8.0dB/0,1nm a 33.0dB/0,1nm, oferecendo um detalhamento fino com 51 níveis distintos. Este estudo foca em receptores ópticos coerentes flexíveis, explorando o potencial das técnicas de aprendizado para aprimorar a eficiência e adaptabilidade desses sistemas. Para isso, desenvolveu-se uma metodologia robusta para simulação e análise de dados, empregando um *framework* especialmente criado para simular efeitos de canal óptico, além de gerar, transmitir e regenerar sinais com base em modulações e valores de OSNR previamente definidos. Este processo incluiu a adoção de estratégias específicas utilizando uma variedade de modelos de aprendizado de máquina, como *Random Forest*, *AdaBoost*, *Gradient Boost*, *Árvore de Decisão*, *CART* e *perceptron multicamada*. Esta abordagem permitiu, inicialmente, a classificação eficiente dos tipos de modulação e, subsequentemente, a predição dos valores de OSNR, ilustrando a aplicabilidade e eficácia dessas técnicas no campo da óptica coerente. Os modelos *Random Forest*, *Gradient Boost* e *Perceptron Multicamada* demonstraram alta eficácia na classificação do tipo de modulação, alcançando acurácias acima de 99%. Contudo, na tarefa de predição de OSNR, os modelos *Random Forest* e *Perceptron Multicamada* apresentaram desempenho apenas moderado, com acurácias variando entre as modulações. Para melhorar esses resultados, expandiu-se a margem de aceitação nas predições para $\pm 5,0$ dB e adicionou-se o *Random*

Forest como uma abordagem de regressão, buscando diversificar as estratégias de modelagem. A metodologia adotada permitiu uma análise comparativa equitativa entre os modelos de aprendizado de máquina em cenários complexos de predição de OSNR. Mediante uma abordagem baseada em perguntas específicas, identificou-se o *Perceptron Multicamada* como o modelo mais eficaz, destacando-se pela sua acurácia flexível superior em todos os tipos de modulação estudados, sendo de 95,55,81%, 98,63%, 90,48%, 90,48%, 64,39% e 41,39%. Contudo, observou-se que o *Random Forest*, quando aplicado como regressor, apresentou resultados promissores em certas classes específicas de OSNR, embora com uma generalização inferior em comparação ao *Perceptron Multicamada*. Esta dissertação avança significativamente o campo dos sistemas de comunicação óptica ao desenvolver e compartilhar um *framework* de simulação inovador. Este *framework* não só facilita a pesquisa futura com transmissores ópticos coerentes flexíveis, mas também valida a estratégia para identificar automaticamente o tipo de modulação e o valor de OSNR. Ao automatizar a identificação dessas variáveis críticas, elimina-se a necessidade de intervenção manual especializada, potencializando a eficiência e a autonomia na regeneração de sinais. Essa capacidade de reconfiguração autônoma dos DSPs (Processadores Digitais de Sinais) em dispositivos ópticos contribui para melhorar a gestão das redes de transporte óptico e facilitar a implementação de redes definidas por software, promovendo uma maior flexibilidade e controle operacional.

5.1 Estudos futuros

Para futuras investigações, sugere-se a exploração de novos modelos e estratégias voltadas especificamente para a melhoria da precisão na predição dos valores de OSNR. Isso pode incluir a análise detalhada dos parâmetros utilizados nos modelos atuais, bem como a introdução de novas técnicas e abordagens alternativas para enriquecer a capacidade preditiva, visando aprimorar significativamente os resultados obtidos.

Uma abordagem futura interessante seria avaliar o impacto de efeitos não lineares adicionais inerentes às fibras ópticas. Isso poderia ser feito expandindo o *framework* de simulação para incluir cenários com distâncias de fibra maiores.

Essa expansão revisitaria a necessidade de simular o uso de amplificadores e outros componentes de sistemas de comunicação óptica, cruciais em configurações que vão além do esquema *back-to-back*. Esta análise adicional permitiria uma compreensão mais aprofundada das complexidades envolvidas na transmissão óptica de longa distância.

