



UNIVERSIDADE FEDERAL FLUMINENSE
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DE
TELECOMUNICAÇÕES

ALLYSON ANDRADE DA SILVA

Classificação sonar passiva de Navios
utilizando *Machine Learning*

NITERÓI

2024

UNIVERSIDADE FEDERAL FLUMINENSE
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DE
TELECOMUNICAÇÕES

ALLYSON ANDRADE DA SILVA

Classificação sonar passiva de Navios utilizando *Machine Learning*

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Telecomunicações da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica e de Telecomunicações. Área de concentração: <Sistemas de Telecomunicações>.

Orientador:

Prof. Tadeu Nagashima Ferreira, D.Sc.

Co-orientador:

Prof. Lisandro Lovisolo, D.Sc.

NITERÓI

2024

Ficha catalográfica elaborada pelo Sistema de Bibliotecas da UFF - SDC/UFF
com os dados fornecidos pelo(a) autor(a)

S586t Silva Junior, José Lins da
Título do Trabalho: subtítulo do trabalho / José Lins da
Silva Junior; Orientador Sobrenome, orientador; Coorientador
Sobrenome, co-orientador. Niterói, 2017.
120 f.

Monografia (Especialização em Engenharia de Produção) -
Universidade Federal Fluminense, Laboratório de Tecnologia,
Gestão de Negócios e Meio Ambiente, Niterói, 2017.

1. Primeiro assunto. 2. Produção intelectual. I. Título
II. Sobrenome, Orientador, orientador. III. Sobrenome,
Coorientador, co-orientador. IV. Universidade Federal
Fluminense. Laboratório de Tecnologia, Gestão de Negócios e
Meio Ambiente.

CDD -

ALLYSON ANDRADE DA SILVA

Classificação sonar passiva de Navios utilizando *Machine Learning*

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Telecomunicações da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica e de Telecomunicações. Área de concentração: Sistemas de Telecomunicações.

Aprovada em 3 de abril de 2024.

BANCA EXAMINADORA

Prof. Tadeu Nagashima Ferreira, D.Sc. – Orientador, UFF

Prof. Lisandro Lovisolo, D.Sc. – Coorientador, UERJ

Prof. Edson Luiz Cataldo Ferreira, D.Sc. – UFF

Prof. Jorge Luis Machado do Amaral, D.Sc. – UERJ

Dr. Cap. de Mar e Guerra André Paim Gonçalves, D.Sc. –
Marinha do Brasil

Niterói

2024

Dedicatória

A todos: aqueles me ajudaram e me ajudam;

A todos aqueles que não só ouvem, mas escutam;

Aos amigos que ganhei no caminho.

Àqueles que sempre estiveram comigo.

Ao meu filho Lucas por ser meu melhor amigo.

À minha filha Júlia de exemplar cuidado comigo.

À minha mãe pelas tardes de estudo.

Ao meu pai por pensar em nós acima de tudo.

À minha irmã exemplar.

Aos meus amigos de Marinha por me aturar.

Aos professores por me ensinar.

À minha esposa e amiga

A ti dedico a minha vida

Assim com a tua se tornou minha.

A todos o meu muito obrigado.

À familia

Agradecimentos

Agradeço primeiramente a Deus que me fez nascer numa época tão promissora e de novas tecnologias.

Agradeço a meus pais por seus ensinamentos e convívio. A minha mãe pelas tardes de estudo e a meu pai pelas inúmeras correções em nossos trabalhos familiares que já se tornou pré-requisito para a entrega.

Agradeço a minha irmã pelos planos discutidos para nosso futuro.

Agradeço a minha esposa por ser minha companheira nessa jornada tão interessante que é a vida.

Agradeço meus filhos por ser minha maior motivação.

Agradeço à Marinha do Brasil pela oportunidade de estudar em instituição tão distinta.

Agradeço ao prof. Tadeu Nagashima pelas orientações de trabalho e vida.

Agradeço ao prof. Lisandro Lovisolo por seu interesse pelo meu bem estar.

Agradeço ao Centro de Guerra Acústica e Eletrônica da Marinha (CGAEM) pela confiança e ajuda.

Agradeço ao Centro de Apoio a Sistemas Operativos (CASOP) - Arraial do Cabo por me fornecer os dados, sem os quais seria impossível pesquisar!

Agradeço ao Instituto de Pesquisas da Marinha (IPqM) por me colocar em contato com as novas pesquisas da Marinha e me fazer pertencer ao esforço conjunto acadêmico que norteará nossa Marinha.

Agradeço a meus companheiros de mestrado pela colaboração em inúmeros momentos e amizade!

Resumo

Esta dissertação centra-se na aplicação de algoritmos de *Machine Learning* para identificar assinaturas acústicas (via sonares) de embarcações. O estudo visa aprimorar e ampliar as capacidades de detecção naval quer seja em superfície ou subaquática, no reconhecimento e a categorização de meios aquáticos, navios ou submarinos, a partir de assinaturas acústicas. Investiga-se a eficácia de sete algoritmos distintos de aprendizado de máquina na identificação de assinaturas acústicas; analisa-se o desempenho de cada algoritmo usando métricas apropriadas para determinar suas adequações para implantações no mundo real. Com isso, pretende-se contribuir para o avanço da tecnologia de sistemas autônomos inteligentes, especificamente em navegação subaquática e detecção de objetos. Visando o uso em ambiente real, utiliza-se um banco de dados de assinaturas acústicas coletadas em ambientes subaquáticos, com variações nos tipos de navios, totalizando doze classes de navio, em condições dinâmicas e condições ambientais também variadas, perfazendo um banco de dados composto de 290 assinaturas. Essas assinaturas acústicas necessitam ser pré-processadas para utilizá-las como vetores de características de entrada de classificadores, portanto um método para isso é também apresentado e instrumentalizado. Os sete algoritmos de aprendizado de máquina são escolhidos com base em suas relevâncias e sucesso em tarefas de reconhecimento de padrões. Os algoritmos incluem: Análise de Discriminantes Lineares, Conjuntos de Árvores de Decisão (Xgboost), K-Vizinhos Mais Próximos (KNN), Máquinas de Vetores de Suporte (SVM), *Multi-Layer Perceptron*, Regressão Logística e Regressão *Ridge*. Cada algoritmo é ajustado usando uma metodologia padronizada, R^2 score e só então é treinado e avaliado com métrica da acurácia. Técnicas de validação cruzada são empregadas para garantir robustez e generalização dos resultados. Além disso, os algoritmos são testados de três diferentes formas: com um vetor de características apenas das assinaturas acústicas; esse mesmo vetor submetida à técnica de Análise de Componentes Principais (PCA); e ainda o vetor de características submetido à técnica de Análise não-linear de Componentes Principais (NLPCA). Objetivando concluir qual a melhor técnica para classificação, a sua acurácia e ainda se é possível reduzir o vetor de características mantendo a acurácia da máquina.

Palavras-chave: Classificação, Sonar Passivo, Identificação, *Machine Learning*

Abstract

This master thesis focuses on the application of Machine Learning algorithms to identify acoustic sonar signatures. The study aims to improve the detection capabilities of underwater vehicles, both surface and subsurface, in recognizing and categorizing underwater objects, ships and submarines, based on their acoustic signatures.

The objective is to investigate the effectiveness of seven different machine learning algorithms in accurately identifying acoustic signatures, to analyze the performance of each algorithm, and to determine its suitability for real-world deployment. Additionally, the study aims to contribute to the advancement of intelligent autonomous vehicles technology, specifically in underwater navigation and object detection.

To facilitate real-world applicability, the study utilizes a database of acoustic signatures collected in underwater environments, capturing variations in object types, and environmental conditions, encompassing twelve ship classes in a database with 290 signatures. These acoustic signatures need to be pre-processed, so that they can be used as a feature vector for classifier inputs, therefore a method of achieving this feat is presented and instrumented.

Seven machine learning algorithms are selected based on their relevance to pattern recognition tasks and prior success in similar domains. These algorithms include Linear Discriminant Analysis, Decision Trees (XGBoost), K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Multi-Layer Perceptron, Logistic Regression, and Ridge Regression.

Each algorithm undergoes tuning using a standardized methodology, including R-score evaluation, before being trained and evaluated using accuracy metrics. Cross-validation techniques are employed to ensure the robustness and generalization of the results.

Keywords: Classification, Passive Sonar, Identification, Machine Learning.

Lista de Figuras

1.1	Amazônia Azul inclui o Mar Territorial, a Plataforma continental, a Zona Econômica Exclusiva e as águas interiores brasileiras[1].	3
1.2	Identificação de assinatura acústica de contato de interesse em ambiente ruidoso.	4
2.1	Componentes de um sistema <i>Sound Navigation and Ranging</i> (SoNaR). Adaptado de [2].	13
2.2	SoNaR rebocado que é uma aplicação de sensoriamento remoto.	15
2.3	Refração negativa do som[3].	19
2.4	Refração positiva do som[3].	19
2.5	Velocidade do som em relação à profundidade [4].	20
2.6	SOFAR, zonas de convergência e de sombra.	21
2.7	Capacidade de Guerra eletrônica.	28
2.8	Atividades de Guerra Eletrônica.	29
2.9	Assinaturas e ruídos.	30
3.1	Diferenças entre Sonar de Varredura Lateral (SSS) e Radar de Abertura Sintética (SAS). fonte: https://oceanexplorer.noaa.gov/technology/sonar/sas.html	32
3.2	Towed Array SoNaR - SoNaR com múltiplas entradas e múltiplas saídas. Fonte: https://en.mercopress.com/2022/01/10/royal-navy-frigate-underwater-sonar-struck-by-russian-submarine	35
4.1	Epícciclo da análise de dados, adaptado de [5].	40
4.2	Exemplo de banco de corridas com assinaturas acústicas de diferentes navios, com diferentes quantidades e de diferentes durações	45
4.3	Segmentação do banco de corridas.	45

4.4	Banco de dados equiprovável.	46
4.5	Etapas de pré-processamento de dados.	47
4.6	Exemplo de Lofargrama com frequência, tempo e intensidade [6].	49
4.7	Fluxograma do pré Processamento <i>Low Frequency Analysis and Recording</i> (LOFAR).	50
4.8	Áudio captado pelo hidrofone quando há passagem de um navio.	50
4.9	Janela de Hamming.	51
4.10	Diagrama esquemático do algoritmo TPSW.	52
4.11	Periodograma do Sinal da figura 4.8.	53
4.12	Periodograma do Sinal da figura 4.8 com dados retirados acima de 3 desvio padrão.	54
4.13	Periodograma do Sinal da figura 4.8 com dados retirados com magnitude abaixo de 1% do maior valor.	54
4.14	Periodograma do Sinal da figura 4.8 com TPSW aplicado.	55
5.1	Ajuste de hiperparâmetros seguido por uma avaliação da acurácia.	59
5.2	Resultados acústicos para o conjunto de validação e de teste retirados do mesmo cenário acústico.	61
5.3	Validação cruzada.	64
5.4	Fluxograma de Sinais para avaliação de diferentes técnicas de representação de sinal.	71
5.5	O conjunto de dados tridimensional, onde as setas vermelha, azul e verde indicam a direção do primeiro, segundo e terceiro componentes principais, respectivamente [7].	71
5.6	O truque do <i>kernel</i>	74
6.1	Perceptron, onde uma constante faz a curva de operação não passar pela origem e os pesos ponderam a importância de cada variável da entrada x_i	80
6.2	MLP - configuração 1×1 - Percurso da estimação de otimização de parâmetros, taxa de aprendizagem (α), tamanho das camadas ocultas, inicio do (α) e iterações.	86

6.3	MLP - $1 \times$ Todos - Densidade de Distribuição de Probabilidade e Inter-relacionamento de variáveis.	87
6.4	Algoritmo de classificação KNN, onde $K=3$ [8].	88
6.5	KNN - Percurso da estimação de otimização de parâmetros, número de vizinhos e métrica de distância.	91
6.6	KNN - Densidade de Distribuição de Probabilidade e Inter-relacionamento de variáveis.	92
6.7	Diferenças entre a regressão linear e a ridge.	93
6.8	Ridge - Percurso da estimação de otimização de parâmetros, taxa de aprendizagem α , grau do polinômio e tipo de regularização.	95
6.9	Ridge - Densidade de Distribuição de Probabilidade e Inter-relacionamento de variáveis.	96
6.10	SVM - O hiperplano maximiza a distância entre os vetores suporte, através deles é calculada a classe de pertencimento de uma entrada não-classificada.	97
6.11	SVM- 1×1 - Percurso da estimação de otimização de parâmetros apenas o inverso taxa de aprendizagem (C).	100
6.12	SVM - 1×1 - Densidade de Distribuição de Probabilidade e Inter-relacionamento de variáveis.	100
6.13	Diferenças entre regressão Logística e Linear, a regressão linear varia continuamente com a inclinação da reta e a logística possui um limiar e divide a classificação em antes e depois desse limiar.	101
6.14	Logit- 1×1 - Percurso da estimação de otimização de parâmetros, apenas o inverso da taxa de aprendizagem (C) a ser otimizado.	103
6.15	Logit- 1×1 - Densidade de Distribuição de Probabilidade e Inter-relacionamento de variáveis.	104
6.16	LDA \times PCA - fonte: https://miro.medium.com	105
6.17	LDA - Percurso da estimação de otimização de parâmetro, número de discriminantes lineares.	107
6.18	LDA - Densidade de Distribuição de Probabilidade e Inter-relacionamento de variáveis.	108

6.19	XgBoost	109
6.20	XGBoost - Percurso da estimação de otimização de parâmetros, taxa de aprendizagem (α), aleatoriedade do treinamento, segmentação de árvore, regularização, ponderação do nó, tamanho máximo do nó, número mínimo de ocorrências para iniciar um nó, número de árvores e tamanho da partição usada no treinamento.	111
6.21	XGBoost- Densidade de Distribuição de Probabilidade e Inter-relacionamento de variáveis.	112
7.1	Classificador MLP com características Lofar, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por <i>Kernel</i> em vermelho e eixo à direita.	114
7.2	Classificador MLP com características PCA, histograma com frequência de ocorrência em preto com eixo à esquerda, e estimativa de densidade por <i>Kernel</i> em vermelho e eixo à direita.	115
7.3	KNN-Lofar, histograma com frequência de ocorrência em preto com eixo à esquerda e estimativa de densidade por <i>Kernel</i> em vermelho e eixo à direita.	117
7.4	KNN-PCA	118
7.5	KNN-NLPCA, histograma com frequência de ocorrência em preto com eixo à esquerda e estimativa de densidade por <i>Kernel</i> em vermelho e eixo à direita.	119
7.6	Ridge - Lofar, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por <i>Kernel</i> em vermelho e eixo a direita.	120
7.7	Ridge - PCA, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por <i>Kernel</i> em vermelho e eixo à direita.	121
7.8	Ridge - PCA, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por <i>Kernel</i> em vermelho e eixo a direita.	122
7.9	Classificador SVM com características LOFAR, à esquerda, usa-se a estratégia 1×1 , e à direita a estratégia $1 \times T$, histograma com frequência de ocorrência em preto com eixo à esquerda e estimativa de densidade por <i>Kernel</i> em vermelho e eixo à direita.	123

7.10 Classificador SVM com características PCA, à esquerda, usa-se a estratégia 1×1 , e à direita a estratégia $1 \times T$, histograma com frequência de ocorrência em preto com eixo à esquerda e estimativa de densidade por *Kernel* em vermelho e eixo à direita. 123

7.11 Classificador SVM com características NLPCA, à esquerda, usa-se a estratégia 1×1 , e à direita a estratégia $1 \times T$, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de One more step before you proceed... densidade por *Kernel* em vermelho e eixo à direita. 124

7.12 Classificador Logit com características LOFAR, à esquerda, usa-se a estratégia 1×1 , e à direita a estratégia $1 \times T$, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo a direita. 125

7.13 Classificador Logit com características LOFAR, à esquerda, usa-se a estratégia 1×1 , e à direita a estratégia $1 \times T$ 126

7.14 Classificador Logit com características LOFAR, à esquerda, usa-se a estratégia 1×1 , e à direita a estratégia $1 \times T$, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo à direita. 126

7.15 LDA - Lofar, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo a direita. 127

7.16 LDA - PCA, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo a direita. 128

7.17 LDA - NLPCA,, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo a direita. 129

7.18 XGBoost - Lofar, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo a direita. 130

7.19 XGBoost - PCA, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo a direita. 131

7.20 XGBoost - NLPCA, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por <i>Kernel</i> em vermelho e eixo a direita.	132
7.21 Compilação dos resultados de todas as máquinas apresentadas para LO-FAR, PCA e NLPCA.	134

Lista de Tabelas

2.1	Unidades do sistema internacional para várias quantidades de ondas acústicas e propriedades pertinentes da água no oceano.	21
6.1	Parâmetros otimizados obtidos através de uma otimização bayesiana para o algoritmo MLP.	82
6.2	Parâmetros otimizados obtidos através de otimização Bayesiana para o algoritmo MLP.	84
6.3	Parâmetros obtidos no KNN.	89
6.4	Parâmetros obtidos no Ridge	94
6.5	Parâmetros obtidos no SVM.	99
6.6	Parâmetros obtidos Regressão Logística.	103
6.7	Parâmetros obtidos no LDA.	106
6.8	Parâmetros obtidos no XGBoost.	110
7.1	Compilação dos resultados apresentados nas seções anteriores, com acurácia e desvio padrão (std).	134

Lista de Abreviaturas e Siglas

AI	Inteligência Artificial	39
AoA	ângulo de chegada	33
AUVs	submarinos autônomos	32
CASOP	Centro de Apoio a Sistemas Operativos em Arraial do Cabo	42
DI	Índice de diretividade	16
DNN	Rede Neural profunda	32
DoA	<i>Direction of Arrival</i>	5
DT	Limiar de detecção	18
DFT	Transformada discreta de Fourier	41
FFT	Transformada rápida de Fourier	38
LDA	Análise Discriminante Linear	104
LOFAR	<i>Low Frequency Analysis and Recording</i>	viii
MAEr	Erro médio Absoluto	76
MAGE	Medida de Apoio a Guerra Eletrônica	30
MAPE	Percentual do erro médio absoluto	76
MB	Marinha do Brasil	2
MBES	Ecobatímetro multi-feixe	32
MCA	Motor de Combustão Auxiliar	43
MCP	Motor de Combustão Principal	43
MIMO	Entrada múltipla e saída múltipla	34
ML	Machine Learning	10
MLP	<i>Multi Layer Perceptron</i>	80
MSE	Erro médio quadrático	76
MST	Média da soma total dos quadrados	76

NLPCA	Análise de Componentes Principais não-linear	69
ONU	Organização das Nações Unidas	2
PCA	Análise de Componentes Principais	69
PDS	Processamento Digital de Sinais	7
PIM	Posição e Intenção de Movimento	7
PQM	Pressão quadrática média	22
RMSE	Raiz do erro médio quadrático	76
RaDaR	Radio Detection and Ranging	16
SAS	Radar de Abertura Sintética	vii
SL	lóbulo lateral	35
SL	Nível do Sinal	35
SNR	Razão Sinal-ruído	15
SOFAR	Canal Sonoro Profundo	18
SoNaR	<i>Sound Navigation and Ranging</i>	vii
SSP	Perfil de velocidade do som	16
SSS	Sonar de Varredura Lateral	vii
SVM	<i>Support vector Machine</i>	74
TL	Perda de transmissão ou propagação	16
TOM	Teatro de Operação Marítimo	5
TPSW	<i>Two-Pass split window</i>	39
URA	Unidade de Resfriamento de Água	43

Sumário

1	Introdução	1
1.1	Formulação e Delimitação do Problema	4
1.2	Justificativa e Relevância	6
1.3	Objetivos da Dissertação	7
1.3.1	Objetivos Específicos	8
1.4	Metodologia	8
1.5	Estrutura do Documento	9
2	SoNaR	11
2.1	O SoNaR e Sua História	11
2.2	Descrição Analítica do SoNaR	12
2.2.1	SoNaR Passivo	15
2.2.2	SoNaR Ativo	17
2.2.3	Excesso de Sinal	17
2.3	Propagação do Som na Água	18
2.3.1	Campo Acústico e Potência	21
2.3.2	Perdas por Absorção	24
2.3.3	Atenuação	25
2.4	Classificação SoNaR	27
2.4.1	Guerra Eletrônica	27
3	Diferentes tipos e usos de SoNaR e realizações de classificação SoNaR	

passiva na atualidade	31
3.1 SoNaR	31
3.1.1 SoNaR Sidescan	32
3.1.2 SoNaR de Abertura Sintética	33
3.1.3 SoNaR MIMO	34
3.2 Classificação Passiva de Contatos SoNaR	36
3.3 Comunicação Submarina	37
4 Banco de Corridas e Pré-Processamento de sinais acústicos	38
4.1 Metodologia de Análise dos Dados de Entrada	39
4.1.1 Aplicação da Metodologia de Análise dos Dados de Entrada	42
4.2 Banco de Sinais de Corridas	42
4.2.1 Dataset	43
4.2.2 Implementação do Algoritmo de Balanceamento do Banco de Corridas	44
4.3 Pré-Processamento	46
4.3.1 LOFAR	48
4.3.2 Pré-Processamento de Sinais Acústicos	49
4.3.3 TPSW	51
4.3.4 Sobre o Desempenho do TPSW	53
4.4 DataFrame - TPSW	56
5 Processamento de Sinais acústicos	57
5.1 Aspectos Metodológicos	60
5.1.1 Ajuste e Avaliação de Modelos de ML	60
5.1.2 Particionamento de Dados: Treino, Validação e Teste	63
5.1.2.1 <i>K-fold Cross Validation</i>	64
5.1.3 Ajuste de Hiperparâmetros	65

5.1.3.1	Otimização Matemática	66
5.1.3.2	Otimização Bayesiana	67
5.2	Uso de Classificadores Binários para Problemas Multiclasses	68
5.3	Redução de Atributos	69
5.3.1	Comprimento do Vetor de Características do Lofargrama	69
5.3.2	PCA	70
5.3.3	NLPCA	73
5.4	Medidas de Desempenho	75
5.4.1	R^2 Score	76
5.4.2	Acurácia	77
6	Resultados	79
6.1	MLP	79
6.1.1	MLP - Parâmetros Não-Otimizados	81
6.1.2	MLP - Otimização Bayesiana	82
6.1.3	MLP - Resultados da Otimização Bayesiana	85
6.2	KNN	87
6.2.1	KNN - Parâmetros Não-Otimizados	88
6.2.2	KNN - Otimização Bayesiana	88
6.2.3	KNN - Resultado da Otimização Bayesiana	89
6.3	Ridge	92
6.3.1	Ridge - Otimização Bayesiana	93
6.3.2	Ridge - Resultado da Otimização Bayesiana	94
6.4	SVM	96
6.4.1	SVM- Parâmetros Não-Otimizados	97
6.4.2	SVM- Otimização Bayesiana	98
6.4.3	SVM- Resultado da Otimização Bayesiana	99

6.5	Regressão Logística	101
6.5.1	Logit - Parâmetros Não-Otimizados	101
6.5.2	Logit - Otimização Bayesiana	102
6.5.3	Logit - Resultado da Otimização Bayesiana	103
6.6	LDA	104
6.6.1	LDA - Parâmetros Não-Otimizados	105
6.6.2	LDA - Otimização Bayesiana	106
6.6.3	LDA - Resultado da Otimização Bayesiana	107
6.7	XgBoost	108
6.7.1	XgBoost- Parâmetros Não-Otimizados	109
6.7.2	XgBoost- Otimização Bayesiana	109
6.7.2.1	XgBoost- Resultado da Otimização Bayesiana	111
7	Resultados	113
7.1	MLP	113
7.1.1	Classificador MLP e Características Lofar	113
7.1.2	MLP - Resultados PCA	114
7.1.2.1	Resultados NLPCA	115
7.2	KNN	117
7.2.1	KNN - Resultados em Lofar	117
7.2.2	KNN - Resultados com Pré-Processamento PCA	118
7.2.3	KNN - Resultados NLPCA	118
7.3	Ridge	120
7.3.1	Ridge - Resultados Lofar	120
7.3.2	Ridge - Resultados PCA	121
7.3.3	Ridge - Resultados NLPCA	121
7.4	SVM	122

7.4.1	SVM- Resultados no Lofar	122
7.4.2	SVM- Resultados com Pré-Processamento PCA	123
7.4.3	SVM- Resultados NLPCA	124
7.5	Regressão Logística	124
7.5.1	Logit - Resultados em Lofar	124
7.5.2	Logit - Resultados PCA	125
7.5.3	Logit - Resultados NLPCA	126
7.6	LDA	127
7.6.1	LDA - Resultados Lofar	127
7.6.2	LDA - Resultados PCA	128
7.6.3	LDA - Resultados NLPCA	128
7.7	XgBoost	129
7.7.0.1	XgBoost- Resultados Lofar	129
7.7.0.2	XgBoost- Resultados PCA	130
7.7.0.3	XgBoost- Resultados NLPCA	131
7.8	Comparação dos Modelos de Aprendizado de Máquina para Classificação .	133
8	Conclusão	136
	Referências	141

Capítulo 1

Introdução

Este trabalho explora o uso da inteligência artificial, e, em particular, a etapa de aprendizado de máquinas, para reconhecer padrões em sinais sonoros. O objetivo é criar um protótipo de um sistema que possa ser utilizado a bordo de um submarino ou navio para auxiliar o operador sonar. Para construir tal sistema, serão estudados tanto o campo da inteligência artificial como o campo de processamento de sinais, uma vez que o sistema terá que utilizar técnicas em ambos os campos.

Neste capítulo, serão discutidos os antecedentes e a motivação para isso, seguidos de uma apresentação dos objetivos deste projeto, com uma breve introdução à abordagem científica adotada. Finalmente, a estrutura do projeto restante será delineada.

Um operador de sonar humano tem um grau de incerteza quanto à classificação que ele realiza. A solução para esse problema é um sistema de apoio que utilize dados acústicos brutos como entrada, seguido de processamento e extração de características, e finalmente, rotule uma classe, isto é, um tipo de navio.

A classificação passiva de sonar contribui para os sistemas de alerta precoce, proporcionando um tempo vital para resposta em caso de ameaças potenciais. Ao classificar com precisão um contato de interesse, permite-se planejar as medidas adequadas para mitigá-lo, em caso de ameaça.

Muitas nações possuem ativos estratégicos localizados debaixo d'água, como cabos submarinos ou instalações de energia *offshore*. A classificação sonar passiva ajuda a proteger esses ativos contra sabotagem ou acessos não-autorizados, identificando e monitorando ameaças potenciais em suas proximidades.

Além disso, o sonar passivo pode ser usado para fins de coleta de inteligência, moni-

torando as atividades de embarcações e submarinos estrangeiros. A classificação precisa dessas embarcações fornece informações valiosas sobre as capacidades e intenções de outras nações, contribuindo para a estratégia global de segurança nacional.

Este tema é relevante para a segurança nacional, pois permite a detecção, rastreamento e classificação de ameaças subaquáticas, salvaguardando assim os interesses marítimos e os ativos estratégicos. Nacionalmente falando, este tema está alinhado com o Plano Estratégico da Marinha (PEM 2040), publicado em setembro de 2020 [9].

Um estado nacional, em sua política de estado, assim entendida como os meios que um estado nacional encontra para alcançar objetivos de longo prazo, possui objetivos que, idealmente, se desdobram em políticas de governo. A capacidade de alcançar esses objetivos é conhecida como poder nacional. Esse poder se manifesta em cinco expressões: a política, a econômica, a psicossocial, a militar e a científico-tecnológica [10].

O Almirante da Marinha estadunidense Alfred Thayer Mahan cunhou o termo *poder marítimo* em seu livro [11]. Em sentido amplo, tal poder inclui não apenas a força militar que governa o mar ou qualquer parte dele pela força das armas, mas também o comércio pacífico e transporte marítimo. A parcela militar do *poder marítimo* é o *poder naval*, exercido em nosso país pela Marinha do Brasil (MB).

A soberania de um país pode ser traduzida por sua efetiva presença no território. Nosso território marítimo é conhecido como Amazônia Azul, que é uma região que compreende a superfície do mar, águas acima ao leito do mar, o solo e subsolo marinhos contidos na extensão atlântica que se projeta a partir do litoral até o limite exterior da Plataforma Continental Brasileira. A Amazônia azul, representada na figura 1.1, possui 3,6 milhões de quilômetros quadrados, com possibilidade de ampliação para até 5,7 milhões de quilômetros quadrados a partir do Estudo do Limite Exterior da Plataforma Continental Brasileira encaminhado à Organização das Nações Unidas (ONU).

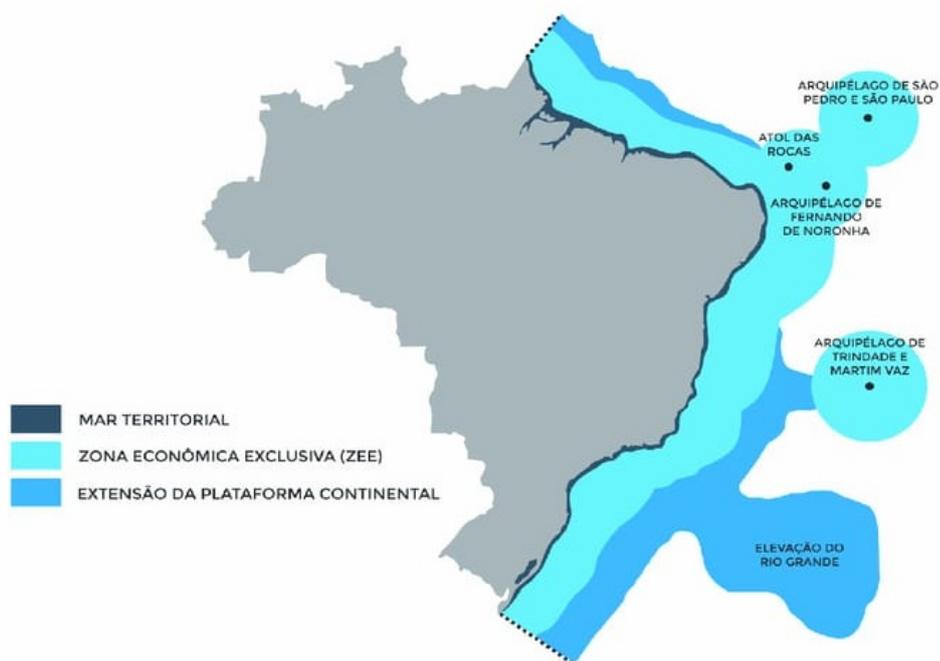


Figura 1.1: Amazônia Azul inclui o Mar Territorial, a Plataforma continental, a Zona Econômica Exclusiva e as águas interiores brasileiras[1].

O tamanho da Amazônia Azul, conforme representado na Figura 1.1, ilustra o desafio de patrulhar e o tamanho da tarefa da marinha Nacional, ressalta-se, por exemplo, que somente o porto de Santos recebeu um fluxo de navios de 5202 atracações em 2022 [12].

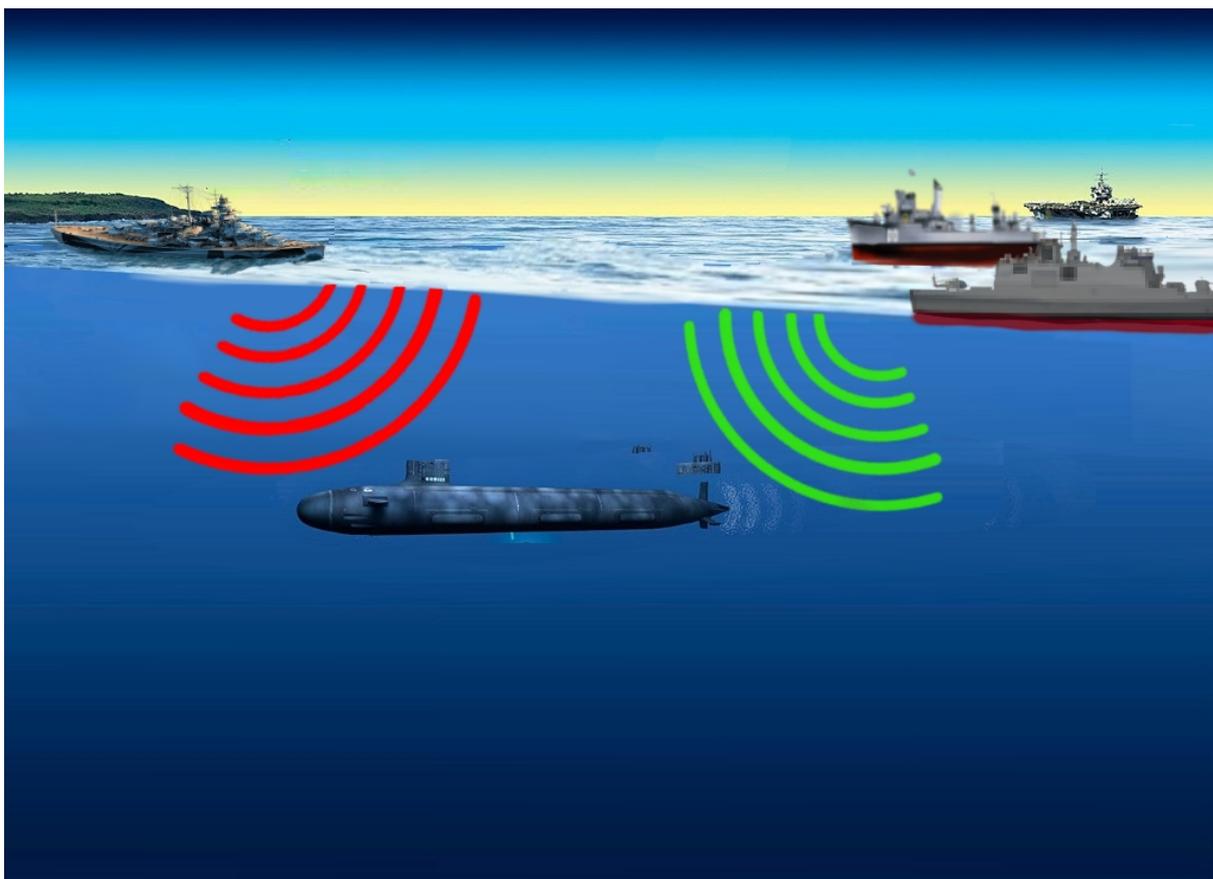


Figura 1.2: Identificação de assinatura acústica de contato de interesse em ambiente ruidoso.

1.1 Formulação e Delimitação do Problema

Nos últimos anos, as preocupações com a segurança marítima aumentaram devido a proliferação de atividades ilegais, como o contrabando, a pirataria e o tráfico. Um aspecto crucial da segurança marítima é a capacidade de identificar com precisão os navios que operam numa determinada área. Os métodos tradicionais de identificação de embarcações, como a inspeção visual e os sistemas de radar, têm limitações, especialmente em condições meteorológicas adversas ou à noite. As assinaturas acústicas são padrões sonoros únicos emitidos pelas embarcações e oferecem uma alternativa promissora para fins de identificação, pois cada embarcação possui a sua assinatura.

A MB realiza constante vigilância em toda a Amazônia Azul, interrogando contatos via rádio, verificando a derrota, isto é, o curso do navio interrogado, provendo constantes treinamentos e inspeções. Normalmente, um navio, ao adentrar num porto, passa por um canal que é dragado para garantir a profundidade de calado necessária, ou seja, a profundidade abaixo d'água do navio, tal que, esse navio não arraste o fundo. Nesse

ambiente é possível a instalação de hidrofones sub-aquáticos para captar a assinatura acústica do navio e identificá-lo.

Outra forma de obter assinaturas acústicas ocorre no caso de contato com outras marinhas, como na figura 1.2. Nesta aplicação, um submarino participante de um Teatro de Operação Marítimo (TOM), mesmo recebendo um somatório de perturbações acústicas em seu SoNaR, deve perceber a existência de uma ameaça ou contato desconhecido. Para isso, consegue distinguir as fontes, através de suas *Direction of Arrivals* (DoAs) e, processando os sinais, consegue classificar quais as ameaças e incerteza à sua segurança.

Tanto a instalação dos hidrofones nos portos quanto a aquisição de sinais SoNaR pelo submarino alimentariam um banco de dados, permitindo que se obtenha um registro da embarcação e que se possa consultar esse registro no futuro para fins de comparação com um sinal acústico recebido.

Formula-se o problema como a identificação de embarcações por assinatura acústica explorando técnicas para classificar com precisão e eficiência, diferentes tipos de embarcações com base em suas características acústicas únicas. Para tal, a pesquisa visa enfrentar os seguintes desafios:

- Variabilidade das assinaturas acústicas: As assinaturas acústicas podem variar significativamente dependendo de fatores como tipo de embarcação, tamanho, características do motor e condições ambientais. Essa variabilidade representa um desafio no desenvolvimento de algoritmos de identificação robustos, capazes de distinguir com precisão entre diferentes tipos e condições de embarcações;
- Interferência de Ruído: Os sinais acústicos no ambiente marítimo são frequentemente contaminados por ruído de fundo de fontes naturais (por exemplo, ondas ou vida marinha) e atividades humanas (por exemplo, outras embarcações ou instalações *offshore*). Essa interferência de ruído obscurece as características distintivas das assinaturas acústicas, comprometendo a precisão dos algoritmos de classificação.
- Coleta de dados: Adquirir conjuntos de dados abrangentes de assinatura acústica contendo vários tipos de embarcações e condições operacionais é escassa. A disponibilidade limitada de dados dificulta o desenvolvimento e a validação de modelos de identificação eficazes.
- Requisitos de processamento em tempo real: Em cenários de segurança marítima, a identificação oportuna de embarcações é crucial para a tomada de decisões e ações

de resposta. O desenvolvimento de algoritmos de processamento de assinatura acústica em tempo real, capazes de analisar rapidamente os fluxos de dados recebidos, apresenta desafios técnicos em termos de eficiência e precisão computacional.

A limitação deste problema reside na complexidade e variabilidade das assinaturas acústicas produzidas por diferentes tipos de embarcações, para garantir a viabilidade e o foco da pesquisa, são estabelecidos certos limites:

- **Âmbito:** Esta dissertação centra-se principalmente na identificação de embarcações com base nas suas assinaturas acústicas no contexto de aplicações de segurança marítima. Embora campos relacionados, como detecção de mamíferos marinhos ou sistemas de comunicação subaquática, também utilizem sinais acústicos, esses campos não estão dentro do escopo deste estudo;
- **Foco Tecnológico:** O estudo se concentra no desenvolvimento e avaliação de algoritmos de processamento de sinais e técnicas de aprendizado de máquina para análise e classificação de assinaturas acústicas. Embora as considerações sobre o hardware do sensor sejam importantes, elas não são o foco principal desta pesquisa;
- **Condições Ambientais:** A pesquisa considerará o impacto dos fatores ambientais na análise da assinatura acústica, com ênfase particular nas técnicas de mitigação de ruído. Contudo, estudos detalhados sobre fenômenos oceanográficos ou modelos de propagação acústica estão além do escopo desta dissertação.

Ao delinear a formulação e delimitação do problema, esta dissertação aborda os desafios associados à utilização de assinaturas acústicas para identificação de embarcações e contribui para o avanço de tecnologias de segurança marítima. Portanto, essa pesquisa é relevante para o incremento nacional do poder naval da marinha brasileira e oportunidade de melhoria para fiscalização da Amazônia Azul.

1.2 Justificativa e Relevância

Esta dissertação justifica-se por dois fatores principais que conduziram à escolha do tema. O primeiro deles é que a classificação de contatos de interesse, assim entendido como qualquer meio naval, cuja identificação produza conhecimento a nível tático ou estratégico, através da assinatura acústica. Outro fator é a criação de um banco de dados de

aquisição automática de características acústicas de meios navais, em outras palavras, as características são obtidas sem a interferência humana.

O primeiro fator é de relevância para o setor militar naval, tanto pelo conhecimento da teoria de processamento de sinais acústicos quanto pelo conhecimento técnico a ser adquirido como um fator estratégico pela MB. O processamento da assinatura acústica através de características extraídas do sinal permite o uso imediato desta informação mas também facilitar o uso deste conhecimento para detecção de características da assinatura eletromagnética em estudos futuros.

O segundo fator é que a pesquisa pode contribuir para a efetiva criação de um banco de dados acústicos, com aquisição de sinais que são automaticamente classificados e inseridos nestes banco. A pesquisa visa motivar a instalação subaquática destes transdutores ao longo de nossa costa proporcionando identificação de contatos em um possível conflito bélico.

Conhecendo-se a dificuldade de identificar com o ouvido humano através de um operador SoNaR, um contato somente pela assinatura acústica, classificar um contato é uma tarefa difícil. Obter a classificação de contatos automaticamente feito através do Processamento Digital de Sinais (PDS) destes sinais, adquiridos em tempos de paz, é de grande interesse para segurança nacional e justificativa para tal estudo.

A identificação acústica de navios é de extrema relevância em várias áreas como: na segurança marítima, visando identificar navios e rastrear suas posições, movimentos e atividades; na defesa nacional, prevenindo incidentes e permitindo uma resposta rápida em caso de ameaças potenciais à segurança nacional. É fundamental ainda na aplicação da lei e fiscalização, a identificação acústica pode ser instrumentalizada para monitorar atividades ilegais, como pesca não-autorizada, contrabando, tráfico de drogas e outros crimes marítimos; e, no estudo da fauna marinha, onde a identificação acústica de navios é usada em estudos oceanográficos e de comportamento animal marinho.

1.3 Objetivos da Dissertação

Uma vez recebida uma assinatura acústica, a primeira pergunta é se essa assinatura é proveniente de uma força amiga ou inimiga e, após isso, a quem se deve aquela assinatura. O primeiro problema é resolvível pois normalmente um submarino ou um navio conhece a mensagem Posição e Intenção de Movimento (PIM) de outros navios, então se dispõe das posições e estimativas de movimento de cada navio. Pode-se assim fazer projeções

para saber se é provável haver uma força amiga naquela região. Caso não haja, deve se entender a força detectada como inimiga.

O objetivo militar desta dissertação é analisar e apresentar abordagens, técnicas e métodos para identificar e classificar eficientemente contatos detectados por sistemas de SoNaR. O foco está na aplicação de tecnologias de SoNaR para identificar diferentes tipos de alvos, como navios e submarinos.

O Objetivo acadêmico é conduzir uma análise comparativa de sete algoritmos de aprendizado de máquina para classificação sonar passiva, com o objetivo de avaliar seu desempenho, a fim de identificar o algoritmo mais adequado para implantação no mundo real no reconhecimento de alvos através da assinatura acústica.

1.3.1 Objetivos Específicos

Avaliar Algoritmos e Técnicas: avaliar o desempenho de algoritmos de processamento de sinais, aprendizado de máquina e outras técnicas

Extrair características: distintivas nos sinais de SoNaR e classificar diferentes tipos de contatos.

Testar Modelos de Classificação: criar modelos de classificação que utilizem características acústicas extraídas de sinais de SoNaR, como assinaturas, para diferenciar e classificar tipos de contatos subaquáticos.

Avaliar o Desempenho: realizar testes e simulações para avaliar a precisão e confiabilidade dos métodos de extração de características e identificação pesquisados e propostos, considerando variáveis como ruído ambiental e condições da água.

Aplicar em sinais reais: discutir as implicações práticas das técnicas de identificação de contatos SoNaR em contextos reais, como operações navais, pesquisa científica subaquática, monitoramento ambiental, dentre outros.

1.4 Metodologia

Para alcançar os objetivos supracitados utiliza-se um banco de dados oriundo de gravações de medições de ruído irradiado realizadas na raia acústica Centro de Análise de Sistemas Operacionais (CASOP) de Arraial do Cabo, Brasil. Nele constam assinaturas acústicas

de doze navios, em diferentes condições e diferentes estados do mar. Assim o algoritmo de classificação surge com o desafio de identificar o navio em meio a diferentes ambientes.

A finalidade deste banco é coletar as assinaturas acústicas de meios militares para identificação e classificação dos meios. O banco de corridas possui 6,06 GB de dados e 4770 arquivos divididos em 709 pastas contendo 12 navios. As assinaturas acústicas são gravadas em formato *wav* ou *mp3*, por hidrofones. Os dados são coletados em dois tipos de corridas estáticas, onde somente as principais máquinas do navio estão funcionando e ele não está se movendo; e dinâmica onde o navio percorre um caminho enquanto é gravado. Estes dados são de uso reservado e por isso o nome dos navios é suprimido nesta dissertação.

- A linguagem de programação escolhida foi Python, isso porque é de alto nível e oferece uma ampla gama de recursos, desde análise e visualização de dados até desenvolvimento, automação web e prototipagem. Além disso, está entre as linguagens de programação mais utilizadas na atualidade.

Para validar os resultados são utilizadas métricas como R^2 score, validação cruzada, *k-fold* e a repetibilidade dos resultados é testada por mais de 30 vezes buscando uma confiabilidade média e não um resultado particular.

Existem vários estudos realizados neste sentido, cita-se como exemplo [13, 14, 15], porém, em geral, trabalham com uma quantidade menor de classes de navios ou com classes completamente diferentes umas das outras. No banco de dados desta dissertação há mais de uma ocorrência de dois navios do mesmo tipo tornando mais fácil o algoritmo se confundir e tornando mais exitoso ainda que possa obter a mesma acurácia que outros estudos que utilizaram bancos de dados bem mais simples.

1.5 Estrutura do Documento

A dissertação está dividida em 5 capítulos. O capítulo 1 procura familiarizar o leitor com alguns dilemas militares navais e elucidar a importância do tema não só para MB mas para nação como um todo. Situa-se essa pesquisa na área militar naval.

No capítulo 3, se encontra uma pesquisa sobre os usos que a tecnologia SoNaR adquiriu com o passar do tempo. Avaliando-se aproximadamente quinze artigos atuais, mostra-se como diferentes tecnologias foram empregadas tornando o SoNaR um sistema muito mais complexo e seu uso fora do meio militar para pesquisas dos mais diversos tipos.

No capítulo 2, há uma apresentação geral sobre a tecnologia SoNaR, com os livros mais utilizados na área e com alguns livros mais recentes. O intuito deste texto é perceber as vantagens e desvantagens do SoNaR, entender descritivamente a onda sonora e como o som se comporta no meio subaquático. É trazida uma discussão histórica sobre o SoNaR e as equações SoNaR, tanto ativas quanto passivas.

O capítulo 4 traz uma metodologia de pré-processamento desenvolvida por [16] que tem sido largamente empregada por sua eficácia. Apresenta as etapas necessárias para se tornar o sinal utilizável, demonstrando graficamente e analiticamente através de equações o tratamento que o sinal sofre em cada etapa.

O capítulo 5 apresenta as técnicas de Análise de Componentes Principais, Análise não-Linear de Componentes Principais. Resolução de problemas multi-classe em algoritmos de Machine Learning (ML) binários através de um contra um e um contra todos. Além de validação cruzada e *k-fold* e as métricas que definem o ajuste de cada algoritmo.

O capítulo 6 apresenta sete algoritmos de ML, os hiperparâmetros envolvidos em seu ajuste e o uso a otimização Bayesiana para tal feito e os resultados de tal implementação.

Por fim, o Capítulo 7 conclui a dissertação, revisitando os principais pontos discutidos no trabalho, destacando as principais contribuições e estabelecendo os trabalhos futuros.

Capítulo 2

SoNaR

2.1 O SoNaR e Sua História

A ideia da utilização de ondas mecânicas sonoras no meio subaquático para reconhecimento de obstáculos e adversários é antiga. Em 1490, Leonardo da Vinci utilizou um tubo inserido na água para detectar contatos quais sejam navios ou animais[17]. Desde a tragédia do RMS *Titanic*, em 1912, diversos inventores depositaram registros de patentes de dispositivos anteriores ao SoNaR [18].

Um SoNaR se baseia na captura e interpretação de informações contidas em ondas acústicas, provenientes de alvos de interesse. Assim, como um golfinho que se localiza pelo som [19], um submarino se orienta pelo SoNaR. Um SoNaR pode ser passivo ou ativo. Um SoNaR passivo apenas escuta os sons emanados pelo meio em que coabita. O SoNaR ativo transmite uma onda sonora, a qual, ao encontrar um obstáculo, se choca com ele e é, em parte, refletida.

Uma assinatura acústica é obtida por meio de dispositivos, SoNaRes ou hidrofones instalados no assoalho submarino, que fornecem o registro de áudio da atividade do campo acústico. Os sinais acústicos subaquáticos são usados em muitas aplicações, como comunicações acústicas, detecção e localização de objetos superficiais e subterrâneos, sondas de profundidade, perfiladores sub-fundos e outras aplicações. A identificação e reconhecimento de sinais tornaram-se uma questão central.

Na arte da guerra, o submarino é um tipo de navio que possui como principal característica a dissuasão, isto é, demover o oponente de algum intento. Durante a Segunda Guerra Mundial, após o rompimento das relações diplomáticas do Brasil com o Eixo, a Alemanha enviou à costa brasileira uma frota de submarinos, os quais afundaram cinco

navios mercantes, que foram Bapendi, Araraquara, Aníbal Benévolo, Itagiba e Arará, totalizando 607 brasileiros mortos. Com a escolta da Marinha Brasileira aos nossos navios mercantes, esses ataques cessaram [20].

Esse episódio histórico denota que o submarino é uma arma de dissuasão e a ocultação é a sua melhor tática. Informalmente, os submarinistas dizem entre si que só existem dois tipos de navios: os submarinos e os alvos. Por mais que essa frase tenha um fundo de verdade, após atacar um navio, um submarino denuncia sua posição. Isso explica o motivo do sucesso da Marinha brasileira em agosto de 1942, pois os submarinos abdicaram dos ataques à nossa frota mercante com a escolta da marinha brasileira.

A segurança e a vigilância marítima dependem fortemente da identificação e classificação precisas dos navios que operam em vários domínios marítimos. A tecnologia SoNaR, com a sua capacidade de detectar e analisar sinais acústicos subaquáticos, apresenta uma excelente abordagem para a classificação de embarcações, com base nas suas assinaturas acústicas únicas e, por isso, é tema deste capítulo.

2.2 Descrição Analítica do SoNaR

SoNaR é o acrônimo de *Sound Navigation and Ranging*, quer dizer, em tradução livre, navegação e mensuração de distância pelo som. Atualmente, utiliza-se a tecnologia de SoNaR para detectar alvos submarinos. A sua motivação inicial, devido ao naufrágio do *RMS Titanic* ao colidir com um iceberg em 14 de Abril de 1912, era a de funcionar como uma ferramenta de auxílio à navegação. Lewis Richardson [21] patenteou um sistema de ecolocalização para aeronaves no mesmo mês do naufrágio e um mês depois um sistema submarino. Reginald Fessenden patenteou um transdutor eletromagnético em 1913 e em 1914 demonstrou a detecção de um iceberg a uma distância de aproximadamente 4 km, esse sistema ficou conhecido como o *Oscilador Fessenden*[21].

Além de ser uma ferramenta de auxílio à navegação, são diversas as aplicações do SoNaR. O SoNaR pode ser utilizado também na pesca, para detectar cardumes, na determinação do mapeamento do assoalho submarino e para determinação da profundidade quando em portos, através do ecobatímetro. Seu diagrama de blocos está representado na figura 2.1.

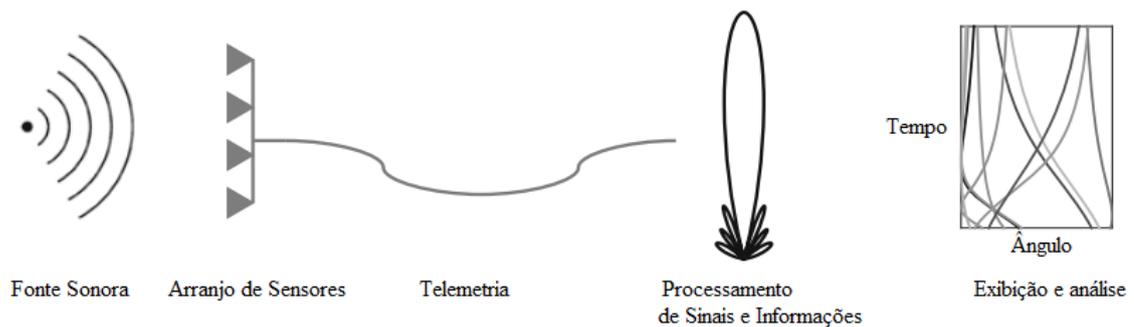


Figura 2.1: Componentes de um sistema SoNaR. Adaptado de [2].

Os componentes básicos de um sistema de SoNaR, conforme é ilustrado na Fig. 2.1, incluem: fonte sonora, arranjo de sensores, telemetria, processamento de sinais e de informações, e equipamentos de exibição e análise.

Um sistema SoNaR é categorizado em dois grandes grupos, ativo ou passivo. Para sistemas SoNaR do tipo ativo, a fonte sonora é um transdutor, comumente chamada de hidrofone, que é um altofalante subaquático. O hidrofone emite um pulso sonoro modulado temporalmente, para facilitar seu reconhecimento a partir de seu eco.

Em sistemas de SoNaR passivos, a *fonte sonora* adquirida é oriunda do meio que está sendo explorado. Não há um componente físico no sistema de SoNaR passivo que emita uma onda sonora. Ao monitorar o ambiente passivamente, o submarino não interfere no meio, apenas o monitora, mantendo assim sua discricção.

Uma onda sonora apresenta um comprimento de onda $\lambda = c_w/f$, onde c_w , é a velocidade de propagação da onda de som na água, que é $c_w \approx 1500$ m/s, e f é a frequência central da onda. Observe-se que, para uma frequência de 1 kHz, $\lambda = 1500/1000$, a detecção de um navio fica bastante prejudicada, pois qualquer peixe com pouco mais de 1,5 m aparece na apresentação do *display*, poluindo a imagem e tornando mais difícil a detecção de um navio.

Quando um hidrofone é pequeno, isto é, cerca de 10% do comprimento de onda do sinal, relativamente ao comprimento de onda (por exemplo, um hidrofone com diâmetro de 2 cm é pequeno para uma transmissão em 10 kHz, que possui um comprimento de onda de 15 cm), sua resposta é quase constante em frequência e em relação ao ângulo, possuindo assim uma resposta omnidirecional[2].

Os hidrofones são normalmente organizados em grupos chamados de matrizes de sensores ou de *arranjos de sensores*. A aquisição espacial das fontes acústicas depende de uma

separação entre os hidrofones, pois, conforme diferentes hidrofones recebem um mesmo sinal acústico com diferença de fase e de amplitude, pode-se estimar a sua DoA, ou, em outros termos, localizar a direção de uma fonte sonora específica. O uso de matrizes de sensores também melhora o desempenho do SoNaR quando os sinais do objeto de interesse são estáveis, em magnitude maior que a do ruído ambiente, ou quando há uma interferência no conjunto do sensor, vindo de uma direção diferente do sinal do objeto de interesse[2].

A configuração física dos sensores na matriz depende tanto da aplicação, ou não, de sensoriamento remoto, quanto das restrições físicas da plataforma do SoNaR. Por exemplo, matrizes rebocadas, conforme a figura 2.2, que viajam atrás de navios, são geralmente configuradas com sensores igualmente espaçados em uma linha. Rebocar a matriz com o navio, a mantém longe do ruído gerado pelo navio, o que melhora o desempenho do sensor remoto, pois esse é ligado ao navio através de um fio, mas cria um problema adicional, pois os sinais do sensor precisam ser transmitidos para uma unidade de processamento no navio [2].

O processo combinado de medir automaticamente a pressão acústica e comunicar os sinais a um processador fisicamente deslocado é chamado de *telemetria* e é uma parte comum dos sistemas de sensoriamento remoto, conforme apresentado na figura 2.1.

Os sinais elétricos recebidos são amplificados e filtrados para remover ruído e melhorar a qualidade do sinal. Isso é seguido de uma conversão analógico-digital para possibilitar processamento digital adicional onde técnicas digitais são aplicadas aos sinais recebidos, através de algoritmos para processar, analisar e interpretar os sinais. As técnicas PDS envolvem filtragem, formação de feixe, processamento Doppler e análise espectral, entre outros.

A *exibição e análise*, objetiva a interpretação dos dados, uma vez processados os sinais, os dados são interpretados para extrair informações sobre os objetos detectados. Determinam a distância, tamanho, forma e movimento dos alvos subaquáticos.

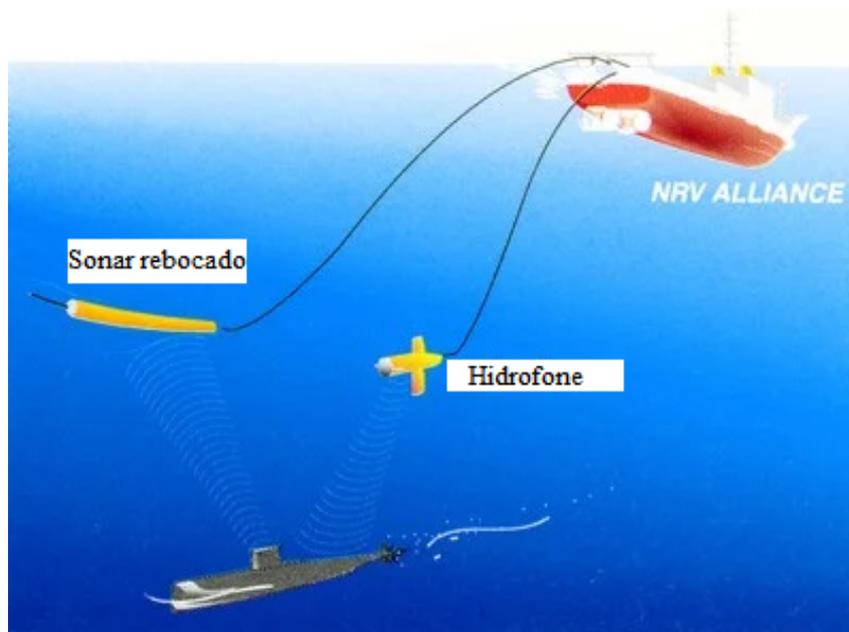


Figura 2.2: SoNaR rebocado que é uma aplicação de sensoriamento remoto.

2.2.1 SoNaR Passivo

Num SoNaR passivo, a razão sinal-ruído, quer dizer, a razão entre a potência do sinal e a do ruído, é dada por:

$$\frac{S}{N} = \frac{S}{TL} \times \frac{AG}{N} \quad (2.1)$$

Notadamente, na equação (2.1), S é a potência do sinal de interesse, recebido passivamente juntamente com N , que representa o ruído total. Um arranjo de sensores bem diretivo consegue trazer algum AG , ganho devido ao arranjo de sensores, e diminuindo esse sinal de interesse estão as TL , perdas de propagação que ocorrem por sinal ter que se deslocar da fonte até o arranjo de hidrofones. O termo AG não é o ideal uma vez que ganho é um processo ativo que se realiza no sinal de interesse. Tanto a equação do SoNaR Passivo, quanto a do Ativo, são comumente apresentadas da forma descrita acima e alguns termos nessas equações não são precisos em termos de processamento de sinal, mas são mantidos no texto em deferência aos precursores no estudo de SoNaR.

É comum usar a definição em decibel (dB) [22] da Razão Sinal-ruído (SNR). Um decibel é a décima parte do Bell. É uma unidade de medida adimensional, que compara a intensidade de um sinal a uma referência, conforme definido na equação (2.2). Recebe esse nome em homenagem ao físico Alexander Graham Bell.

$$Q_P = \log \frac{P_1}{P_2} B = 10 \log \frac{P_1}{P_2} dB. \quad (2.2)$$

Utiliza-se a equação (2.1) em:

$$SNR = S - TL - N + G[\text{dB}], \quad (2.3)$$

na qual S é a potência da onda acústica proveniente do alvo a ser identificado. Suas características dependem de fatores como o tipo de alvo, o modo de operação, a velocidade, a profundidade, sua aceleração do alvo, a frequência das ondas produzidas e, ainda, da posição relativa entre o alvo e o hidrofone. Este último fator é mitigado pelo ganho G .

A Perda de transmissão ou propagação (TL) inclui efeitos do espalhamento de energia, perdas de transmissão por absorção ou espalhamento, no percurso até o hidrofone, e interação com fronteiras, como o assoalho submarino, diferentes zonas de temperatura d'água e a superfície do mar. Esse termo é altamente dependente de parâmetros do ambiente, como velocidade do som versus profundidade, Perfil de velocidade do som (SSP), absorção/dispersão de volume, perda de fundo, perda de superfície, frequência e profundidades de transmissor, alvo e receptor.

É notório que o SoNaR e o Radio Detection and Ranging (RaDaR) possuem semelhanças, ambos se orientam através de um sinal recebido de uma onda, seja ela eletromagnética ou acústica visando detectar, identificar e classificar algo, um através de som e o outro através de ondas eletromagnéticas. No entanto, a interação com fronteiras torna o SoNaR muito mais difícil de prever e operar em comparação com o radar, pois há grande espalhamento e reverberação, ou ainda dutos sonoros submarinos [23].

Por sua vez, o ruído (\mathbf{N}) engloba tudo o que não é o sinal de interesse. Por exemplo, peixes são uma fonte de ruído quando se procura por submarinos e os submarinos são uma fonte de ruído quando se procuram peixes. Ele contém ainda uma parcela de ruído próprio e de ruído ambiente, o ruído total é composto de muitos componentes, tornando-o uma função do sensor específico, estado do mar, transporte nível, orientação relativa, ângulo e frequência.

Índice de diretividade (DI) é o ganho contra ruído de campo distante omnidirecional. Leva em consideração algumas características direcionais e de correlação dos vários componentes do campo de ruído. Os ganhos de ruído observados aqui são uma função das dimensões da matriz, frequência, ângulo de elevação e o ângulo horizontal relativo ao arranjo de sensores. É medido através da razão entre a potência máxima, P_{max} , da antena real e a potência de uma antena isotrópica, P_{iso} , que irradia uniformemente em todas as

direções do espaço para uma antena hipotética:

$$DI = 10 \frac{P_{max}}{P_{iso}}. \quad (2.4)$$

2.2.2 SoNaR Ativo

Para o SoNaR ativo monoestático [22], ou seja, um SoNaR no qual o transmissor e o receptor estão localizados no mesmo invólucro, vale a equação abaixo:

$$\frac{S}{N} = \frac{S}{TL^2} \times \frac{AG}{N}, \quad (2.5)$$

ou, em escala dB,

$$SNR = S - 2TL - NL + AG, \quad (2.6)$$

onde S é o sinal de interesse; TL são as perdas de propagação, oriundas de reverberação, relaxamento iônico, espalhamento e atenuação; AG é o ganho devido a arranjo de sensores. Rigorosamente falando, AG não é um ganho, uma vez que não é ativo e se deve apenas a captação mais eficiente, mas, se comparado com um único hidrofone fornece um sinal melhor; e N é o ruído total, tudo aquilo que atrapalha a leitura do sinal, seja do próprio equipamento ou do ambiente.

Num SoNaR ativo, o sinal é emitido pelo transmissor, propaga-se até o alvo, é refletido, e depois se propaga de volta ao receptor. Esse sinal gerado, normalmente é muito potente, uma vez que, em seu percurso até o possível alvo, sofre diversas perdas e também é refletido por outros elementos do ambiente, como o assoalho submarino ou a superfície do oceano.

Essas interações entre fronteiras, regiões com variação de temperatura, salinidade ou pressão, causam reverberação, uma cópia do sinal com menor magnitude e defasada em relação a onda original, caso a demodulação deste sinal não seja coerente, com a onda transmitida, essa reverberação prejudicará a percepção do sinal recebido.

A atenuação ocorre tanto do hidrofone até o alvo e novamente do alvo até o receptor e se deve a lei da propagação esférica [24].

2.2.3 Excesso de Sinal

Para se executar uma detecção, onde apenas se emite um alarme de presença de um alvo; ou rastreamento, onde os movimentos do alvo são continuamente acompanhados, há

uma SNR mínima necessária, para se atingir um nível especificado de desempenho, em cada caso. O excesso de sinal é simplesmente a quantidade que se tem em relação a esse requisito, o valor acima dessa SNR mínima. Para a detecção, a quantidade necessária é referida como Limiar de detecção (DT). A definição mais comum desse termo é a razão sinal-ruído necessária para uma probabilidade de detecção de 50% em um período de tempo especificado. Em uma especificada probabilidade de alarme falso, esse conceito pode ser empregado pensando num determinado sistema e nos sinais que detectará e largura de banda específicas. A equação do SoNaR passivo [22] é geralmente escrita como:

$$SE = SL - TL - NL + DI - DT, \quad (2.7)$$

onde **SE** é o Excesso de sinal em dB.

A equação do SoNaR ativo manipulada para exibir o **SE** é

$$SE = SL - 2TL - NL + AG - DT. \quad (2.8)$$

2.3 Propagação do Som na Água

Embora tanto um morcego quanto um golfinho se localizem através do uso do som, o último tem uma tarefa mais complexa, por estar num meio muito peculiar para a transmissão de uma onda mecânica, que é o meio subaquático. O som é uma onda mecânica, que necessita de um meio para se propagar, que pode ser sólido, líquido ou gasoso, através da vibração das moléculas.

A peculiaridade da propagação do som na água se deve a alguns fatores, como variações de temperatura, salinidade e pressão. O Canal Sonoro Profundo (SOFAR) é um fator de suma importância nesta peculiaridade bem como os diferentes efeitos de reverberação, relaxamento iônico, que são agrupados sob o nome genérico de atenuação, embora a reverberação por exemplo não seja atenuação, e sim uma retransmissão do sinal, ao se chocar em um obstáculo que gera um cópia minorada do sinal defasada.

A onda sonora ainda sofre fenômenos de refração positiva ou negativa. A figura 2.3 demonstra a refração negativa que tende a acontecer no verão. Ocorre com a diminuição da temperatura em relação a profundidade, assim a água se torna mais densa e conseqüentemente a onda mecânica que necessita de um meio físico para interagir, tem mais moléculas para agitar, fazendo com que esta se propague mais lentamente.

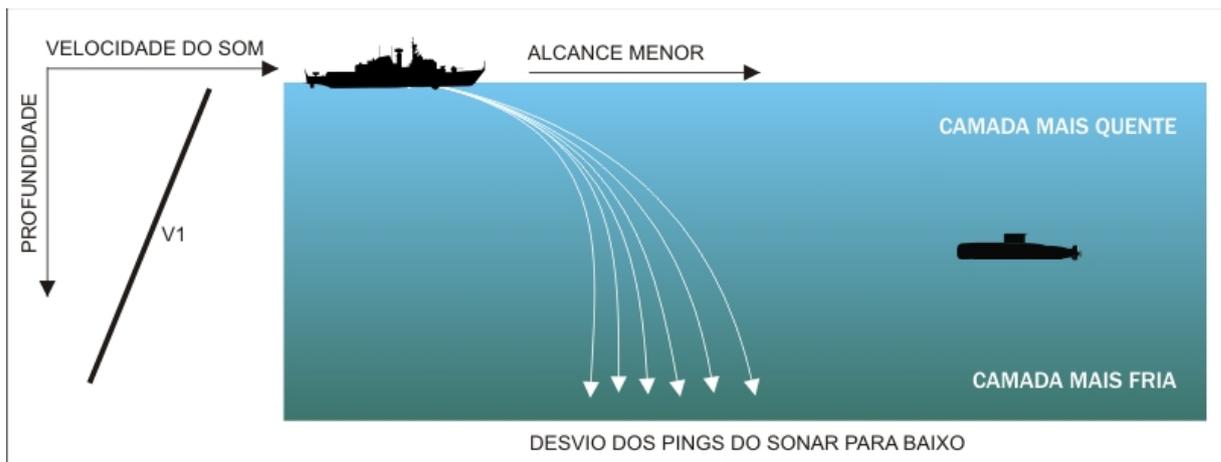


Figura 2.3: Refração negativa do som[3].

A refração positiva tende a acontecer no inverno, ela torna o alcance do SoNaR maior, as ondas se curvam pra cima pois é a região mais fria, conforme mostrado na figura 2.4.



Figura 2.4: Refração positiva do som[3].

O SOFAR é uma camada horizontal no oceano, onde o som pode viajar longas distâncias com perda mínima de intensidade. É formado através da interação das ondas sonoras com as diferentes camadas de densidade e temperatura d'água, o que é bom para detecção mas péssimo para acompanhamento, pois a diferença de eficiência de transmissão do som neste canal fará com que todos os alvos pareçam estar mais próximos.

Nesse canal, representado na figura 2.5, as ondas sonoras viajam com mais eficiência, devido à forma como a densidade e a temperatura da água afetam a velocidade do som. Geralmente, o som viaja mais rápido em águas mais quentes e densas. Perto da superfície, onde as variações de temperatura e a turbulência são comuns, as ondas sonoras podem dobrar-se e espalhar-se, perdendo energia. Porém, a uma certa profundidade, geralmente

em torno de 1.000 metros, existe uma camada de água onde a temperatura e a pressão fazem com que a velocidade do som atinja um mínimo. Isso cria um canal sonoro onde as ondas ficam presas e podem viajar longas distâncias sem muita perda de potência.

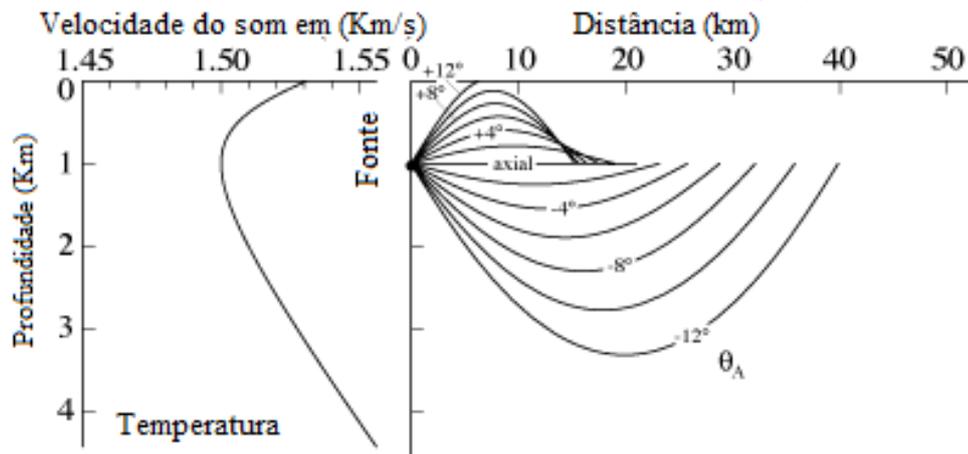


Figura 2.5: Velocidade do som em relação à profundidade [4].

Esse canal único SOFAR tem aplicações práticas, especialmente na comunicação submarina e no estudo de mamíferos marinhos que utilizam o som para comunicar a grandes distâncias. É um conceito essencial na acústica oceânica e tem implicações tanto para a investigação científica como para aplicações militares.

A onda acústica pode ser entendida como uma variação de pressão que se propaga no meio, da fonte ou do transmissor ao alvo. Para a descrição da onda, uma ferramenta que mapeie essa alteração de pressão no tempo e no espaço faz-se necessária, assim como também outras ferramentas como pressão quadrática média, intensidade acústica e potência ou energia.

O som pode percorrer caminhos bem diferentes se originado de fonte próxima à superfície do oceano até um dado ponto. Se o canal sonoro profundo se estender até a superfície, os raios que partem da fonte quase horizontalmente não atingirão a superfície do oceano ou o fundo do mar. Os sons que viajam por esses caminhos podem ser detectados a longas distâncias, assim como acontece com os sons que viajam para longe de uma fonte profunda que não interage com a superfície do oceano ou com o fundo do mar. Os caminhos sonoros de uma fonte próxima à superfície se unem ou convergem, criando regiões de maior pressão sonora aproximadamente na mesma profundidade da fonte a cada 50 a 60 km de distância dela.

Essas regiões de maior pressão sonora são chamadas de zonas de convergência. Entre

as zonas de convergência, existem regiões de menor pressão sonora chamadas zonas de sombra. E entre essas está a camada termoclina. Na zona de sombra não há nenhum som proveniente da fonte, caso um navio posicionasse seu SoNaR ativo próximo a superfície um submarino que encontre uma zona de sombra, jamais seria detectado. A figura 2.6 exemplifica essa situação.

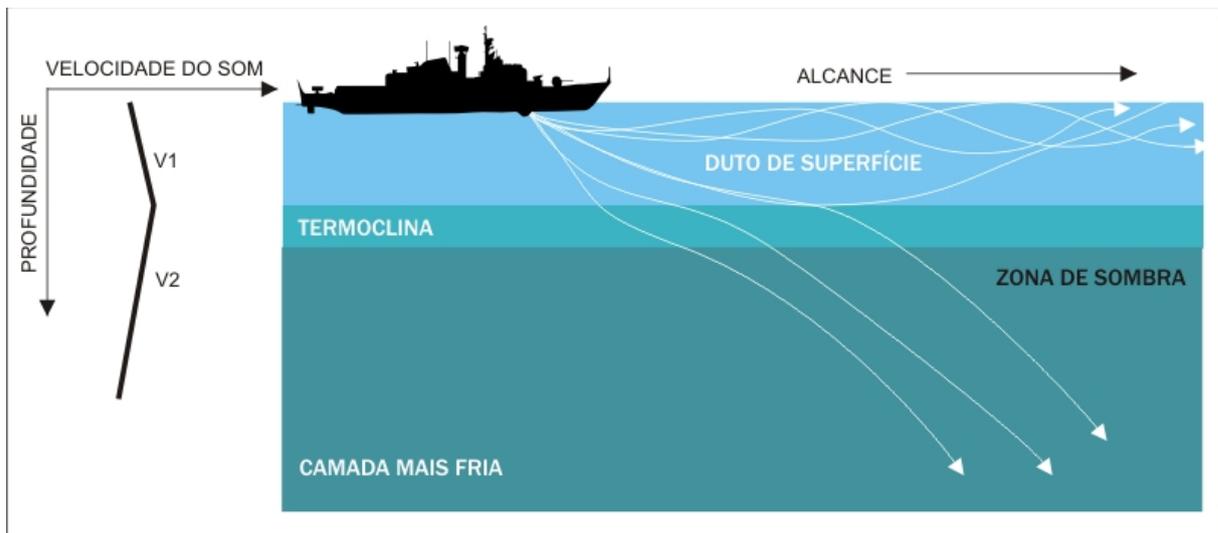


Figura 2.6: SOFAR, zonas de convergência e de sombra.

Tabela 2.1: Unidades do sistema internacional para várias quantidades de ondas acústicas e propriedades pertinentes da água no oceano.

Descrição	Variável	Unidades	Breve explicação
Pressão acústica	$p(t, \mathbf{x})$	μPa	Quantidade escalar de tempo e espaço
Pressão média ao quadrado	$P(\vec{x})$	μPa^2	Uma função do tempo para pulsos
Velocidade da partícula	$\vec{v}(t, \vec{x})$	m/s	Quantidade vetorial de tempo e espaço
Intensidade acústica	$\vec{J}(\vec{x})$ ou $J(\vec{x})$	pW/m^2	Densidade de fluxo de potência ou energia
Potência acústica	P	pW	Potência de Energia
Densidade do fluxo de energia acústica	$\varepsilon(\vec{x})$	pJ/m^2	Potência por área
Velocidade do som na água	c_w	m/s	O valor nominal é 1500 m/s
Densidade da água	ρ_w	kg/m^3	O valor nominal é 1027 kg/m^3

2.3.1 Campo Acústico e Potência

Ondas acústicas são uma forma de energia conhecida pelos humanos que podem viajar por grandes distâncias na água e geralmente são considerados o melhor portador de informações, para detectar e reconhecer alvos subaquáticos[25].

O sinal de pressão acústica $p(t, \vec{x})$ é composto por um somatório de compressões e rarefações. Por definição, sua média temporal, ao longo de um intervalo de tempo T para

um dada posição é nula:

$$\bar{p} = \frac{1}{T} \int_t^{t+T} p(\tau, \vec{x}) d\tau. \quad (2.9)$$

A Pressão quadrática média (PQM) $P(\vec{x})$ é usada para determinar o desempenho de um sistema de SoNaR. PQM é definida como a média da pressão ao quadrado ao longo de um intervalo de tempo T :

$$P(\vec{x}) = \frac{1}{T} \int_t^{t+T} p^2(\tau, \vec{x}) d\tau \quad (2.10)$$

onde t e T são considerados para abranger as características pertinentes do sinal. Por exemplo, a extensão de um eco de SoNaR ativo ou muitos ciclos de um processo de ruído de banda estreita. Ao se referir a sinais de pressão, é comum que o nível de pressão represente o PQM em decibéis. Por exemplo, se $P(\vec{x})$ é o PQM em \vec{x} expresso em μPa^2 , o nível de pressão acústica ou nível de pressão sonora é de $10 \log_{10} P(\vec{x})$ em μPa^2 .

A perda por transmissão é medida em dB entre um ponto de interesse e um ponto de referência. Sejam I_0 a intensidade mensurada num ponto de referência; TL , a perda de transmissão; R , a distância e D , a profundidade, então

$$TL = -10 \log \frac{I(R, D)}{I_0} = -20 \log \frac{|p(R, D)|}{|p_0|}. \quad (2.11)$$

Na equação acima, $I(R, D)$ é a intensidade acústica em um intervalo, R , e profundidade D de interesse; I_0 é a intensidade acústica na distância de referência; $p(R, D)$ é a pressão sonora em uma faixa R , e profundidade D de interesse; e p_0 é a pressão sonora na distância de referência.

As condições que afetam a perda de propagação são espalhamento, absorção, e reflexão seja no meio, ou na superfície, ou nos limites do fundo, múltiplos caminhos e efeitos de onda, por exemplo, vazamento de duto[24].

Quando uma onda acústica se propaga através da água, as partículas de água oscilam na mesma direção em que a onda se propaga. A velocidade da partícula é uma grandeza vetorial $\vec{v}(t, x)$, enquanto a pressão é uma grandeza escalar. A relação entre a velocidade da partícula e a pressão depende do tipo de propagação da onda (como, por exemplo, esférica, plana ou cilíndrica) e pode ser derivada dos princípios da mecânica dos fluidos produzindo a equação da onda. Sob várias suposições simplificadoras, que são satisfeitas pelo modelo básico do oceano, a porção da velocidade da partícula atribuível à pressão acústica está relacionada ao sinal de pressão espaço-temporal através da equação (2.12).

$$\vec{v}(t, \vec{x}) = \frac{-1}{\rho_w} \int \nabla p(t, \vec{x}) dt, \quad (2.12)$$

onde ∇ é gradiente espacial, e ρ_w é a densidade de potência.

A velocidade de partícula da onda plana é

$$\vec{v}(t, \vec{x}) = \left[\frac{p(t, \vec{x})}{\rho_w c_w} \right] \vec{u}. \quad (2.13)$$

Na equação (2.13), \vec{u} é a direção de propagação da onda, perpendicular à frente de onda, e c_w é a Velocidade do som.

A intensidade de uma onda acústica é uma medida da densidade de potência. É o produto médio no tempo da pressão e da velocidade da partícula,

$$\vec{J}(\vec{x}) = \frac{1}{T} \int_t^{t+T} p(\tau, \vec{x}) \vec{v}(\tau, \vec{x}) d\tau. \quad (2.14)$$

De modo semelhante à formação do PQM, a integral deve abranger tempo suficiente para capturar as características pertinentes do sinal. Explorando a relação entre pressão e velocidade da partícula da equação 2.13 e considerando apenas a componente na direção da propagação da onda como na equação (2.14), a intensidade de uma onda plana é vista como sendo a pressão média ao quadrado dividida pela impedância característica de água (ρ_w, c_w),

$$\vec{J}(\vec{x}) = \frac{\frac{1}{T} \int_t^{t+T} p^2(\tau, \vec{x}) d\tau}{\rho_w c_w} = \frac{P(\vec{x})}{\rho_w c_w} \quad (2.15)$$

O integrando da Equação (2.14) é a potência instantânea:

$$\vec{I}(t, \vec{x}) = p(t, \vec{x}) \vec{v}(t, \vec{x}) d\tau. \quad (2.16)$$

Particularmente, para uma onda plana,

$$I(t, \vec{x}) = p(t, \vec{x}) v_{\vec{u}}(t, \vec{x}) = \frac{p^2(t, \vec{x})}{\rho_w c_w}. \quad (2.17)$$

A potência acústica, que é a energia transferida através de uma superfície por unidade de tempo é usada para caracterizar uma fonte de som. Esta, pode ser obtida a partir da intensidade como a integral sobre uma superfície \mathbf{S} envolvendo a fonte,

$$P = \int \int_S \vec{J}(\vec{x}) \cdot \vec{n} dS, \quad (2.18)$$

onde \vec{n} é o vetor unitário normal a superfície.

Muitos sinais acústicos são de curta duração e têm amplitudes variáveis, por exemplo, emissões acústicas de um mamífero marinho, um eco de SoNaR ativo ou um pacote de comunicações. Na prática, esses sinais são chamados de sinais transitórios; no entanto, Transitório é um sinal que transaciona entre as condições iniciais e regime permanente [26] e em um sentido matemático, eles são sinais de energia porque sua energia total é finita [26],

$$\int_{-\infty}^{\infty} p^2(t, \vec{x}) < \infty. \quad (2.19)$$

A Lei de Espalhamento Esférico é definida num meio homogêneo e sem perdas. É um princípio físico que descreve como a potência de uma fonte radiante diminui à medida que se afasta dela. A lei é válida para radiação eletromagnética e ondas mecânicas quando a fonte é pontual e se espalha uniformemente no meio que se encontra. Conforme a potência é distribuída em uma área cada vez maior, a intensidade da radiação diminui pontualmente com o afastamento da fonte.

A potência total P irradiada pela fonte permanece constante, mas a intensidade I , que é a potência por unidade de área, diminui à medida que a distância aumenta. A relação entre a intensidade I e a potência P é dada por:

$$P = 4\pi r_1^2 I_1 = 4\pi r_2^2 I_2 = 4\pi r_3^2 I_3 = \dots = 4\pi r_n^2 I_n, \quad (2.20)$$

onde

$r =$ é a distância entre a fonte e o ponto de interesse.

À medida em que r aumenta, a intensidade diminui de acordo com o inverso do quadrado da distância. Portanto,

$$P = I \times A. \quad (2.21)$$

2.3.2 Perdas por Absorção

Nas ondas sonoras, o deslocamento das partículas do meio ocorre ao se comprimir e se rarefazer alternadamente. Quando a fonte sonora vibra, cria-se uma região de compressão, onde as partículas do meio ficam mais próximas umas das outras, gerando uma maior pressão local. Após a região de compressão, ocorre a rarefação, onde as partículas se afastam, resultando em uma menor pressão local.

Se a fonte sonora emana um sinal $u_o(t)$ na origem do sistema de coordenadas, ou seja, $x_o = (0, 0, 0)$, então o campo de pressão acústica resultante é função do tempo e do

espaço, $p(t, x)$, deve satisfazer a equação de onda.

$$\nabla^2 p = \frac{1}{c_w^2} \frac{\delta^2 p}{\delta t^2}. \quad (2.22)$$

Na equação (2.22) o laplaciano, ∇^2 é o produto escalar do operador *nabla* por ele mesmo conforme a expressão:

$$\nabla^2 = \nabla \cdot \nabla. \quad (2.23)$$

Quando uma onda sonora viaja pela água do mar, as perdas por absorção ocorrem por meio de três mecanismos.

O primeiro mecanismo foi investigado teoricamente por Rayleigh que calculou a absorção devido à viscosidade de cisalhamento [27]. Com o avanço da ciência, após umas previsões de Rayleigh, o valor esperado representava apenas um terço da absorção observada em água destilada. A absorção adicional observada é devido à viscosidade do volume. As fontes dominantes de absorção na água do mar acima de 1 MHz são dadas por

$$\alpha = 16\pi^2(\mu_s + 0,75\mu_v)f^2/(3\rho c^3), \quad (2.24)$$

onde α é coeficiente de absorção de intensidade (cm^{-1}), μ_s é a viscosidade de cisalhamento, μ_v é o coeficiente de viscosidade volumétrica, f a frequência em Hz, ρ a densidade em (g/cm^3), e c é a velocidade do som em $1,5 \times 10^5 cm/s$.

As primeiras medições mostraram que a absorção na água do mar era de duas ordens de magnitude maior do que para água doce em frequências abaixo de 100 kHz. Em 1948, Liebermann [28] propôs que isso se devia ao relaxamento iônico, que domina a absorção em frequências mais baixas e tem a forma

$$\alpha_r = af^2 f_r / (f^2 + f_r^2) \quad (2.25)$$

onde f_r é a frequência de relaxação dos íons.

2.3.3 Atenuação

Em 1967, Thorpe [29] publicou uma equação para atenuação, que se tornou a equação mais comumente vista para atenuação:

$$\alpha = 0,1f^2/(1 + f^2) + 40f^2/(4100 + f^2), \quad (2.26)$$

onde f é frequência em kHz e α a atenuação em dB/kyd.

Essa equação é quase inteiramente baseada nas medições do Atlântico Norte e carece das dependências de temperatura, pH, salinidade e pressão de modelos posteriores. A Marinha dos EUA atualmente usa o modelo Francois–Garrison [30] para atenuação, que inclui relaxamento iônico para ácido bórico e sulfato de magnésio mais viscosidade. Há dependência da temperatura, salinidade, pH, profundidade e pressão:

$$\alpha = \frac{A_1 P_1 F_1 F^2}{F_1^2 + F^2} + \frac{A_2 P_2 F_2 F^2}{F_2^2 + F^2} + A_3 P_3 F^2, \quad (2.27)$$

$$A_1 = (8,86/c)10^{0,78pH-5},$$

$$P_1 = 1,$$

$$F_1 = 2,8\left(\frac{S}{35}\right)^{0,5}10^{4-1245/T+273},$$

$$c = 1412 + 3.2T + 1.19S + 0.0167D,$$

$$A_2 = 21,44(S/c)(1 + 0,025T),$$

$$P_2 = 1 - 1,37 \times 10^{-4}D + 6,2 \times 10^{-9}D^2,$$

$$F_2 = \frac{8,17 \times 10^8 - 1990/(T+273)}{1+0,0018(S-35)}.$$

Para $T < 20^\circ C$:

$$A_3 = 4,937 \times 10^{-4} - 2,59 \times 10^{-5}T + 9,11 \times 10^{-7}T^2 - 6,5 \times 10^{-8}T^3.$$

Para $T > 20^\circ C$:

$$A_3 = 3,964 \times 10^{-4} - 1,146 \times 10^{-5}T + 1,45 \times 10^{-7}T^2 - 6,5 \times 10^{-10}T^3,$$

$$P_3 = 1 - 3,83 \times 10^{-5}D + 4,9 \times 10^{-10}D^2,$$

S— salinidade (ppt),

C— velocidade do som (m/s),

D— profundidade (m).

Atualmente os três tipos de absorção juntamente com a viscosidade, explicam a maior parte da absorção nos oceanos. Esses três relaxamentos foram identificados como aqueles associados ao sulfato de magnésio, ácido bórico e carbonato de magnésio. O coeficiente de absorção α em dB por km obedece a:

$$\alpha = A_1(MgSO_4) + A_2(B(OH)_3) + A_3(MgCO_3) + 0,0003f^2, \quad (2.28)$$

$$A_n = \frac{a_n f^2 f_n}{f^2 + f_n^2}, \quad (2.29)$$

onde: $a_1 = 0,5 \times 10^{-D/20}$, $f_1 = 50 \times 10^{T/60}$, $a_2 = 0,10 \times 10^{pH-8}$, $f_2 = 0,90 \times 10^{T/70}$, $a_3 = 0,03 \times 10^{pH-8}$; e $f_3 = 4,5 \times 10^{T/30}$.

D – profundidade(km),

F – frequência (kHz),

T – temperatura ($^{\circ}C$).

O modelo acima é da Mellen [31]. Modelos sofisticados de perda de propagação modelarão a absorção em função do alcance e da profundidade, levando em consideração as mudanças de temperatura, pH, salinidade e pressão [2].

2.4 Classificação SoNaR

O processamento rápido e preciso dos dados do SoNaR é essencial na guerra submarina. A capacidade de detectar rapidamente e classificar os alvos do SoNaR é crucial para o desempenho e capacidade de sobrevivência de todos os navios de superfície e submarinos da Marinha. Com o advento da tecnologia de rede neural, novas oportunidades surgiram, pois agora é possível explorar novas relações entre o sinal adquirido, o limiar de detecção, o ruído ambiente e realizando fusão de diversas informações para melhor entender os sinais recebidos [32].

Identificar um contato SoNaR somente por sua assinatura acústica é uma tarefa extremamente difícil, pois é necessário perceber um ruído específico em meio a um ambiente ruidoso. É mais simples através do espectrograma, exemplificado na figura 4.6, perceber um sinal contínuo na frequência e então utilizar o áudio para determinar a presença ou não de um contato de interesse. A classificação do contato exige o conhecimento de, pelos menos, uma das suas muitas assinaturas. Esse conhecimento é normalmente abordado campo do saber da Guerra Eletrônica e será apresentado na seção 2.4.1.

2.4.1 Guerra Eletrônica

A guerra é uma luta armada entre dois ou mais povos, ou a pré-disposição para tê-la. Conforme Hobbes, não consiste num combate efetivo, mas numa disposição patente nesse

sentido, enquanto não houver garantia do contrário. Qualquer outro tempo, que não seja de guerra, se chama paz [33].

Embora em termos gerais a guerra deva ser evitada, existe um conceito de guerra justa, quando for da: defesa de um direito líquido e certo, injustamente desrespeitado; são esgotadas todas as tentativas de uma composição pacífica do litígio; e há uma equivalência moral entre o bem que se espera, a reparação do direito do ofendido, e os males inerentes ao desencadear da violência [34].

A guerra eletrônica é o uso de ondas eletromagnéticas para controlar ou perturbar o espectro eletromagnético. Inclui atividades como bloquear comunicações inimigas, detectar sinais de radar e proteger sistemas amigos de ataques eletrônicos. A guerra eletrônica desempenha um papel crítico nas operações militares modernas, proporcionando um meio de obter vantagens táticas e estratégicas [35].

A guerra eletrônica é avaliada para um dado ente militar e a totalidade das suas capacidades denomina-se Capacidade de Guerra eletrônica, conforme figura 2.7 que se subdivide em Atividades de Guerra Eletrônica e Medidas de Apoio à Guerra Eletrônica.



Figura 2.7: Capacidade de Guerra eletrônica.

As Atividades de Guerra Eletrônica (AGE) são todas aquelas de caráter estratégico, tático, logístico e de pesquisa que contribuam para o estabelecimento, a exploração, a reformulação, ou verificação da CGE e para o apoio ao planejamento do seu emprego em operação de guerra naval.

As Medidas de Apoio à Guerra Eletrônica (MAGE) são o conjunto de ações visando à busca, interceptação, identificação e localização eletrônica das fontes de energia eletromagnética irradiada no ambiente eletrônico, de uma Força ou unidade, a fim de permitir a análise, o imediato reconhecimento de uma ameaça ou sua posterior exploração. As Atividades de Guerra Eletrônica se subdividem em Reconhecimento Eletrônico e Aprestamento Eletrônico, representados na figura 2.8.

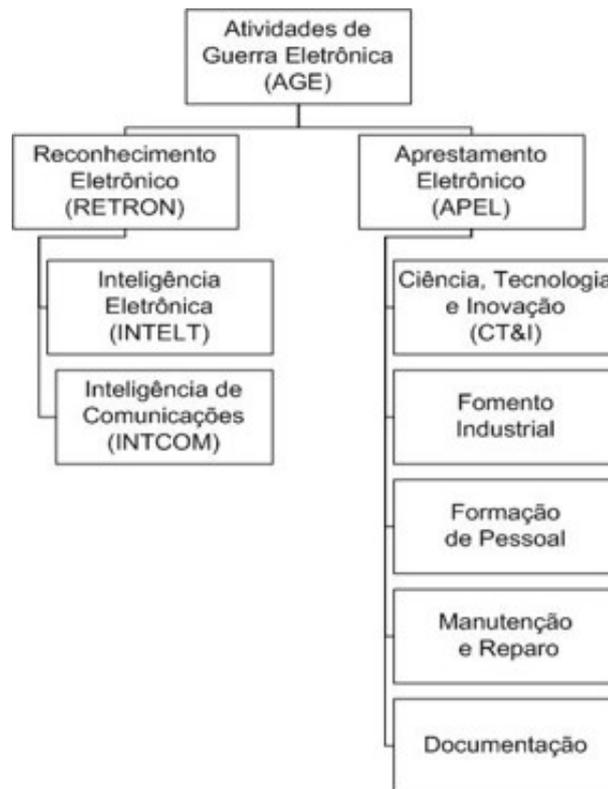


Figura 2.8: Atividades de Guerra Eletrônica.

O Reconhecimento Eletrônico (Retron) se refere às atividades voltadas à captação de informações sobre as Capacidades de Guerra Eletrônica (CGE) do inimigo e possibilita o emprego dos recursos mais efetivos contra uma determinada ameaça, esta dissertação se insere neste campo.

O Aprestamento Eletrônico (Apel) é o ramo responsável por garantir o bom preparo do pessoal e material envolvidos nessa guerra.

Um ser humano pode ser identificado univocamente por sua impressão digital, sua voz, seu rosto, ou seu DNA. Uma plataforma, qual seja um navio, ou helicóptero, ou submarino possui assinaturas acústicas, visual, seção reta-radar, por usas emissões e infravermelha. Associado a todos esses sinais está o ruído, o qual mascara o real formato dessas assinaturas, dificultando a identificação destes sinais. A Figura 2.9 apresenta cinco assinaturas, os parâmetros que as afetam bem como o ruído ambiental ou do equipamento que interfere em cada uma delas.

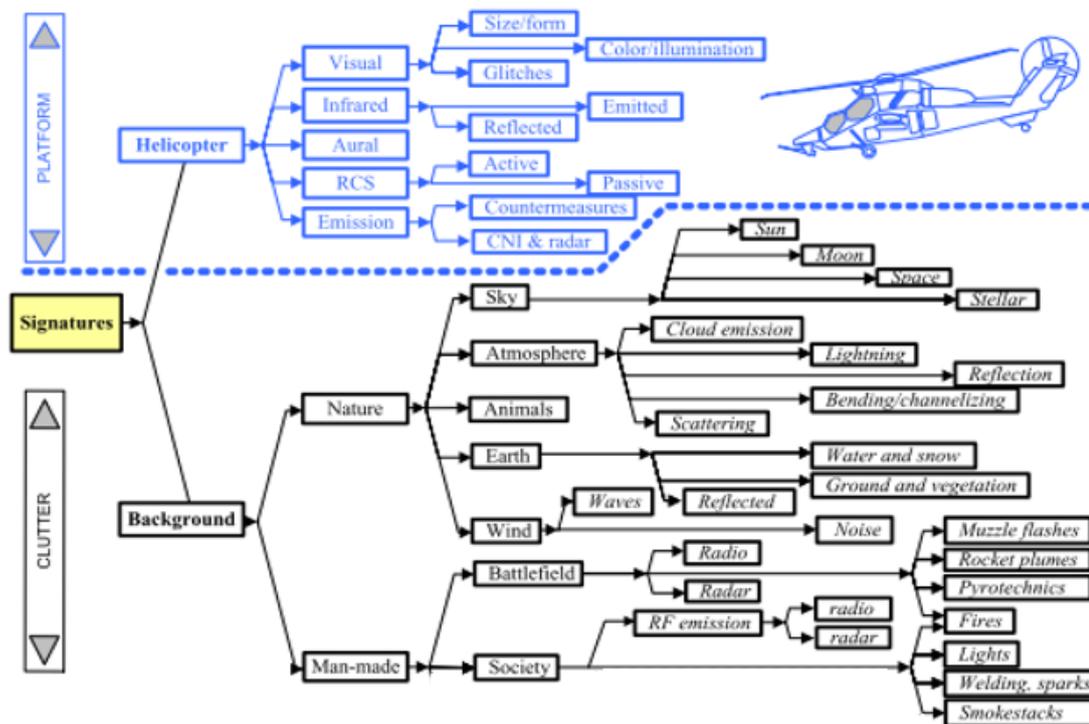


Figura 2.9: Assinaturas e ruídos.