Implementar o sistema em um ambiente real seria um passo crucial para validar a aplicabilidade prática das técnicas desenvolvidas. Desafios como o tempo necessário para o pré-processamento e a execução das predições seriam avaliados em condições operacionais reais. Tal implantação permitiria testar a robustez e a eficiência do sistema sob variáveis adicionais não contempladas no ambiente de simulação, fornecendo percepções valiosas para refinamentos futuros e ajustes para otimizar o desempenho em aplicações reais.

Explorar a integração dos modelos em ambientes de *Edge Learning* ao incorporá-los diretamente nos Processadores Digitais de Sinais (DSPs) pode abrir novas fronteiras para o processamento de sinais ópticos. Essa abordagem permitiria a execução de algoritmos de aprendizado de máquina diretamente no hardware, próximo ao ponto de coleta de dados, otimizando o desempenho e reduzindo a latência nas decisões tomadas pelo sistema. A adaptação para esse cenário demanda considerações técnicas específicas, mas pode significativamente aumentar a eficiência e a velocidade do processamento de sinais em redes ópticas.

Referências Bibliográficas

- [1] CISCO, U., “Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper”, *Cisco: San Jose, CA, USA*, 2020.
- [2] BOGDAN-MARTIN, D., *Facts and Figures 2020*, Tech. rep., Cisco: San Jose, CA, USA, 2020.
- [3] “Brasil terá 5G em 2022”, <https://www.gov.br/pt-br/noticias/transito-e-transportes/2021/03/brasil-tera-5g-em-2022>, March 2021.
- [4] “Liberação Do 5G Pode Ocorrer Para Algumas Capitais Antes de Agosto, Afirmam Anatel - Notícias - Portal Da Câmara Dos Deputados”, <https://www.camara.leg.br/noticias/885603-liberacao-do-5g-pode-ocorrer-para-algumas-capitais>, June 2022.
- [5] SKUBIC, B., BOTTARI, G., ROSTAMI, A., et al., “Rethinking Optical Transport to Pave the Way for 5G and the Networked Society”, *Journal of Lightwave Technology*, v. 33, pp. 1084–1091, March 2015.
- [6] AGIWAL, M., ROY, A., SAXENA, N., “Next Generation 5G Wireless Networks: A Comprehensive Survey”, *IEEE Communications Surveys & Tutorials*, v. 18, n. 3, pp. 1617–1655, 2016.
- [7] VPIPHOTONICS, “Citations of VPIphotronics Design Suite”, *VPIphotronics*, Dec. 2022.
- [8] ITU, *G.694.1 : Spectral Grids for WDM Applications: DWDM Frequency Grid*, Tech. rep., ITU - International Telecommunication Union, Jan. 2021.