Sendo assim, em termos de Guerra Eletrônica, esta dissertação visa fornecer uma ferramenta de Reconhecimento eletrônico, que é um campo estratégico da guerra eletrônica, através de uma das muitas assinaturas de um meio, no caso a assinatura acústica mesmo em ambiente ruidoso visando diferenciar e especificar o ruído feito pelo homem de um ruído oriundo do meio ambiente e da fauna marinha. Portanto uma Medida de Apoio a Guerra Eletrônica (MAGE), observada do ponto do sinal pré-processado e não da aquisição acústica deste que auxilia na busca, interceptação e localização de um alvo.

Os usos da tecnologia SoNaR englobam os campos militar, auxílio a navegação, detecção de objetos, comunicação submarina e mapeamento do assoalho submarino. Desde os primórdios dessa invenção, todos esses usos foram possíveis graças a esse equipamento, o capítulo 3 apresenta uma demonstração de artigos que representam o estado da arte nos campos supracitados.

Capítulo 3

Diferentes tipos e usos de SoNaR e realizações de classificação SoNaR passiva na atualidade

O objetivo deste capítulo é observar os usos em que a tecnologia SoNaR tem sido empregada. Inicialmente, o equipamento visava a detecção de alvos submarinos para auxílio a navegação, depois tomou o uso militar que é o interesse primário desta dissertação.

Há novas configurações SoNaR que, embora derivem da configuração básica, utilizam outros conceitos e agregam novas tecnologias para uso em detecção de perigos e objetos subaquáticos; navegação e comunicação em submarinos; arqueologia subaquática e caça ao tesouro; na indústria pesqueira para localização de peixes; exploração de petróleo e gás submarino; e monitoramento da vida marinha e dos ecossistemas [36].

3.1 SoNaR

Atualmente, há uma grande discussão no mundo científico sobre a utilização do sinal recebido pelo SoNaR. A discussão está em pauta em áreas como otimização da navegação do navio; comunicação acústica subaquática; reconhecimento de sinal acústico subaquático; e detecção e posicionamento subaquático [37]. Há três modelos de SoNaR mais visualizados na literatura, conforme especificados abaixo:

SSS - SoNaR de Varredura Lateral

SAS - SoNaR de Abertura Sintética

SoNaR MIMO - SoNaR com múltiplas entradas e múltiplas saídas

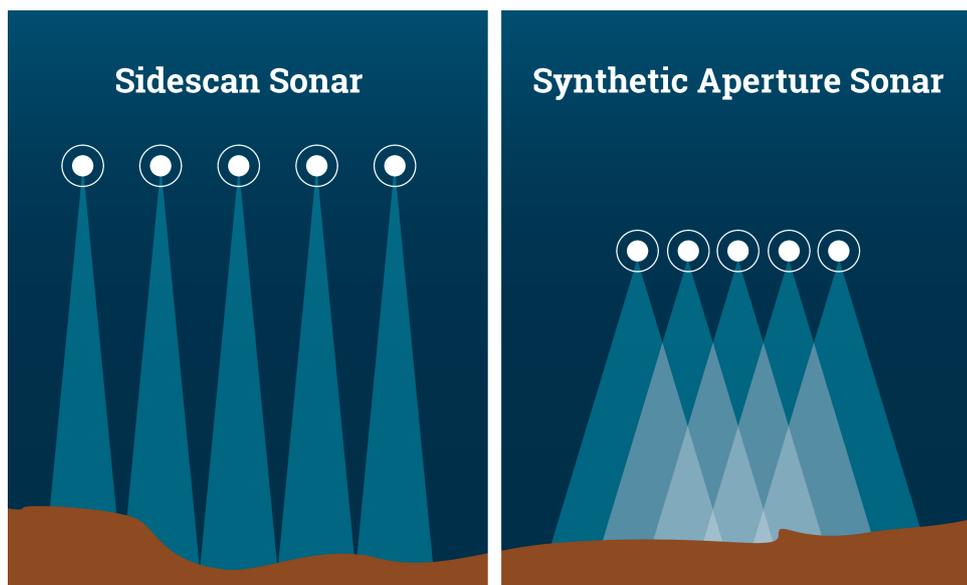


Figura 3.1: Diferenças entre SSS e SAS.

fonte: <https://oceanexplorer.noaa.gov/technology/sonar/sas.html>

3.1.1 SoNaR Sidescan

SSS e o Ecobatímetro multi-feixe (MBES), representados na figura 3.1, são sonares comumente usados para pesquisar o fundo do mar. SSS é usado para obter imagens detalhadas do fundo do mar devido à sua alta resolução e ampla cobertura de faixa, enquanto o MBES é usado na construção de um mapa batimétrico devido à sua capacidade para medir diretamente a geometria 3-D do fundo do mar. MBESs são normalmente montados em navios, mas também podem ser mobilizados em grandes submarinos autônomos (AUVs), porém esses MBES são de menor porte que os montados em navios e, assim, não têm grande resolução além de possuir alto custo em comparação com o SSS de matriz única [38].

No artigo [39], é proposto um modelo de discriminação de alvo subaquático de SoNaR de varredura lateral baseado na convolução dilatada híbrida combinada e no algoritmo de atenção dividida em pirâmide UNet (BHP-UNet). O modelo BHP-UNet exibiu melhor adaptabilidade ao complexo ambiente marinho, apresentando desempenho superior, capacidade de generalização e de segmentação, quando é comparado com modelos consagrados na literatura.

A varredura lateral fornece uma projeção 2-D do fundo do mar 3-D. O artigo[38] prova a melhoria da batimetria obtida usando profundidades esparsas com estimativas de varredura lateral apenas com SSS, através de uma Rede Neural profunda (DNN) convolucional para estimar o contorno de profundidade e sua incerteza. É mostrado o benefício

da ponderação de confiança ao fundir várias estimativas batimétricas em um único mapa.

Um nova arquitetura proposta em [40] para reconhecimento de alvo utilizando um SSS, usando a rede ResNet152. A ResNet (Rede Residual) é uma rede neural convolucional que democratizou os conceitos de aprendizagem residual (*Transfer learning*) e de pular conexões. Isso, permite treinar modelos muito mais profundos, com esforço computacional muito menor, pois se utiliza uma rede previamente treinada que necessita fazer apenas uma associação para um novo uso e não um treino do zero[41]. Essa rede é utilizada como a rede principal do modelo. A estratégia de transferência faseada, onde os dados ImageNet e SAR foram utilizado para aprender os parâmetros da rede e assim utilizar a técnica de *Transfer Learning*, onde uma rede já treinada é usada excluindo a última camada, inserindo uma nova e treinando somente esta. Isso resultou em taxas de reconhecimento significativamente melhores do que outras redes de backbone. O desempenho foi melhor do que outros métodos de reconhecimento com *Transfer Learning*.

3.1.2 SoNaR de Abertura Sintética

Em [42], os autores desenvolveram um algoritmo, denominado omega-K, no qual SASs de vários receptores são decompostas nas imagens SASs biestáticas, isto é, um par transmissor-receptor. Com o método apresentado, é possível entender a variação de fase com qual a onda é acometida durante seu trajeto melhorando a imagem e identificação de objetos.

As variações da arquitetura UNet em [43] são propostas e investigadas para tradução de domínio de intensidade para relevo do assoalho oceânico num SAS. UNet é um tipo especial de autoencoder convolucional que contém conexões com camadas não-adjacentes. Essas conexões trazem características de camadas convolucionais anteriores para as posteriores para diminuir a perda de informações.

Uma arquitetura DNN capaz de resolver o ângulo de chegada (AoA) de retornos acústicos, por meio do processamento do espaço de dimensão k é explorada em [44]. São K diferentes imagens formadas por diferentes AoA permitindo a extração de características temporais e espaciais do fundo do mar que são úteis para discriminar certos tipos de textura do fundo do mar e naturalmente objetos em movimento.

A interferometria de SAS é uma técnica para imagens de alta resolução e mapeamento do fundo do mar, dependente da estimativa da profundidade do fundo do mar. É uma função do sistema, da geometria, da razão sinal-ruídoSNR e do comprimento do filtro. Em [45], os autores sugerem usar um tamanho de filtro de estimativa de fase adaptável,

de modo que o desvio padrão da estimativa de profundidade seja proporcional à resolução horizontal. Esse filtro adaptativo pode diminuir o tamanho da janela de correlação para alta SNR e intervalos relativamente curtos, o que diminui a resolução horizontal em detrimento do aumento do desvio padrão das estimativas de profundidade. Em longas distâncias e baixos ângulos rasantes, onde os sistemas SASs geralmente são limitados por ruído, o filtro adaptativo aumenta a complexidade do filtro, visando manter o desvio padrão das estimativas de profundidade em um nível aceitável às custas de uma resolução horizontal reduzida.

3.1.3 SoNaR MIMO

Alguns artigos classificam o SAS como um SoNaR Entrada múltipla e saída múltipla (MIMO). Se SoNaR tem uma abertura sintética, como denota a figura 3.1, esse SoNaR envia em diferentes momentos um eco em diferentes posições, o que não configura MIMO, mas sim a de entradas somadas no tempo. Consequentemente nesta dissertação esses equipamentos são apresentados distintamente.

O SAS opera sintetizando uma grande abertura através do movimento da plataforma SoNaR, usa o movimento da plataforma para criar um *array* (vetor) virtual que atinge alta resolução na direção transversal. Um SoNaR MIMO utiliza vários elementos de transmissão e recepção para formar um conjunto espacialmente diversificado. O SoNaR MIMO obtém imagens de alta resolução e melhora a detecção de alvos.

No artigo [46], o uso de formas de onda codificadas ortogonalmente é modelado por um sistema MIMO. Essas ondas compartilham a mesma largura de banda, e têm o potencial de aumentar a taxa de cobertura de área do SAS. O autor obteve êxito em duplicar a taxa de cobertura da área com um pequeno impacto na complexidade do hardware, apesar da interferência entre as formas de onda.

Um sistema com alta resolução angular é proposto em [47]. A técnica utilizada consiste numa convolução em duas dimensões ortogonais. Essa imagem é duplamente convoluída, na qual cada eixo é um componente dessa convolução. O método proposto pode melhorar as resoluções angulares e de alcance e suprimir os lóbulos laterais dos receptores significativamente, aumentando assim a discriminação angular e, consequentemente, a diretividade.

A utilização de múltiplos feixes suaviza, torna mais esparsa a matriz de correlação, o que, por sua vez, simplifica a identificação de contatos. Em [48], propõe-se que matriz de

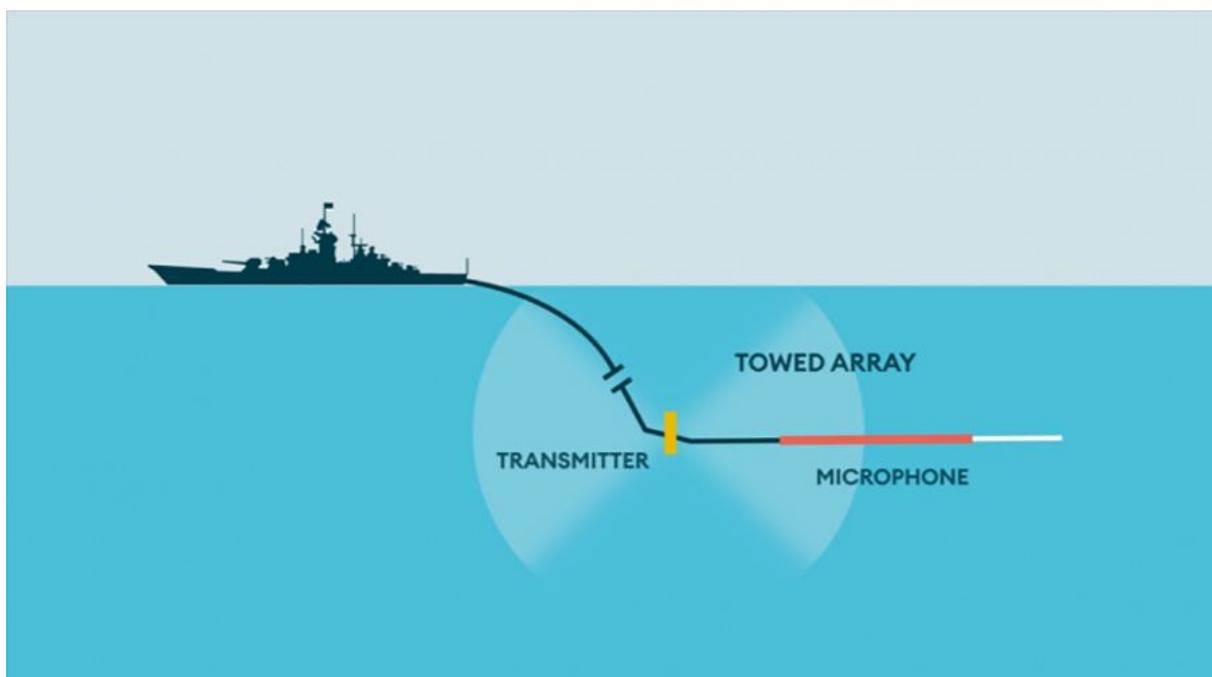


Figura 3.2: Towed Array SoNaR - SoNaR com múltiplas entradas e múltiplas saídas. Fonte: <https://en.mercopress.com/2022/01/10/royal-navy-frigate-underwater-sonar-struck-by-russian-submarine>

covariância de dados MIMO conjugada seja multiplicada à esquerda e à direita, respectivamente, pela matriz de troca, obtém-se uma matriz de covariância de dados suavizada que é então regularizada com a matriz de covariância original. Com efeito, esse método é aplicado a alvos reais e denota significativa melhora.

Os Nível do Sinais (SLs) na imagem do SoNaR MIMO são altos devido à ortogonalidade não-ideal das formas de onda de transmissão. Os autores em [49] propõem um método de desconvolução para suprimir os SLs. Os SLs de alcance no resultado da imagem são compostos de funções de correlação cruzada e de funções de autocorrelação (ACFs) de formas de onda de transmissão. Eles podem ser tratados como o termo de “ruído” na saída da convolução entre as distribuições originais. Assim, usamos o método de desconvolução, que neste caso corresponde ao algoritmo de Richardson-Lucy [49]. Para suprimir esses SLs Levando em consideração a distorção de eco, foi possível encontrar a melhor função de dispersão de pontos tipo Dirac, resultando em SL de menor alcance.

O algoritmo Richardson-Lucy, também conhecido como desconvolução Lucy-Richardson, é um procedimento iterativo para recuperar uma imagem subjacente que foi desfocada por uma função de propagação de pontos conhecida. Foi nomeado em homenagem a William Richardson e Leon B. Lucy, que o descreveram de forma independente [50], [51].

3.2 Classificação Passiva de Contatos SoNaR

Os sistemas de SoNaR são usados em diversas aplicações para fins militares e civis, como vigilância subaquática, detecção, localização e identificação, determinação da profundidade da água, contornos e composição do fundo e localização de cardumes de peixes. O SoNaR ativo emite pulsos acústicos e analisa os ecos recebidos. O SoNaR passivo registra e processa os sinais acústicos emitidos por outras embarcações, visando a detecção e identificação de alvos.

Para detectar e classificar sinais em relação ao ruído de fundo, os sistemas de SoNaR passivos ouvem o ruído irradiado por alvos usando uma série de hidrofones. O ruído de fundo pode ser produzido pelo ruído ambiente do mar, ou pelo ruído próprio do plataforma SoNaR. A partir dos sinais adquiridos, estima-se a direção de chegada (DoA), a fim de informar a eventual presença de um alvo em uma determinada direção. Após a estimativa do DoA, características relevantes do alvo podem ser extraídas de uma dada direção. Para sistemas de SoNaR passivos, a identificação do alvo é geralmente realizada por operadores humanos experientes.

O LOFAR (*Low Frequency Analysis and Recording*), que é uma análise de banda larga geralmente empregada em sistemas de SoNaR passivo, estima a vibração sonora do maquinário alvo. A análise LOFAR é baseada em estimação espectral e suporta classificação de alvos. Abaixo, há uma discussão da sua utilização, para fins de classificação na atualidade.

Em [14], utiliza-se um perceptron multicamada classe-modular (CM-MLP), em que cada neurônio da saída (especialista) é responsável por reconhecer uma classe E, para combinar a saída dos especialistas, é utilizada a técnica de votação que é uma abordagem atraente para construir Sistemas de classificação automática (ACS), uma vez que se beneficia da decomposição de problemas e dos pontos fortes do perceptron multicamada (MLP) na resolução de padrões complexos. Foram utilizados sinais digitalizados usando 16 bits e a uma taxa de amostragem de 22 kHz. No total, 8 classes, 263 viagens de navio e 19 a 66 viagens por cada classe. A maior acurácia obtida foi de aproximadamente 85%.

No artigo [52], usam-se redes adversárias generativas (GANs) para sintetizar dados, objetivando que um volume maior de dados possa ser produzido para treinar redes neurais convolucionais (CNNs), que são usadas para a tarefa de classificação. Diferentes abordagens de projeto de GAN foram avaliadas e tanto a probabilidade máxima quanto as estratégias de classe especialista foram exploradas para classificação de sinal. Foi dada especial atenção à forma como o conhecimento especializado pode ajudar na análise do

desempenho dos vários modelos de aprendizagem profunda através de testes que espelham a implementação real. Uma precisão de até $99,0 \pm 0,4\%$ foi alcançada usando dados experimentais, o que melhora projetos anteriores de aprendizado de máquina em campo.

Em [13], uma estrutura generativa não-supervisionada utilizando um autoencoder variacional (VAE) é proposto nesse trabalho para criar representações melhor separadas para a tarefa de classificação. A análise experimental de várias configurações de classificador produziu resultados encorajadores em termos de eficiência de dados e classificação. Alcançou-se uma precisão de até $99,3 \pm 0,4\%$.

3.3 Comunicação Submarina

Além de ser utilizado para levantamento do assoalho submarino e classificação passiva, há aplicações de redes de comunicações acústicas submarinas. Os sinais acústicos usados para comunicação subaquática normalmente ficam na faixa de 10 Hz a 1 MHz. Os sinais de baixa frequência podem viajar distâncias mais longas, mas sofrem com baixas taxas de dados, enquanto os sinais de alta frequência podem transportar mais dados, mas têm alcance limitado [53].

Em [54], há um estudo que fornecer um levantamento abrangente das pesquisas mais recentes no campo da comunicação UWA MIMO-OFDM. Os trabalhos anteriores são resumidos, revisados e comparados de acordo com os anos de publicação, enquanto são destacados os problemas enfrentados pela comunicação UWA MIMO-OFDM.

Em [55], analisa-se a aplicação de diversidade na transmissão acústica em canal submarino através de um simulador que usa parâmetros extraídos de campanhas de medições realizadas no canal, utilizando simulações que basearam em parâmetros extraídos de uma campanha de medições em águas rasas realizadas em Arraial do Cabo, RJ. Os resultados mostram ganhos obtidos no desempenho em termos de BER para os sistemas com transmissão *Multiple Input Single Output* (MISO) recepção em diversidade *Single Input Multiple Output* (SIMO), que diminuem com o crescimento da E_b/N_0 .

Em [56], os autores propõem um novo esquema de modulação de caos MC, que combina chaveamento diferencial de caos com mudança de código (CS-DCSK) com multiplexação por divisão de frequência ortogonal (OFDM), nomeadamente MC-CS-DCSK. Os resultados da simulação nos canais UWA mostram que o sistema proposto tem melhor desempenho que MC-DSSS, MM-DCSK e OFDM-CS-DCSK.

Capítulo 4

Banco de Corridas e Pré-Processamento de sinais acústicos

Este capítulo objetiva discutir as transformações num banco de corridas que sejam necessárias para que o sinal de entrada possa ser mais bem processado durante a classificação. O capítulo também aborda as etapas do pré-processamento em si. Uma corrida é a gravação da assinatura acústica de um navio por um dado intervalo de tempo em condição estática ou dinâmica.

As transformações que precedem a classificação visam a organização do banco de corridas; que é o conjunto de diversas assinaturas acústicas de doze tipos de navios gravadas em movimento ou estáticas; e da apresentação desse banco. A assinatura acústica do alvo de interesse nesta dissertação é captada em hidrofones conjuntamente com o ruído, portanto se faz necessário retirar esse ruído do sinal, pois no caso do mesmo navio se encontrar em outra condição de mar ou com um ruído de fundo distinto daquele que foi gravado, ainda possa ser reconhecido. Assim, o sinal é contaminado por ruído e, portanto, é necessário extraí-lo realizando um branqueamento no sinal.

O vetor de características utilizado é oriundo de uma Transformada rápida de Fourier (FFT). Deve-se determinar o número de pontos utilizados na FFT, trabalhando com o conteúdo espectral e não no domínio do tempo. Para ser coerente, a determinação temporal do comprimento do vetor, que será entendido como uma unidade de assinatura acústica, é imperativa. Determina-se, quanto tempo de corrida, deverá ser entendido como uma realização do sinal. O banco de corridas é ainda desbalanceado, possui um quantitativo de corridas diferentes por navio, assim um classificador tende, na dúvida entre duas assinaturas, eleger a mais frequente, mascarando seu real poder discriminatório. Por isso, é necessário o balanceamento do quantitativo de sinais, tornar o banco equiprovável,

ou utilizar um ponderador para fornecer o resultado de classificação de acordo com a probabilidade de ocorrência das classes. Visando inserir sinais no futuro, balancear o bando de dados é o método escolhido, pois, cada vez que se inserisse um sinal, precisaria que fossem recalculadas as probabilidades.

Este capítulo se inicia com uma breve análise dos dados relevantes e as etapas necessários ao pré-processamento. O método de pré-processamento é originário da tese de Soares Filho[16], e destaca-se a técnica *Two-Pass split window* (TPSW), responsável pelo branqueamento espectral. Por fim, analisa-se o resultado das transformações no banco de corridas e o pré-processamento que resultará no *dataset* ou banco de dados, conforme o jargão da área de ML e Inteligência Artificial (AI), possibilitando o uso do *dataset* nos algoritmos de ML.

4.1 Metodologia de Análise dos Dados de Entrada

A análise de dados é um processo não-linear e iterativo [5], onde, a cada iteração, busca-se entender a correlação entre os dados, isto é, padrões e tendências que tornam previsíveis as assinaturas acústicas de um mesmo navio. Uma forma de agrupar esses dados num conjunto que, quanto mais denso for, mais preciso estará. Objetiva-se, a partir de tais características, uma representação mais compacta e amigável dos dados, ou seja, obter conjuntos reduzidos dos atributos mais significativos dos dados.

Cada assinatura acústica de cada classe y_i é descrita como um vetor \mathbf{x} no espaço dimensão N , tal que, $\mathbf{x} = [x_1, x_2, \dots, x_N] \in \mathbb{R}^N$. O objetivo é encontrar uma representação alternativa que contenha características proeminentes, $k_i, 1 \leq i \leq M$, tal que $M < N$, onde $K = \zeta[\mathbf{x}]$ e $\mathbf{x}' = \zeta^{-1}[K]$, e ζ é um operador matricial, que realiza um mapeamento elementos de um espaço para produzir outros elementos do mesmo espaço. Conseqüentemente, o vetor de características é descrito por $\mathbf{k} = [k_1, k_2, \dots, k_M] \in \mathbb{R}^M$. Supõe-se que há um erro na representação, ϵ , onde $\epsilon = \|\mathbf{x} - \mathbf{x}'\|$, e o objetivo é que esse erro seja o menor possível. Esse processo não é assintoticamente monótono, pois se pode encontrar um mínimo local, o que limita o decréscimo do ϵ e torna necessária uma parada antecipada, e por causa dos parâmetros inseridos na busca, mais especificamente a taxa de aprendizagem, pode não ser grande o suficiente para um incremento, que retire o gradiente deste vale, por isso, várias iterações, com pontos de partida diferentes. O gradiente da função de erro em relação aos parâmetros podem, às vezes, se tornar muito pequenos (*vanishing gradients*) ou muito grandes (*exploding gradients*), culminando num

processo de otimização que pare ou divirja, respectivamente. A não-linearidade se deve ao fato de que, após um certo número de iterações, busca-se uma outra condição inicial, na esperança de iniciar num ponto próximo do mínimo global e distante de vales [5].

Numa visão holística [5], a análise dos dados de entrada visando a classificação pode ser entendida conforme a figura 4.1. Suas etapas principais são: *Formulando e refinando a questão*; *Explorando os dados*; *Construindo modelos estatísticos formais*; *Interpretando os resultados*; e *Comunicando os resultados*.

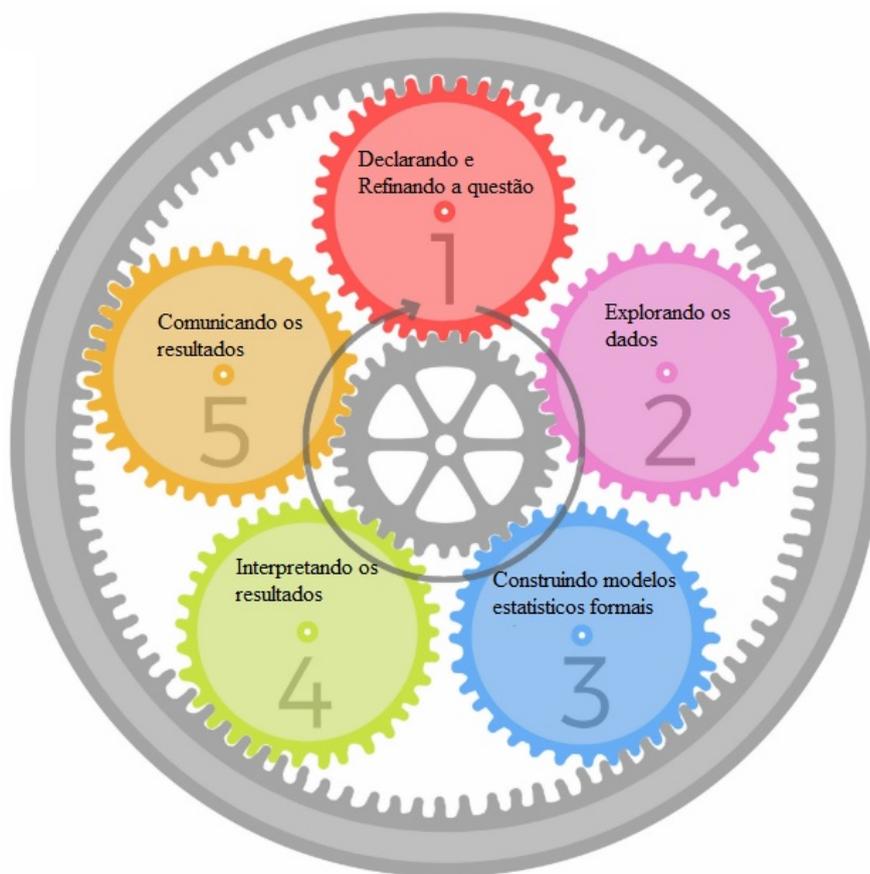


Figura 4.1: Epiciclo da análise de dados, adaptado de [5].

Declarando e refinando a questão é a primeira etapa. Parte-se do pressuposto que, através dos dados coletados, é possível responder a diferentes questões. Por exemplo, ao se recolher dados alimentares sobre a vida de um indivíduo é possível responder, quantas calorias consome, se se alimenta de modo saudável, quanto gasta com sua alimentação, quantas refeições faz ao dia, entre outros. Com o banco de dados instrumentalizado nesta dissertação, é possível estudar a assinatura acústica dos navios, mas também a paisagem acústica ambiente, o ruído ambiental.

Na etapa de *exploração dos dados*, faz-se uma representação gráfica de parte dos

dados, pois, devido à sua dimensão, em geral, é inviável representar graficamente todo o espaço no qual esses estão contidos. Isso obriga a utilizar técnicas apropriadas para esta representação. Notadamente são técnicas oriundas dos campos de Estatística, como medidas de tendência central e dispersão; e de Processamento Digital de Sinais, como amostragem, filtragem e Transformada discreta de Fourier (DFT). Essas técnicas permitem avaliar a dispersão dos dados, em seu espaço amostral, e auxiliam a representar uma porção dos dados que contenha as características importantes dos mesmos. Nesta dissertação, visualiza-se os dados em função do tempo e da frequência já que interessa a informação da banda de frequência do sinal de interesse.

Ainda na procura de melhor representar esse conjunto de dados, *construir modelos estatísticos* é uma ferramenta valiosa pois reduzem o banco de dados a um conjunto de equações. Quando são bem sucedidos, é possível reduzir um banco de dados a um conjunto de equações. Mesmo que sejam necessárias equações para mais de um subconjunto dos dados. A obtenção de tais modelos, facilita qualquer cômputo e a efetividade da representação. Nesta dissertação, buscou-se representar os dados da FFT sem processamento, com média e desvio padrão, para melhor entender o TPSW, conforme Figura 4.12.

A *interpretação dos resultados* é a força motriz na evolução dos modelos. Mesmo que, para um dado específico, um modelo não seja satisfatório, o entendimento da causa precede a sua correção. Assim, uma boa análise do que se pode interpretar e de suas falhas é imprescindível para a retroalimentação que visa tornar o modelo mais confiável.

A *comunicação dos resultados* é uma parte frequentemente subestimada. Essa etapa pode ser o início de um refinamento, pois a comunicação ao usuário final permite ser assertivo sobre a qualidade da análise e modelo produzido. Esse resultado também pode ser útil a outro sistema. Tomando o caso específico dos muitos sistemas que compõem um navio moderno, a fusão de dados dos vários sistemas é complexa, não por falta da informação disponível, mas pela adequação dela a vários sistemas. Assim, é preciso disponibilizar a mesma informação em vários formatos diferentes. Por exemplo, ao contra-atacar um míssil que se dirige para colidir com o navio, esse alvo poderia ser acompanhado por uma alça optrônica, que é um aparelho que fornece, detecta e controla luz com objetivo de acompanhar um alvo visualmente ou com câmera infravermelha. Sua marcação deve ser referenciada com a proa do navio, através da agulha giroscópica do navio. Essa informação deve alarmar nos consoles táticos do navio, esse míssil pode ser neutralizado por um canhão, por isso é importante a fusão de dados num sistema de sistemas, isto é, sistemas compostos por sistemas constituintes independentes, que agem juntos em direção

a um objetivo comum por meio do sinergismo entre eles, tal qual um navio moderno.

Nesta dissertação, os resultados finais serão apresentados individualmente e comparativamente visando perceber as especificidades de cada algoritmo de ML e as diferenças que culminam num melhor resultado.

Cada um dos processos de análise da dados pode ser pormenorizado em três passos cíclicos: *a definição de expectativas*, onde há uma ideia prévia do resultado das ações empreendidas; *a coleta dos dados*, que comprova, ou não, as expectativas; e *a comparação entre os dados coletado e as expectativas*, a fim de verificar se as expectativas são satisfeitas e como parte integrante do terceiro passo a revisão das expectativas reavalia os dados em conjunto com as premissas.

4.1.1 Aplicação da Metodologia de Análise dos Dados de Entrada

A aplicação desta metodologia, tal qual apresentada na seção 4.1, nesta dissertação, é realizada de forma que a questão a ser formulada, para análise de dados, na etapa de *Formular e refinar a questão* é a de definir quais e quantas características são necessárias.

Na fase de *exploração dos dados*, principalmente ao se trabalhar com dados reais, é necessária a remoção do ruído, e eleger conforme critérios quantitativos e/ou qualitativos qual é mais adequada, dentre as diversas técnicas possíveis, através do processamento digital dos sinais. Uma vez escolhida a técnica, para trabalho em laboratório, resta avaliar se ela responde em tempo hábil no caso de um confronto bélico.

Na terceira fase, *construir modelos estatísticos*, o modelo estatístico que mais se ajusta aos dados deve ser encontrado através de métricas e análises diversas. Há diversas possibilidades para extrair e agregar características, que são exploradas na fase de *interpretação dos resultados*. Há ainda diferentes técnicas de AI para classificação que devem ser avaliadas a partir de métricas e de indicadores de desempenho adequados para avaliar os resultados dos modelos. Finalmente, deve-se enfrentar a questão de como os resultados devem ser comunicados.

4.2 Banco de Sinais de Corridas

O banco de corridas com sinais acústicos foi cedido pela Marinha do Brasil, através da Organização Militar do Centro de Apoio a Sistemas Operativos em Arraial do Cabo (CASOP). Essa organização militar, possui hidrofones instalados no assoalho submarino,

gravando as assinaturas acústicas de navios, que percorrem a raia acústica em Arraial do Cabo (RJ). Esses sons são armazenados e catalogados num banco de corridas.

A metodologia de aquisição dessas assinaturas acústicas é peculiar. Inicialmente, os navios são gravados estaticamente, atracados e sem se movimentar. São as “corridas estáticas” que duram cerca de cinco minutos. Nessa fase, são gravadas as máquinas que estão sempre em linha, em funcionamento, no Navio, como Motor de Combustão Auxiliar (MCA), que é o gerador elétrico do navio e sempre estará em funcionamento e a Unidade de Resfriamento de Água (URA), o ar condicionado do navio que também sempre estará na linha, entre outros equipamentos vitais. Posteriormente, o navio segue para o mar para ser gravado em movimento. São as “corridas dinâmicas” com todas as suas configurações de máquinas possíveis na linha, por vezes, o navio tem mais de uma configuração de máquinas, por exemplo apenas um Motor de Combustão Principal (MCP) rodando ou ambos. Para cada uma dessas configurações, o navio tem uma assinatura acústica distinta. Em todos os casos, tem-se ainda o ruído ambiental e o do próprio equipamento de gravação presentes no sinal capturado.

O banco de dados de corridas para esta pesquisa possui 6,06 GB de dados e 4770 arquivos divididos em 709 pastas contendo 12 navios. Cada corrida foi digitalizada utilizando uma frequência de amostragem de 66 kHz aproximadamente e 16 bits para a quantização das amplitudes. O maior quantitativo de corridas diferentes por navio é de cinco unidades, alguns possuem apenas uma corrida. Algumas gravações, principalmente da fase dinâmica são longas com mais de quarenta e cinco minutos e algumas da fase estática são curtas com somente quarenta segundos.

4.2.1 Dataset

Para tornar todas as assinaturas acústicas equiprováveis é necessário aumentar o quantitativo de dados. Esse processo transforma o banco de corridas em um banco de dados ou *dataset*. A técnica escolhida foi a repetição com *deslocamento temporal*, na qual diferentes trechos do sinal são utilizados como diferentes entradas e casos. Cada navio teve sua assinatura acústica particionada, num tamanho padrão, com 2048 amostras, assim se criam vários extratos de todos os sinais representados no banco de dados.

Para tal fim, o banco de corridas foi representado como um tensor. Tensores são entidades geométricas introduzidas na matemática e na física que generalizam a noção de escalares, vetores e matrizes [57]. Assim, um tensor de dimensão zero seria um escalar, um de dimensão unitária é um vetor, um de dimensão binária é uma matriz, e extrapolando

esse conceito para a dimensão ternária seria como uma “matriz tridimensional”.

Visando atendimento de um ambiente operacional, escolheu-se trabalhar com 40 extratos de sinais diferentes do mesmo navio. Como, nesse banco de corridas, a menor ocorrência de um mesmo sinal é de quatro vezes e objetivando ter quarenta sinais de cada navio, então se necessita-se de 10 diferentes extratos do banco de dados.

Em cada extrato de representação, há pelo menos quatro corridas diferentes do mesmo navio. Em caso de inserção de mais um sinal no banco de dados, para pesquisas futuras, apenas $40 \times 93ms$, isto é, menos de 4 segundos de gravação são necessários para que o novo sinal seja também equiprovável em relação aos anteriormente adquiridos. Atende-se assim a uma inserção de sinal no banco de dados ágil.

Um tensor de três dimensões foi utilizado para representação. Duas delas são o quantitativo de sinais no banco $290 \text{ sinais} \times 2048 \text{ pontos da FFT}$. A terceira dimensão são os diferentes extratos dos sinais, deslocados temporalmente. Assim o tensor é composto por 10 vezes as duas primeiras. Uma “matriz de três dimensões” corresponde a um conjunto de 10 matrizes onde a primeira tem as 2048 $[0 - 2047]$ primeiras ocorrências dos 290 sinais, a segunda das 10 matrizes teria as 2048 seguintes, ou seja, as entradas $[2048 - 4095]$, e assim sucessivamente. Naturalmente, para um mesmo navio, cada sinal é fatiado e inicia ao término do anterior.

4.2.2 Implementação do Algoritmo de Balanceamento do Banco de Corridas

O banco de corridas original tem 290 sinais aproveitáveis no total, onde todas as gravações foram testadas, para verificar sua viabilidade como dado de entrada do pré-processamento. Alguns sinais oriundos de gravação estática, ou seja, somente com os equipamentos do navio, que em condições normais estarão em linha, e em condição dinâmica onde os navios estão se movimentando na raia acústica. O banco de corridas possui, por navio, um diferente quantidade de gravações, com diferentes durações.

Visando mostrar o processo que culminará num banco de dados equiprovável, segue um exemplo. A Figura 4.2 representa um banco de corridas desbalanceado com assinaturas acústicas de diferentes navios, com diferentes quantidades e diferentes durações. No caso, há três navios: o **A** com quatro assinaturas, o **B** com uma e o navio **C** com duas.

Banco de Dados - Original							
Sinal	Duração						
A 1	0s a 40 s	40s a 80s	80s a 120s	-	-	-	-
A 2	0s a 40 s	40s a 80s	80s a 120s	120s a 160s	160s a 200s	200s a 240s	240s a 280s
A 3	0s a 40 s	40s a 80s	-	-	-	-	-
A 4	0s a 40 s	40s a 80s	80s a 120s	-	-	-	-
B1	0s a 40 s	40s a 80s	80s a 120s	120s a 160s	160s a 200s	200s a 240s	240s a 280s
C1	0s a 40 s	40s a 80s	80s a 120s	-	-	-	-
C2	0s a 40 s	40s a 80s	80s a 120s	120s a 160s	-	-	-

Figura 4.2: Exemplo de banco de corridas com assinaturas acústicas de diferentes navios, com diferentes quantidades e de diferentes durações

O intuito é obter quatro assinaturas distintas de cada um dos navios, priorizando a diversidade. Para auxiliar neste processo serão implementados três contadores: um *contador de sinal*, um *contador de linha* e o *contador de banco de dados*. Do banco de corridas original criam-se sete banco de corridas segmentados, conforme a Figura 4.3.

Banco de Dados - Segmentado		Banco de Dados - Segmentado		Banco de Dados - Segmentado		Banco de Dados - Segmentado	
Sinal	Banco 1	Sinal	Banco 2	Sinal	Banco 3	Sinal	Banco 4
A 1	0s a 40 s	A 1	40s a 80s	A 1	80s a 120s	A 1	-
A 2	0s a 40 s	A 2	40s a 80s	A 2	80s a 120s	A 2	120s a 160s
A 3	0s a 40 s	A 3	40s a 80s	A 3	-	A 3	-
A 4	0s a 40 s	A 4	40s a 80s	A 4	80s a 120s	A 4	-
B1	0s a 40 s	B1	40s a 80s	B1	80s a 120s	B1	120s a 160s
C1	0s a 40 s	C1	40s a 80s	C1	80s a 120s	C1	-
C2	0s a 40 s	C2	40s a 80s	C2	80s a 120s	C2	120s a 160s

Banco de Dados - Segmentado		Banco de Dados - Segmentado		Banco de Dados - Segmentado	
Sinal	Banco 5	Sinal	Banco 6	Sinal	Banco 7
A 1	-	A 1	-	A 1	-
A 2	160s a 200s	A 2	200s a 240s	A 2	240s a 280s
A 3	-	A 3	-	A 3	-
A 4	-	A 4	-	A 4	-
B1	160s a 200s	B1	200s a 240s	B1	240s a 280s
C1	-	C1	-	C1	-
C2	-	C2	-	C2	-

Figura 4.3: Segmentação do banco de corridas.

Iniciando no **Banco 1**, conforme a Figura 4.3, buscam-se quatro assinaturas distintas para cada navio. No caso do sinal **A**, todas estão no **Banco 1**. O sinal **A1** é lido e registrado, então se incrementa o *contador de linha*, o sinal **A2** é lido e registrado e assim sucessivamente até o **A4**, quando chega no quarto sinal, o *contador de sinal* estoura, ou, em outros termos, alcança seu máximo valor. Ao estourar o *contador de sinal*, inicia-se a busca pelo próximo sinal, o sinal **B**, ao incrementar o *contador de linha* a comparação mostra que o sinal é diferente de **B**, no caso o sinal **C 1**, então se incrementa o *contador de banco de dados*, passando ao **Banco 2**. Mais uma vez o sinal **B1** é lido e registrado

sob o nome de **B1'** e assim sucessivamente até terminar no **Banco 4**, registrando o sinal **B1'''**. Ao se ler o quarto sinal, o *contador de sinal* estoura e busca-se o próximo sinal no primeiro Banco.

Seguindo então para sinal **C**, no **Banco 1**, há apenas duas assinaturas, o sinal **C1** é lido e registrado, então se incrementa o *contador de linha*, o sinal **C2** é lido e registrado ao incrementar o *contador de linha* a comparação mostra que o sinal é diferente de **C**, já que neste caso, não há sinal. Incrementa-se o *contador de banco de dados*, passando o **Banco 2**. O sinal **C1** é lido e registrado sob o nome de **C1'** e, após isso, o sinal **C2** é lido e registrado sob o nome de **C2'**. Termina-se assim o banco de dados agora equiprovável, onde todos os sinais possuem a mesma probabilidade de ocorrência, conforme a Figura 4.4.

Banco de Dados - Equiprovável	
A 1	0s a 40 s
A 2	0s a 40 s
A 3	0s a 40 s
A 4	0s a 40 s
B1	0s a 40 s
B1'	40 s a 80 s
B1''	80 s a 120 s
B1'''	120 s a 160s
C1	0s a 40 s
C2	0s a 40 s
C1'	40 s a 80 s
C2'	40 s a 80 s

Figura 4.4: Banco de dados equiprovável.

4.3 Pré-Processamento

O pré-processamento de dados é uma etapa básica e primária para converter dados brutos em informações úteis. Em geral, os dados brutos podem ser incompletos, redundantes ou ruidosos. Através do pré-processamento de dados, todos esses problemas mencionados podem ser resolvidos e usados para gerar modelos de aprendizado de máquina [58]. A Figura 4.5 mostra as etapas que foram seguidas nesta dissertação para converter dados num vetor de características que, de fato, traga informação sobre a identidade de cada navio.

A *discretização de dados* é um processo que envolve a conversão de dados contínuos



Figura 4.5: Etapas de pré-processamento de dados.

em discretos. Imprescindível para diversas tarefas de análise de dados, como criação de histogramas, simplificação de conjuntos de dados complexos ou preparação de dados para determinados tipos de algoritmos de aprendizado de máquina.

A *transformação de dados* envolve a conversão de dados de um formato ou estrutura para outro, muitas vezes para torná-los mais adequados para análise, armazenamento ou processamento. Esse processo inclui tarefas como normalização, dimensionamento, codificação de variáveis, preenchimento de dados ausentes e criação de novas características derivadas das originais.

A *limpeza de dados* é o processo de corrigir ou remover dados incorretos, corrompidos, formatados incorretamente, ou duplicados em um conjunto de dados. Se os dados estiverem incorretos, os resultados e os algoritmos não serão confiáveis, porém podendo parecer corretos. O processo de limpeza de dados varia de conjunto de dados para conjunto de dados.

A *integração de dados* refere-se ao processo de reunir dados de múltiplas fontes em uma organização para fornecer um conjunto de dados completo, preciso e atualizado.

Nesta dissertação, o pré-processamento é oriundo de um método consagrado e será apresentado na subseção 4.3.2. Os dados que serão pré-processados são oriundos de uma análise conhecida como LOFAR.

4.3.1 LOFAR

Os dados de entrada são oriundos da análise LOFAR, a cada sinal do banco de dados esta análise é aplicada. A análise consiste em uma representação espectral de banda larga, cobrindo a faixa esperada do ruído. A sequência da análise LOFAR é a seguinte: *Seleção de direção de interesse*, qual a direção se encontra o contato de interesse; *Processamento do sinal de entrada com a janela de Hanning*, para suavizar o vazamento espectral no início e final do vetor a ser analisado; *Processamento adicional do sinal resultante com a FFT*, que é o vetor de interesse; e esse resultado é disposto numa representação gráfica conhecida como Lofargrama, que possui esse nome por motivos históricos[59].

Um Lofargrama 4.6 é uma representação gráfica de frequência *versus* tempo em função da marcação, que é o termo naval para azimute. Cada ponto no eixo das abscissas, mostra uma frequência e no eixo das ordenadas, a duração do sinal na raia de frequência. A coloração denota a magnitude do sinal.

Na figura 4.6, nota-se um sinal contínuo de frequência em aproximadamente 2200 Hz e outro em 2700 Hz, indicando uma ruído de máquina nestas frequências. O azul escuro, que marca 0 dB, é a paisagem acústica ambiente. Quanto mais “quente” (isto é, vermelho, laranja e o amarela) for a cor da raia de frequência, maior é a perturbação nesse ambiente. Percebe-se ainda que não há marcação nesta apresentação gráfica, pois esse *display* é apresentado para uma dada marcação, termo naval, para azimute, de uso mais comum.