- [9] “Demonstration of World Record: 319 Tb/s Transmission over 3,001 km with 4-core optical fiber | 2021”, <https://www.nict.go.jp/en/press/2021/07/12-1.html>, July 2021.
- [10] “World Record 301 Tb/s Transmission in a Standard Commercially Available Optical Fiber | 2024”, <https://www.nict.go.jp/en/press/2024/01/29-1.html>.
- [11] LETAIEF, K. B., CHEN, W., SHI, Y., et al., “The Roadmap to 6G: AI Empowered Wireless Networks”, *IEEE Communications Magazine*, v. 57, n. 8, pp. 84–90, Aug. 2019.
- [12] DE CASTRO, L. G. M., SANTOS, A. E., LUIZ, D. F., et al., “A Survey of 5G Network Systems: Trends and Deep Learning Approaches”. In: *2022 IEEE International Conference on Consumer Electronics - Taiwan*, pp. 25–26, July 2022.
- [13] FETTWEIS, G., BOCHE, H., WIEGAND, T., et al., *The Tactile Internet*, Tech. rep., International Telecommunication Union (ITU), 2014.
- [14] FETTWEIS, G. P., “The Tactile Internet: Applications and Challenges”, *IEEE Vehicular Technology Magazine*, v. 9, n. 1, pp. 64–70, March 2014.
- [15] MATA, J., DE MIGUEL, I., N, R. N. J. D., et al., “Artificial Intelligence (AI) Methods in Optical Networks: A Comprehensive Survey”, *Optical Switching and Networking*, v. 28, pp. 43–57, April 2018.
- [16] MUSUMECI, F., ROTTONDI, C., NAG, A., et al., “An Overview on Application of Machine Learning Techniques in Optical Networks”, *IEEE Communications Surveys & Tutorials*, v. 21, n. 2, pp. 1383–1408, 2019.
- [17] AMIRABADI, M. A., “A Survey on Machine Learning for Optical Communication [Machine Learning View]”, Aug. 2019.
- [18] MUSUMECI, F., ROTTONDI, C., NAG, A., et al., “A Survey on Application of Machine Learning Techniques in Optical Networks”, *ArXiv*, March 2018.

- [19] THAJEEL, T. G., ABDULHASSAN, A., “A Comprehensive Survey on Software-Defined Networking Load Balancers”. In: *2021 4th International Iraqi Conference on Engineering Technology and Their Applications (IICETA)*, pp. 1–7, Sept. 2021.
- [20] TOMKOS, I., MUKHERJEE, B., KOROTKY, S. K., et al., “The Evolution of Optical Networking [Scanning the Issue]”, *Proc. IEEE*, v. 100, n. 5, pp. 1017–1022, May 2012.
- [21] HUYNH-THE, T., PHAM, Q.-V., NGUYEN, T.-V., et al., “Automatic Modulation Classification: A Deep Architecture Survey”, *IEEE Access*, v. 9, pp. 142950–142971, 2021.
- [22] FURTADO, R. S., TORRES, Y. P., SILVA, M. O., et al., “Automatic Modulation Classification in Real Tx/Rx Environment Using Machine Learning and SDR”. In: *2021 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–4, Jan. 2021.
- [23] ZHAO, Y., SHI, C., YANG, T., et al., “Low-Complexity and Joint Modulation Format Identification and OSNR Estimation Using Random Forest for Flexible Coherent Receivers”, *Optics Communications*, v. 457, pp. 124698, Feb. 2020.
- [24] ZHANG, Q., ZHOU, H., JIANG, Y., et al., “A Simple Joint Modulation Format Identification and OSNR Monitoring Scheme for IMDD OOFDM Transceivers Using K-Nearest Neighbor Algorithm”, *Applied Sciences*, v. 9, n. 18, pp. 3892, Jan. 2019.
- [25] HAMDAN, M., HASSAN, E., ABDELAZIZ, A., et al., “A Comprehensive Survey of Load Balancing Techniques in Software-Defined Network”, *Journal of Network and Computer Applications*, v. 174, pp. 102856, Jan. 2021.
- [26] BORGES, L., COSTA, C., DE PAULA, R., et al., “K-Means Clustering for Mitigation of Nonlinear Phase Noise in Digital Coherent Optical Systems Using 16-QAM Modulation Format”. In: *XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT2020)*, pp. 1–5, 2020.