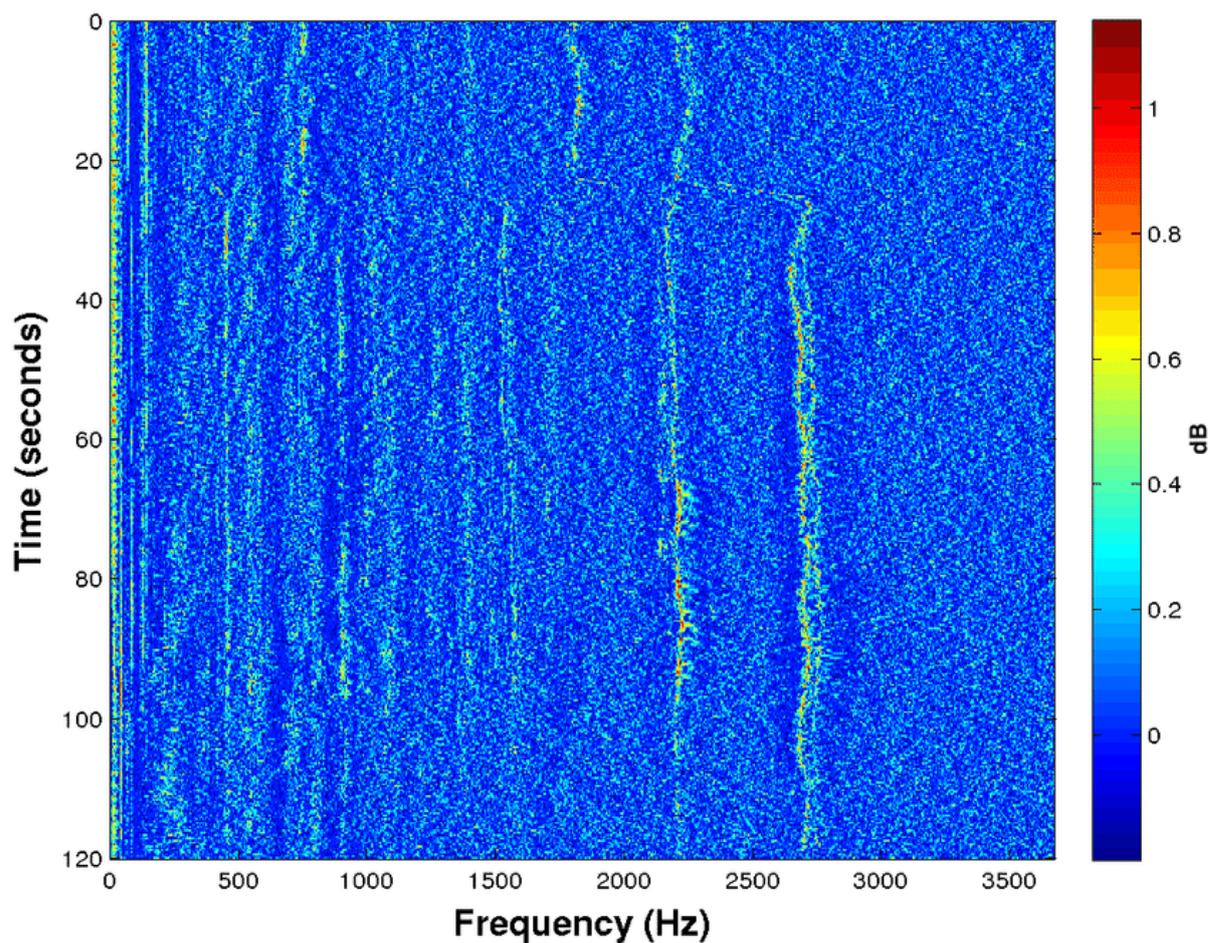


Figura 4.6: Exemplo de Lofargrama com frequência, tempo e intensidade [6].

4.3.2 Pré-Processamento de Sinais Acústicos

O pré-processamento visa adequar o sinal LOFAR, apresentado na seção 4.3.1, pois esse é adquirido em um ambiente ruidoso. Esse método de pré-processamento foi desenvolvido em [16]. A Figura 4.7 elucida o processo de branqueamento do sinal adquirido.

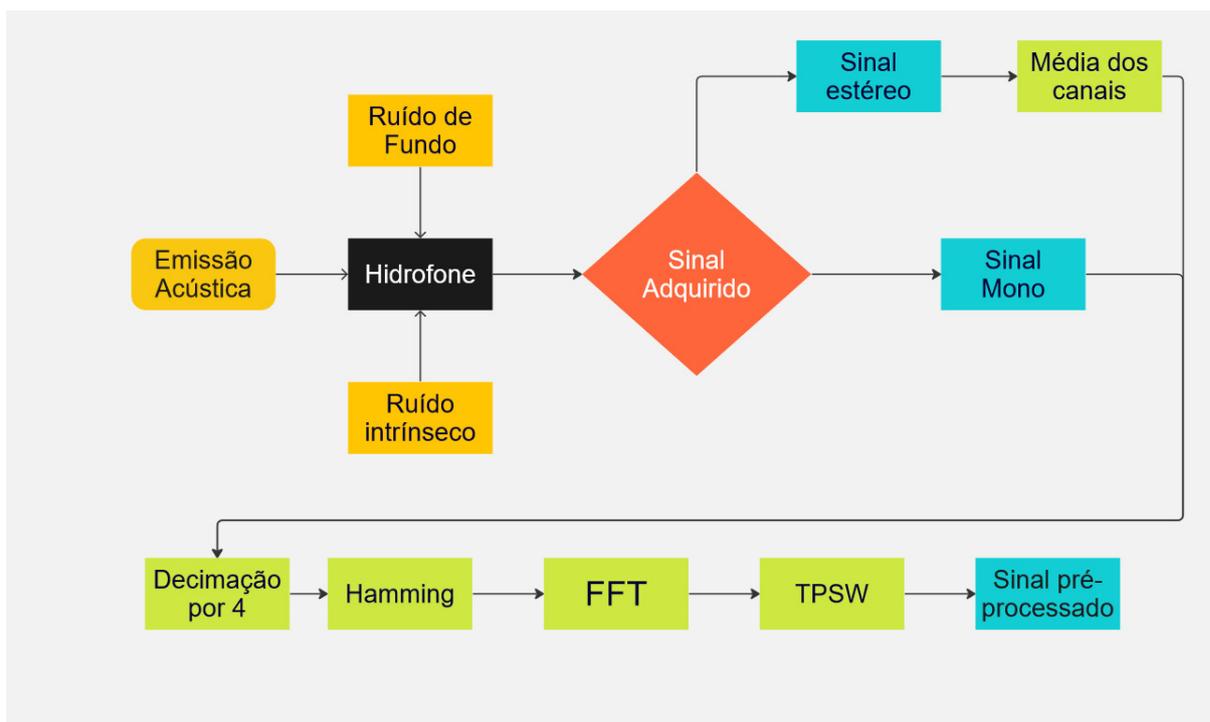


Figura 4.7: Fluxograma do pré Processamento LOFAR.

A figura 4.8 ilustra um sinal captado pelo **Hidrofone**, que é um transdutor sonoro subaquático. Ao gravar esse sinal, grava-se também o ruído ambiente, bem como o ruído intrínseco ao equipamento. O **Sinal adquirido** pode ser gravado em canal **mono** ou **estéreo**. No caso do sinal estéreo faz-se uma **média dos canais**, conforme apresentado na Figura 4.7.

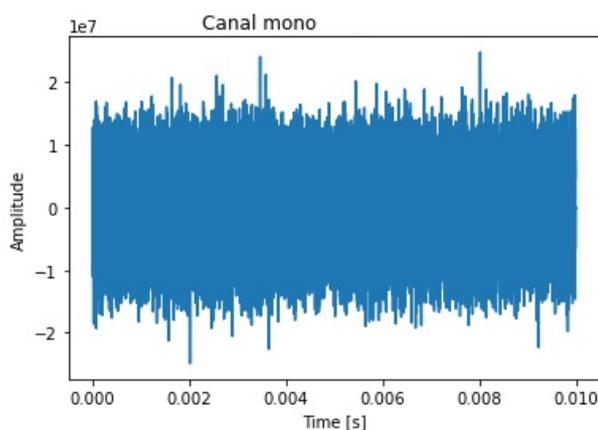


Figura 4.8: Áudio captado pelo hidrofone quando há passagem de um navio.

O bloco de **Decimação por 4**, conforme Figura 4.7, a cada quatro amostras, retém uma. Visa reduzir a resolução temporal do sinal, conseqüentemente, converte-se arquivos de áudio com alta taxa de amostragem em uma taxa de amostragem menor, economizando

espaço de armazenamento e facilitando um pré-processamento online, porém, mantendo-se características do sinal.

O janelamento suaviza o impacto do vazamento espectral, presente em qualquer sinais de duração finita no domínio da frequência, ademais inserido pela **Decimação por 4**. A janela afunila as bordas do sinal, reduzindo a amplitude no início e no final do sinal. Essa redução gradual ajuda a minimizar o vazamento espectral e, assim, potencializa análises de frequência. Aplicamos a janela de Hamming, conforme mostrado na Figura 4.9, de tamanho N , multiplica-se o pontualmente o vetor original x_n ponto-a-ponto por w_n , conforme está definida para cada ponto n através da equação a seguir.

$$w(n) = 0,54 - 0,46 \cos\left(\frac{2\pi n}{N}\right). \quad (4.1)$$

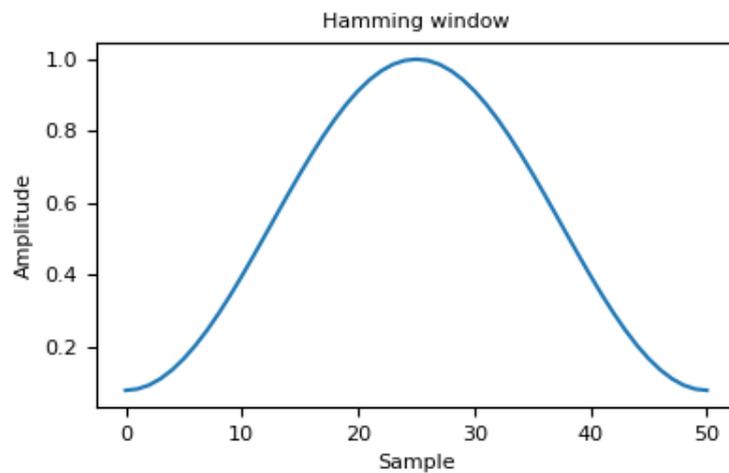


Figura 4.9: Janela de Hamming.

Após o janelamento, é aplicada a FFT ao sinal janelado transformando o sinal para o domínio da frequência. É de particular interesse a magnitude do sinal da FFT, pois será dado de entrada para o TPSW.

4.3.3 TPSW

Grandes picos no espectro de magnitude indicam a presença de altos valores de ruído e sinais. Para facilitar a busca dos sinais, o ruído de fundo deve ser branqueado, geralmente na altura da unidade. O bloco TPSW [60], que também é conhecido como *the split three-*

pass mean [61], é uma técnica de branqueamento de ruído, na qual o ruído é estimado pelo contraste com o espectro, o que nada mais é que uma normalização.

Dada uma sequência $\mathbf{x} = [x_1, x_2, \dots, x_N]$, onde N é o comprimento, a estimativa do nível de ruído da amostra $X(k)$, $k = 1, 2, \dots, N$ é obtida a partir de duas janelas de vizinhança, uma à esquerda e outra à direita da amostra, conforme mostrado na Figura 4.10.

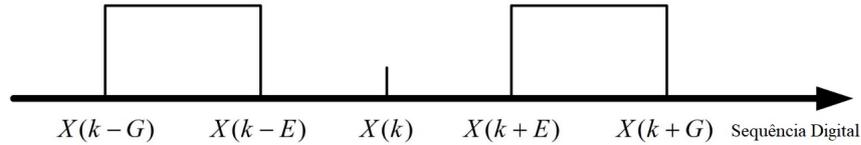


Figura 4.10: Diagrama esquemático do algoritmo TPSW.

Se as janelas têm comprimento $G - E + 1$ e estão centradas em k , os índices das amostras contidas nas das duas janelas, representadas na Figura 4.10, são:

$$R_k = \{k - G, k - G + 1, \dots, k - E - 1, k - E, k + E, k + E + 1, \dots, k + G - 1, k + G\}, \quad (4.2)$$

onde G e E são as bordas das janelas.

Para a situação em que k está próximo do início ou do final da sequência de dados, a janela correspondente é truncada, ou mesmo abandonada, nos limites da sequência. Então, o valor médio local dos elementos nas duas janelas é dado por

$$\tilde{X}(k) = \sum_{i \in \mathbf{X}_{R_k}} \frac{\mathbf{X}_i}{K}. \quad (4.3)$$

O ruído presente, r , é calculado implicitamente, na k -ésima raia espectral, a partir da média do sinal, das janelas adjacentes à essa raia $\mu(k)$. Cria-se uma sequência auxiliar $\Phi(k)$:

$$\Phi(k) = \begin{cases} X(k) & \text{Se } X(k) < r\tilde{X}(k) \\ \tilde{X}(k) & \text{Se } X(k) \geq r\tilde{X}(k) \end{cases} \quad (4.4)$$

onde $l = 1 + C[(4/\pi - 1)/A]^{1/2}$ é um regulador de limiar; C é uma constante maior ou igual a 1; e $A = G - E + 1$ é o comprimento de cada janela. A partir de $\Phi(k)$, obtêm-se os valores normalizados de TPSW,

$$Z_{TPSW}(k) = X(k)/\mu(k). \quad (4.5)$$

4.3.4 Sobre o Desempenho do TPSW

Visando elucidar o resultado obtido com o algoritmo TPSW, realizou-se uma comparação gráfica com outras técnicas de interpretação mais intuitivas. A partir da magnitude da FFT, apresentado na Figura 4.11 do sinal acústico (sem tratamento, a menos da decimação e do janelamento) foram avaliadas três técnicas de supressão de ruído: na primeira delas retirou-se todos os valores acima de três desvios padrão, σ , Figura 4.12; na segunda anulou-se todas as magnitudes menores que 1% do maior valor, mostrado na Figura 4.13; e finalmente o TPSW, visto na Figura 4.14.

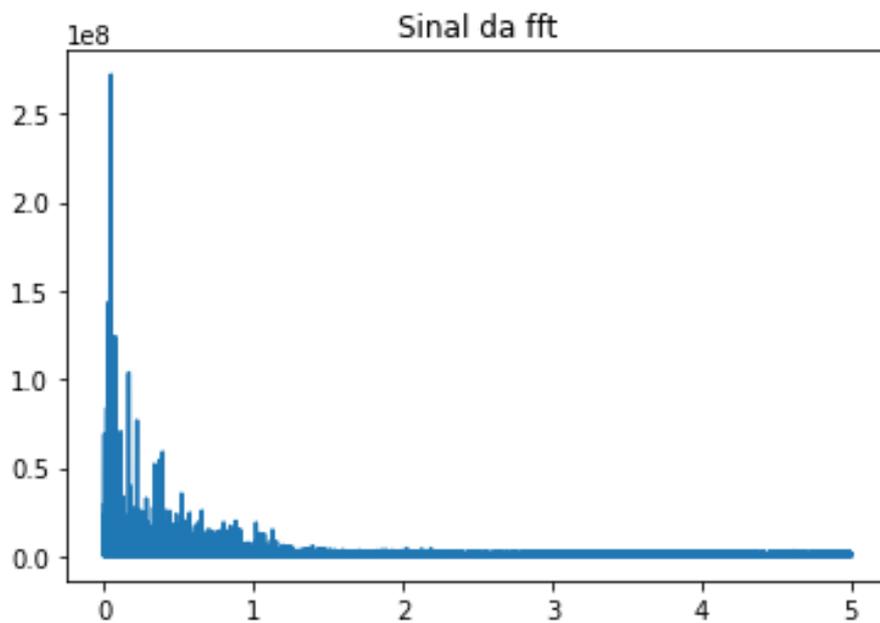


Figura 4.11: Periodograma do Sinal da figura 4.8.

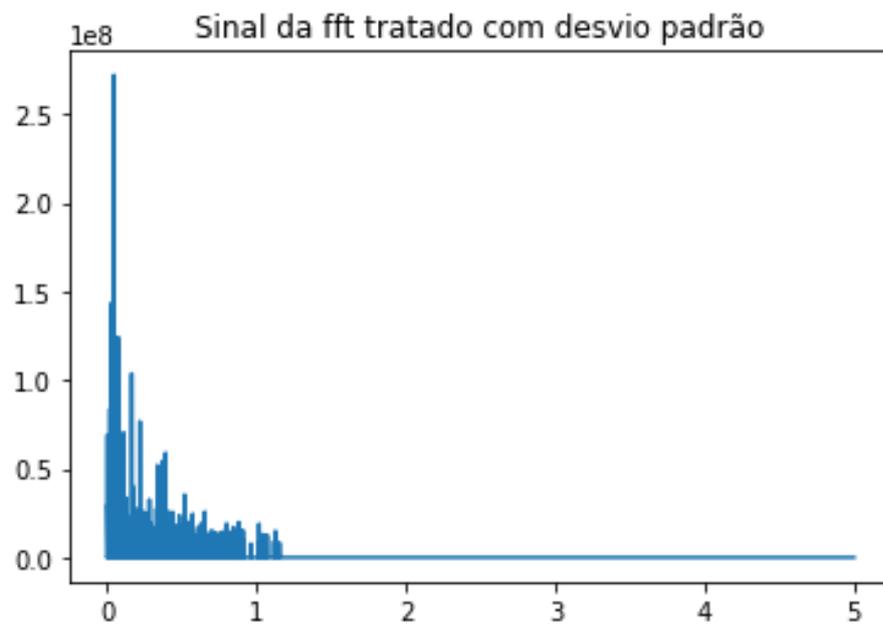


Figura 4.12: Periodograma do Sinal da figura 4.8 com dados retirados acima de 3 desvio padrão.

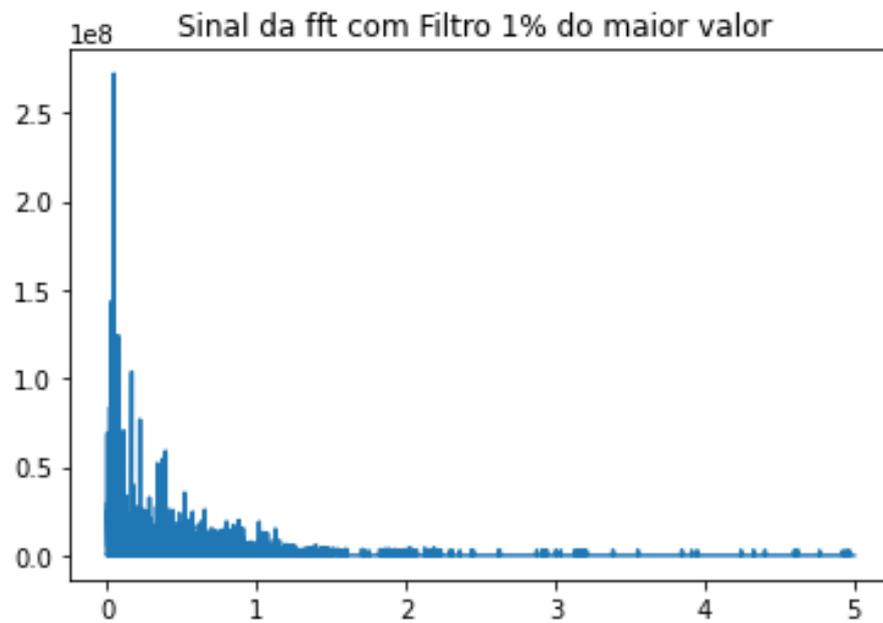


Figura 4.13: Periodograma do Sinal da figura 4.8 com dados retirados com magnitude abaixo de 1% do maior valor.

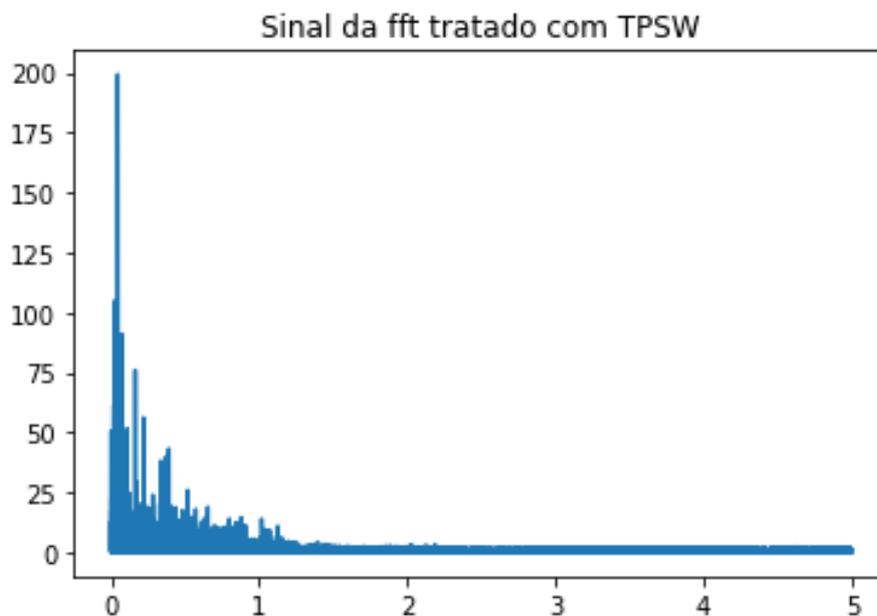


Figura 4.14: Periodograma do Sinal da figura 4.8 com TPSW aplicado.

Nota-se que a FFT tratada com a supressão de raias de frequência acima de três desvios padrões, visto na Figura 4.12; apagou todos os valores acima de uma dada frequência, o que não é interessante pois se perde a capacidade de discriminação.

A FFT com retirada de 1% do valor da magnitude mais alta encontrada, como visto na Figura 4.13; mantém alguns valores de alta frequência mas acabou com outros pontualmente, possuindo um desempenho superior ao anterior mas ainda apagando informação.

A FFT tratada com TPSW, visto na Figura 4.14; mostrou-se visualmente semelhante ao sinal da figura 4.11. Analisando-se a técnica, vê-se que essa não torna nula nenhuma raia de frequência, apenas leva a sua magnitude para a média das janelas adjacentes se o seu valor for maior que $r\tilde{X}(k)$, conforme explicado na seção anterior.

Na obtenção da FFT, foram testados comprimentos de 512, 1024, 2048, 4096, 8192, 16384 e 32768 amostras. Esse comprimento do vetor para a FFT será o mesmo utilizado para o comprimento da janela de Hanning. Essa diferença de tamanho de amostras, altera a largura de banda das raias frequências analisadas. O melhor resultado em termos de acurácia obtido foi com 2048 raias de frequência, o que corresponde a uma janela de duração de, aproximadamente, 93 ms. Para tal resultado, foram testadas a acurácia do mesmos sinais, submetidos aos mesmos algoritmos, com as mesmas configurações, apenas alterando o número de amostras da FFT.

4.4 DataFrame - TPSW

As transformações apresentadas, tanto no banco de corridas como no sinal LOFAR, explicadas ao longo deste capítulo, são implementadas na linguagem de programação *Python* e geram o *DataFrame - TPSW*, temo cunhado nessa dissertação. O resultado delas é gravado num *Python - Panda DataFrame* que é definido como um conjunto de dados tabulares bidimensionais, mutáveis em tamanho e potencialmente heterogêneos (o que traz muita flexibilidade). Possui uma estrutura de dados que contém eixos rotulados (linhas e colunas). As operações aritméticas se alinham nos rótulos de linha e coluna. A estrutura de dados primária do *Pandas*.

Pandas é uma biblioteca *Python* usada para trabalhar com conjuntos de dados. Possui funções para analisar, limpar, explorar e manipular dados. O nome *Pandas* faz referência a *Panel Data* e *Python Data Analysis* e foi criado por Wes McKinney em 2008 [62].

O arquivo *DataFrame - TPSW* do banco de dados equiprovável desta dissertação, possui 480 entradas, 40 para cada uma das doze classes. Possuindo em cada entrada um vetor de 2048 pontos advindo da magnitude transformada rápida de Fourier, o qual foi branqueado utilizando a técnica de *Two-Pass split window*, será denominado *DataFrame - TPSW*, e é a entrada adequada aos algoritmo de ML.

Capítulo 5

Processamento de Sinais acústicos

No capítulo 4, descreveu-se o pré-processamento de sinais acústicos de SoNaR. O sinal resultante de tal processo é aplicado na entrada do sistema classificador de embarcações, que é o tema central desta dissertação. Mais especificamente, são testados sete algoritmos de *Machine Learning* (ML), visando-se avaliar qual deles “entende” melhor o vetor de características, rotulando-o na classe de navio correta. Os algoritmos incluem: Análise de Discriminantes Lineares (LDA), Árvores de Decisão (Xgboost), K-Vizinhos Mais Próximos (KNN), Máquinas De Vetores De Suporte (SVM), Multi-Layer Perceptron, Regressão Logística e Regressão Ridge.

Este capítulo se inicia com a discussão dos aspectos metodológicos para validar o desempenho de um algoritmo de ML. Primeiramente, discute-se a necessidade de se particionar o banco de dados em conjuntos distintos, visando treinar uma máquina, validá-la e testá-la. Em seguida, comenta-se sobre o uso da técnica *K-fold Cross validation* que, a partir da formação de partições distintas do banco de dados, permite aumentar a confiabilidade sobre o desempenho dos classificadores desenvolvidos.

Em ML, classificadores e regressores são dois tipos fundamentais de algoritmos usados para diferentes tipos de tarefas preditivas. O Classificador é um tipo de algoritmo que, através de dados de entrada (\mathbf{x}), realiza uma classificação, o que significa atribuir uma classe y dentre o conjunto de classes predefinidas. O objetivo de um classificador é aprender um mapeamento das características de entrada para a determinação da classe correta (\mathbf{x}, y).

O Regressor é um tipo de modelo usado quando a variável alvo y possui valores contínuos. O objetivo de um regressor é aprender um mapeamento das características de entrada para uma variável de destino contínua. Tanto os classificadores quanto os re-

gressores podem ser treinados usando várias técnicas de aprendizado de máquina, como aprendizado supervisionado, aprendizado não-supervisionado ou aprendizado semissupervisionado, dependendo da disponibilidade ou não de dados rotulados.

O aprendizado supervisionado é um tipo de ML em que o modelo é treinado em dados rotulados, o que significa que os dados de entrada já estão marcados com a saída correta. O modelo aprende a mapear a entrada para a saída com base nos exemplos fornecidos durante o treinamento [63].

O aprendizado não-supervisionado é um tipo de ML em que o modelo é treinado em dados não-rotulados. O objetivo da aprendizagem não-supervisionada é encontrar padrões ou relações nos dados sem orientação explícita ou exemplos rotulados. Na aprendizagem não-supervisionada, o algoritmo tenta aprender a estrutura dos dados sem a necessidade de respostas rotuladas. Algumas técnicas comuns usadas na aprendizagem não-supervisionada incluem agrupamento *k-means*, agrupamento hierárquico, Análise de Componentes Principais (PCA) e incorporação estocástica de vizinho distribuída em t (t-SNE) [63].

Na aprendizagem semissupervisionada, o modelo é treinado em uma combinação de dados rotulados e não-rotulados. A disponibilidade de dados rotulados é muitas vezes limitada ou dispendiosa de se obter, enquanto os dados não-rotulados são mais abundantes. A aprendizagem semissupervisionada aproveita ambos os tipos de dados para melhorar o processo de aprendizagem e o desempenho do modelo. A ideia da aprendizagem semissupervisionada é que o modelo possa usar as informações dos dados não-rotulados para melhorar sua compreensão da distribuição dos dados e fazer melhores previsões sobre os dados rotulados. Ao incorporar os dados não-rotulados durante o treinamento, o modelo pode potencialmente generalizar melhor e alcançar um desempenho superior do que se fosse treinado apenas nos dados rotulados limitados[64].

Regressores podem ser avaliados usando métricas apropriadas para suas respectivas tarefas, como exatidão, precisão, recall, pontuação F1 e acurácia para classificadores e erro quadrático médio (MSE), erro quadrático médio raiz (RMSE) ou R^2 score.

Como critério de avaliação do treinamento das máquinas usa-se a métrica R^2 score ou acurácia, Acc, para classificadores. Com isso, obtêm-se uma estratégia para obter um algoritmo de classificação empregado em sintonia com o *dataset* ou banco de dados. Uma vez ajustada cada uma das máquinas, avalia-se o desempenho em função da taxa de acerto da classe estimada, através da métrica da acurácia Acc_t , que é o total de acertos sobre o total de estimações em várias iterações. A Figura 5.1 mostra esse processo.

Embora o desempenho de diferentes classificadores possa ser mensurado pelo R^2 score ou Acc, o objetivo final de um classificador é estimar a classe y correspondente ao dado em sua entrada \mathbf{x} , que é aquela com maior probabilidade de tê-lo gerado, ou melhor, que o classificador seja eficaz, ou seja, produza o maior número de acertos. Assim, uma vez treinado o modelo, para avaliar seu desempenho e saber a quem se deve uma dada assinatura, utilizamos a acurácia. A busca é pela máquina com maior efetividade, a máquina cujo R^2 score seja maior; e a máquina com maior Acc_t , que é a que mais vezes encontra o resultado correto.

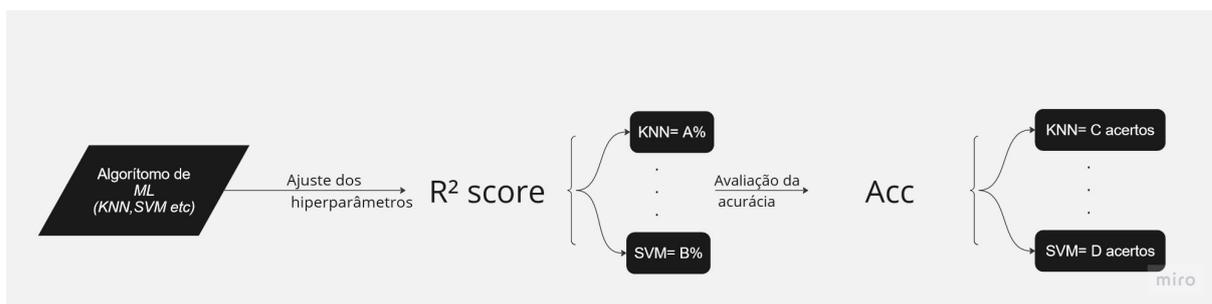


Figura 5.1: Ajuste de hiperparâmetros seguido por uma avaliação da acurácia.

São avaliados sete classificadores, que empregam as seguintes técnicas: *Multi-Layer Perceptron*, *K-Nearest Neighbours*, Ridge, Regressão Logística, *Support Vector Machine*, *Linear Discriminator Analysis* e XGBoost.

O *Multi-Layer Perceptron* (MLP) é um algoritmo de aprendizado supervisionado que aprende uma função $f(\cdot) : R^M \rightarrow R^O$, treinando em um conjunto de dados, onde M é o número de dimensões para entrada e O é o número de dimensões para saída. Considere um conjunto de características \mathbf{x} e um alvo y .

O *K-Nearest Neighbours* possui como princípio encontrar um número pré-definido de amostras de treinamento mais próximas do novo ponto a ser classificado e prever a classe y_p a partir delas. O número de amostras é definido como k . A distância pode, em geral, ser qualquer medida métrica, sendo que a distância euclidiana padrão é a escolha mais comum. Os métodos baseados em vizinhos são conhecidos como métodos de aprendizado de máquina não generalizantes, pois “lembram” todos os seus dados de treinamento.

A regressão Ridge resolve um modelo de regressão onde a função de perda é a função linear de mínimos quadrados e a regularização é dada pela norma ℓ_2 , também conhecida como Regressão de Ridge ou Regularização de Tikhonov.

A regressão Logística classifica, entre duas possibilidades de classe, atribuindo valores para classe predita y_p como 0 ou 1, ou seja, classe ou não-classe, usando a curva logística. É

um método estatístico para classificação binária. É modelada a relação entre uma variável binária dependente e uma ou mais variáveis independentes, estimando-se a probabilidade do resultado binário.

A *Support Vector Machine* (SVM) utiliza vetores de suporte para classificar, obtendo-se comparações entre os vetores de suporte a classe da entrada a ser predita. É capaz de realizar classificação binária e multi-classe em um conjunto de dados.

O LDA é um classificador com limite de decisão linear, que é gerado ajustando-se densidades condicionais de classe aos dados e usando o Teorema de Bayes. O modelo ajusta uma curva de estimada de densidade para cada classe, assumindo que todas as classes compartilham a mesma matriz de covariância.

O XGBoost constrói um modelo incremental e progressivamente permite a otimização de funções de perda diferenciáveis arbitrárias. Em cada estágio, n árvores de regressão são ajustadas de acordo com gradiente negativo da função de perda. A classificação binária é um caso especial onde apenas uma única árvore de regressão é induzida.

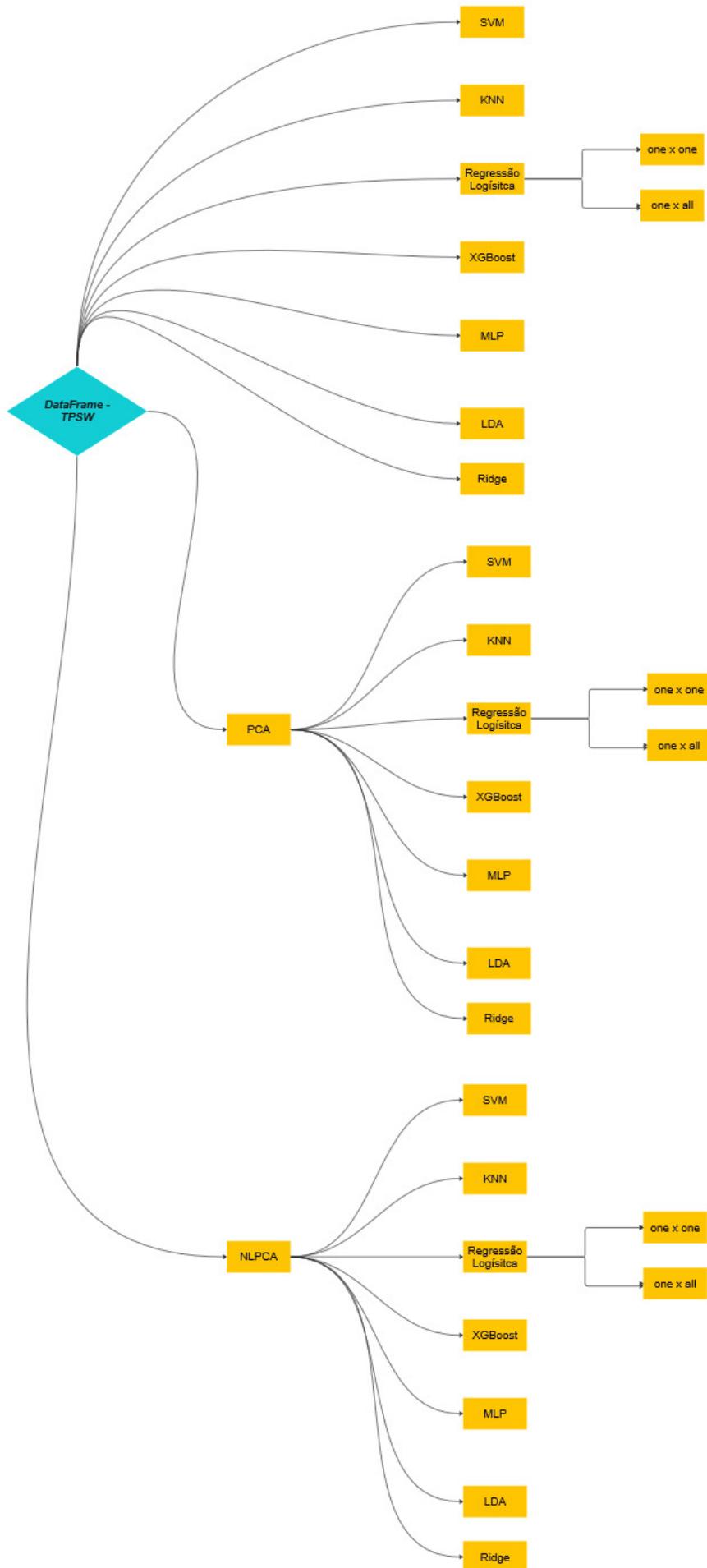
Dos sete algoritmos utilizados, a Regressão Logística não suporta sem o devido tratamento o caso multi-classe. Porém, pode ser utilizado se empregadas as técnicas conhecidas como *um contra todos* e *um contra um*, descritas na seção 5.2.

5.1 Aspectos Metodológicos

Nesta seção, são discutidas as escolhas tomadas para exploração dos dados e os motivos dessa predileção, com a divisão do banco de dados em três subconjuntos, validação cruzada e a otimização de hiperparâmetros através do Teorema de Bayes.

5.1.1 Ajuste e Avaliação de Modelos de ML

Diversos modelos de ML são avaliados para eleger qual deles obtém os melhores resultados a partir dos dados usando as três configurações de entrada: Dataframe-TPSW, Dataframe-TPSW com PCA e Dataframe-TPSW com NLPCA; para avaliar o mais eficiente e mais eficaz. A Figura 5.2 ilustra esses casos, considerando ainda o uso de estratégias *um contra um* e *um contra todos* que são discutidas na Subseção 5.2 adiante.



Numa visão holística, a função de um classificador é predizer corretamente a classe y , correspondente aos atributos em sua entrada, que é o vetor de características \mathbf{x} . Para isso, deve-se encontrar uma relação entre \mathbf{x} e y . O modelo responsável por tal feito é um regressor. Um classificador pode ser entendido como um tipo de regressor com uma camada adicional que mapeia entre o número obtido e a classe correspondente[63]. Logo, um classificador pode ser implementado como um regressor cuja saída é comparada a um limiar para apresentar 0 ou 1 como saída.

Assim, a princípio, o modelo que mais aprender a relação entre \mathbf{x} e y fornece a maior quantidade de respostas corretas. Para avaliar e comparar diferentes classificadores, de forma a dizer qual é o mais eficiente, utiliza-se uma métrica comum, o R^2 score ou acurácia dependendo do classificador.

Após a utilização da métrica correta para auxiliar o encontro dos hiperparâmetros que mais se ajustam ao conjunto de treinamento, através de uma otimização Bayesiana, treina-se a máquina com os hiperparâmetros otimizados. Ao ser treinada, a máquina torna-se pronta para receber um conjunto de atributos e classificá-los. Supondo que um submarino observe um sinal acústico em alto-mar e o classifique, essa informação será um forte apoio decisório, portanto o grau de confiabilidade dessa resposta necessita ser conhecida e, para isso, utiliza-se a acurácia Acc . De outra forma, quantas vezes esse classificador aferido através do R^2 score ou Acc , rotulou a variável dependente predita y_p em diversas iterações Acc_t . Esse processo está exemplificado conforme a Figura 5.1.

O bom ajuste da máquina ocorre em pontos entre o *overfitting* e *underfitting*. O *overfitting* ocorre quando um modelo aprende a ajustar muito bem os dados de treinamento, capturando ruído ou flutuações aleatórias nos dados em vez do padrão entre (\mathbf{x}, y) . Como resultado, o modelo tem um bom desempenho nos dados de treinamento, mas não consegue generalizar para dados não vistos. Uma máquina que enfrenta esse mal, o *overfitting* tem alta precisão nos dados de treinamento, mas baixa precisão na validação ou nos dados de teste.

As causas do *overfitting* incluem: modelos complexos com muitos parâmetros relativos à quantidade de dados de treinamento, má escolha do vetor de características contaminado de dados ruidosos ou irrelevantes, e ajuste insuficiente de parâmetros.

O *underfitting* ocorre quando um modelo é muito simples para capturar as relações entre \mathbf{x} e y . O modelo tem um desempenho insatisfatório nos dados de treinamento e validação, porque não consegue capturar os relacionamentos entre a variável dependente e a variável de destino. As causas de *underfitting* incluem: modelos muito simples ou

com poucos parâmetros para capturar a complexidade dos dados, dados de treinamento insuficientes ou representação inadequada de características, e uma escolha inadequada de hiperparâmetros.

O equilíbrio entre *overfitting* e *underfitting* envolve encontrar o nível certo de complexidade e regularização do modelo para alcançar um bom desempenho de generalização em dados desconhecidos. Técnicas de regularização e seleção, além de avaliação cuidadosa de modelos, são essenciais para mitigar esses problemas e construir modelos que generalizem bem.

5.1.2 Particionamento de Dados: Treino, Validação e Teste

Os algoritmos de ML precisam ser treinados para que possam ser utilizados. O Perceptron, que inaugurou a área de ML, surge da analogia com o neurônio humano e aprendizado, portanto emprega-se a mesma analogia para analisar o poder de generalização. A mente humana emprega processos de associação corriqueiramente e inconscientemente. Exemplificando, para explicar para uma criança o que é um tigre, basta dizer que é um gato maior e com listras. Por mais que essa descrição tenha uma série de imprecisões, ao se deparar com um tigre, num zoológico por exemplo, ou na representação deste felino num desenho ou *anime*, muito provavelmente a criança o identificará. Pode-se também explicar, mesmo sem ela ter visto o tigre, que uma pantera é um tigre todo preto e ao ver uma, ela também a identificará. Vemos que há uma grande capacidade de generalização a partir de uma pequena quantidade de informação. Imagina-se que tal poder de associação em ML é uma fronteira do conhecimento e da técnica, cada vez mais palpável. Resumindo, o objetivo de uma máquina de aprendizado é, através de um conhecimento prévio (treino), generalizar e adquirir novos conhecimentos.

Para treinar modelos clássicos de ML, é necessário dividir o banco de dados em mais de um conjunto. Pode ser em dois: treino e teste, no *k-fold*, ou em três: treino, teste e validação, como comumente é realizado.

O conjunto de treino oferece à máquina material de consulta e de aprendizado, a partir dos quais aprenderá, se estiver bem ajustada, relações entre a variável dependente \mathbf{x} (entrada) e a independente y (saída), isto é, aprenderá um modelo. O conjunto de validação permite avaliar se a máquina obteve boas relações entre \mathbf{x} e y , e, principalmente, se a mesma não está se especializando demais, ou, melhor dizendo, perdendo a capacidade de generalização.

O conjunto de validação faz com que a máquina possa testar as relações que ela aprendeu durante o treino em dados aos quais ela não tinha sido exposta antes. Por isso, recebe o nome de validação.

O conjunto de teste contém dados desconhecidos pela máquina durante o treinamento e a validação. Ele é aplicado para realizar um derradeiro teste e avaliar o desempenho do modelo obtido. Muito se discute sobre quais as razões adequadas para distribuição entre os três a partir de um conjunto de dados, muitos pesquisadores usam 70% para treinamento, 15% para validação e 15% para teste [64].

5.1.2.1 *K-fold Cross Validation*

Na validação cruzada *K-fold*, o conjunto de dados é dividido em K partições. Utilizam-se todas as partições, exceto uma, para treino e teste, com uma proporção muito maior para treino. A partição que sobra é utilizada para o teste. Esse procedimento é repetido k vezes. A figura 5.3 mostra tal princípio para $k = 5$.



Figura 5.3: Validação cruzada.

O *K-fold* permite um melhor ajuste, variando-se os conjuntos de treino e de teste. Supondo, na Figura 5.3, que numa iteração (correspondente a um *fold*) consegue-se um resultado muito melhor que com as outras combinações de subconjuntos (os outros *folds*), naturalmente essa seria a escolha da máquina a ser utilizada. Por outro lado, se todas as iterações resultam em desempenhos muito próximos, pode-se ter certeza que a máquina realmente tem uma performance em torno de uma média. A variação de desempenho entre os *folds* permite avaliar a confiabilidade dos modelos que podem ser obtidos. Em outros termos, se o desempenho da máquina varia muito ou não em função dos dados, desde que os mesmos sejam suficientemente representativos do problema e vice-versa.

Neste trabalho, após o ajuste dos hiperparâmetros, será utilizado *K-fold* com $k = 10$. Assim, a máquina terá uma acurácia (Acc) média, buscando uma confiabilidade maior e não dependente de uma conjunção aleatória de fatores, como, por exemplo, qual semente foi utilizada para divisão dos conjuntos no treino, influenciando quais entradas foram utilizadas para o treino e conseqüentemente a quais realizações entre as variáveis dependente e independente o classificador foi exposto.

5.1.3 Ajuste de Hiperparâmetros

Os hiperparâmetros definem como se dará o processo de treinamento e, conseqüentemente, o aprendizado de uma dada máquina. Cada uma das sete máquinas testadas nesta dissertação para classificação possuem seu próprio conjunto de hiperparâmetros, que serão discutidos no capítulo 6. Para obter modelos adequados, é necessário se ajustar os hiperparâmetros de cada modelo.

Nem todos os hiperparâmetros necessitaram passar por uma otimização, em parte por causa do problema, em parte devido ao pré-processamento realizado e às premissas utilizadas ao organizar o banco de dados nesta dissertação. Por exemplo, para o algoritmo *KNN*, o tamanho da folha é de 30, por padrão da biblioteca empregada na implementação em Python [65]. O algoritmo utiliza normalmente trinta elementos para entender o que é uma classe. No pré-processamento, foram definidas todas as classes com 40 amostras, por isso, o tamanho da folha que busca entender a diferença entre classes desbalanceadas e *outliers*, não agrega nada à acurácia do modelo, pois não há desbalanceamento.

No classificador XGBoost, por exemplo, que usa árvores de decisão em série, para se classificar a partir dos resíduos de regressão de árvores anteriores, a contribuição de cada árvore para a resposta final e a profundidade das ramificações das árvores são hiperparâmetros que, quando não são otimizados, levam à avaliação de padrões complexos.

Árvores muito profundas geram *overfitting*. Pois é possível sub-categorizar uma classe em agrupamentos cada vez menores mas que não necessariamente interessam à pergunta formulada; por exemplo, imaginando que já foi possível agrupar todos os integrantes de uma dada classe sob um ramo da árvore, não há ganho em seguir subdividindo esse ramo.

Árvores de decisão funcionam particionando recursivamente o espaço de entrada em regiões, com cada partição correspondendo a uma regra de decisão simples baseada em uma das características de entrada x_i . São formadas por nós e utilizam critérios de decisão.

5.1.3.1 Otimização Matemática

A otimização matemática é o processo de encontrar a melhor solução a partir de um conjunto de candidatos disponíveis para maximizar ou minimizar a função objetivo [66]. Geralmente, os problemas de otimização podem ser classificados como restritos (também chamados de problemas com restrição) ou irrestritos, caso possua ou não restrições para as variáveis de decisão ou solução. Em problemas de otimização irrestrita, uma variável de decisão, x , pode assumir quaisquer valores do espaço unidimensional de todos os números reais, \mathbb{R} . Um problema de otimização restrita pode ser denotado por [66]:

$$\min_{x \in \mathbb{R}} f(x), \quad (5.1)$$

onde $f(x)$ é a função objetivo.

A maioria dos problemas de otimização na vida real são problemas modelados como sendo de otimização restrita. A variável de decisão x estará sujeita a certas restrições que poderiam ser matematicamente igualdades ou desigualdades físicas. Portanto, problemas de otimização restritos ou problemas gerais de otimização podem ser expressos como

$$\min_x f(x) \begin{cases} g_i(x) \leq 0, i = 1, 2, \dots, m, \\ h_j(x) = 0, i = 1, 2, \dots, p, \\ x \in X, \end{cases} \quad (5.2)$$

onde $g_i(x) \leq 0, i = 1, 2, \dots, m$, são restrições de desigualdade e $h_j(x) = 0, i = 1, 2, \dots, p$, e são restrições de igualdade e X é o domínio de x .

Na área de ML, devido à dificuldade de otimizar matematicamente entre diversas possibilidade de hiperparâmetros, algumas metodologias para ajustar hiperparâmetros de ML são comumente empregadas: aleatória, estocástica e Bayesiana.

Na pesquisa aleatória, especifica-se uma grade no espaço de hiperparâmetros e, a partir

de métricas de desempenho, testam-se todas as combinações possíveis para determinar a de melhor desempenho. A pesquisa alcança o melhor conjunto de hiperparâmetros dentre os pontos da grade, mas não é eficiente já que testa todos os pontos da grade, especialmente quando o número de hiperparâmetros é grande e quando eles podem assumir valores contínuos[67].

Na pesquisa estocástica, utiliza-se o mesmo princípio da pesquisa em grade, um espaço de busca pré-definido para os hiper-parâmetros, mas seleciona-se um grupo de hiperparâmetros aleatoriamente em cada iteração dentro do espaço de hiperparâmetros fornecido, visando uma eventual ocorrência de uma configuração melhor, i.e., que obtenha um modelo treinado com melhor desempenho. Apesar de que tal abordagem possa ser vista simplesmente como tentativa e erro, essa abordagem evita mínimos locais[67].

5.1.3.2 Otimização Bayesiana

A otimização Bayesiana [68] é uma técnica baseada no teorema de Bayes para otimização de modelos. O Teorema de Bayes descreve a probabilidade a *posteriori* $P(A|B)$ de um evento A condicionado a um evento B ocorrer, dadas as probabilidades dos eventos individuais $P(A)$ e $P(B)$ e a probabilidade de um evento condicionado, $P(B|A)$, previamente conhecida, através da equação (5.3):

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \quad (5.3)$$

Esse teorema é um corolário do Teorema da Probabilidade Total[69]. Se $P(B|A)$, $P(A)$ e $P(B)$ são bem estimadas ou conhecidas, o teorema de Bayes descreve a probabilidade condicional de um evento ocorrer.

A busca pelo conjunto de hiperparâmetros que maximiza a figura de mérito escolhida como alvo necessita de um espaço de busca adequado para os hiperparâmetros individualmente, um subconjunto do espaço de busca geral; a determinação do subespaço adequado só pode ser alcançada pelo pleno conhecimento do significado do hiperparâmetro. Com o Teorema de Bayes, cria-se um modelo probabilístico para o conjunto de hiperparâmetros em função de uma métrica específica, no caso, R^2 *score* ou *Acc*, para selecionar iterativamente o melhor conjunto de hiperparâmetros.

Isso significa que, na otimização Bayesiana, determina-se o espaço de busca futuro, através de eventos (testes) que já ocorreram, atualizando as probabilidades estimadas conforme os testes são realizados. Conseqüentemente, tem-se uma pesquisa estocástica

na qual se ajusta a densidade de probabilidade do espaço de busca conforme o desempenho do modelo observado durante a própria pesquisa.

Além disso, na otimização Bayesiana, é necessário especificar um critério de parada. Nesta dissertação adota-se, inicialmente, um mínimo de trinta iterações e para-se na eventualidade de, após se testar dez pontos adicionais, houver um aumento menor que 1% na figura de mérito. No caso do R^2 score não se comportar de forma crescente, são adicionados mais 10 pontos até que ele passe a apresentar tal comportamento até o máximo de 100 pontos adicionais.

A otimização Bayesiana trata variáveis discretas como contínuas, depois aplica um método de otimização Bayesiana normal e finalmente arredonda o ponto contínuo sugerido antes das avaliações da função. A otimização Bayesiana é o método escolhido para esta dissertação no ajuste de hiperparâmetros.

5.2 Uso de Classificadores Binários para Problemas Multiclasses

Alguns classificadores têm uma estrutura que permite aplicá-los apenas em problemas binários, ou seja, classifica apenas dentre duas classes, y e \bar{y} (o inverso de y). Para estender o uso dessas técnicas ao problema multiclasses, são utilizadas estratégias *One × One* (Um contra Um) e *One × All* (Um contra Todos).

Na estratégia Um contra Um, possuindo um total de γ classes, treinam-se $[\gamma(\gamma - 1)]/2$ classificadores binários, isso quer dizer que cada classificador distingue entre duas classes específicas das γ classes possíveis. Por exemplo, se há três classes y_A , y_B e y_C , implementam-se três classificadores binários: $C_{y_A} \times C_{y_B}$; $C_{y_A} \times C_{y_C}$; e $C_{y_B} \times C_{y_C}$ [70]. Cada classificador binário faz uma previsão distinta. Classifica-se a entrada como pertencente à classe com mais votos nas saídas dos classificadores do conjunto de classificadores.

Diferentemente, na abordagem um contra todos treinam-se γ classificadores binários. Cada um deles distingue entre uma classe e as demais. Por exemplo, para um problema de três classes y_A , y_B e y_C , implementam-se três classificadores binários: $C_{y_A} \times (C_{y_{(B+C)}})$; $C_{y_B} \times (C_{y_{(A+C)}})$; e $C_{y_C} \times (C_{y_{(A+B)}})$. Essa abordagem requer que cada modelo preveja uma probabilidade de pertencimento à classe ou uma pontuação semelhante à probabilidade. O argumento máximo dessas pontuações, que é o índice de classe com a maior pontuação, é então usado para prever uma classe [70].

5.3 Redução de Atributos

Conforme a Figura 5.4, o *DataFrame-TPSW* puro, o *DataFrame-TPSW* aliado ao Análise de Componentes Principais (PCA) e o *DataFrame-TPSW* aliado ao Análise de Componentes Principais não-linear (NLPCA) são aplicados como entrada em algoritmos de ML, em um total de sete máquinas distintas testadas, e registra-se os resultados para avaliar e comparar seu desempenho. Com o intuito de verificar a possibilidade de redução do vetor de características de entrada, aplica-se o *DataFrame-TPSW* aliado ao PCA e *DataFrame-TPSW* aliado ao NLPCA, registrando os resultados e avaliando seus impactos na acurácia das máquinas resultantes.

Visando responder às perguntas declaradas na Seção 4, se é possível obter técnicas que permitiriam uma resposta adequada e em tempo hábil, no caso de um conflito bélico, são avaliadas duas técnicas que visam diminuir a dimensão do vetor a ser analisado pelo modelo de ML. Parte-se de classificadores que recebem com entrada o vetor *DataFrame-TPSW*, cujas especificidades são descritas no início do Capítulo 4.

Uma PCA permite a reconstrução do sinal $\mathbf{x}' = (\text{operador})^{-1}\mathbf{k}$ com um erro de representação mensurável $\epsilon = \|\mathbf{x} - \mathbf{x}'\|$ usando menos elementos, $\mathbf{k} = [k_1, k_2, \dots, k_M] \in \mathbb{R}^M$ pois $\mathbf{k} = \text{operador}(\mathbf{x})$ e $\mathbf{x}' = (\text{operador})^{-1}\mathbf{k}$ que os da entrada ($\mathbf{x} = [x_1, x_2, \dots, x_N] \in \mathbb{R}^N$) de quem se extraiu as M componentes principais, por isso, diz-se que é compressiva. Complementarmente, testa-se a técnica conhecida como NLPCA, que deriva da anterior, mas como se aplica um processamento linear do vetor de características fornecido pela NLPCA, não é possível a reconstrução do sinal. Essas técnicas possivelmente permitirão reduzir a complexidade do modelo ou aprender mais rapidamente, embora isso não necessariamente mude tempo de resposta do modelo a uma entrada, pode torná-lo mais rapidamente de forma retreinável.

5.3.1 Comprimento do Vetor de Características do Lofargrama

A assinatura acústica do maquinário do navio, situa-se em frequências de até 5,0 kHz [71]. Trabalhar na banda correta elimina um processamento desnecessário para remoção de ruído e raias frequenciais que não agregam informação para a resolução do problema considerado-se o que está descrito em [71].

O sinal de áudio é digitalizado através da técnica de amostragem, onde a forma de onda analógica é representado por um vetor discreto. Essas amostras são iguais ao valor do sinal em instantes determinados, que são os instantes de amostragem. O intervalo de

tempo entre amostras chama-se intervalo de amostragem T_s . O seu inverso é a frequência de amostragem $f_s = 1/T_s$ amostras por segundo. A frequência de amostragem mínima chama-se frequência de Nyquist e, deve ser maior que o dobro da frequência máxima contida no sinal. A taxa de amostragem utilizada foi de 1411 ou 2116 kbits/s, com 16 bits para variação de volume, permitindo frequências de gravação de 44 kHz e 66 kHz, respectivamente.

Para representar o espectro do sinal, conforme descrito no capítulo 4, foram utilizados 2048 pontos para representar a totalidade da banda de frequência. Para cobrir a banda de 0 a 6 kHz, uma faixa um pouco maior que a do sinal de interesse, no caso de 66 kHz, em que 2048 pontos representam 44 kHz, logo 187 primeiros pontos representam 6 kHz. Semelhantemente, para 44 kHz, os 280 primeiros pontos representam 6 kHz. Visando abarcar os dois casos e ainda taxas de amostragem maiores no futuro, arbitrou-se trabalhar com as primeiras 400 raiais de frequência.

Ainda assim, considera-se que 400 seja um número razoavelmente grande de atributos, o que impacta na complexidade dos modelos classificadores em termos de parâmetros do modelo, os valores a serem armazenados, e das operações que o modelo deve realizar. Consequentemente, há a necessidade de uma grande quantidade de dados, para o treinamento dos modelos.

Aplicando-se técnicas de redução de dimensionalidade, que mantenham as informações mais relevantes, simplifica-se o problema e reduz-se a complexidade computacional, e torna-se possível treinar o modelo, a princípio, com menos dados. Para tal fim, avaliam-se as técnicas PCA e NLPCA, com 60 características principais, uma média de 5 características principais por classe.

5.3.2 PCA

PCA é uma técnica que permite reduzir o número de dimensões de dados representados num espaço vetorial. A técnica PCA obtém uma representação com um número menor de dimensões do que a dimensionalidade original dos dados de seu espaço. Isso facilita a visualização e a interpretação de possíveis padrões nos dados. Em situações em que as características dos dados são altamente correlacionadas, a PCA pode ajudar a decorrelacioná-las, a Figura 5.5 elucida esse processo tornando os dados mais adequados para modelos de regressão e classificação que assumem, em geral, a independência entre as características[72].

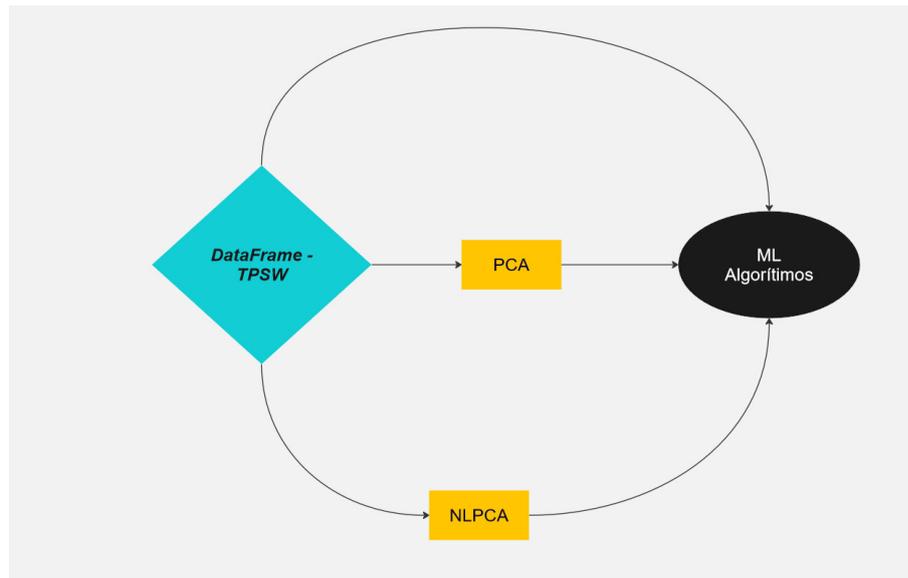


Figura 5.4: Fluxograma de Sinais para avaliação de diferentes técnicas de representação de sinal.

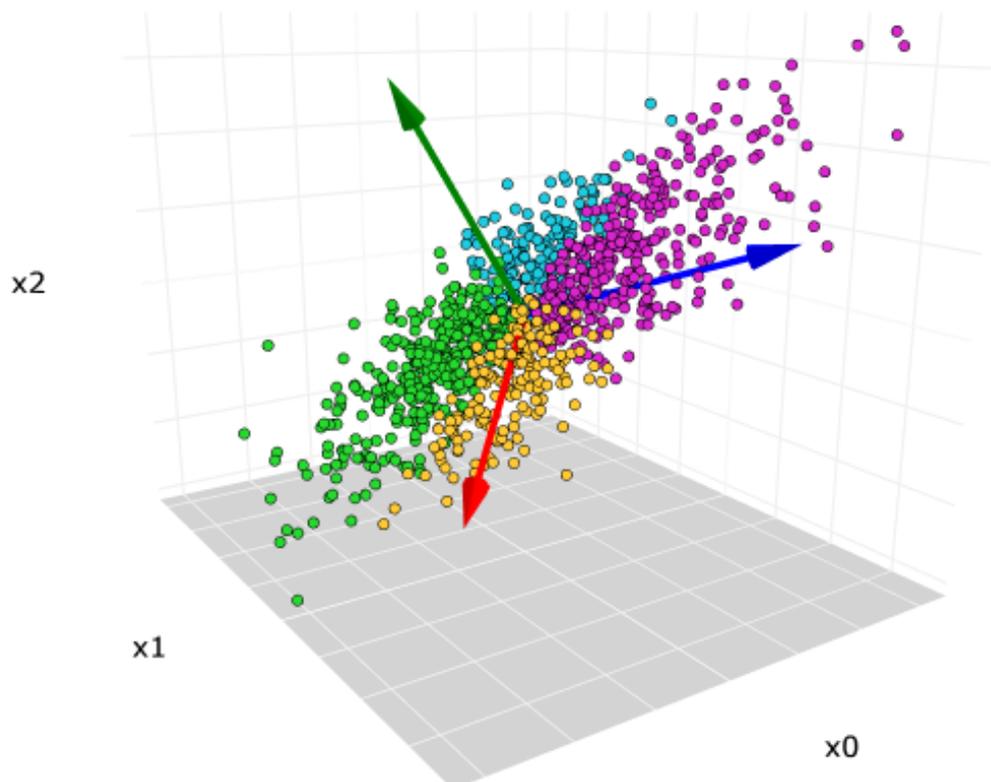


Figura 5.5: O conjunto de dados tridimensional, onde as setas vermelha, azul e verde indicam a direção do primeiro, segundo e terceiro componentes principais, respectivamente [7].

As limitações do PCA residem na suposição de linearidade, espera-se que os dados sejam gerados a partir de combinações lineares de elementos no espaço de dimensão reduzida, na necessidade de selecionar um número apropriado de componentes principais capazes de representar os dados com um erro definível, e na perda de interpretabilidade das características originais.

O algoritmo PCA pode ser implementado em quatro passos. O primeiro deles consiste em padronizar o conjunto de dados para garantir que cada coordenada tenha média 0 e desvio padrão unitário. O objetivo desta etapa é padronizar o intervalo das variáveis iniciais contínuas para que cada uma delas contribua igualmente na análise. Mais especificamente, a razão pela qual é fundamental realizar a padronização, é a sensibilidade do processamento realizado pelo PCA às variâncias das variáveis originais. Se houver grandes diferenças entre os intervalos das variáveis iniciais, aquelas com maiores excursões predominarão, o que levará a resultados tendenciosos [73]. Transformar os dados para escalas comparáveis evita esse problema. Matematicamente, isso pode ser feito subtraindo a média e dividindo pelo desvio padrão de cada valor de cada variável, i.e., seja x_i uma coordenada de \mathbf{x} , faz-se

$$z_i = \frac{x - \bar{x}_i}{\sigma_i}, \quad (5.4)$$

onde a média \bar{x}_i e a variância σ_i de x_i são obtidas a partir das amostras do vetor \mathbf{x} conhecidas.

O segundo passo é o cálculo da matriz de covariância entre as variáveis do vetor padronizado \mathbf{z} , i.e,

$$\sigma_{(z_i, z_j)} = \frac{\sum_{i=i}^n (z_i - \mu_i)(z_i - \mu_j)}{r - 1}, \quad (5.5)$$

para o que se consideram as r realizações de \mathbf{z} obtidas a partir das r realizações de \mathbf{x} .

O objetivo desta etapa é entender como as variáveis do conjunto de dados de entrada variam entre si, ou em outras palavras, verificar se existe alguma relação entre elas. A justificativa é que às vezes há variáveis altamente correlacionadas de tal forma que contêm informações altamente similares ou redundantes e, portanto, compressíveis.

A matriz de covariâncias é uma matriz simétrica de $n \times n$ elementos, onde n é o número de dimensões. Suas entradas são as covariâncias entre pares de variáveis padronizadas. Por exemplo, para um conjunto de dados tridimensional com 3 variáveis x_i , x_j e x_k , a

matriz de covariância é uma matriz de dados 3×3 de

$$K = \begin{bmatrix} \sigma(x_i, x_i) & \sigma(x_i, x_j) & \sigma(x_i, x_k) \\ \sigma(x_j, x_i) & \sigma(x_j, x_j) & \sigma(x_j, x_k) \\ \sigma(x_k, x_i) & \sigma(x_k, x_j) & \sigma(x_k, x_k) \end{bmatrix}$$

O terceiro passo para encontrar os componentes principais é o cálculo dos autovalores e autovetores da matriz de covariância. Seja K uma matriz quadrada $n \times n$ e \mathbf{v} um vetor diferente de zero para o qual

$$K\mathbf{v} = \lambda\mathbf{v}, \quad (5.6)$$

onde cada λ é um autovalor da matriz K e \mathbf{v} é seu autovetor correspondente.

As componentes principais são novas variáveis construídas como combinações lineares das variáveis iniciais. Tais combinações são feitas de tal forma que as novas variáveis, as componentes principais, não são correlacionadas e a maior parte da informação dentro das variáveis originais é comprimida na primeira componente, o segundo maior quantitativo de informação na segunda componente principal e assim sucessivamente. Geometricamente falando, os componentes principais representam as direções dos dados que explicam uma quantidade máxima de variância, ou seja, as direções que capturam a maior parte da informação dos dados. A suposição de uma relação entre variância e informação reside em que, quanto maior a variância transportada por uma linha (reta no espaço), maior será a dispersão dos pontos de dados ao longo dela, e quanto maior a dispersão ao longo de uma linha, mais informação será representada. Assim, os componentes principais são eixos de uma nova base na qual as diferenças entre as observações originais sejam mais visíveis.

Do exposto, o quarto e último passo é a construção do vetor de características. Calcula-se os autovetores e os ordenamos decrescentemente permitindo encontrar os componentes principais em ordem decrescente de significância. Então se pode escolher manter todos esses componentes ou descartar aqueles de menor significância de baixos autovalores, e formar com os demais uma matriz de vetores que se chama de vetor de características.

5.3.3 NLPCA

A Análise de Componentes Principais Não-Lineares de Kernel (Kernel NLPCA) é uma variante da Análise de Componentes Principais (PCA) que estende o algoritmo PCA básico para capturar relacionamentos não-lineares nos dados. O truque do *kernel*, comumente

usado em máquinas de vetores de suporte *Support vector Machine* (SVM) e outros algoritmos de aprendizado de máquina, é aplicado ao PCA para mapear os dados em um espaço de recursos de dimensão superior, onde relações não-lineares podem ser separadas linearmente. Essa característica é empregada para reduzir os atributos a ser avaliados do sinal *DataFrame-TPSW*.

O Kernel NLPCA amplia os recursos do PCA tradicional para lidar com relacionamentos de dados não-lineares. Como principal desvantagem do NLPCA, está a não-recuperação dos dados; em outros termos, não é possível reconstituir os dados originais. Enquanto isso, com o PCA é possível a reconstrução, porém com um erro que depende da quantidade de componentes principais empregada. Visando contornar algumas das deficiências do PCA, o NLPCA utiliza o *Kernel Trick* (o truque de Kernel).

O Kernel é definido via

$$k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle \quad (5.7)$$

na qual $\mathbf{x}, \mathbf{z} \in X$ e usa-se uma função de mapeamento $\phi : X \rightarrow \mathbb{R}^n$, o espaço de dados originais (X). Tem-se assim uma projeção a partir do mapeamento não-linear ϕ .

Com o *Kernel Trick*, projetam-se os dados em um espaço de alta dimensão, mas não se calcula explicitamente o novo espaço; calculam-se apenas os produtos internos, as projeções no novo espaço dos pontos de dados no espaço de características criado. Mapeamentos não-lineares das características numa dimensão maior, podem separar dados que são inseparáveis em dimensões mais baixas. A Figura 5.6 ilustra dados inseparáveis em dimensão 1, mas separáveis quando $X_2 = X_1^2$ neste espaço de características de dimensão superior.

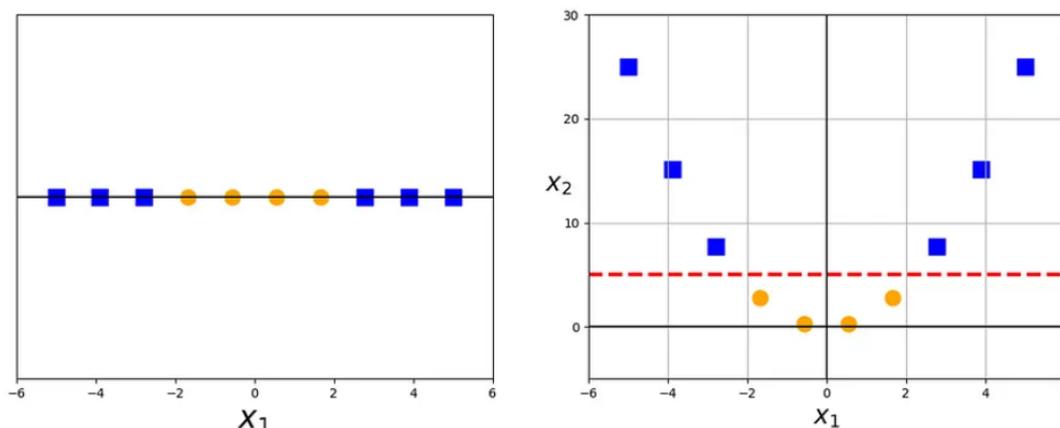


Figura 5.6: O truque do *kernel*.

Em aplicações reais, pode haver um banco de dados com muitas características e a aplicação de transformações que envolvem muitas combinações polinomiais desses recursos levaria a custos computacionais extremamente altos e impraticáveis. O truque do *kernel* fornece uma solução para esse problema. O truque é representar os dados apenas através de um conjunto de comparações de similaridade, produtos vetoriais, aos pares com os dados originais \mathbf{x} , relacionando-os com as coordenadas originais no espaço dimensional inferior, com as projetadas no espaço de dimensão superior. Ao invés de aplicar explicitamente as transformações, conforme mostrado na equação (5.7), representam-se os dados por essas coordenadas transformadas $\Phi(x)$ no espaço de características de dimensão superior. Esse espaço não é calculado explicitamente.

5.4 Medidas de Desempenho

Em aprendizado de máquina, algumas métricas são usadas para avaliar o desempenho de modelos e algoritmos. A escolha das métricas depende do tipo de tarefa que está sendo executada, como classificação ou regressão.

As métricas comuns usadas no aprendizado de máquina para classificação estão listadas a seguir.

- **Acurácia:** É a proporção de instâncias classificadas corretamente em relação ao total de instâncias;
- **Precisão:** É a proporção de previsões positivas verdadeiras entre todas as previsões positivas feitas;
- **Recall (Sensibilidade):** É a proporção de previsões positivas verdadeiras de todas as instâncias positivas reais;
- **Pontuação F1:** É a média harmônica de precisão e recuperação, proporcionando um equilíbrio entre os dois;
- **ROC-AUC:** É a característica operacional do receptor, que está relacionada a área sob a curva mede a área sob a curva ROC, que representa graficamente a taxa de verdadeiros positivos em relação à taxa de falsos positivos em várias configurações de limite; e
- **Matriz de confusão:** É uma tabela que mostra o número de verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos.

As métricas comuns usadas no aprendizado de máquina para regressão:

- **Erro Médio Absoluto (MAE)**: É a média das diferenças absolutas entre as previsões e os valores reais.
- **Erro Quadrático Médio (MSE)**: É a média das diferenças quadráticas entre as previsões e os valores reais.
- **Root Mean Squared Error (RMSE)**: É a raiz quadrada do MSE, fornecendo uma escala interpretável.
- R^2 (*score*): É a proporção da variância na variável dependente que é previsível a partir das variáveis independentes.

Para este trabalho serão utilizados R^2 *score*, acurácia e Matriz de Confusão.

5.4.1 R^2 Score

O coeficiente de determinação da classificação R^2 é definido como a unidade subtraída da razão do Erro médio quadrático (MSE) e Média da soma total dos quadrados (MST) [74]. Assim, o R^2 *score* é definido como

$$R^2 = \left(1 - \frac{MSE}{MST}\right) = 1 - \frac{\sum_{i=1}^m (y_p - \bar{y})^2}{\sum_{i=1}^m (y_r - \bar{y})^2}, \quad (5.8)$$

onde y_p é a classe predita e y_r é a classe real.

Na equação (5.8), MST é dado por

$$MST = \frac{1}{m} \sum_{i=1}^m (y_r - \bar{y})^2, \quad (5.9)$$

na qual, \bar{y} é a media dos valores reais,

$$\bar{y} = \frac{1}{m} \sum_{i=1}^m y_r. \quad (5.10)$$

O erro médio quadrático, MSE, é definido como

$$MSE = \frac{1}{m} \sum_{i=1}^m (y_p - \bar{y})^2. \quad (5.11)$$

Muitos estudos empregam o MSE [75, 76], Raiz do erro médio quadrático (RMSE) [77, 78, 79], ou Erro médio Absoluto (MAEr) [80, 81] e ainda Percentual do erro médio

absoluto (MAPE)[82, 83] para validar seus resultados. Embora sejam úteis, essas figuras de mérito partilham uma desvantagem comum, uma vez que os seus valores podem variar no intervalo $[0, +\infty)$, um único valor não explica o desempenho da regressão no que diz respeito à distribuição dos elementos verdadeiros, mas a comparação desse valor com algum padrão, pois qualquer resultado que não seja infinito, ainda estará infinitamente distante do ideal.

No caso do R^2 score, a melhor pontuação possível é 1,0 e ele pode assumir valores negativos, pois um modelo pode ser arbitrariamente pior do que a média, $(-\infty, 1]$. Assim existe um valor finito, que é 1, que representa o perfeito ajuste aos dados. Exemplificando, um modelo constante, em outras palavras, que sempre prevê o valor esperado, médio, \bar{y} , desconsiderando a entrada, receberia uma pontuação de 0,0 [84]. Nesse caso, não é em nada melhor do que prever a média de eventos.

O R^2 score mede o quanto da variável dependente pode ser explicada pelas variáveis independentes. Por exemplo, um valor de R^2 score igual a 0,60 indica que 60% da variabilidade na variável dependente (y) pode ser explicada pelas variáveis independentes (\mathbf{x})[85]. Assim, esse coeficiente é frequentemente usado como uma medida da qualidade de ajuste de um modelo de regressão linear. Um valor de R^2 maior geralmente sugere que o modelo se ajusta melhor aos dados.

Caso um estimador recebesse apenas entradas \mathbf{x} correspondentes a y , isto é, $\mathbf{x} = y$, então a melhor estimativa seria avaliar a frequência de ocorrência de cada classe e apostar sempre na mais frequente. No caso binário de classe y_A , ou classe y_B , essa estratégia seria apostar sempre no caso de maior probabilidade. Observa-se que este princípio pode ser estendido para o caso de multiclasse. Ao receber valores de \mathbf{x} , isto é, ao receber as variáveis dependentes, pode-se utilizar um classificador treinado com pares ordenados de (\mathbf{x}, y) para estimar corretamente a quantidade total de $R^2\%$ score atinente a classe y .

5.4.2 Acurácia

Para avaliação dos classificadores será utilizada a acurácia Acc . Ela é dependente da quantidade de acertos A e do total dos eventos T , sendo definida como

$$Acc = \frac{A}{T}. \quad (5.12)$$

Treinando-se um mesmo algoritmo ML, uma vez obtêm-se um escalar para o Acc . No entanto, a divisão dos conjuntos de teste, treino e validação é aleatória, mantêm-se o equi-

líbrio entre o quantitativo de sinais em cada conjunto, mas a divisão é aleatória. Então, provavelmente ocorrerá aprendizado de diferentes relações (\mathbf{x}, y) levando possivelmente a diferentes respostas para o regressor e, conseqüentemente, para Acc .

Uma maneira de contornar essa imprecisão é treinar o mesmo algoritmo com diferentes segmentações de conjunto de treino e teste. Para isso, utilizaremos o *k-fold*, apresentado neste capítulo. Ajustando uma distribuição conforme o histograma para a acurácia, de média \overline{Acc} , é possível se estimar o intervalo de confiança Acc_t da classificação, de um mesmo algoritmo ML, sendo treinado T vezes com um mesmo banco de dados. O intervalo de confiança da acurácia é função da média da acurácia \overline{Acc} e do desvio padrão z , onde

$$\overline{Acc} = \frac{1}{n} \sum_{i=1}^N Acc_i, \quad (5.13)$$

e o intervalo

$$\overline{Acc} \pm (z \times SE), \quad (5.14)$$

tal que z é o número de desvios padrão que um valor dista da média da distribuição normal, onde SE é o erro padrão do parâmetro estimado. Para um intervalo de confiança de 95%, z é aproximadamente 1,96.

Utilizando a acurácia como média do intervalo de confiança, obtemos

$$SE = \sqrt{\frac{1}{n} Acc(1 - Acc)}. \quad (5.15)$$

Portanto,

$$Acc_t = Acc \pm z \sqrt{\frac{1}{n} Acc(1 - Acc)}, \quad (5.16)$$

na qual, Acc é a acurácia medida em uma época de classificação, isto é, cada um dos *k-fold*, utilizando a metodologia de *cross-validation*, conforme descrito na subseção 5.1.2.1 e Acc_t é a média de Acc ao longo de todas as épocas.

Capítulo 6

Resultados

Neste capítulo, apresenta-se cada algoritmo usado para classificação, com uma breve explicação sobre seu funcionamento, com o intuito de apresentar os hiperparâmetros que controlam o processo de aprendizado de cada máquina, e que são otimizados nesta pesquisa. Apresentam-se os resultados dessas otimizações e ajustes e os hiperparâmetros que não necessitam ser otimizados para cada uma das máquinas. São descritos os seguintes algoritmos e seus hiperparâmetros: Análise de Discriminantes Lineares, Árvores de Decisão (Xgboost), K-Vizinhos Mais Próximos (KNN), Máquinas De Vetores De Suporte (SVM), Multi-Layer Perceptron, Regressão Logística e Regressão Ridge .

Duas técnicas de processamento de sinais aplicadas ao *DataFrame - TPSW* são investigadas, sendo uma de compressão, que é a PCA. Essa técnica determina as direções que melhor representam a distribuição dos dados e a outra é uma transformação não-linear, NLPCA, que é comumente vista como uma generalização não-linear da análise de componentes principais padrão PCA, ambas com apenas 60 direções principais, diminuindo em mais de 6 vezes o comprimento do vetor de características. Há o intento de levar para o classificador um vetor de características menor, contudo ainda altamente significativo, o que resulta numa menor complexidade para classificação e em menor tempo de processamento, conforme descrito no capítulo 4, o vetor de características nesse caso será

$$K = \zeta[\mathbf{x}], \mathbf{k} = [k_1, k_2, \dots, k_M] \in \mathbb{R}^M. \quad (6.1)$$

6.1 MLP

O perceptron foi a primeira rede neural descrita algoritmicamente [64]. Sua invenção por Rosenblatt, em 1957, denota a importância de conexão entre diferentes áreas do

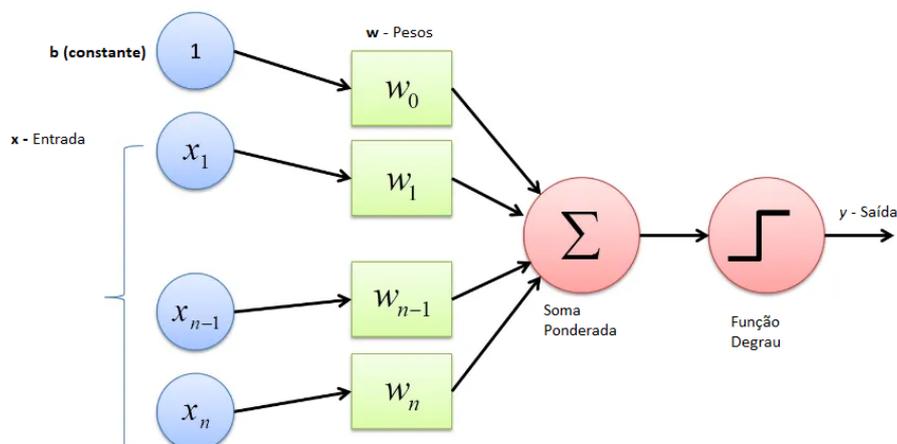


Figura 6.1: Perceptron, onde uma constante faz a curva de operação não passar pela origem e os pesos ponderam a importância de cada variável da entrada x_i .

conhecimento. Inaugurou e faz parte da área de *Machine Learning* (ML) inspirada na biologia e na aprendizagem humana[86].

O *Multi Layer Perceptron* (MLP) consiste em uma rede neural multicamadas totalmente conectada através de sinapses. Cada sinapse tem um peso distinto que indica a força da saída de um neurônio de uma camada na entrada de um neurônio da camada seguinte. Cada neurônio aplica uma função não-linear à combinação ponderada dos sinais em sua entrada para produzir a sua saída.

A Figura 6.1 exemplifica uma rede de um único perceptron. A ideia de Rosenblatt é que a saída y pode ser representada como uma combinação linear das componente do vetor de entrada $\mathbf{x} = [x_1, x_2, x_3]$ multiplicada por um peso \mathbf{w} e acrescida de um viés b ,

$$y = \begin{cases} 0 & \text{Se } \mathbf{w} \cdot \mathbf{x} + b < 0, \\ 1 & \text{Se } \mathbf{w} \cdot \mathbf{x} + b \geq 0. \end{cases} \quad (6.2)$$

A saída do neurônio é 0 ou 1, dependendo de um valor ajustado para o limiar. Basicamente, o perceptron é um decisor que pesa \mathbf{w} , diferentes condições \mathbf{x} , para fornecer uma resposta y .

O teorema da universalidade [87] afirma que um MLP com apenas uma camada oculta pode aproximar qualquer função contínua, naturalmente alterando o número de neurônios da camada oculta e ajustando os parâmetros das conexões entre os neurônios. A aprendizagem do MLP ocorre da seguinte forma:

1. Propagação direta: um *batch*, uma época, subconjuntos de amostras são apresentados um-a-um na camada de entrada e propagam-se para a camada de saída;
2. Cálculo do erro: para cada amostra no *batch*, na camada de saída, calcula-se o erro, que é a diferença entre o resultado previsto e o conhecido;
3. Retropropagação do erro: visando minimizar o erro, obtêm-se as derivadas em relação a cada peso, para cada amostra do *batch* e assim, estocasticamente, atualizam-se os pesos na direção contrária ao do erro médio;
4. Repetem-se essas três etapas (o que corresponde ao gradiente descendente estocástico) em várias épocas (varreduras de todo o conjunto de dados) para aprender os pesos.

Quando usados em problemas de classificação na camada de saída, é aplicado um limite ou uma função de comparação entre as saídas correspondentes a cada classe, para prever a classe correspondente a cada entrada.

O MLP funciona como uma combinação linear das entradas, com um viés adicionado que decide com base numa função degrau permitindo mudar grandemente a resposta com pequenos incrementos na entrada. Basicamente, o algoritmo perceptron executa uma soma ponderada das características de entrada, aplica uma função de limite e gera uma classe prevista. Por isso, nesta dissertação, ainda que o MLP não tenha boa performance, servirá para entender se os dados podem ou não ser representados por uma combinação linear, uma vez que o vetor de características, oriundo de uma FFT, provavelmente não o será.

6.1.1 MLP - Parâmetros Não-Otimizados

O ajuste de hiperparâmetros é um processo recursivo, então eleger previamente quais deles trazem a maior contribuição para a resposta salva recursos computacionais e de tempo. É perceptível que todo hiperparâmetro deve ser adequado ao problema e ao banco de dados, porém, alguns já apresentam valores *defaults* adequados para esta pesquisa.

1. Tamanho do lote: O número de exemplos de treinamento utilizados em uma iteração. Tamanhos maiores de lote podem aumentar a velocidade do treinamento, mas requerem mais memória;

2. Épocas: o número de vezes que todo o conjunto de dados de treinamento é transmitido para frente e para trás pela rede neural. Uma época consiste em um ciclo de passagens para frente, onde os pesos foram atualizados conforme a taxa de aprendizagem e o gradiente dos pesos e para trás, onde se observa o erro e quais pesos foram mais significativos no resultado; e
3. Inicialização de peso: A estratégia usada para definir os pesos iniciais da rede neural. Os métodos comuns incluem inicialização aleatória e inicialização *Xavier/Glorot* [65].

Os hiperparâmetros descritos acima alteram a forma de calcular a resposta, mas as restrições impostas a eles são mitigadas pela otimização bayesiana.

6.1.2 MLP - Otimização Bayesiana

Para construir um modelo de ML ideal, uma gama de possibilidades deve ser explorada. O ajuste de hiperparâmetros é o processo de projetar a arquitetura do modelo ideal com uma configuração ideal de hiperparâmetros. Os hiperparâmetros abaixo são otimizados para o aprendizado e obtenção do modelo e seus parâmetros, nesta dissertação, são:

1. *Hidden layer sizes* - Especifica o número de camadas ocultas e o número de neurônios em cada camada. A tabela 6.1 denota as correlações entre camadas e seu poder discriminatório.

Tabela 6.1: Parâmetros otimizados obtidos através de uma otimização bayesiana para o algoritmo MLP.

Número de camadas ocultas	Resultado
0	Capaz apenas de representar funções ou decisões linearmente separáveis.
1	Pode aproximar qualquer função que contenha um mapeamento contínuo de um espaço finito para outro.
2	Pode representar um limite de decisão arbitrário com precisão arbitrária com funções de ativação racional e pode aproximar qualquer mapeamento suave com qualquer precisão.
>2	Camadas adicionais podem aprender representações complexas, que é utilizada em algoritmos de otimização.

2. *Activation* - Define a função de ativação utilizada em cada neurônio da rede, particularmente. O espaço de busca considera a *relu* (Unidade Linear Retificada), a *tanh* (Tangente Hiperbólica) e a *logistic* (Sigmóide). A função de ativação muda a

resposta dos neurônios de forma não-linear e, conseqüentemente, o comportamento de aprendizagem da rede.

3. *alpha* - Termo de regularização da norma ℓ_2 usado para evitar *overfitting*. A regularização da norma ℓ_2 penaliza a ocorrência de pesos muito grandes para promover um modelo mais simples. Um valor alfa pequeno significa que a regularização tem força menor, enquanto um valor grande significa mais regularização. Tanto a norma ℓ_1 como a ℓ_2 combatem o *overfitting*, mas a regularização ℓ_1 combate o *overfitting* reduzindo os parâmetros para 0. Isso torna obsoletos alguns componentes da entrada ou combinações desta, se estão na camada oculta. A regularização ℓ_2 , ou Ridge (em problemas de regressão), combate o *overfitting* forçando os pesos a serem pequenos, mas não os tornando exatamente zero [88]. A equação (6.3) define a norma ℓ_1 e a (6.4) define a ℓ_2 .

$$\ell_1 = 1/N \sum_{i=1}^N (\hat{Y} - Y)^2 + \lambda \sum_{i=1}^N |W_i|, \quad (6.3)$$

$$\ell_2 = 1/N \sum_{i=1}^N (\hat{Y} - Y)^2 + \lambda \sum_{i=1}^N W_i^2. \quad (6.4)$$

4. *learning rate init* - É a taxa de aprendizagem, isto é, de atualização dos pesos na retropropagação. Se a taxa de aprendizado estiver muito baixa, o treinamento progride muito lentamente, mas se estiver muito alta, isso pode causar um comportamento divergente indesejável na função de perda. Além disso, têm-se as opções de comportamento *constante*, isto é, a taxa de aprendizagem é constante durante todas as épocas, *invscaling*, ou seja, a taxa de aprendizagem diminui gradualmente e de forma *adaptativa*, ou melhor, a taxa de aprendizagem é adaptativa.

A taxa adaptativa mantém a taxa de aprendizagem constante em '*learning-rate-init*' enquanto o aprendizado continuar constante. Essa constância na taxa é interrompida se duas épocas consecutivas não conseguiram aumentar o aprendizado, em pelo menos o fator Tolerância *tol* para a otimização. Se *early-stopping* estiver ativado e duas épocas consecutivas não conseguem aumentar a pontuação de validação (R^2 score e acurácia) em pelo menos *tol*, a taxa de aprendizado atual é dividida por 5. A escolha da estratégia de taxa de aprendizagem pode afetar a velocidade e a convergência do treinamento. Nesta dissertação foi utilizado um aprendizado adaptativo.

5. *solver* - O solucionador especifica o algoritmo de otimização usado para treinar a

rede neural. As opções comuns são *adam* (que é uma escolha popular com boas configurações padrão), *sgd* (*Stochastic Gradient Descent*) e *lbfgs* (que é um método de otimização Quasi-Newton). A escolha do solucionador impacta a convergência e a velocidade de treinamento.

6. *maximum iteration* - Especifica o número máximo de iterações (ou épocas). Se o modelo não convergir dentro desse limite, o treinamento é interrompido. O solucionador itera até a convergência (que é determinada por *tol*) ou esse número de iterações.

Na tabela 6.2 seguem os hiperparâmetros otimizados do MLP considerando a classificação diretamente a partir do vetor *DataFrame-TPSW* (LOFAR), e o mesmo submetido as técnicas PCA e NLPCA para extração de características, assim a cada sinal de entrada foi aplicada a otimização bayesiana e não somente utilizados os dados de um tipo de sinal em outro.

A Tabela 6.2 traz nove colunas, na qual a coluna *Sinal* mostra a origem do vetor de características considerada, seguida dos oito hiperparâmetros otimizados. A maior Acc é de 61%. As configurações com entradas PCA ou NLPCA alcançaram um Acc ligeiramente maior do que a entrada com o vetor processado *DataFrame-TPSW* que obtêm um Acc 1% menor. O multiprocessador na tabela 6.2 refere-se às duas formas de utilização de um classificador binário, conforme subseção 5.2.

Tabela 6.2: Parâmetros otimizados obtidos através de otimização Bayesiana para o algoritmo MLP.

Sinal	maximum iteration	activation	alpha	hidden layer sizes	learning rate init	max iter	solver	Acc
LOFAR	30	<i>relu</i>	0,001	76	0,0300	100	<i>adam</i>	0,4635
LOFAR	40	<i>tanh</i>	0,0036	94	0,004	1000	<i>adam</i>	0,5441
LOFAR	50	<i>logistic</i>	0,001	100	0,012	459	<i>adam</i>	0,5729
LOFAR	60	<i>logistic</i>	0,001	58	0,00117	100	<i>adam</i>	0,5937
PCA	30	<i>logistic</i>	0,0001	100	0,00562	249	<i>adam</i>	0,60936
PCA	40	<i>logistic</i>	0,002711	95	0,00431	967	<i>adam</i>	0,6042
NLPCA	30	<i>tanh</i>	0,00097	66	0,00100	689	<i>adam</i>	0,5829
NLPCA	40	<i>logistic</i>	0,0656	100	0,00369	100	<i>adam</i>	0,6049

Com a tabela 6.2, os hiperparâmetros que obtiveram a maior Acc serão utilizados para o treinamento da máquina.

6.1.3 MLP - Resultados da Otimização Bayesiana

Na subseção anterior, foram obtidos os hiperparâmetros ótimos. Nesta subseção, procura-se observar como esse processo ocorre, verificando o conhecimento descritivo da otimização bayesiana com o percurso utilizado pelo algoritmo de otimização para tal feito.

A Figura 6.2 mostra o comportamento da otimização via Bayes dos hiperparâmetros por iteração e a Acc resultante. No eixo das abscissas, têm-se as iterações e, no das ordenadas, os valores dos hiperparâmetros testados na iteração. Na Figura 6.2, é mostrado o hiperparâmetro α , que ajuda a evitar o *overfitting*.

O número de camadas ocultas foi pré-estabelecido como dois. Variou-se apenas o tamanho da camada e não a quantidade de camadas, observando a Figura 6.2, o *hidden layer sizes* e *learning rate in*, o início da taxa de aprendizagem, variaram conjuntamente, após algumas iterações há uma predileção em manter o primeiro entre 80 e 100 neurônios e o segundo abaixo de 0,05.

Esse algoritmo apenas foi capaz de explicar 61% da variável dependente y , com base nas variáveis independentes \mathbf{x} . O número máximo de interações ficou alto para treinar, levando muito tempo para convergir, o que denota que, para o fim proposto nesta dissertação, esse algoritmo de ML não é adequado pela demora a se adaptar no caso de um sinal novo e pela baixa assertividade.

A Figura 6.3 denota os hiperparâmetros contínuos apenas. É um conjunto de cinco gráficos, onde, no eixo da abscissa, está o espaço no qual cada hiperparâmetro variará e no eixo das ordenadas as épocas são 30 no total. Os hiperparâmetros mais frequentes são aqueles em volta do pico da curva. Os hiperparâmetros mais efetivos são encontrados ao olhar para o maior Acc. Portanto, esses são os hiperparâmetros que mais se ajustam ao modelo e serão empregados para o efetivo treinamento.

A Figura 6.3 tem como objetivo entender como a otimização Bayesiana trabalhou, por exemplo, o algoritmo logo na décima tentativa do termo α se convenceu que esse deveria ser próximo de zero. No parâmetro *hidden layer sizes*, houve oscilação por volta de 80 mas não se repousou sobre um valor. Particularmente, o *hidden layer sizes* sugere que o algoritmo precisa de mais do que 30 épocas e na tabela 6.2, vê-se que, de fato, mais 30 foram utilizadas, com mais um corrida no algoritmo de otimização Bayesiana, ajustou-se o número máximo de épocas para 40 esperando um melhor resultado. No entanto, a

diferença do Accentre a primeira e segunda linha de valores numéricos da tabela:

$$\Delta_{\text{Acc}} = 0,58601 - 0,5833 = 0,00271. \quad (6.5)$$

Essa variação é menor que os 1% para cada 10 épocas que foi o critério de parada estabelecido, portando entende-se o último como o resultado adequado.

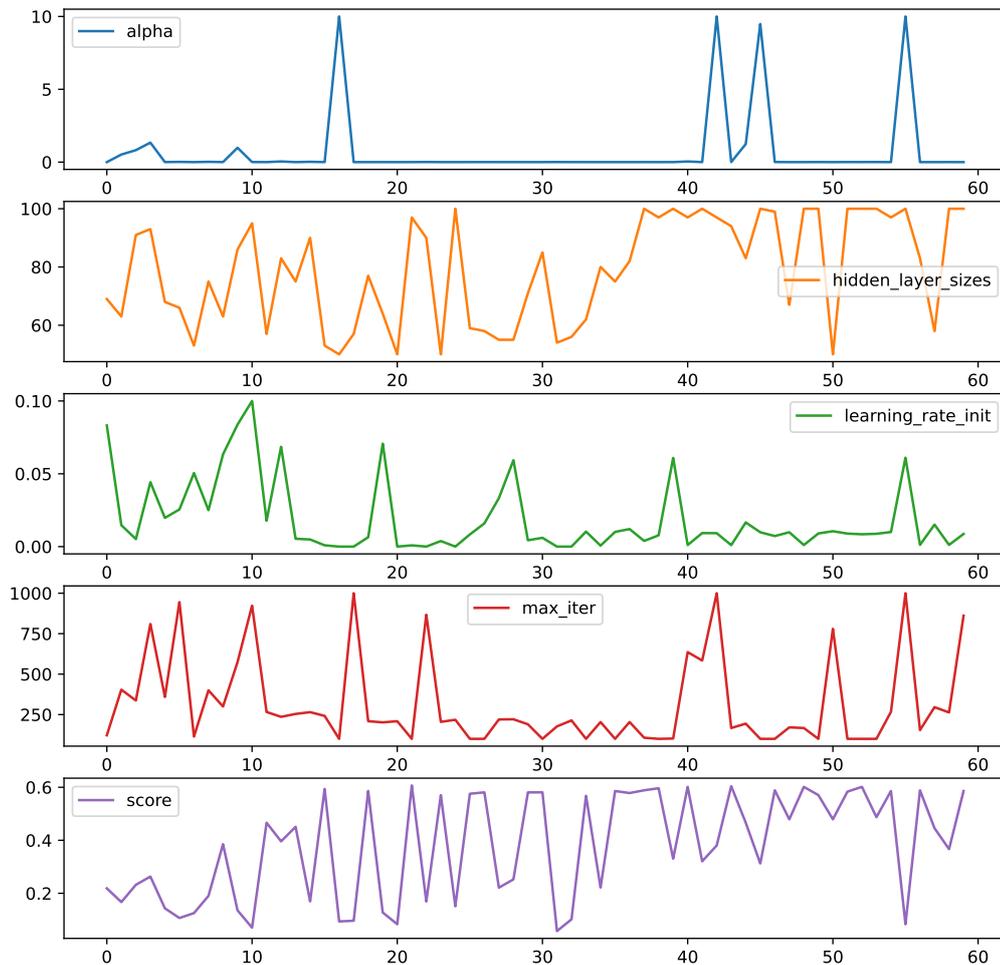


Figura 6.2: MLP - configuração 1×1 - Percurso da estimação de otimização de parâmetros, taxa de aprendizagem (α), tamanho das camadas ocultas, início do (α) e iterações.