- [27] COSTA, A., MARTINS, I., RUDGE BARBOSA, F., “Effects of All-Optical Wavelength Conversion on Signal Quality under Variable-Channel Spacing in Elastic Optical Networks”, *Optical Engineering*, v. 59, pp. 1, June 2020.
- [28] “Submarine Cables | National Oceanic and Atmospheric Administration”, <https://www.noaa.gov/submarine-cables>.
- [29] DE OLIVEIRA, J. M. L., “Padtec vence concorrência e irá fornecer solução DWDM para novas infovias do programa Norte Conectado”, <https://www.padtec.com.br/padtec-vence-concorrencia-e-ira-fornecer-solucao-dwdm-para-novas-infovias-do-programa-norte-conectado/>, Sept. 2023.
- [30] SHALEV-SHWARTZ, S., BEN-DAVID, S., *Understanding Machine Learning: From Theory to Algorithms*. 1st ed. Cambridge University Press, May 2014.
- [31] “API Reference”, <https://scikit-learn/stable/modules/classes.html>, 2023.
- [32] HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. *Springer Series in Statistics*, Springer: New York, NY, 2009.
- [33] ELTAIEB, R. A., FARGHAL, A. E. A., AHMED, H.-D. H., et al., “Efficient Classification of Optical Modulation Formats Based on Singular Value Decomposition and Radon Transformation”, *Journal of Lightwave Technology*, v. 38, n. 3, pp. 619–631, Feb. 2020.
- [34] JONES, R. T., *Machine Learning Methods in Coherent Optical Communication Systems*, Ph.D. Thesis, Technical University of Denmark, 2019.
- [35] MITCHELL, T. M., *Machine Learning*. Nachdr. ed. *McGraw-Hill Series in Computer Science*, McGraw-Hill: New York, 2013.
- [36] GÉRON, A., *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 2nd ed. O’Reilly, 2019.

- [37] ZHANG, C., MA, Y., (eds), *Ensemble Machine Learning: Methods and Applications*. Springer: New York, NY, 2012.
- [38] AZZOUZ, E. E., NANDI, A. K., *Automatic Modulation Recognition of Communication Signals*. Springer US: Boston, MA, 1996.
- [39] ZHU, Z., NANDI, A. K., *Automatic Modulation Classification: Principles, Algorithms and Applications*. 1st ed. Wiley: Chichester, West Sussex, Feb. 2015.
- [40] VALADÃO, M. D. M., SILVA, W., COSTA, A. L. A., et al., “Classificação Automática de Modulações Utilizando Redes Neurais Artificiais Com Regularização Bayesiana e Algoritmo de Retropropagação de Levenberg-Marquardt”. In: *XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT2020)*, pp. 1–5, 2020.
- [41] AMOEDO, D., MELO, W., CORDEIRO, L., et al., “Classificação Automática de Modulações Mono e Multiportadoras utilizando Método de Extração de Características e Classificadores SVM”. In: *Anais de XXXV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, pp. 1–5, Sociedade Brasileira de Telecomunicações, 2017.
- [42] PEREIRA, A. M. C., FURTADO, R. S., AMOEDO, D. A., et al., “Classificação Automática de Modulações DP M-PSK e DP m-QAM Em Receptores Ópticos Coerentes Flexíveis”. In: *XXXIX Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT2021)*, pp. 1–5, 2021.
- [43] KHAN, F. N., ZHONG, K., AL-ARASHI, W. H., et al., “Modulation Format Identification in Coherent Receivers Using Deep Machine Learning”, *IEEE Photonics Technology Letters*, v. 28, n. 17, pp. 1886–1889, Sept. 2016.
- [44] KUMAR, Y., SHEORAN, M., JAJOO, G., et al., “Automatic Modulation Classification Based on Constellation Density Using Deep Learning”, *IEEE Communications Letters*, v. 24, n. 6, pp. 1275–1278, June 2020.
- [45] SIMMONS, J. M., *Optical Network Design and Planning. Optical Networks*, Springer International Publishing: Cham, 2014.