A Figura 6.3 é um conjunto de subfiguras, pois é a combinação de quatro hiperparâmetros contínuos e da Acc. Essa figura é útil para notar como cada componente influencia no resultado e nos outros componentes.

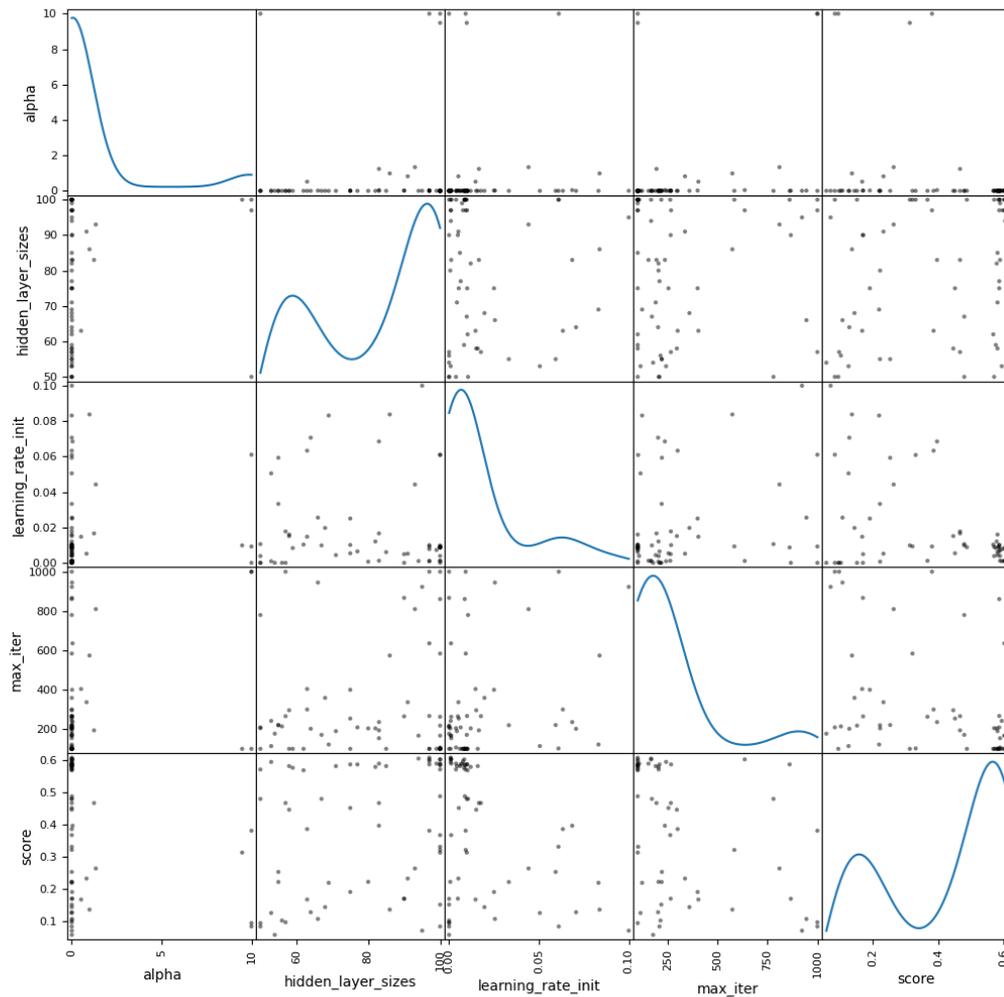


Figura 6.3: MLP - $1 \times$ Todos - Densidade de Distribuição de Probabilidade e Inter-relacionamento de variáveis.

Na Figura 6.3, sobre a diagonal principal, essa função densidade de probabilidade de cada hiperparâmetro e nas interseções entre as variáveis, há um ponto em negrito que denota a melhor correlação entre os dois hiperparâmetros.

6.2 KNN

O algoritmo k-vizinhos mais próximos, também conhecido como KNN ou k-NN, é um classificador de aprendizagem supervisionado. KNN utiliza principalmente dois hiperparâmetros: uma medida de distância e a quantidade de vizinhos mais próximos utilizados

para realizar a classificação a partir das medidas de distância [89]. A Figura 6.4 exemplifica o algoritmo de classificação KNN.

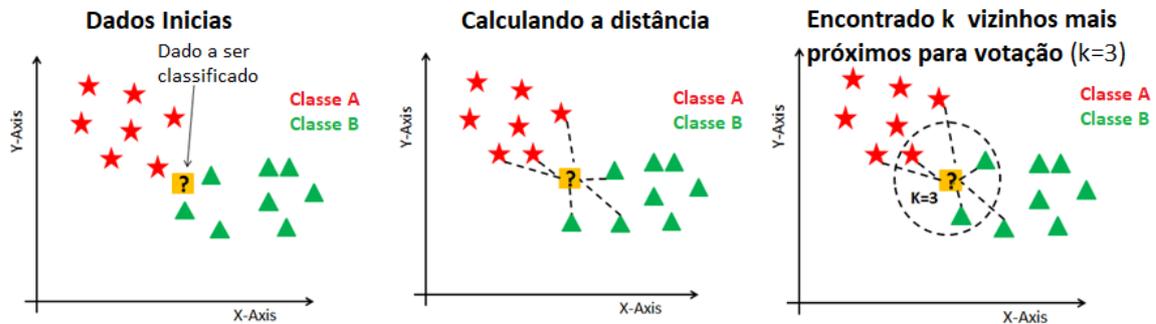


Figura 6.4: Algoritmo de classificação KNN, onde $K=3$ [8].

Há mais de uma métrica possível para calcular a distância, sendo que as mais comuns são: Euclidiana, Manhattan e Minkowski.

A norma Euclidiana é definida por $d(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}$; a Manhattan é $d(x, y) = \sqrt{\sum_{i=1}^n |y_i - x_i|}$; e a Minkowski, $d(x, y) = \sqrt[p]{\sum_{i=1}^n |y_i - x_i|^p}$, $p \in \mathbb{N}$.

6.2.1 KNN - Parâmetros Não-Otimizados

Apenas dois parâmetros são fundamentais para implementar KNN, ou seja, o valor de K e a função de distância. Os parâmetros abaixo foram escolhidos com base nas transformações realizadas no *Dataframe-TPSW*.

Algoritmo - KNN pode usar algoritmos diferentes para calcular os vizinhos mais próximos. As opções comuns incluem: *auto* que escolhe automaticamente o algoritmo mais eficiente com base nos dados de entrada; *ball tree* o qual adequa estrutura em árvore para organizar pontos de dados para consultas mais rápidas; *kd tree*, que é outro algoritmo baseado em árvore; e *bruto*, que calcula distâncias entre todos os pares de pontos.

Tamanho da folha (*leaf size*): Relevante para algoritmos baseados em árvore (*ball tree* ou *kd tree*). Representa o número de pontos em que o algoritmo muda para o modo de força bruta para calcular distâncias.

6.2.2 KNN - Otimização Bayesiana

Os parâmetros a serem otimizados nesta dissertação são:

1. *Número de vizinhos k*- Este hiperparâmetro define o número de vizinhos mais próximos que influenciam a estimacão.

Usar um valor pequeno para k , por exemplo, $k = 1$, pode levar a um limite de decisão mais flexível, mas também pode tornar o modelo mais sensível ao ruído nos dados, resultando em ajuste excessivo, especialmente se o conjunto de dados tiver ruído. Usar um valor grande para k (por exemplo, $k = \sqrt{X}$, onde X é o tamanho do banco de dados) pode suavizar o limite de decisão e reduzir os efeitos do ruído. No entanto, também pode levar à suavização excessiva e à perda de detalhes locais nos dados;

2. *Pesos*- Função de peso usada na previsão podendo ser: *uniforme* onde todos os pontos em cada vizinhança são ponderados igualmente; ou *distância* onde os pesos são ponderados pelo inverso da sua distância. Neste caso, os vizinhos mais próximos de um ponto de busca terão uma maior influência do que vizinhos que estão mais distantes; e *callable*, uma função definida pelo usuário que aceita uma matriz de distâncias e retorna uma matriz da mesma forma contendo os pesos;
3. p - Parâmetro de potência para a métrica Minkowski. Quando p é igual a 1, isso é equivalente a usar Manhattan (ℓ_1) e Euclideana se p é igual a 2. Para p arbitrário, Minkowski é usado.

Abaixo seguem os hiperparâmetros obtidos para o sinal *DataFrame-TPSW* em cada uma das técnicas de compressão de dados, PCA e NLPCA.

Tabela 6.3: Parâmetros obtidos no KNN.

Sinal	Iteraçãõ	n neighbors	weights	p	Acc
LOFAR	30	1	<i>distance</i>	1	0,5831
LOFAR	40	1	<i>distance</i>	1	0,5831
PCA	30	2	<i>distance</i>	1	0,5831
PCA	40	2	<i>distance</i>	1	0,5831
NLPCA	30	1	<i>distance</i>	1	0,5831
NLPCA	40	1	<i>distance</i>	1	0,5831

6.2.3 KNN - Resultado da Otimização Bayesiana

Na Figura 6.5, são expostos os dois maiores problemas do KNN. No caso da **maldição da dimensionalidade**, o algoritmo não funciona bem com entradas de dados de alta

dimensão, isso também é chamado de fenômeno de pico, em que, após o algoritmo atingir o número ideal de características, características adicionais aumentam a quantidade de erros de classificação, especialmente quando o tamanho da amostra é menor.

O segundo problema é que o algoritmo é propenso a *overfitting*. Valores mais baixos de k podem superajustar os dados, enquanto valores mais altos de k tendem a suavizar os valores de previsão, uma vez que está calculando a média dos valores em uma área ou vizinhança maior. No entanto, se o valor de k for muito alto, ele poderá não ajustar os dados [90].

O melhor resultado obtido, conforme Figura 6.5, é de aproximadamente 56%, com apenas um vizinho, varrendo o número de vizinhos de 1 até 10 e tendo a métrica *Manhattan* de medição.

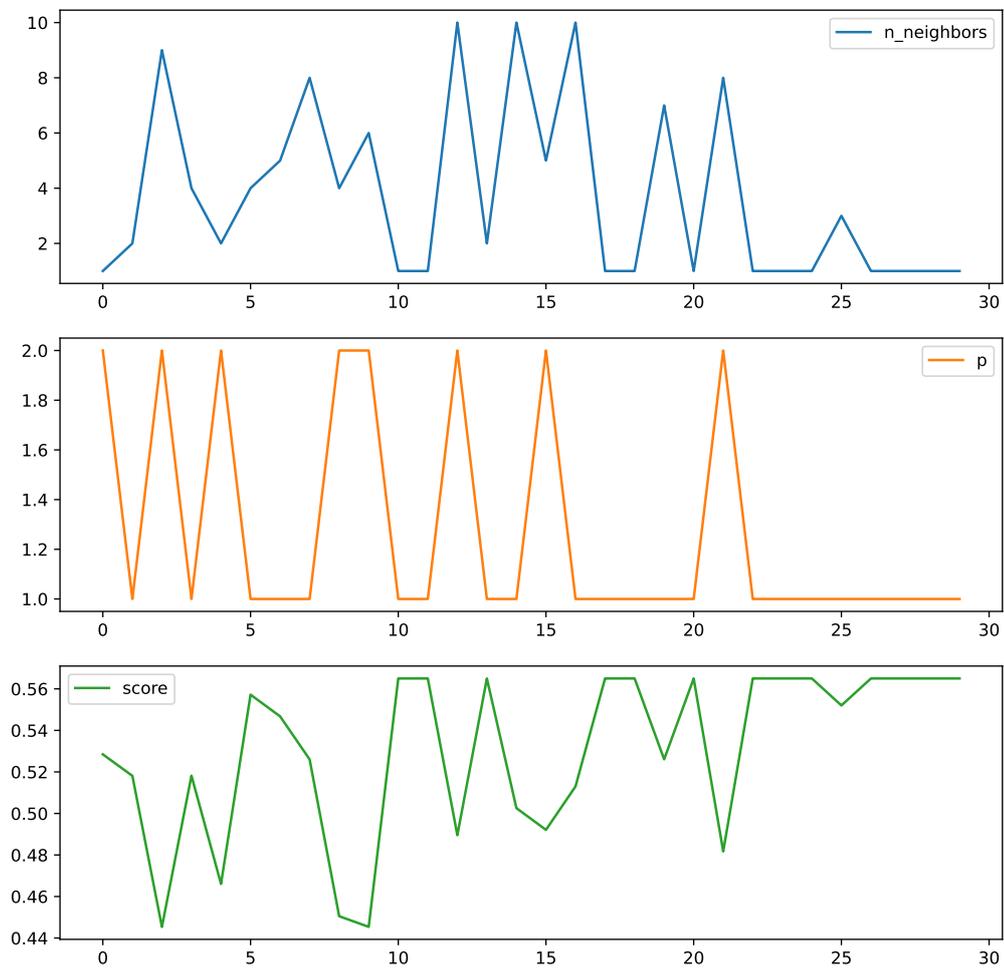


Figura 6.5: KNN - Percurso da estimação de otimização de parâmetros, número de vizinhos e métrica de distância.

A Figura 6.6 é um conjunto de 9 subfiguras pois são todas as combinações dos 3 hiperparâmetros contínuos, em que o ponto preto em negrito simboliza o melhor resultado para cada hiperparâmetro.

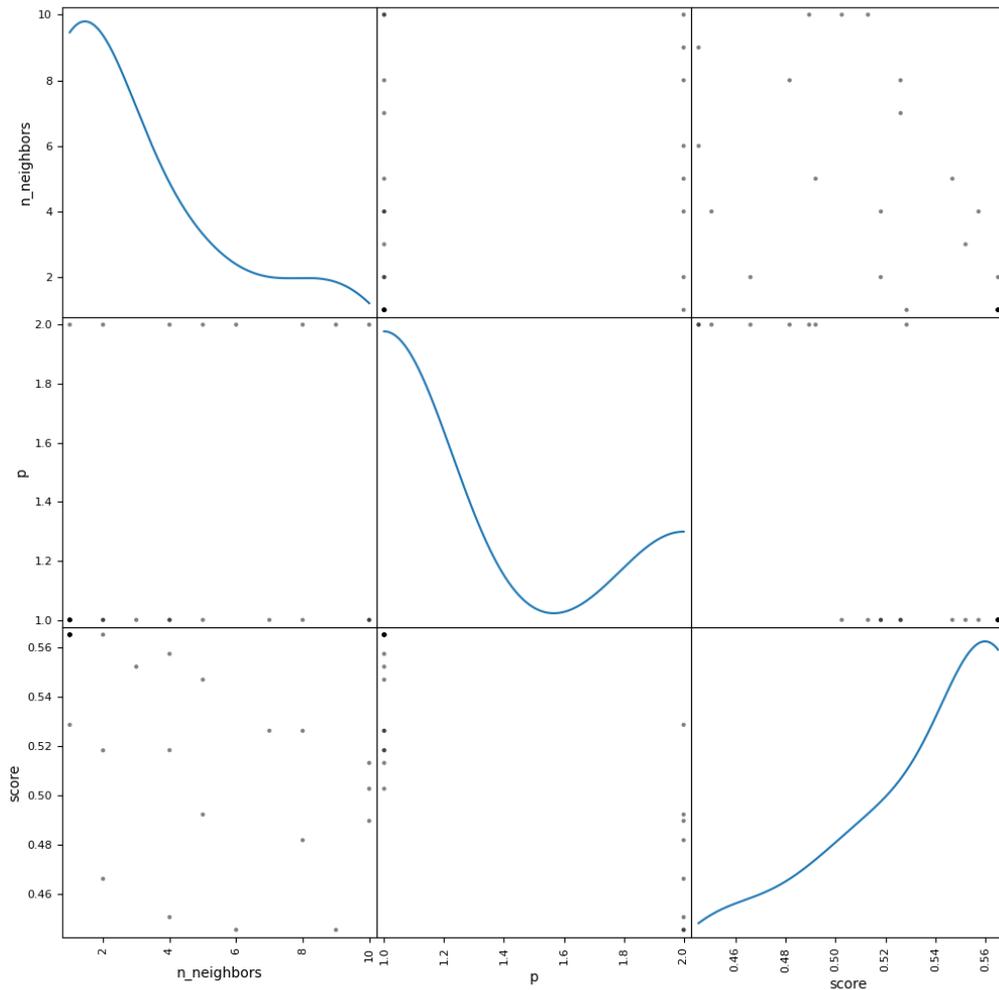


Figura 6.6: KNN - Densidade de Distribuição de Probabilidade e Inter-relacionamento de variáveis.

Na Figura 6.6, sobre a diagonal principal, está a função densidade de probabilidade de cada hiperparâmetro e nas interseções entre as variáveis, há um ponto em negrito que denota a melhor correlação entre os dois hiperparâmetros.

6.3 Ridge

A regressão Ridge [91] é quase idêntica à regressão linear (conforme a equação (6.6))[92], exceto que se introduz um certo viés, inexactidão em reproduzir corretamente a função de transferência entre a entrada e saída. Com efeito obtém-se uma menor variância, um

classificador mais estável em diferentes conjuntos de amostras. A Regressão Ridge pode fornecer melhores previsões de longo prazo em comparação com a regressão linear. A regressão linear minimiza:

$$Arg_{min} = \sum_{i=1}^k (X_i - \hat{X}) \quad (6.6)$$

A regressão ridge:

$$Arg_{min} = \sum_{i=1}^k (X_i - \hat{X}) + \lambda \times \frac{\delta x}{dw_i}, \quad (6.7)$$

onde w são os pesos das conexões dos neurônios, k são os elementos do conjunto amostra e i é cada elemento do conjunto em si.

A figura 6.7, deixa claro como a regressão ridge consegue enviezar a reta, graças ao segundo termo da equação (6.7).

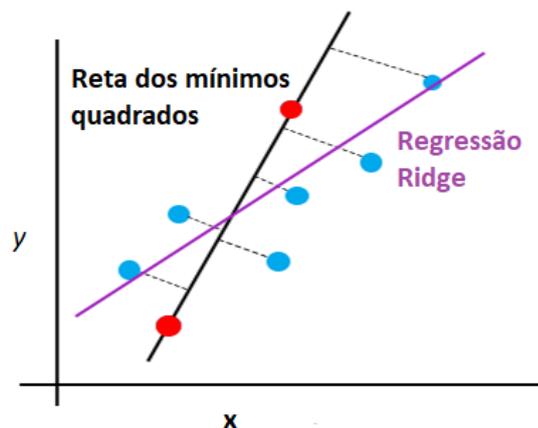


Figura 6.7: Diferenças entre a regressão linear e a ridge.

6.3.1 Ridge - Otimização Bayesiana

Os parâmetros a serem otimizados nesta dissertação são:

1. Parâmetro de regularização (α)- É o principal hiperparâmetro na regressão, controla a força do termo de regularização adicionado à função de perda de mínimos quadrados. Um valor mais alto de α resulta em uma regularização mais forte, levando a valores de coeficientes menores e a um modelo mais simples. Por outro lado, um valor mais baixo de α reduz a quantidade de regularização, permitindo que o modelo se ajuste melhor aos dados de treinamento.

2. *Kernel*- Especifica o algoritmo usado para calcular os coeficientes. Diferentes solucionadores têm diferentes complexidades computacionais e são adequados para diferentes tipos de problemas.
3. Grau do *kernel* polinomial- É ignorado por outros kernels.
4. Parâmetro de regularização (γ)- É utilizado para RBF, laplaciano, polinomial, exponencial, χ^2 e núcleos sigmóides. A interpretação do valor padrão depende do *kernel*.

Seguem os hiperparâmetros obtidos para o sinal *DataFrame-TPSW* em cada uma das técnicas de compressão de dados, PCA e NLPCA.

Tabela 6.4: Parâmetros obtidos no Ridge

Sinal	Iteração	α	<i>Kernel</i>	γ	grau	R^2 Score
LOFAR	30	10	linear	10^{-6}	-	0,97948
PCA	30	0,99999	linear	10^{-6}	-	0,97948
NLPCA	30	0,5352417	linear	0,748465	-	0,96220

6.3.2 Ridge - Resultado da Otimização Bayesiana

A Figura 6.8 relembra que um classificador pode apresentar performance pior que a média com um resultado negativo. O Ridge é um regressor ao qual cada classe se atribuiu um escalar.

Vê-se o parâmetro de aprendizado α é mantido baixo, e o grau do polinômio também. O γ trabalhou por um tempo acompanhando α , mas, após isso, trabalham em oposição, assim quanto menor o aprendizado maior a regularização. Encontrou-se um resultado de 97% de acurácia com o NLPCA e o PCA. O *DataFrame-TPSW* não ficou muito atrás com 96%, mas como o Ridge é um regressor trabalhou melhor com uma menor dimensão dos dados.

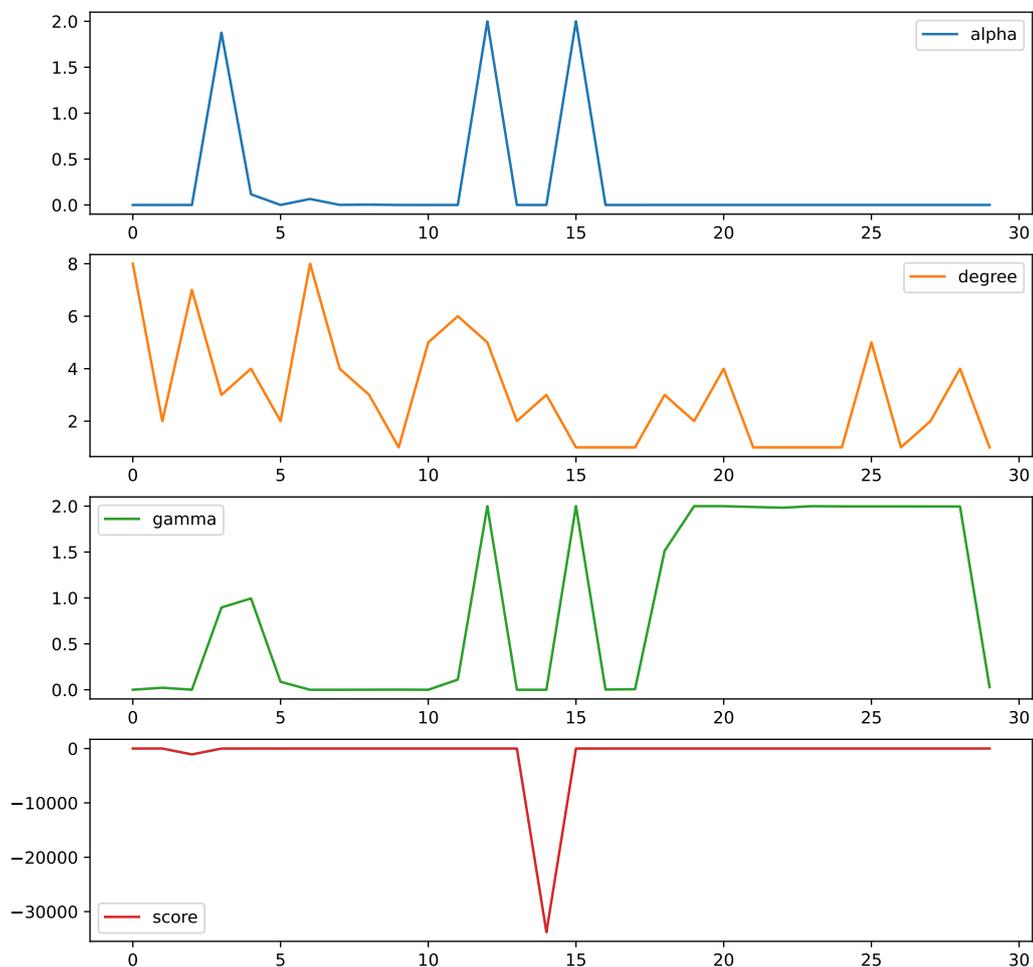


Figura 6.8: Ridge - Percurso da estimação de otimização de parâmetros, taxa de aprendizagem α , grau do polinômio e tipo de regularização.

A Figura 6.8 é um conjunto de 16 subfiguras pois é a combinação de três hiperparâmetros contínuos e do R^2 score. Essa figura é útil para notar como cada componente influencia no resultado e nos outros componentes.

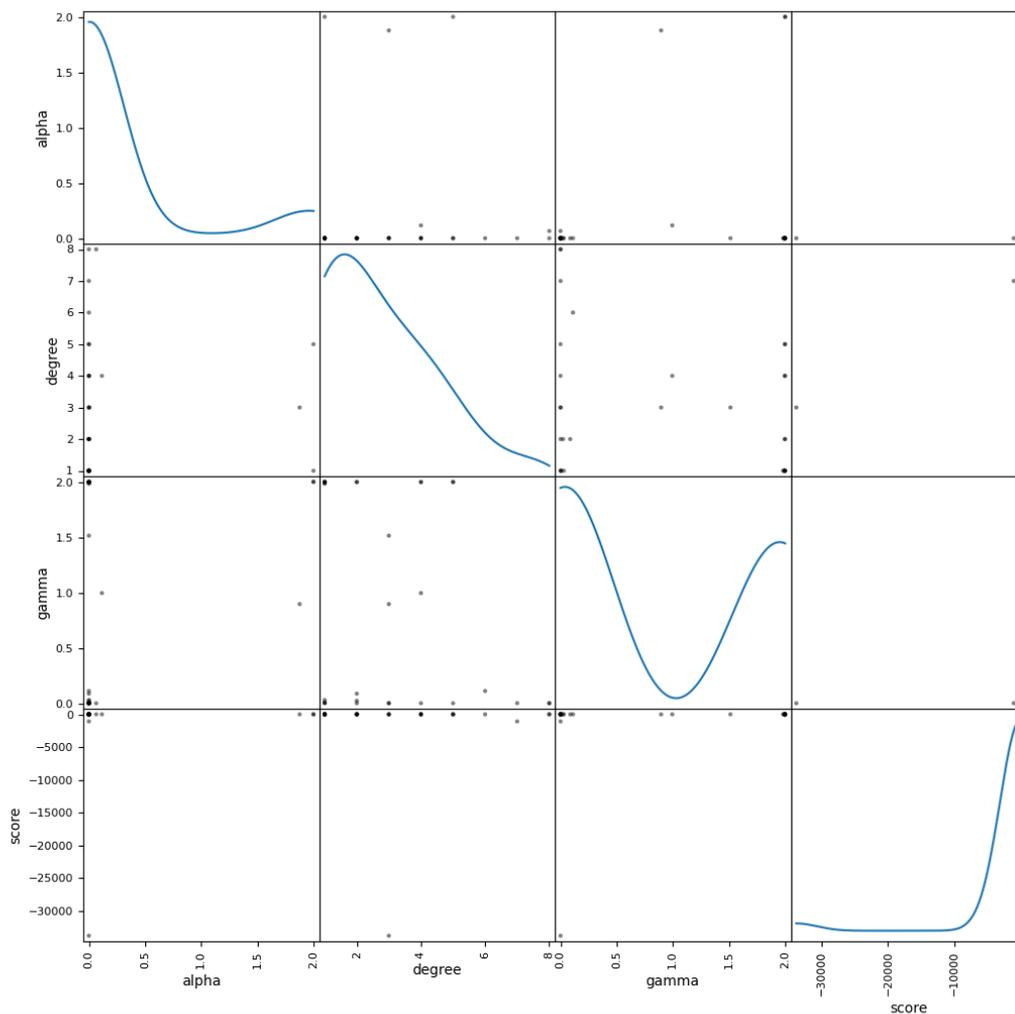


Figura 6.9: Ridge - Densidade de Distribuição de Probabilidade e Inter-relacionamento de variáveis.

Na Figura 6.9, sobre a diagonal principal essa função densidade de probabilidade de cada hiperparâmetro e nas interseções entre as variáveis, há um ponto em negrito que denota a melhor correlação entre os dois hiperparâmetro.

6.4 SVM

A máquina de vetores suporte (SVM) separa através de um hiperplano os dados, seleciona-se um hiperplano com a margem máxima possível entre os vetores de suporte em um determinado conjunto de dados. Nos casos em que os dados não são linearmente separáveis,

o SVM ainda pode ser usado transformando o espaço de características original em um espaço de dimensão superior usando uma técnica chamada truque do kernel. A função kernel calcula os produtos internos entre pontos de dados no espaço de dimensão superior sem transformá-los explicitamente. Isso permite que o SVM encontre limites de decisão não lineares no espaço de características original.

O Kernel pode ser : Linear: $K(x, x_i) = \sum(x \times x_i)$; Polinomial: $K(x, x_i) = 1 + \sum(x \times x_i)^d$; e Base Radial: $K(x, x_i) = e^{-\gamma \times \sum x - x_i^2}$.

O hiperplano pode ser escrito como o conjunto de pontos \mathbf{x} satisfazendo

$$\mathbf{w} \cdot \mathbf{x} + b = 0, \quad (6.8)$$

onde \mathbf{w} é o vetor normal ao hiperplano. O parâmetro $\frac{b}{\|\mathbf{w}\|}$ determina o deslocamento do hiperplano da origem ao longo do vetor normal a \mathbf{w} .

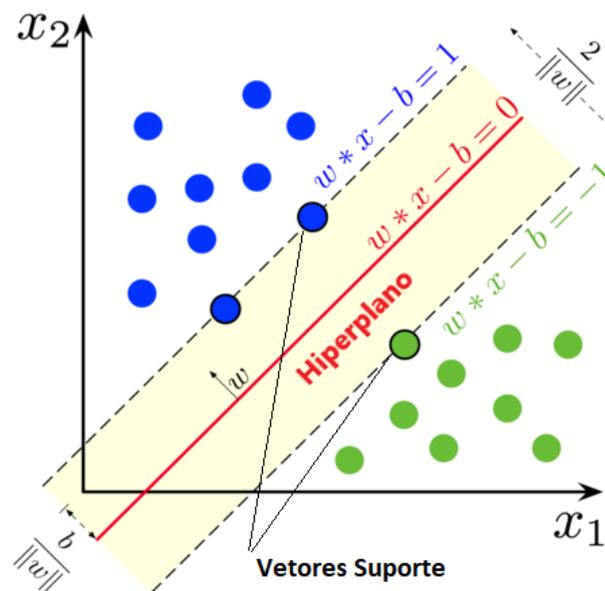


Figura 6.10: SVM - O hiperplano maximiza a distância entre os vetores suporte, através deles é calculada a classe de pertencimento de uma entrada não-classificada.

6.4.1 SVM- Parâmetros Não-Otimizados

Como resultado da otimização, foram obtidos os hiperparâmetros a ser otimizados. Os hiperparâmetros abaixo, nesta dissertação, não necessitam de otimização. Por exemplo, para o *multiclasse*, testou-se todas as possibilidades.

1. perda: O parâmetro perda determina a função de perda a ser usada. O padrão é *squared hinge*, que corresponde à perda ℓ_2 SVM. *Dobradiça* é outra opção, que

corresponde à perda ℓ_1 -SVM.

2. dual: O parâmetro dual determina se o problema de otimização primal ou dual deve ser resolvido. O padrão é *dual = True*, que é adequado para um número menor de amostras com mais características. Para um número maior de amostras com menos características, você pode querer definir *dual=False*.
3. multiclasse: O parâmetro multi classe determina como o algoritmo deve lidar com problemas de classificação multiclasse. O padrão é *ovr* (um contra o resto/todos), onde um problema binário é adequado para cada classe. Outras opções incluem *crammer singer* (otimiza um objetivo conjunto em todas as classes) e *ovr*.

6.4.2 SVM- Otimização Bayesiana

Os parâmetros a serem otimizados nesta dissertação são:

1. Parâmetro de Regularização, *C*: Controla o compromisso entre atingir um erro de treinamento baixo e um erro de teste baixo. É essencialmente o inverso da força de regularização, portanto, um *C* menor significa uma regularização mais forte.
2. Função de perda: O *LinearSVC* usa a perda de dobradiça por padrão, que é uma função de perda padrão para SVMs.

A perda de dobradiça é definida como

$$L(y, f(x)) = \max(0, 1 - y \cdot f(x)), \quad (6.9)$$

onde, *L* é a perda de dobradiça, *y* é o símbolo da classe $y = [-1, 1]$, *f(x)* é a saída bruta da função de decisão antes de aplicar a função de sinal.

Abaixo seguem os hiperparâmetros obtidos para o sinal *DataFrame-TPSW* em cada uma das técnicas de compressão de dados, PCA e NLPCA.

Tabela 6.5: Parâmetros obtidos no SVM.

Sinal	Multipro- cessador	Iteração	C	$Loss$	penalidade	Acc
LOFAR	1 × 1	30	21792	<i>squared hinge</i>	ℓ_2	0,5467
PCA	1 × 1	30	773393	<i>hinge</i>	ℓ_2	0,5260
NLPCA	1 × 1	30	40472	<i>squared hinge</i>	ℓ_2	0,5338
LOFAR	1 × T	30	47730	<i>squared hinge</i>	ℓ_2	0,5184
PCA	1 × T	30	773393	<i>squared hinge</i>	ℓ_2	0,56801
NLPCA	1 × T	30	269308	<i>squared hinge</i>	ℓ_2	0,50397
NLPCA	1 × T	40	36921	<i>squared hinge</i>	ℓ_2	0,51397

6.4.3 SVM- Resultado da Otimização Bayesiana

O hiperparâmetro C , apresentado na figura 6.11 é o inverso do α nas máquinas anteriores, então quanto maior, mais lentamente ele aprende, porém há um ponto de inflexão em 400.000.

A acurácia conforme mostrada na Figura 6.12 é de um percentual muito baixo, associada às várias máquinas que devem ser construídas por esse ser um algoritmo de classificação binária, tornam essa máquina inadequada pra essa aplicação.

A Figura 6.11 é um conjunto de 4 subfiguras, pois são todas as combinações dos 2 hiperparâmetros contínuos, essa figura é útil para notar como cada componente influencia no resultado e nos outros componentes.

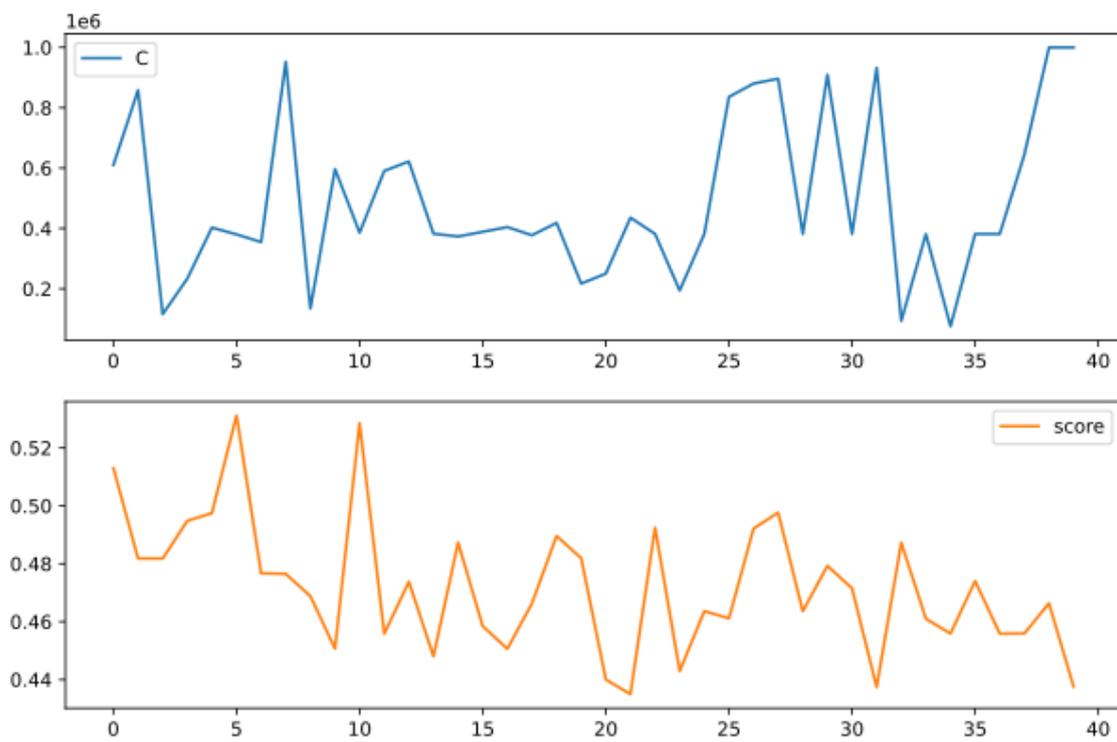
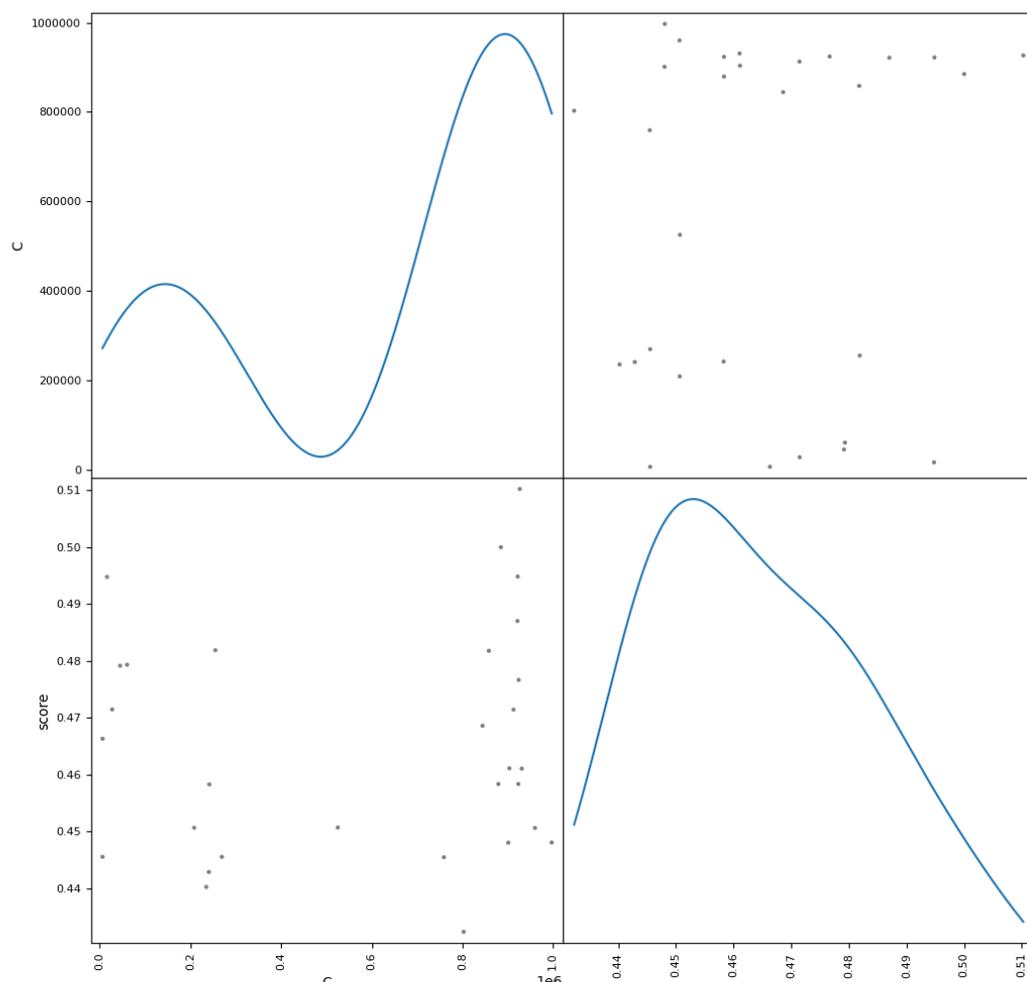


Figura 6.11: SVM- 1 x 1 - Percurso da estimação de otimização de parâmetros apenas o inverso taxa de aprendizagem (C).



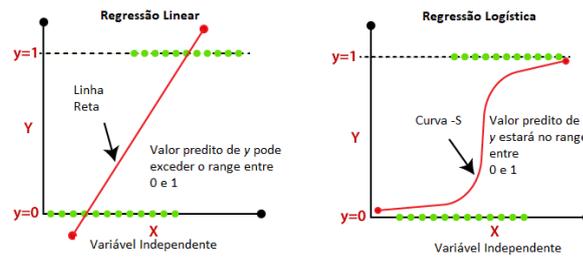


Figura 6.13: Diferenças entre regressão Logística e Linear, a regressão linear varia continuamente com a inclinação da reta e a logística possui um limiar e divide a classificação em antes e depois desse limiar.

Na Figura 6.12, sobre a diagonal principal esta função densidade de probabilidade de cada hiperparâmetro e nas interseções entre as variáveis, há um ponto em negrito que denota a melhor correlação entre os dois hiperparâmetro.

6.5 Regressão Logística

Na regressão logística, uma transformação *logit* é aplicada às probabilidades, que são definidas como a taxa de sucesso dividida pela taxa estimada de fracasso. Matematicamente, o *logit* é o inverso da função logística padrão, notadamente conhecida como *sigmoid*, $\sigma(x) = \frac{1}{1+e^{-x}}$

$$\text{logit}(p) = \sigma^{-1} = \frac{p}{1 - \ln p}. \quad (6.10)$$

A regressão logística visa resolver problemas de classificação, prevendo a probabilidade de resultados (*range* entre 0 e 1), conforme apresentado na Figura 6.13, ao utilizá-lo num problema multi-classes é mister dividi-lo em vários problemas binários, conforme apresentado no capítulo 5.

6.5.1 Logit - Parâmetros Não-Otimizados

1. fit intercept: Este é um parâmetro booleano que determina se deve calcular a interceptação para o modelo. Se definido como 'False', nenhuma interceptação será usada nos cálculos.
2. peso da classe: este parâmetro é usado para lidar com classes desequilibradas. Ele pode ser definido como *balanceado* para ajustar automaticamente os pesos inversamente proporcionais às frequências de classe nos dados de entrada.

3. solucionador: O solucionador é um algoritmo usado para otimização. Diferentes solucionadores são adequados para diferentes tipos de problemas. A escolha do solucionador também influencia outros hiperparâmetros como o tipo de regularização. As escolhas comuns incluem *liblinear*, *newton-cg*, *lbfg*, *sag* e *saga*.
4. max iter: Este parâmetro define o número máximo de iterações para o solucionador convergir. Se o solucionador não convergir dentro deste limite, poderá não encontrar a solução ótima.
5. multiclasse: Este parâmetro determina como o algoritmo lida com problemas de classificação multiclasse. As opções incluem *ovr* (um contra todos) e *multinomial* (*softmax*).
6. estado aleatório: presta-se a conseguir resultados reproduzíveis, pode-se definir uma semente usando o parâmetro *random state*.

6.5.2 Logit - Otimização Bayesiana

Os parâmetros a serem otimizados nesta dissertação são:

1. Penalidade: Especifica o tipo de regularização a ser aplicada. A regularização ajuda a prevenir o *overfitting*. As opções mais comuns são:
 - (a) ℓ_1 : Regularização Lasso, que soma os valores absolutos dos coeficientes à função de custo.
 - (b) ℓ_2 : Regularização de Ridge, que soma os valores quadrados dos coeficientes à função de custo.
 - (c) *none*: Sem regularização.
2. C (Inverso da Força de Regularização): O hiperparâmetro C controla a força da regularização. A relação entre C e a força de regularização é $C = 1/\lambda$, λ é o parâmetro de regularização.

Seguem os hiperparâmetros obtidos para o sinal *DataFrame-TPSW* em cada uma das técnicas de compressão de dados, PCA e NLPCA.

Tabela 6.6: Parâmetros obtidos Regressão Logística.

Sinal	Multipro- cessador	Iteração	C	penalidade	Acc
LOFAR	1x1	30	1,21020	ℓ_1	0,9427
PCA	1x1	30	0,745383	ℓ_1	0,94535
NLPCA	1x1	30	1,43926	ℓ_1	0,9479
LOFAR	1xT	30	1,8454	ℓ_1	0,945352
PCA	1xT	30	1,11829	ℓ_1	0,94535
NLPCA	1xT	30	1,3543	ℓ_1	0,9427

6.5.3 Logit - Resultado da Otimização Bayesiana

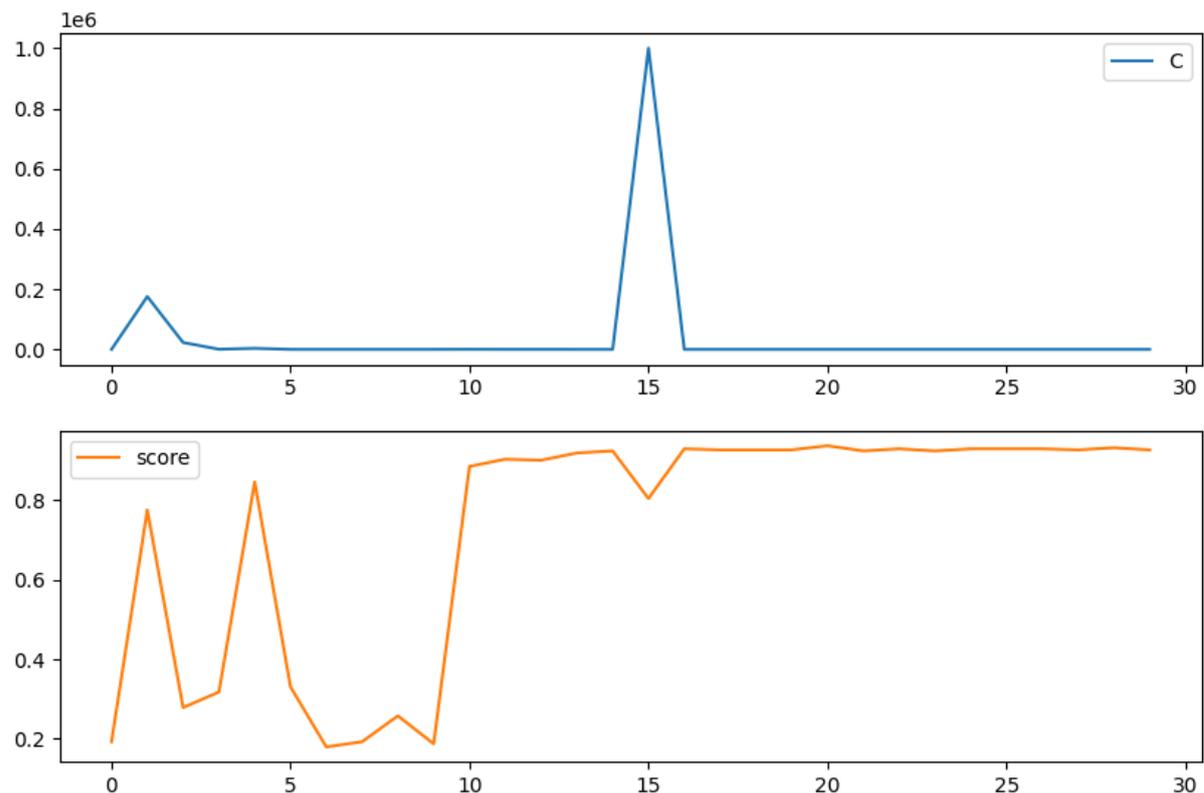


Figura 6.14: Logit- 1 x 1 - Percurso da estimação de otimização de parâmetros, apenas o inverso da taxa de aprendizagem (C) a ser otimizado.

A Figura 6.14 é um conjunto de 4 subfiguras pois é a combinação de um hiperparâmetro contínuo e da Acc. Essa figura é útil para notar como cada componente influencia no resultado e nos outros componentes.

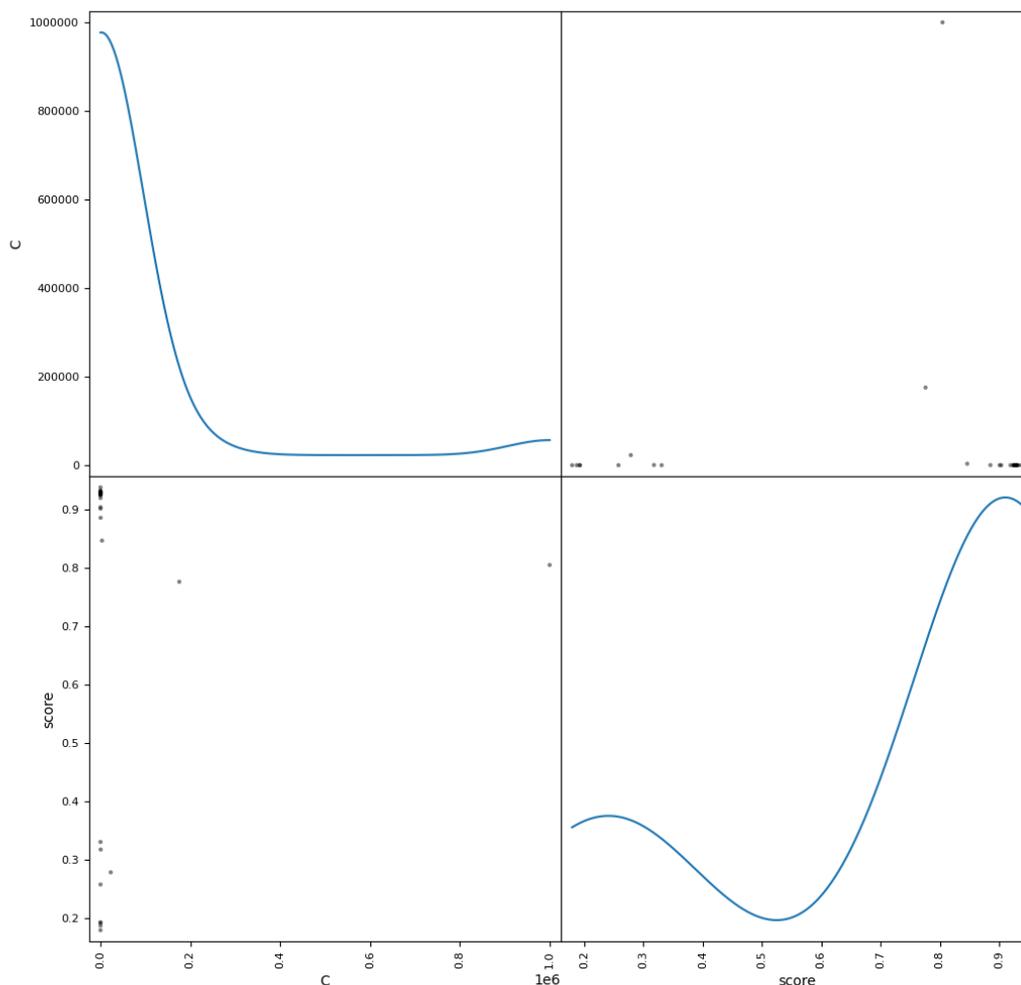


Figura 6.15: Logit- 1 x 1 - Densidade de Distribuição de Probabilidade e Inter-relacionamento de variáveis.

Na Figura 6.15, sobre a diagonal principal esta função densidade de probabilidade de cada hiperparâmetro e nas interseções entre as variáveis, há um ponto em negrito que denota a melhor correlação entre os dois hiperparâmetros.

6.6 LDA

A Análise Discriminante Linear (LDA) projeta os dados em um espaço de menor dimensão que maximiza a separação entre as classes. Encontrando um conjunto de discriminantes lineares que maximizam a razão entre a variância entre classes e a variância dentro da

classe. O LDA fornece as direções no espaço de características que melhor separam as diferentes classes de dados.

O LDA assume que os dados possuem distribuição gaussiana e que as matrizes de covariância das diferentes classes são iguais. Também assume que os dados são linearmente separáveis, o que significa que um limite de decisão linear separaria com precisão as diferentes classes.

Diferenças entre LDA e PCA

O PCA busca reduzir a representação do conjunto de dados fornecendo as direções de maior variação dos dados. O LDA parte do pressuposto que existem classes e tenta representar com o menor número de direções as classes existentes, assim o menor número de componentes dele é o exato número de classes. A figura 6.16 explica visualmente esse conceito.

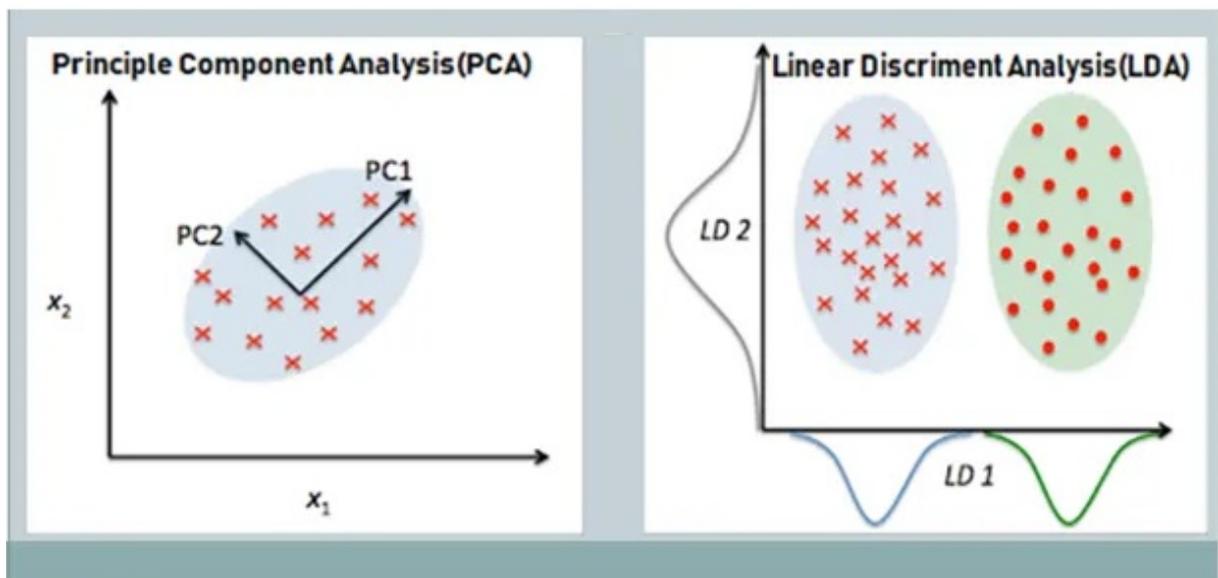


Figura 6.16: LDA × PCA - fonte: <https://miro.medium.com>

6.6.1 LDA - Parâmetros Não-Otimizados

1. encolhimento: Se o solucionador estiver definido como *lsqr* ou *eigen*, este parâmetro comprime os dados para regularizar a matriz de covariância. É um valor entre 0 e 1, onde 0 corresponde a nenhum encolhimento e 1 corresponde a encolhimento completo.
2. Premissas: Este parâmetro permite especificar as probabilidades anteriores das classes. Se não for especificado, serão usadas as proporções de classe nos dados de

treinamento. É necessário um objeto semelhante a um vetor que representa as probabilidades anteriores de cada classe.

3. covariância de armazenamento: Se *True*, a matriz de covariância de cada classe é armazenada. Isso pode ser útil no caso de se inspecionar as matrizes de covariância após ajustar o modelo. Por padrão, está definido como *Falso*.

6.6.2 LDA - Otimização Bayesiana

Os parâmetros a serem otimizados nesta dissertação são:

1. Solucionador: O LDA pode ser resolvido usando diferentes métodos, e a escolha do solucionador pode impactar o desempenho. As opções comuns incluem:
 - *svd*: Decomposição de Valores Singulares. Este é o solucionador padrão e é eficiente para conjuntos de dados de pequeno e médio porte.
 - *lsqr*: solução de mínimos quadrados. Adequado quando os dados estão bem condicionados.
 - *eigen*: decomposição de autovalores. Útil quando o número de características é muito menor que o número de amostras.
2. componentes: Especifica o número de componentes (dimensões) a manter no espaço transformado. É relevante ao usar LDA para redução de dimensionalidade. Se não for especificado, será definido como o mínimo entre o número de classes menos um e o número de características.

Seguem os hiperparâmetros obtidos para o sinal *DataFrame-TPSW* em cada uma das técnicas de compressão de dados, PCA e NLPCA.

Tabela 6.7: Parâmetros obtidos no LDA.

Sinal	Iteração	Componentes	solucionador	Acc
LOFAR	30	7	<i>svd</i>	0,92450
PCA	30	2	<i>svd</i>	0,92450
NLPCA	30	10	<i>svd</i>	0,92450

6.6.3 LDA - Resultado da Otimização Bayesiana

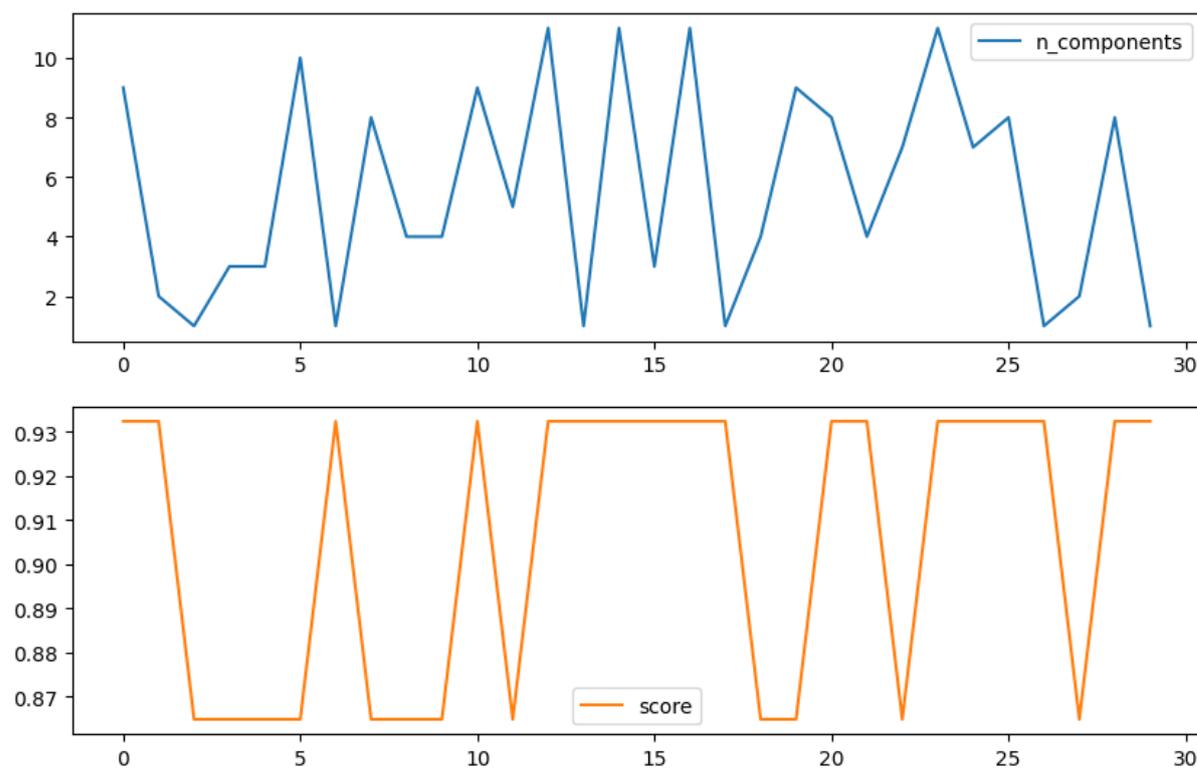


Figura 6.17: LDA - Percurso da estimação de otimização de parâmetro, número de discriminantes lineares.

A Figura 6.17 é um conjunto de 4 subfiguras pois é a combinação de hiperparâmetros contínuos e da Acc. Essa figura é útil para notar como cada componente influencia no resultado e nos outros componentes.

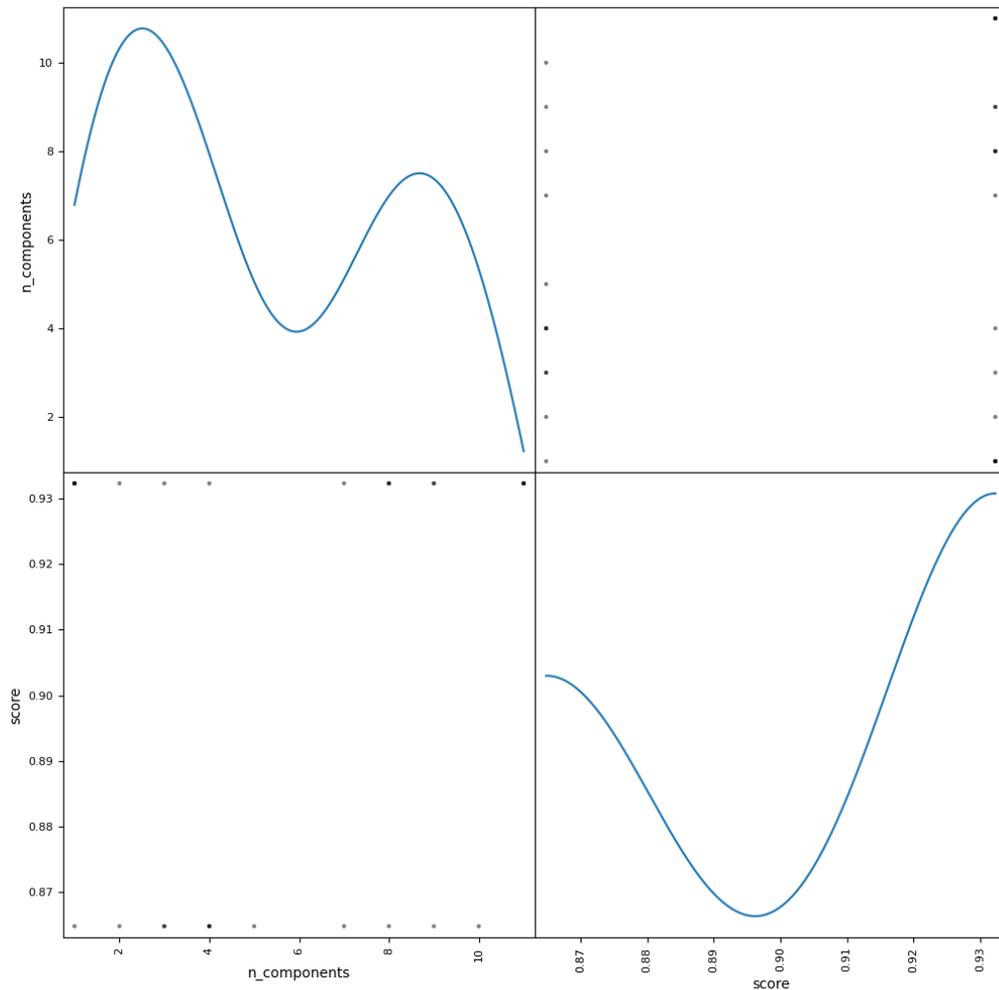


Figura 6.18: LDA - Densidade de Distribuição de Probabilidade e Inter-relacionamento de variáveis.

Na Figura 6.18, sobre a diagonal principal esta função densidade de probabilidade de cada hiperparâmetro e nas interseções entre as variáveis, há um ponto em negrito que denota a melhor correlação entre os dois hiperparâmetro.

6.7 XgBoost

XGBoost é uma implementação de código aberto do algoritmo *gradient boosted trees*. O gradiente *boosting* é um algoritmo de aprendizado supervisionado que busca classificar com precisão uma variável alvo combinando as estimativas de um conjunto de modelos

mais simples e mais fracos.

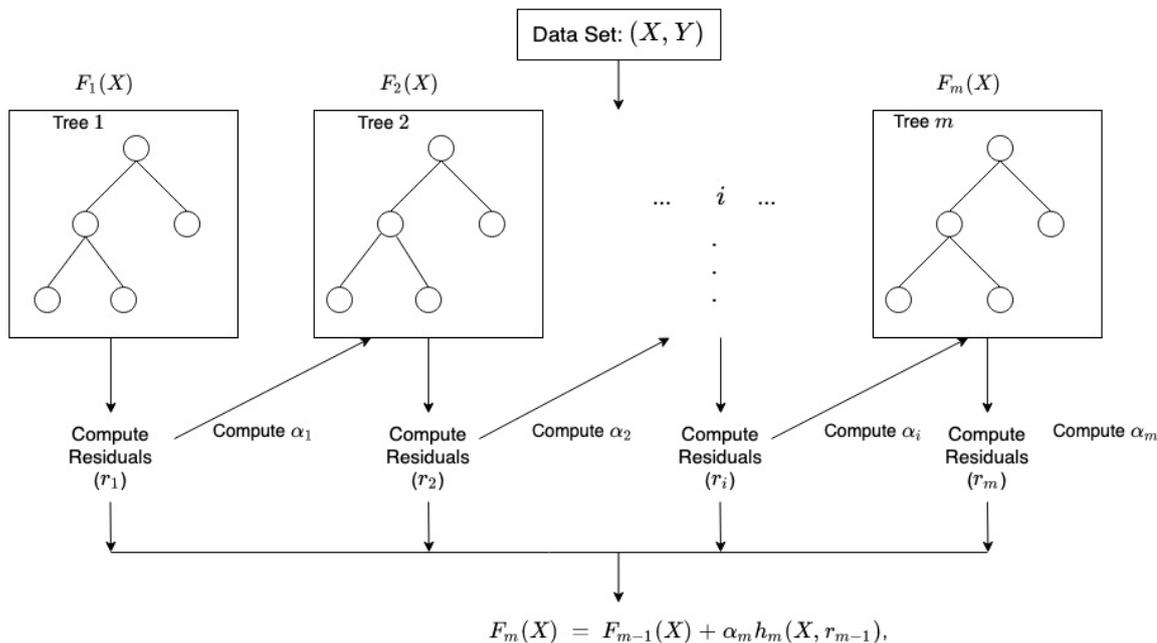


Figura 6.19: XgBoost

6.7.1 XgBoost- Parâmetros Não-Otimizados

1. Redimensionamento de pesos: É utilizado para equilibrar os pesos positivos e negativos, principalmente em conjuntos de dados desequilibrados.
2. Objetivo: Define a função de perda a ser minimizada durante o treinamento. Para classificação binária, *binary:logistic* é comumente usado. Para problemas multi-classe, *multi:softmax* ou *multi:softprob* podem ser usados.
3. Parada Antecipada: Este não é um hiperparâmetro específico, mas uma técnica para evitar *overfitting*. Ele interrompe o treinamento quando o desempenho em um conjunto de validação para de melhorar.
4. Tipo de reforço: XGBoost suporta dois tipos de boosters: modelos baseados em árvore (*gbtree*) e modelos lineares (*gblinear*). A escolha depende do problema específico e do conjunto de dados.

6.7.2 XgBoost- Otimização Bayesiana

Os parâmetros a serem otimizados nesta dissertação são:

1. Taxa de aprendizagem (η): Controla a contribuição de cada árvore para a previsão final. Uma taxa de aprendizagem mais baixa torna o modelo mais robusto, mas requer mais rodadas de reforço.
2. Número de árvores: - Define o número de rodadas de boosting ou árvores a serem construídas.
3. Profundidade máxima: Limita a profundidade máxima de cada árvore. Árvores mais profundas podem capturar padrões mais complexos, mas podem levar a ajustes excessivos.
4. Subamostra: Especifica a fração de dados de treinamento a serem amostrados aleatoriamente para o crescimento de árvores durante cada rodada de reforço. Ajuda a prevenir o *overfitting*.
5. *Colsample Bytree e Colsample Bylevel*: Controlam a fração de características amostrados aleatoriamente para cada árvore e cada nível da árvore, respectivamente. Eles adicionam aleatoriedade ao processo de treinamento, reduzindo o *overfitting*.
6. min child weight γ : É a redução mínima de perdas necessária para fazer uma partição adicional em um nó folha da árvore. Ajuda a controlar o *overfitting*.
7. λ (reg lambda) e Alfa (reg alpha): Estes são parâmetros de regularização. Lambda controla a regularização ℓ_2 em pesos, enquanto alfa controla a regularização ℓ_1 . Eles ajudam a prevenir o *overfitting* ao penalizar pesos grandes.

Abaixo seguem os hiperparâmetros obtidos para o sinal LOFAR em cada uma das técnicas de compressão de dados, PCA e NLPCA.

Tabela 6.8: Parâmetros obtidos no XGBoost.

Sinal	Iteração	α	colsample bytree	γ	λ	Learning rate	max depth	min childweight	n estimators	subsample	Acc
LOFAR	30	0.000287	0.37423	0.142805	0.00010	0.031765	6	1	1000	1,0	0.99740
PCA	30	0.01512	0.47659	0.004465	$9.4274 \cdot 10^{-6}$	0,01	9	1	530	0.3901	0.9974
NLPCA	30	0.05489	0.898444	1,0	1e-06	0,01	1	1	1000	0.93008	0.99213

6.7.2.1 XgBoost- Resultado da Otimização Bayesiana

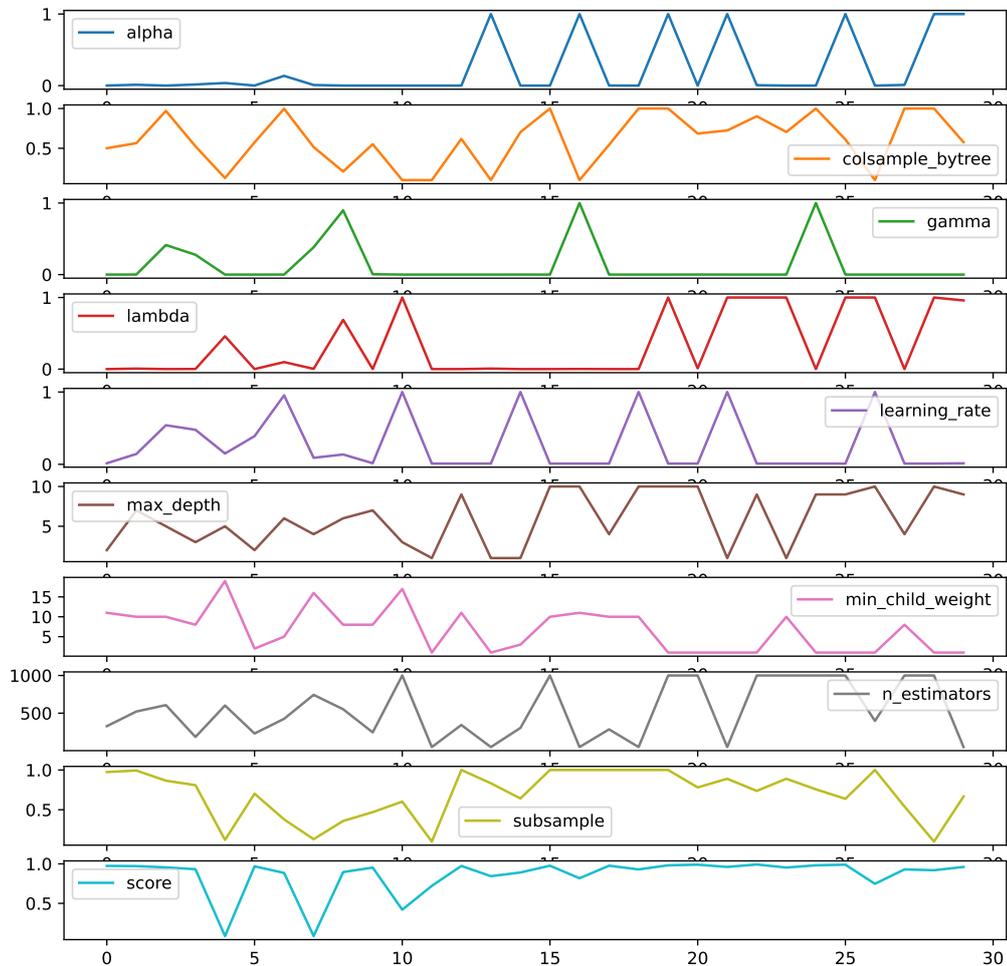


Figura 6.20: XGBoost - Percurso da estimação de otimização de parâmetros, taxa de aprendizagem (α), aleatoriedade do treinamento, segmentação de árvore, regularização, ponderação do nó, tamanho máximo do nó, número mínimo de ocorrências para iniciar um nó, número de árvores e tamanho da partição usada no treinamento.

A Figura 6.20 é um conjunto de 100 subfiguras pois é a combinação de nove hiperpâmetros contínuos e da Acc. Essa figura é útil para notar como cada componente influencia no resultado e nos outros componentes.

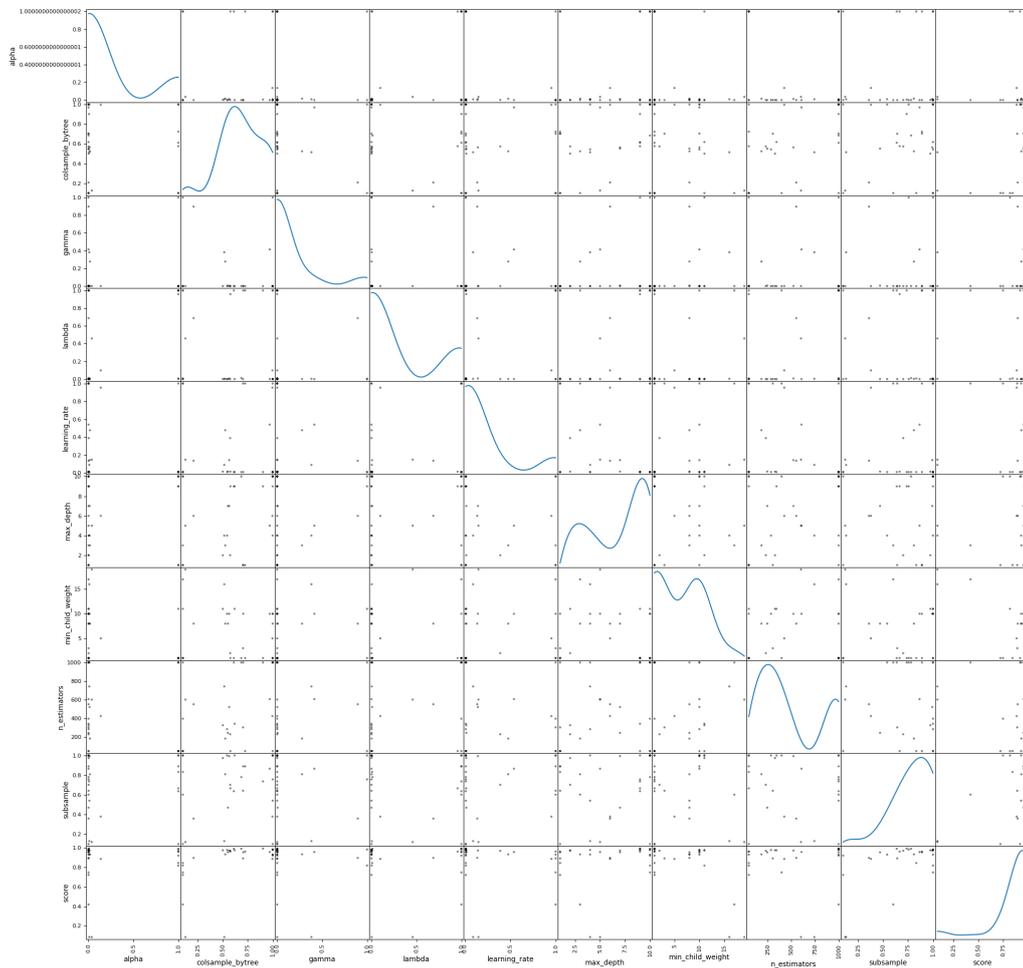


Figura 6.21: XGBoost- Densidade de Distribuição de Probabilidade e Inter-relacionamento de variáveis.

Na Figura 6.21, sobre a diagonal principal esta função densidade de probabilidade de cada hiperparâmetro e nas interseções entre as variáveis, há um ponto em negrito que denota a melhor correlação entre os dois hiperparâmetro.

Capítulo 7

Resultados

Neste capítulo, apresenta-se o culminar dos esforços de pesquisa no domínio da classificação no aprendizado de máquina, com ênfase particular na abordagem de desafios do mundo real e no aprimoramento da interpretabilidade do modelo.

Através de experimentação e avaliação, iluminamos os pontos fortes e as limitações de cada algoritmo sob condições variadas, esclarecendo sua eficácia no tratamento de conjuntos de dados e espaços de características de alta dimensão, no caso LOFAR e de dimensões reduzidas no caso da PCA e NLPCA.

A confiabilidade da metodologia proposta é verificada através de extensa experimentação utilizando, validação cruzada de 10-fold, rodando o algoritmo 100 vezes, no total construindo mil máquinas e verificando o *score* médio que a depender do algoritmo pode ser o R^2 score, para o regressor Ridge ou acurácia média, para todas as outras seis máquinas. Finalmente, concluímos este capítulo sintetizando os resultados e destacando os principais resultados e as contribuições alcançadas para o campo da classificação no aprendizado de máquina.

7.1 MLP

7.1.1 Classificador MLP e Características Lofar

Para o Classificador MLP utilizando como vetor de características o sinal Lofar, sem técnicas de redução, foram obtidos os seguintes resultados em 37 épocas computadas no eixo das abscissas e a acurácia conseguida no eixo das ordenadas. Na subseção 6.1.2, foram encontrados os hiperparâmetros que melhor ajustam a máquina, função de ativação tipo *logistic*, α de 0,001, número de neurônios na camada oculta de 58 unidades, taxa de

aprendizagem inicial de 0,00117, número máximo de iterações de 100, solucionador *adam*; obtendo uma Acc de 0,5937. Com esses hiperparâmetros treina-se a máquina e a submete a um conjunto de teste obtendo os resultados a seguir.

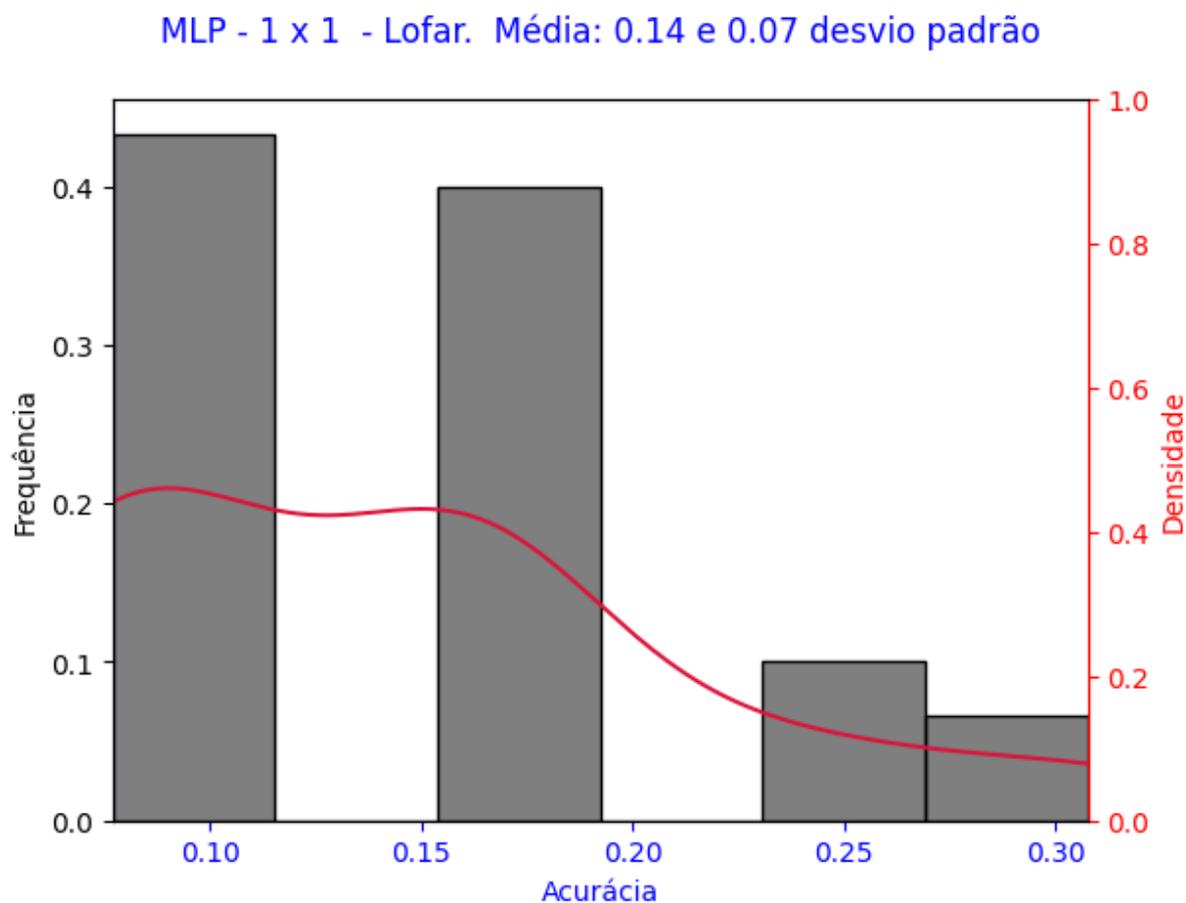


Figura 7.1: Classificador MLP com características Lofar, histograma com freqüência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo à direita.

7.1.2 MLP - Resultados PCA

Para um classificador MLP que, com sinais *Dataframe-TPSW* como vetores de características PCA, os seguintes resultados são calculados para 37 épocas no eixo horizontal e a precisão alcançada é exibida no eixo vertical. A máquina é treinada usando esses hiperparâmetros. A função de ativação tipo *logistic*, α de 0,002711, número de neurônios na camada oculta de 95 unidades, taxa de aprendizagem inicial de 0,00431, número máximo de iterações de 967, solucionador *adam*; obtendo uma Acc de 0,6042. Com esses hiperparâmetros treina-se a máquina e a submete a um conjunto de teste obtendo os resultados a seguir.

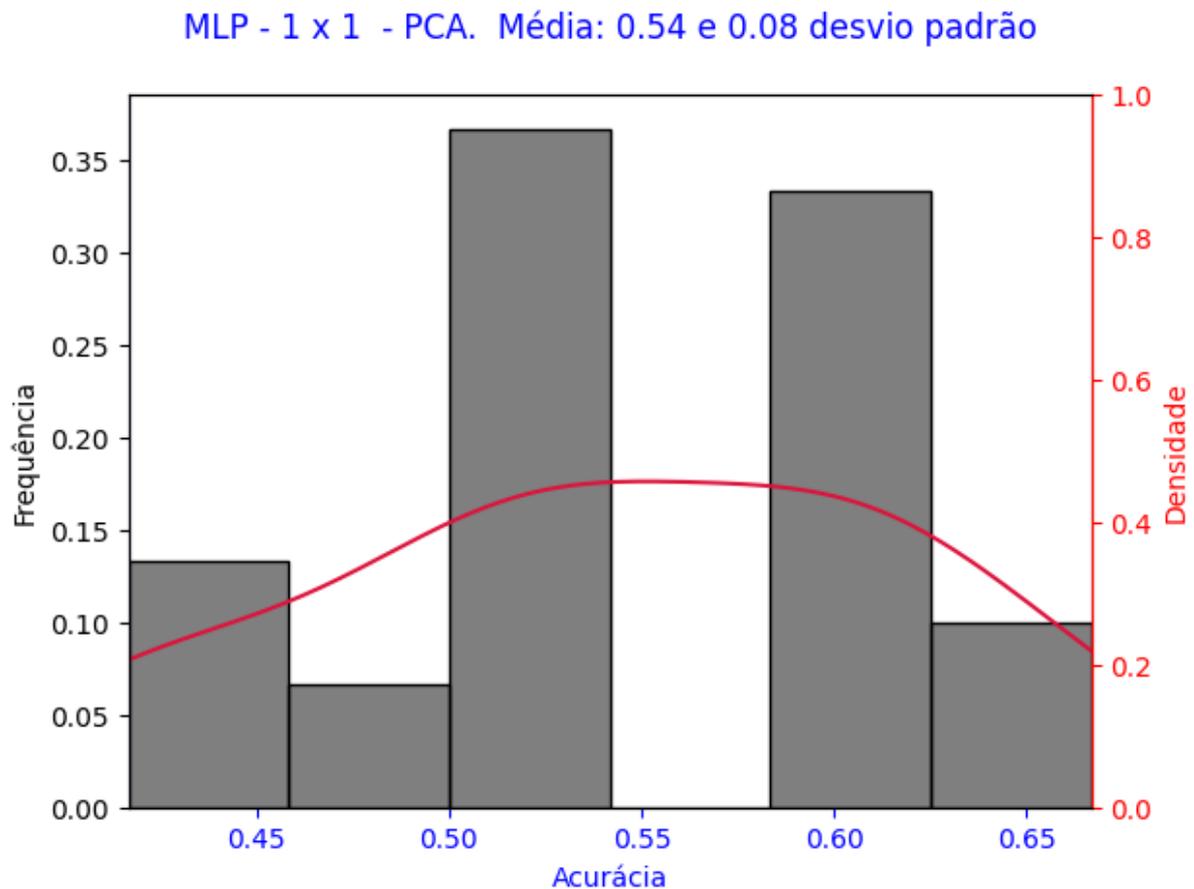


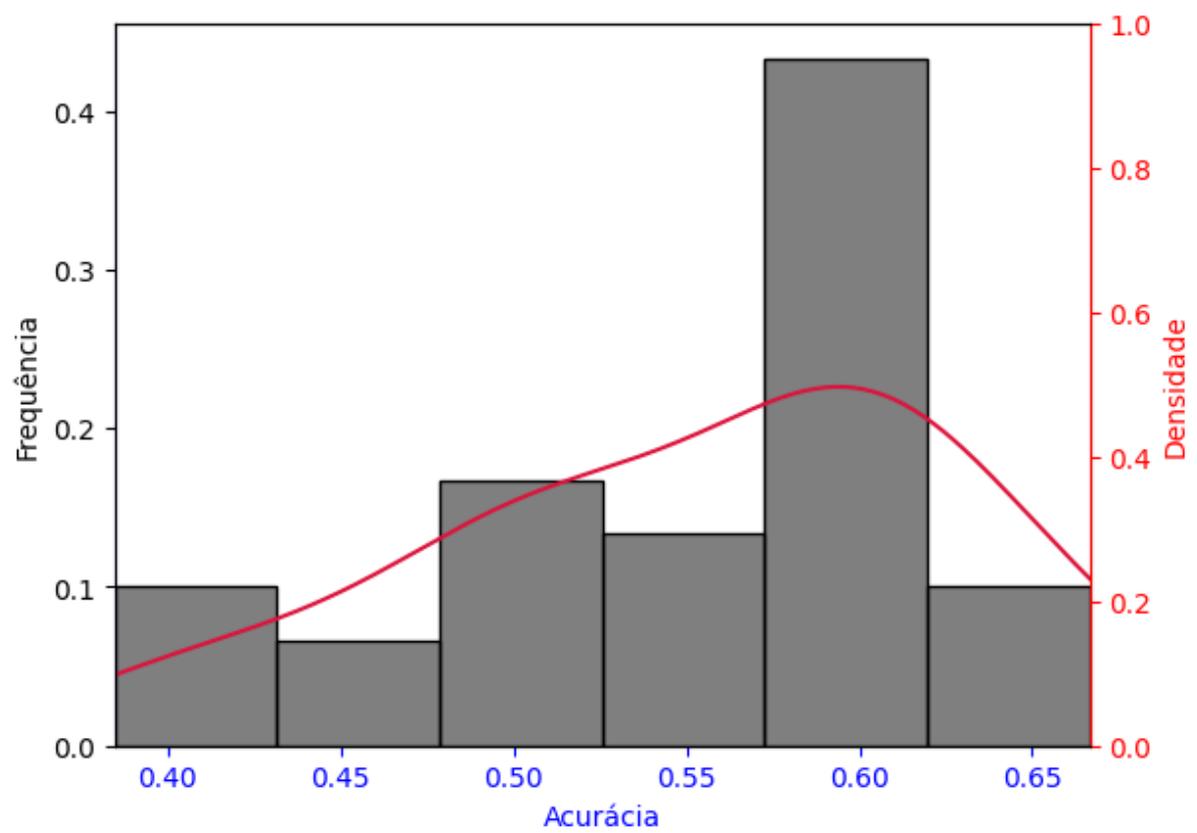
Figura 7.2: Classificador MLP com características PCA, histograma com frequência de ocorrência em preto com eixo à esquerda, e estimativa de densidade por *Kernel* em vermelho e eixo à direita.

7.1.2.1 Resultados NLPCA

Para um classificador MLP que com sinais *Dataframe-TPSW* como vetores de características NLPCA, emprega-se o NLPCA na esperança que a separação das características mais proeminentes auxilie a encontrar um hiperplano que efetivamente separe as classes. Os hiperparâmetros obtidos são a função de ativação tipo *logistic*, α de 0,0656, número de neurônios na camada oculta de 100 unidades, taxa de aprendizagem inicial de 0,00369, número máximo de iterações de 100, solucionador *adam*; obtendo uma Acc de 0,6049. Com esses hiperparâmetros treina-se a máquina e a submete a um conjunto de teste obtendo os resultados a seguir.

Esse algoritmo de ML é o melhor exemplo neste trabalho de que fornecer muitos dados pode atrapalhar significativamente o classificador. Observando-se a média e o desvio padrão de todas as implementações verifica-se que a acurácia dessas estão próximas quanto a Acc. As implementações que conseguiram de alguma forma reduzir o número de

MLP - 1 x 1 - NLPCA. Média: 0.55 e 0.07 desvio padrão

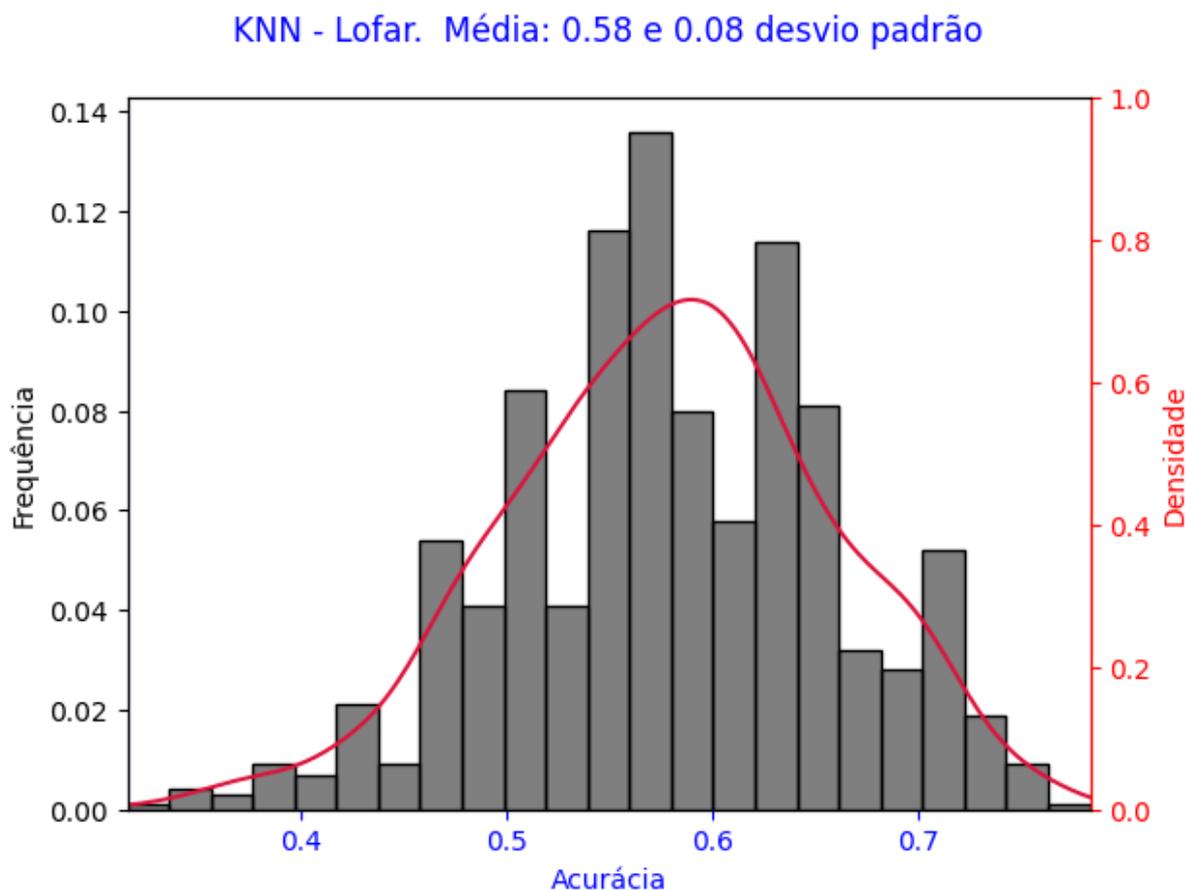


máquinas: com PCA, NLPCA conseguiram um resultado cerca de 10% melhor.

7.2 KNN

7.2.1 KNN - Resultados em Lofar

Para o Classificador KNN, utilizando-se como vetor de características o sinal Lofar sem técnicas de redução, foram obtidos os seguintes resultados em 37 épocas computadas, conforme Figura 6.4, no eixo das abscissas e a acurácia conseguida no eixo das ordenadas. Na subseção 6.2.3, foram encontrados os hiperparâmetros que melhor ajustam a máquina, necessitando de quarenta iterações, utilizando um vizinho e com a distância euclidiana, obtendo uma Acc de 0,5831. Com esses hiperparâmetros treina-se a máquina e a submete a um conjunto de teste obtendo os resultados a seguir.



One

more step before you proceed...

Figura 7.3: KNN-Lofar, histograma com frequência de ocorrência em preto com eixo à esquerda e estimativa de densidade por *Kernel* em vermelho e eixo à direita.

7.2.2 KNN - Resultados com Pré-Processamento PCA

Para um classificador KNN com sinais *Dataframe-TPSW* como vetores de características PCA, os seguintes resultados são calculados para 37 épocas, conforme a Figura 7.4 no eixo horizontal e a precisão alcançada é exibida no eixo vertical. A máquina é treinada com hiperparâmetros obtidos após quarenta iterações, utilizando um vizinho e com a distância euclidiana, obtendo uma Acc de 0,5831. Com esses hiperparâmetros treina-se a máquina e a submete a um conjunto de teste obtendo os resultados a seguir. e é executada em um conjunto de testes que produz os resultados na Figura 7.4.