- [46] YUAN, T., DENG, W., TANG, J., et al., “Signal-to-Noise Ratio: A Robust Distance Metric for Deep Metric Learning”, April 2019.
- [47] SCHANNER, L., “Evaluation of Frequency-Domain Learned Digital Back-Propagation Nonlinear Compensation for Unrepeated Optical Links”. In: *Anais de XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, pp. 1–4, Sociedade Brasileira de Telecomunicações, 2020.
- [48] PINCEMIN, E., KARAKI, J., LOUSSOUARN, Y., et al., “Challenges of 40/100Gbps and Higher-Rate Deployments over Long-Haul Transport Networks”, *Optical Fiber Technology*, v. 17, n. 5, pp. 335–362, Oct. 2011.
- [49] ALWAYN, V., *Optical Network Design and Implementation / Cisco Press*. Cisco Press, March 2004.
- [50] WILLNER, A. E., FALLAHPOUR, A., ALISHAHI, F., et al., “All-Optical Signal Processing Techniques for Flexible Networks”, *Journal of Lightwave Technology*, v. 37, n. 1, pp. 21–35, Jan. 2019.
- [51] NADEAU, T. D., GRAY, K., *SDN: Software Defined Networks*. O’Reilly, Aug. 2013.
- [52] GÖRANSSON, P., BLACK, C., *Software Defined Networks: A Comprehensive Approach*. Elsevier, Morgan Kaufmann: Amsterdam ; Boston, 2014.
- [53] LIN, X., ELDEMERDASH, Y. A., DOBRE, O. A., et al., “Modulation Classification Using Received Signal’s Amplitude Distribution for Coherent Receivers”, *IEEE Photonics Technology Letters*, v. 29, n. 21, pp. 1872–1875, Nov. 2017.
- [54] PEREIRA, D. R., “Redes ópticas definidas por software: um estudo do controle de equipamentos NETCONF/YANG usando os modelos OpenROADM”, *Optical software defined network: a study over the control of NETCONF/YANG equipment using OpenROADM models*, Dec. 2020.
- [55] LOCATELLI, F., CHRISTODOULOPOULOS, K., FÀBREGA, J. M., et al., “Experimental Demonstration of a Machine Learning-Based in-Band

- OSNR Estimator from Optical Spectra”. In: *2020 International Conference on Optical Network Design and Modeling (ONDM)*, pp. 1–4, May 2020.
- [56] THRANE, J., WASS, J., PIELS, M., et al., “Machine Learning Techniques for Optical Performance Monitoring From Directly Detected PDM-QAM Signals”, *Journal of Lightwave Technology*, v. 35, n. 4, pp. 868–875, Feb. 2017.
- [57] RAMASWAMI, R., SIVARAJAN, K., SASAKI, G., *Optical Networks: A Practical Perspective*. 3rd ed. Morgan Kaufmann: Amsterdam ; Boston, July 2009.
- [58] BREIMAN, L., “Random Forests”, *Machine Learning*, v. 45, n. 1, pp. 5–32, Oct. 2001.
- [59] LIAW, A., WIENER, M., “Classification and Regression by RandomForest”, *Forest*, v. 23, Nov. 2001.
- [60] LOH, W.-Y., “Classification and regression trees”, *Wiley interdisciplinary reviews: data mining and knowledge discovery*, v. 1, pp. 14–23, 2011.
- [61] SCHAPIRE, R. E., FREUND, Y., *Boosting: Foundations and Algorithms*. The MIT Press, May 2012.
- [62] FRIEDMAN, J. H., “Greedy Function Approximation: A Gradient Boosting Machine”, *The Annals of Statistics*, v. 29, n. 5, pp. 1189–1232, 2001.
- [63] NATEKIN, A., KNOLL, A., “Gradient Boosting Machines, a Tutorial”, *Frontiers in Neurorobotics*, v. 7, 2013.
- [64] GOODFELLOW, I., BENGIO, Y., COURVILLE, A., *Deep Learning*. MIT Press, 2016.

Apêndice A

Artigos Publicados

Neste apêndice, o artigo desenvolvido nesta dissertação é apresentado.

Pereira, Antonio; Furtado, Rafael; Amoedo, Diego; Valadão, Myke; Carvalho, Celso; Costa, André; Junior, Waldir. Classificação Automática de Modulações DP m-PSK e DP m-QAM em Receptores Ópticos Coerentes Flexíveis. 10.14209/sbrt.2021.1570724020, 2021.