KNN - PCA. Média: 0.58 e 0.08 desvio padrão

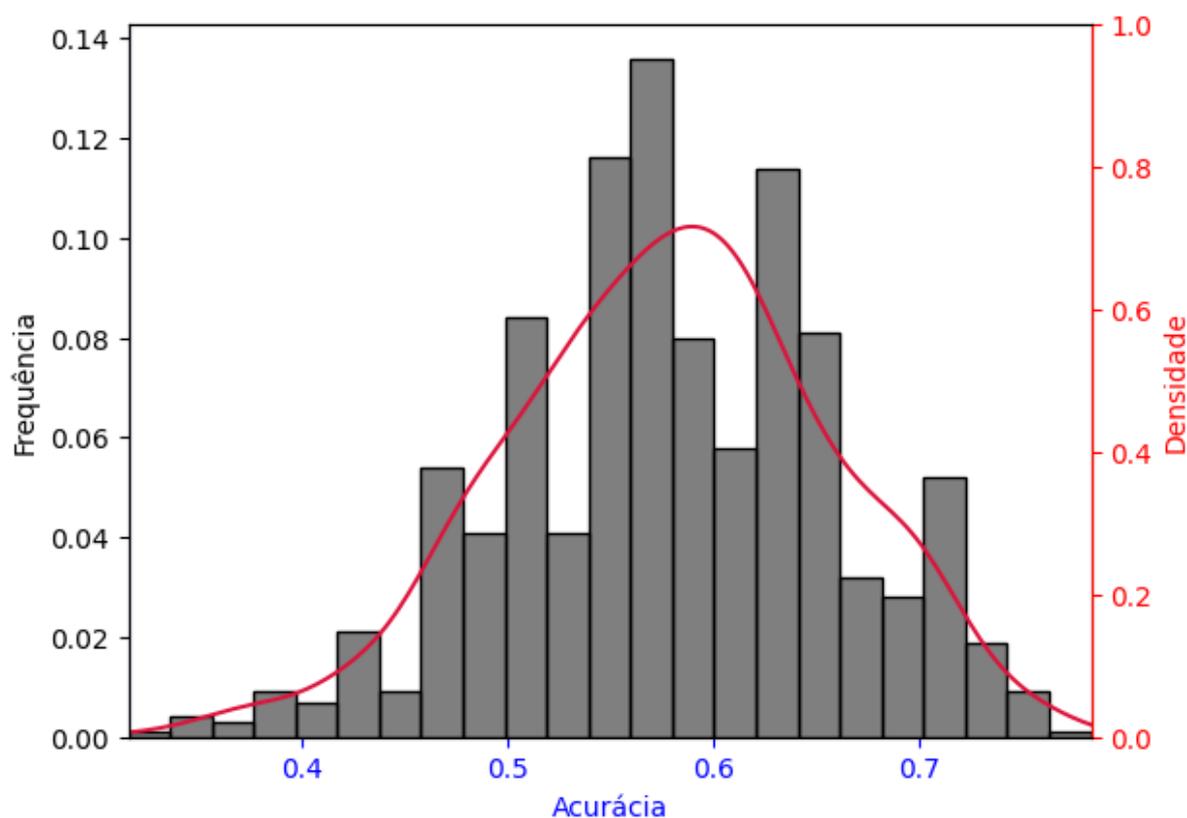


Figura 7.4: KNN-PCA

, histograma com frequência de ocorrência em preto com eixo à esquerda e estimativa de densidade por *Kernel* em vermelho e eixo à direita.

7.2.3 KNN - Resultados NLPCA

Para um classificador KNN com sinais *Dataframe-TPSW* como vetores de características NLPCA são construídas 1000 máquinas e verifica-se a acurácia de cada uma delas,

intentando observar o uso do algoritmo em número suficiente para que seu resultado não fique ao acaso, esse resultado é apresentado na Figura 7.5. O ajuste é feito após quarenta iterações, utilizando um vizinho e com a distância euclidiana, obtendo uma Acc de 0,5831. Com esses hiperparâmetros treina-se a máquina e a submete a um conjunto de teste obtendo os resultados a seguir.

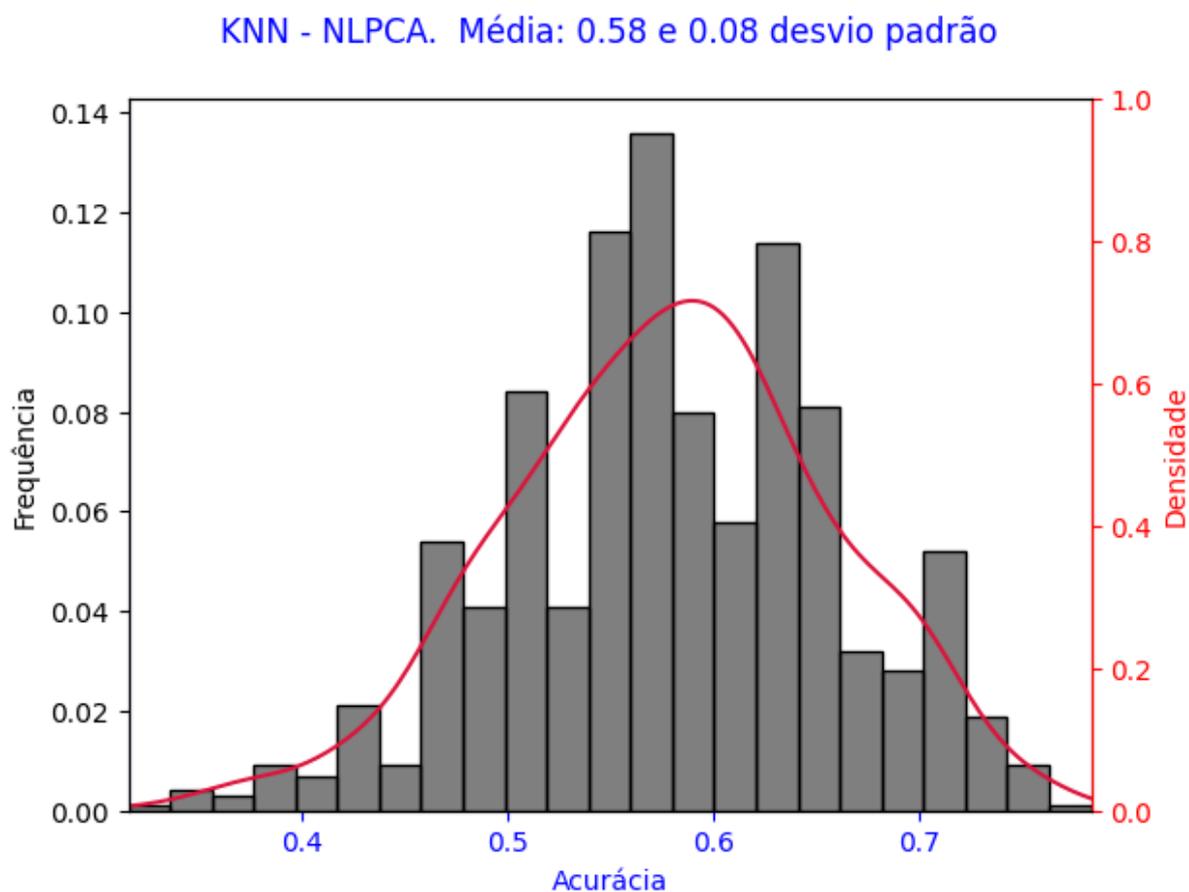


Figura 7.5: KNN-NLPCA, histograma com frequência de ocorrência em preto com eixo à esquerda e estimativa de densidade por *Kernel* em vermelho e eixo à direita.

O algoritmo KNN é treinado no conjunto de dados rotulados. Durante o treinamento, o algoritmo armazena todos os vetores de características e suas classes correspondentes. Quando um novo sinal acústico é recebido, suas características são extraídas e o algoritmo KNN calcula a distância entre o vetor de características do novo sinal e aqueles dos exemplos fornecidos no conjunto de treinamento. Os k vizinhos mais próximos do novo sinal são identificados com base numa métrica de distância. Uma vez identificados os k vizinhos mais próximos, um esquema de votação majoritária é usado para atribuir um rótulo de classe ao novo sinal. O rótulo de classe com a frequência mais alta entre os k vizinhos é atribuído ao novo sinal. O desempenho do classificador KNN obtido é de 58%,

não conseguindo um resultado considerado satisfatório.

7.3 Ridge

7.3.1 Ridge - Resultados Lofar

Para o Classificador Ridge utilizando como vetor de características o sinal LOFAR sem técnicas de redução, foram obtidos os seguintes resultados em 37 épocas computadas no eixo das abscissas da Figura 7.6 e a acurácia obtida no eixo das ordenadas. Na subseção 6.3.2, encontrou-se os hiperparâmetros , taxa de aprendizagem de 10, *kernel* tipo linear, fator de regularização de 10^{-6} , resultando num R^2 score de 0,97948. Com esses hiperparâmetros treina-se a máquina e a submete a um conjunto de teste obtendo os resultados abaixo.

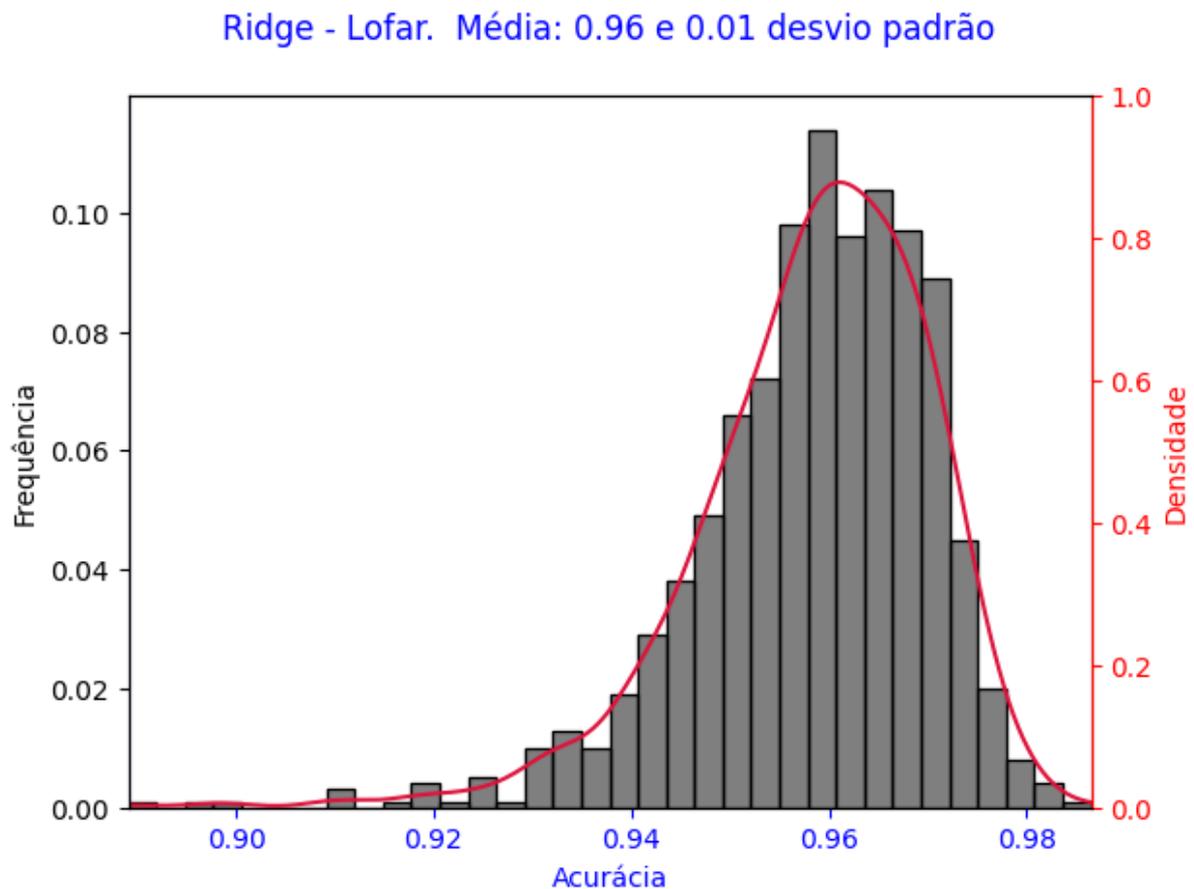


Figura 7.6: Ridge - Lofar, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo a direita.

7.3.2 Ridge - Resultados PCA

Com o bom resultado obtido na seção anterior, testa-se o classificador Ridge que com sinais *Dataframe-TPSW* como vetores de características PCA objetivando confirmar se um vetor de características menor pode resultar em ganho na acurácia, que é uma hipótese verificada na Figura 7.7. O resultado da otimização bayesiana é taxa de aprendizagem de 0,99999, *kernel* tipo linear, fator de regularização de 10^{-6} , resultando num R^2 score de 0,97948.

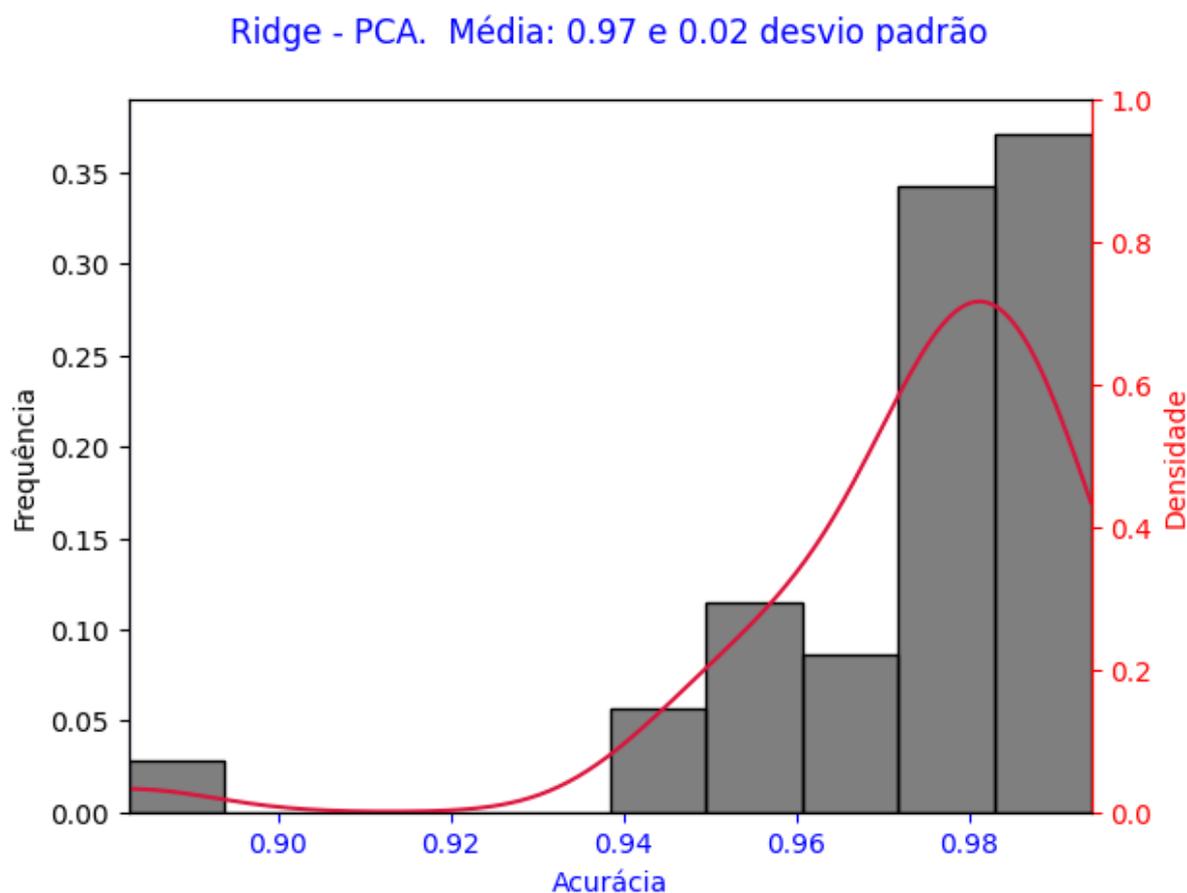


Figura 7.7: Ridge - PCA, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo à direita.

7.3.3 Ridge - Resultados NLPCA

O objetivo desta subseção é semelhante ao da subseção anterior, mas agora utilizando um vetor de entrada com características de separação não-lineares. O resultado do otimização bayesiana é taxa de aprendizagem de 0,5352417, *kernel* tipo linear, fator de regularização de 0,74846, resultando num R^2 score de 0,96220.

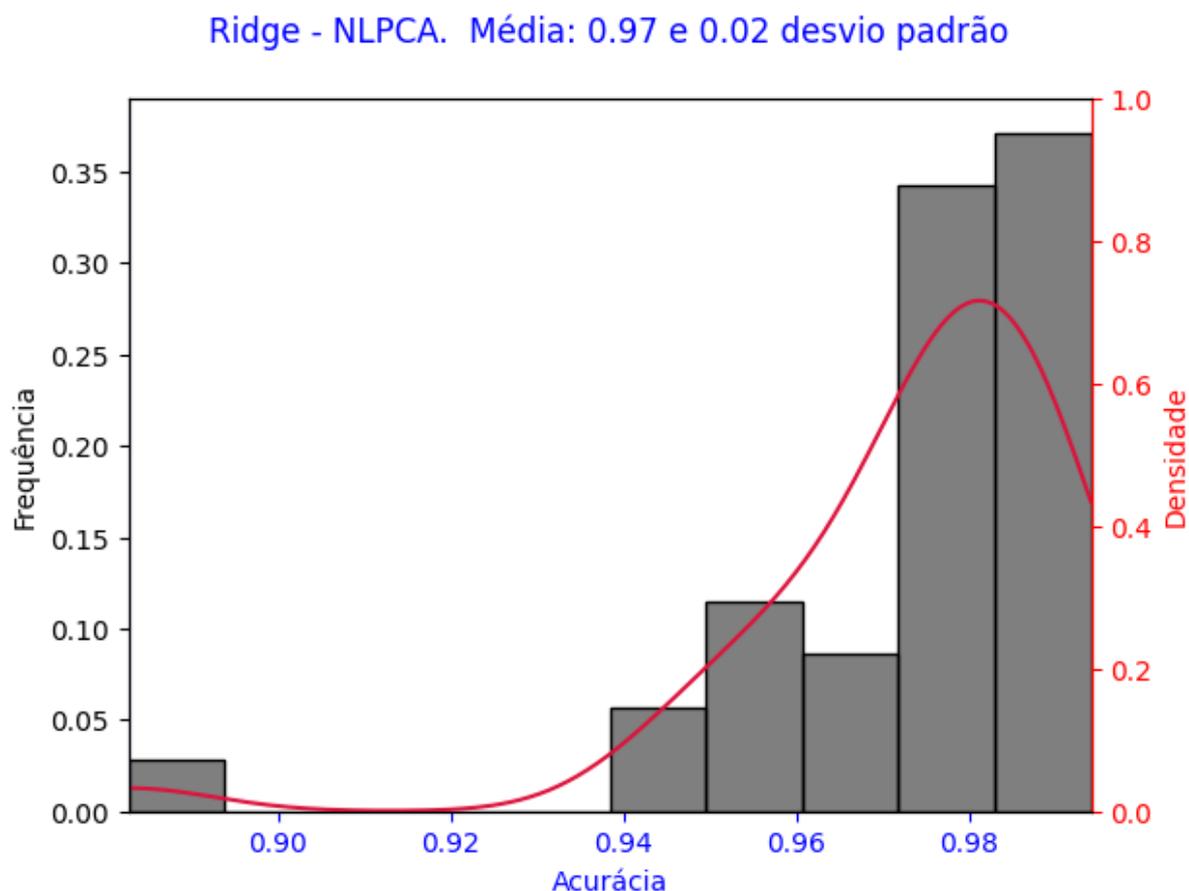


Figura 7.8: Ridge - PCA, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo a direita.

A análise dos resultados da classificação de assinaturas acústicas usando a regressão Ridge denota a capacidade do modelo de generalizar para dados desconhecidos alcançando uma Acc de 96% de classificação correta.

7.4 SVM

7.4.1 SVM- Resultados no Lofar

Para o Classificador SVM -LOFAR, foram encontrados os hiperparâmetros que melhor ajustam a máquina. Na subseção 6.4.3, com esses hiperparâmetros obtidos como resultado do otimização bayesiana para o configuração 1×1 , o inverso da taxa de aprendizagem de 21792, a função de perda tipo *squared hinge*, a perda ℓ_2 , resultando numa Acc de 0,5467; e para o configuração $1 \times T$, o inverso da taxa de aprendizagem de 47730, a função de perda tipo *squared hinge*, a perda ℓ_2 , resultando numa Acc de 0,5184. Treina-se a máquina e a submete a um conjunto de teste obtendo os resultados abaixo.

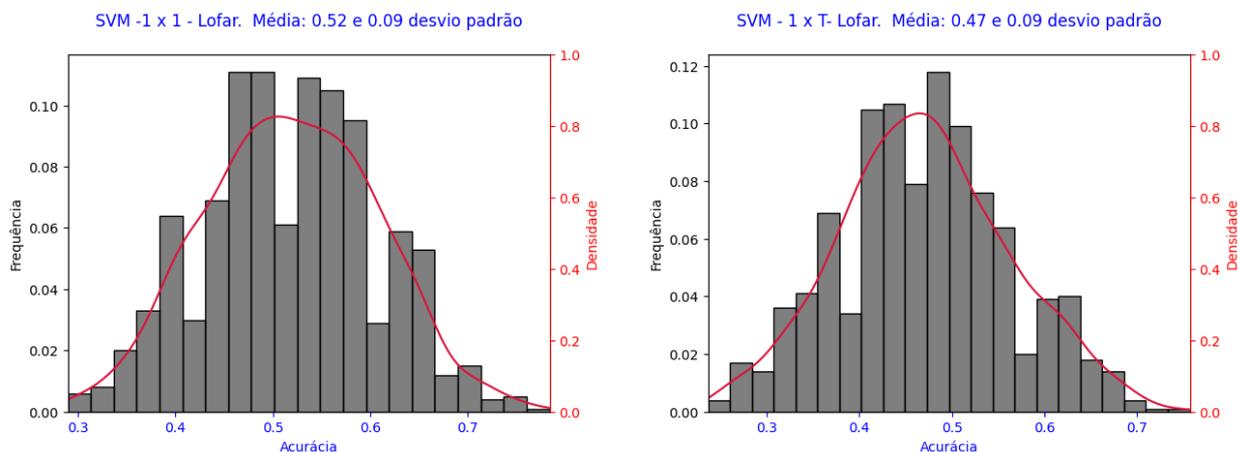


Figura 7.9: Classificador SVM com características LOFAR, à esquerda, usa-se a estratégia 1×1 , e à direita a estratégia $1 \times T$, histograma com frequência de ocorrência em preto com eixo à esquerda e estimativa de densidade por *Kernel* em vermelho e eixo à direita.

7.4.2 SVM- Resultados com Pré-Processamento PCA

Para o classificador SVM com sinais *Dataframe-TPSW-PCA* obtêm-se os resultados da otimização dos hiperparâmetros e executa um conjunto de testes que produz os resultados abaixo. O resultado do otimização bayesiana para o configuração 1×1 , o inverso da taxa de aprendizagem de 773393, a função de perda tipo *hinge*, a perda ℓ_2 , resultando numa Acc de 0,5260; e para o configuração $1 \times T$, o inverso da taxa de aprendizagem de 773393, a função de perda tipo *squared hinge*, a perda ℓ_2 , resultando numa Acc de 0,56801. Treina-se a máquina e a submete a um conjunto de teste obtendo os resultados abaixo.

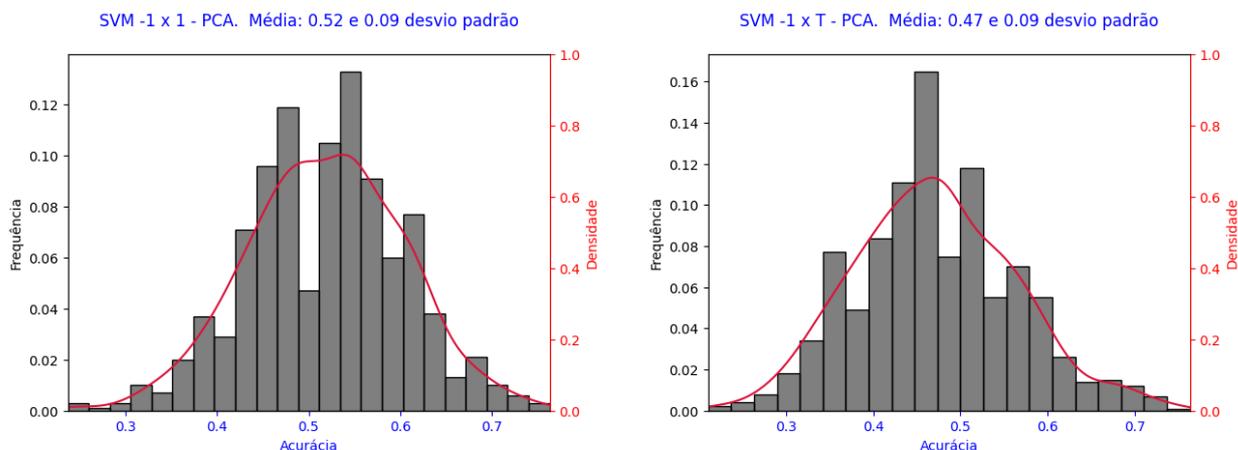


Figura 7.10: Classificador SVM com características PCA, à esquerda, usa-se a estratégia 1×1 , e à direita a estratégia $1 \times T$, histograma com frequência de ocorrência em preto com eixo à esquerda e estimativa de densidade por *Kernel* em vermelho e eixo à direita.

7.4.3 SVM- Resultados NLPCA

Para o classificador SVM com sinais *Dataframe-TPSW* como vetores de características NLPCA, o resultado dos hiperparâmetros ótimos após aferição em cerca de mil máquinas é representada na Figura 7.11. O resultado da otimização bayesiana para a configuração 1×1 , o inverso da taxa de aprendizagem de 40472, a função de perda tipo *hinge*, a perda ℓ_2 , resultando numa Acc de 0,5338; e para a configuração $1 \times T$, o inverso da taxa de aprendizagem de 36921, a função de perda tipo *squared hinge*, a perda ℓ_2 , resultando numa Acc de 0,51397. Treina-se a máquina e a submete a um conjunto de teste obtendo os resultados abaixo.

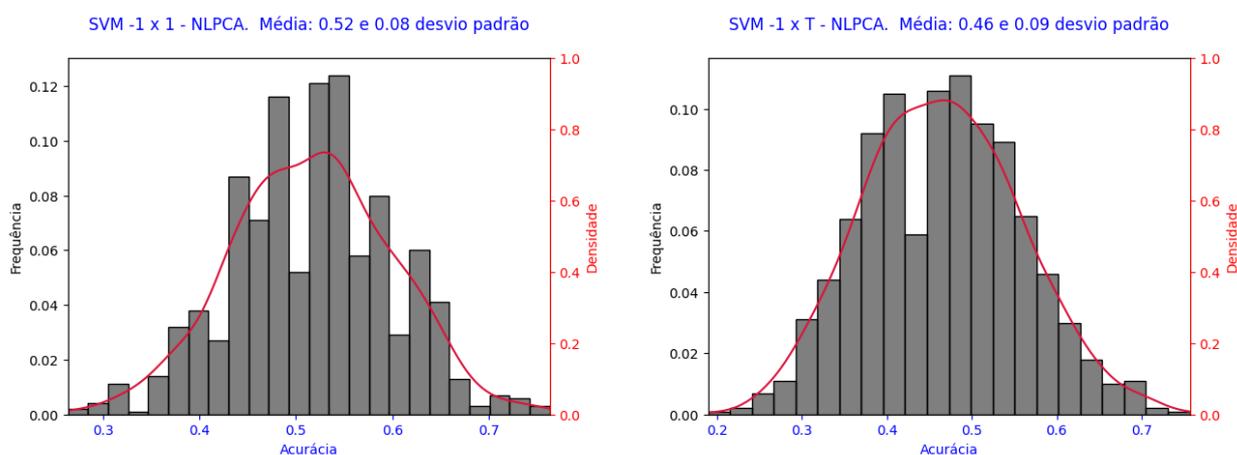


Figura 7.11: Classificador SVM com características NLPCA, à esquerda, usa-se a estratégia 1×1 , e à direita a estratégia $1 \times T$, histograma com frequência de ocorrência em preto com eixo à esquerda e estimativa de One more step before you proceed... densidade por *Kernel* em vermelho e eixo à direita.

A SVM encontra vetores diretores que criam um plano onde, de cada lado do plano, está uma classe. A máquina trabalha melhor em casos um contra um. É interessante entender o quanto os dados estão superpostos e difíceis de separar num hiperplano.

7.5 Regressão Logística

7.5.1 Logit - Resultados em Lofar

Para o Classificador trabalhando com regressão logística -LOFAR sem técnicas de redução foram obtidos os resultados expressos na Figura 7.12. Na subseção 6.5.3, foram encontrados os hiperparâmetros que melhor ajustam a máquina. O resultado da otimização bayesiana para a configuração 1×1 , o inverso da força de regularização 1,21020, a perda

ℓ_1 , resultando numa Acc de 0,9427; e para o configuração $1 \times T$, o inverso da força de regularização 1,8454, a perda ℓ_1 , resultando numa Acc de 0,945352. Treina-se a máquina e a submete a um conjunto de teste obtendo os resultados abaixo.

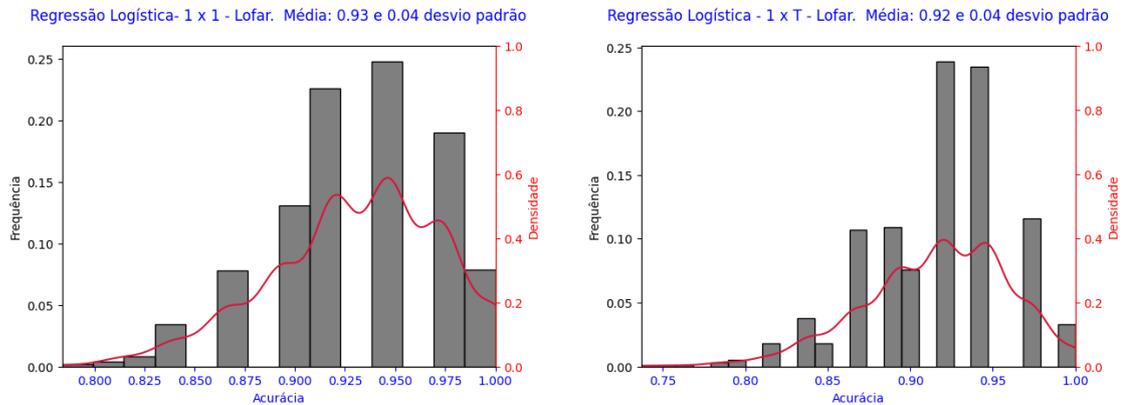


Figura 7.12: Classificador Logit com características LOFAR, à esquerda, usa-se a estratégia 1×1 , e à direita a estratégia $1 \times T$, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo a direita.

7.5.2 Logit - Resultados PCA

O classificador com regressão Logística que com sinais *Dataframe-TPSW-PCA* é submetido a diversas realizações da máquina e ao alcançar ser resultado ótimo, treina-se a máquina usando esses hiperparâmetros. O resultado do otimização bayesiana para o configuração 1×1 , o inverso da força de regularização 0,74538, a perda ℓ_1 , resultando numa Acc de 0,94535; e para o configuração $1 \times T$, o inverso da força de regularização 1,11829, a perda ℓ_1 , resultando numa Acc de 0,94535. Treina-se a máquina e a submete a um conjunto de teste obtendo os resultados abaixo. Com estes dados executa-se um conjunto de testes que produz os resultados a seguir.

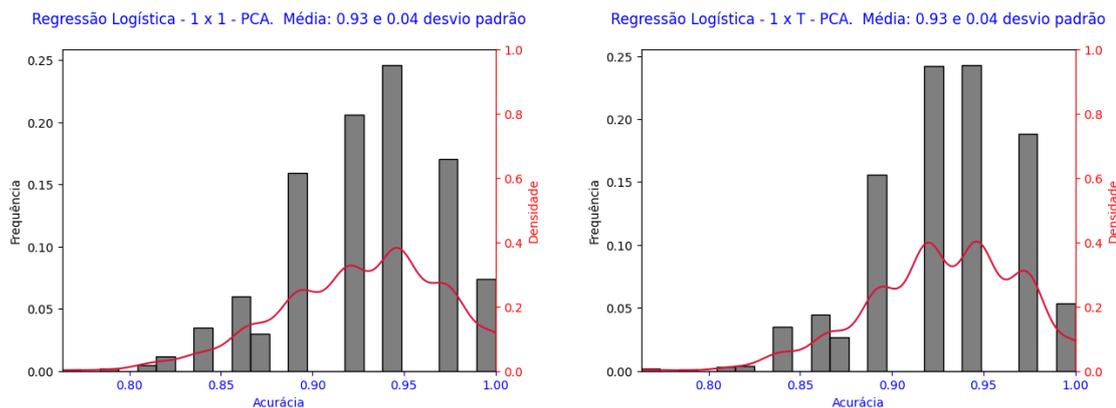


Figura 7.13: Classificador Logit com características LOFAR, à esquerda, usa-se a estratégia 1×1 , e à direita a estratégia $1 \times T$.

7.5.3 Logit - Resultados NLPCA

Para um classificador utilizando regressão Logística que com sinais *Dataframe-TPSW* como vetores de características NLPCA os seguintes resultados são calculados para 37 épocas no eixo horizontal e a precisão alcançada é exibida no eixo vertical. O resultado do otimização bayesiana para o configuração 1×1 , o inverso da força de regularização 1,43926, a perda ℓ_1 , resultando numa Acc de 0,9479; e para o configuração $1 \times T$, o inverso da força de regularização 1,3543, a perda ℓ_1 , resultando numa Acc de 0,9427. Treina-se a máquina e a submete a um conjunto de teste obtendo os resultados abaixo.

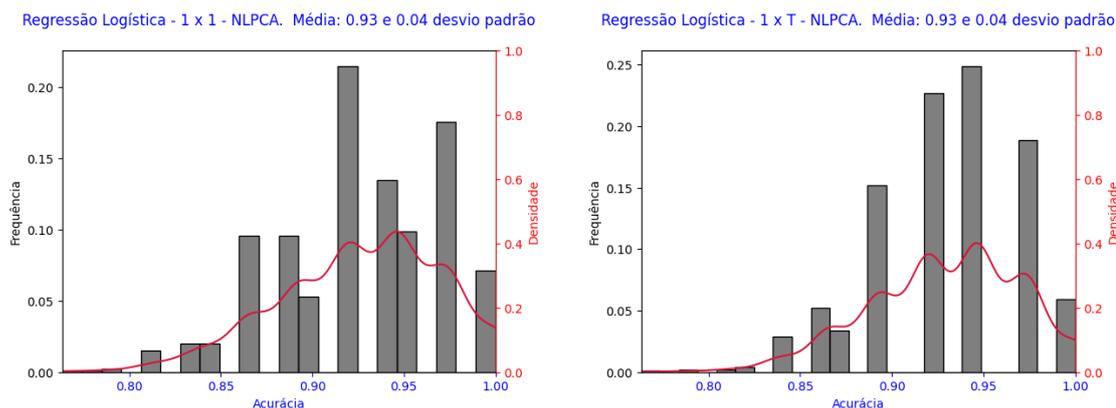


Figura 7.14: Classificador Logit com características LOFAR, à esquerda, usa-se a estratégia 1×1 , e à direita a estratégia $1 \times T$, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo à direita.

Se a SVM mostra o quanto é difícil separar os dados, a regressão logística torna claro que nada melhor que um Kernel não linear para trabalhar com não-linearidades, com

apenas dois parâmetros para otimizar consegue-se um resultado significativo.

7.6 LDA

7.6.1 LDA - Resultados Lofar

Para o ClassificadorLDA utilizando como vetor de características o sinal LOFAR, espera-se um bom resultado uma vez que em sua composição já está implícito o número de classes como discriminante, se elas estiverem bem separadas o classificador será ideal. Na subseção 6.6.3, foram encontrados os hiperparâmetros que melhor ajustam a máquina. O resultado do otimização bayesiana, sete componentes, solucionador tipo *svd*, gerando uma Acc de 0,92450. Assim treina-se a máquina e a submete a um conjunto de teste obtendo os resultados abaixo.

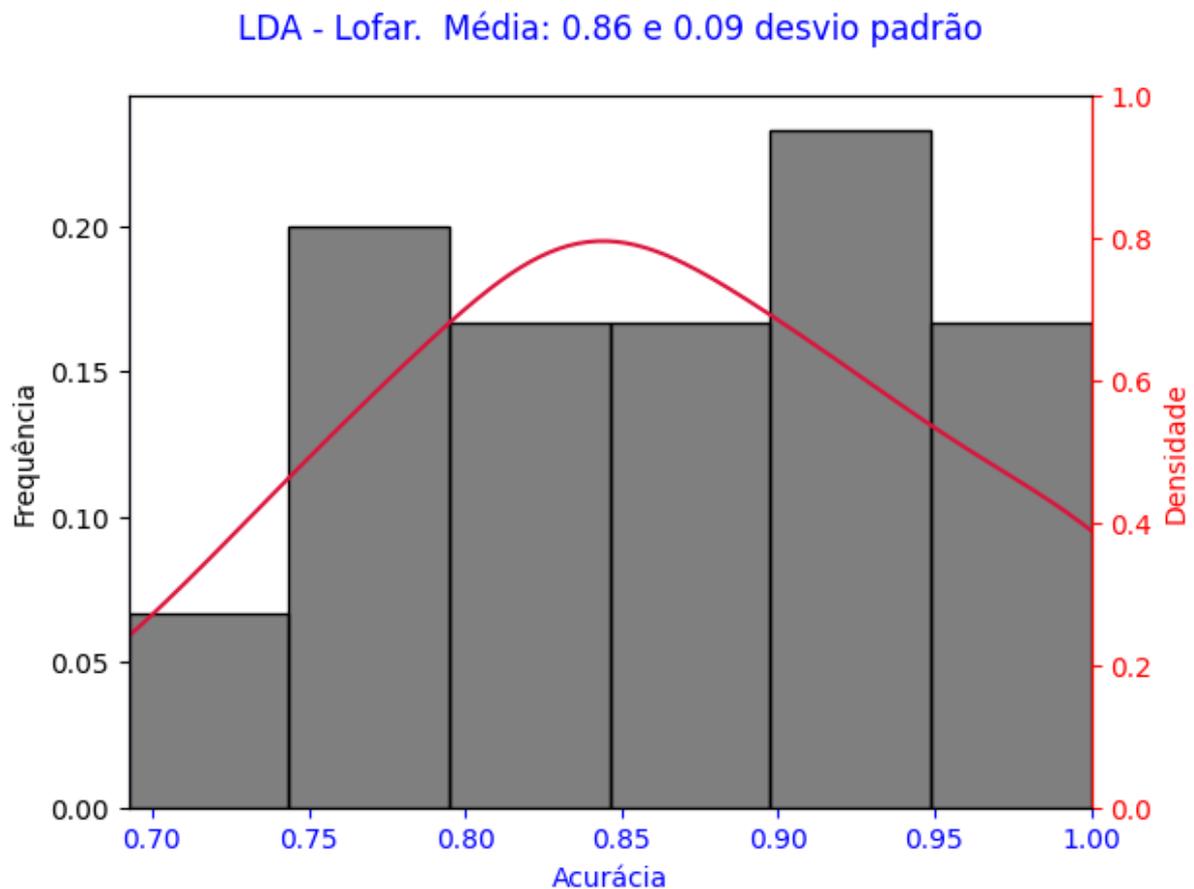


Figura 7.15: LDA - Lofar, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo a direita.

7.6.2 LDA - Resultados PCA

Para um classificador utilizando LDA que com sinais *Dataframe-TPSW* como vetores de características PCA os seguintes resultados são calculados para 37 épocas no eixo horizontal e a precisão alcançada é exibida no eixo vertical. A máquina é treinada usando esses hiperparâmetros. O resultado do otimização bayesiana, dois componentes, solucionador tipo *svd*, gerando uma Acc de 0,92450. Assim treina-se a máquina e a submete a um conjunto de teste obtendo os resultados abaixo.

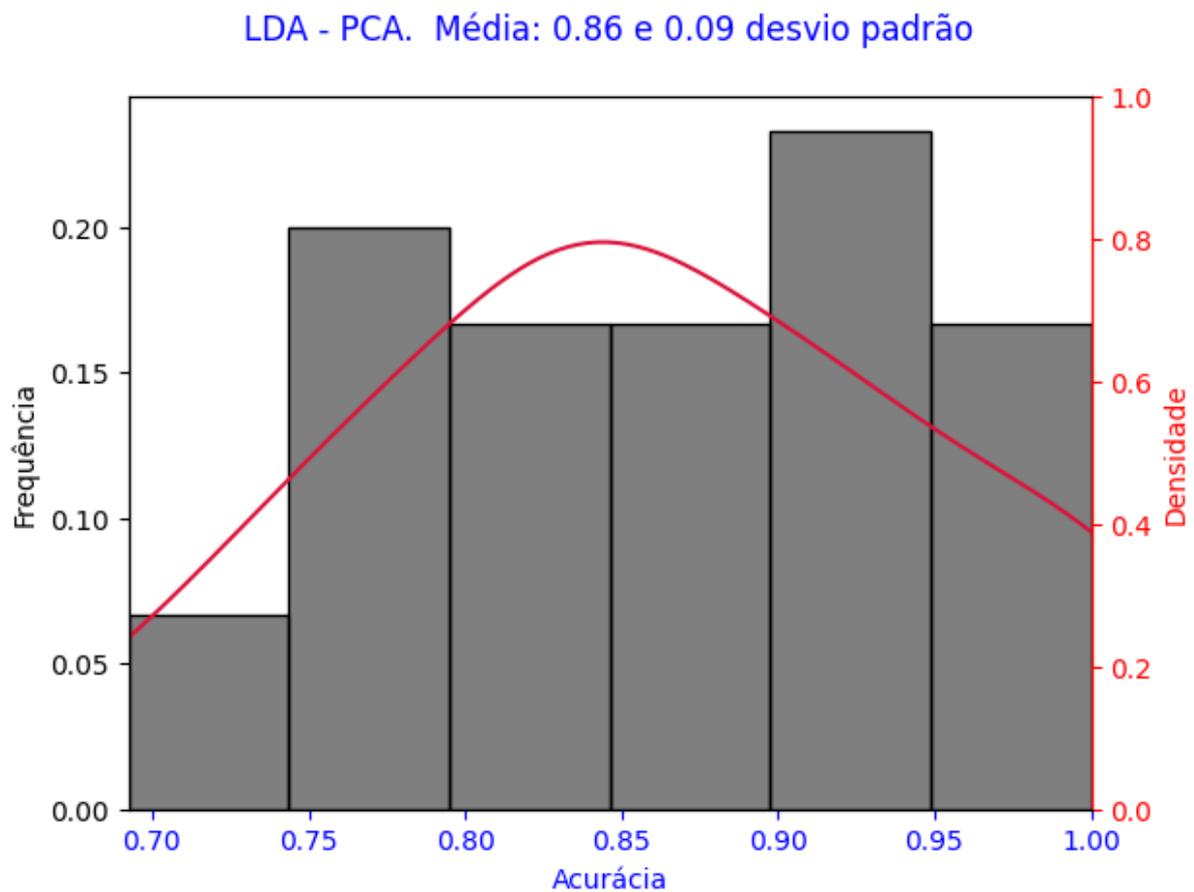


Figura 7.16: LDA - PCA, histograma com freqüência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo a direita.

7.6.3 LDA - Resultados NLPCA

Para um classificador utilizando LDA que com sinais *Dataframe-TPSW* como vetores de características NLPCA os seguintes resultados são calculados para 37 épocas no eixo horizontal e a precisão alcançada é exibida no eixo vertical. A máquina é treinada usando esses hiperparâmetros. O resultado do otimização bayesiana, dez componentes, solucionador tipo *svd*, gerando uma Acc de 0,92450. Assim treina-se a máquina e a submete a

um conjunto de teste obtendo os resultados abaixo.

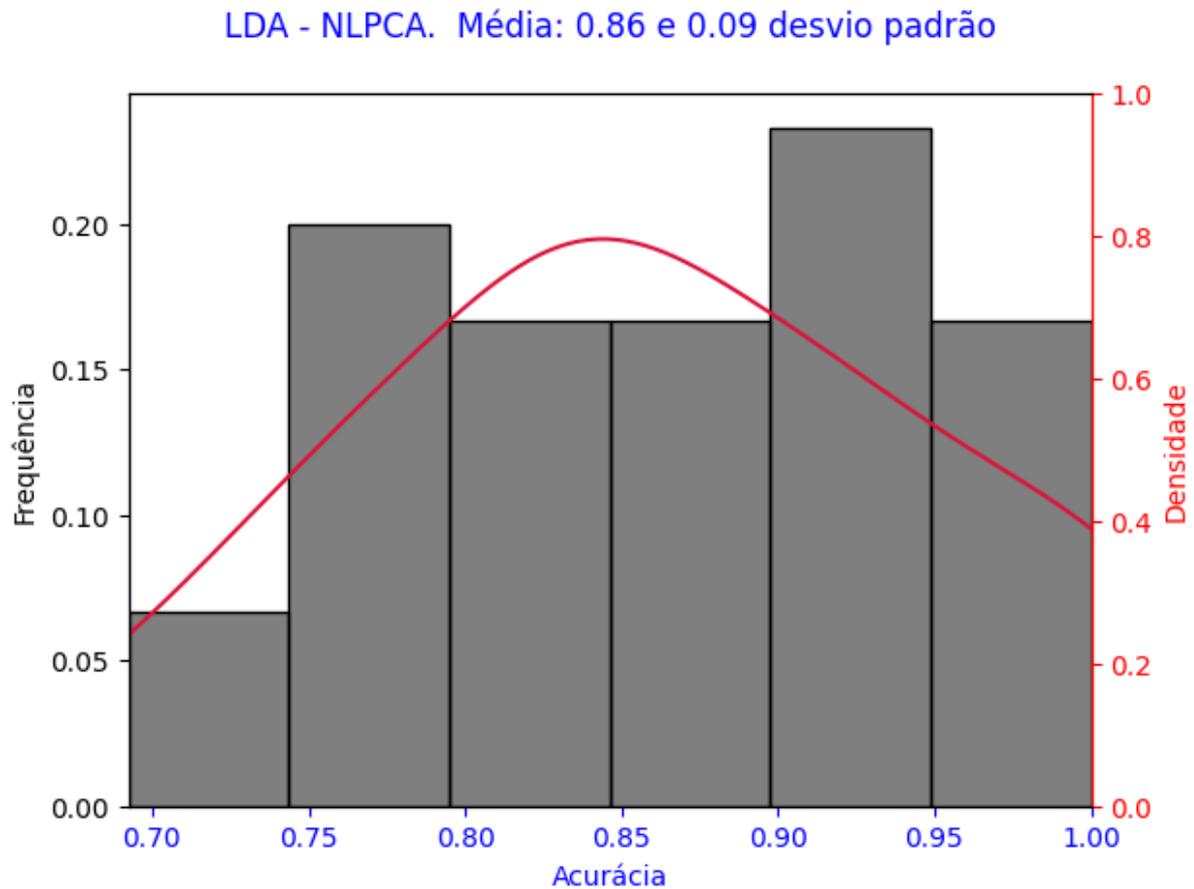


Figura 7.17: LDA - NLPCA,, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo a direita.

O LDA obteve boa acurácia bem com o NLPCA sendo a melhor das três implementações. Esse algoritmo também é interessante para verificar a diferença entre o R^2 Score e o Acc, comparando-se as Figuras anteriores, observa-se que o R^2 Score ficou próximo de 93% mas ao classificar teve um desempenho menor 3% menor.

7.7 XgBoost

7.7.0.1 XgBoost- Resultados Lofar

Para o Classificador XgBoost utilizando como vetor de características o sinal LOFAR sem técnicas de redução, foram obtidos os seguintes resultados em 37 épocas computadas no eixo das abscissas e a acurácia conseguida no eixo das ordenadas. Na subseção 6.7.2.1, foram encontrados os hiperparâmetros que melhor ajustam a máquina. A máquina é trei-

nada usando esses hiperparâmetros. O resultado do otimização bayesiana, α de 0,000287, **colsample bytree** de 0,37423, γ a 0,142805, λ de 0,00010, taxa de aprendizagem de 0,142805, **max depth** de 0,031765, *min childweight* de 0,031765, número de estimador 1000 unidades, uma subamostra gerando uma Acc de 0,99740. Assim treina-se a máquina e a submete a um conjunto de teste obtendo os resultados abaixo.

XGBoost - Lofar. Média: 0.99 e 0.03 desvio padrão

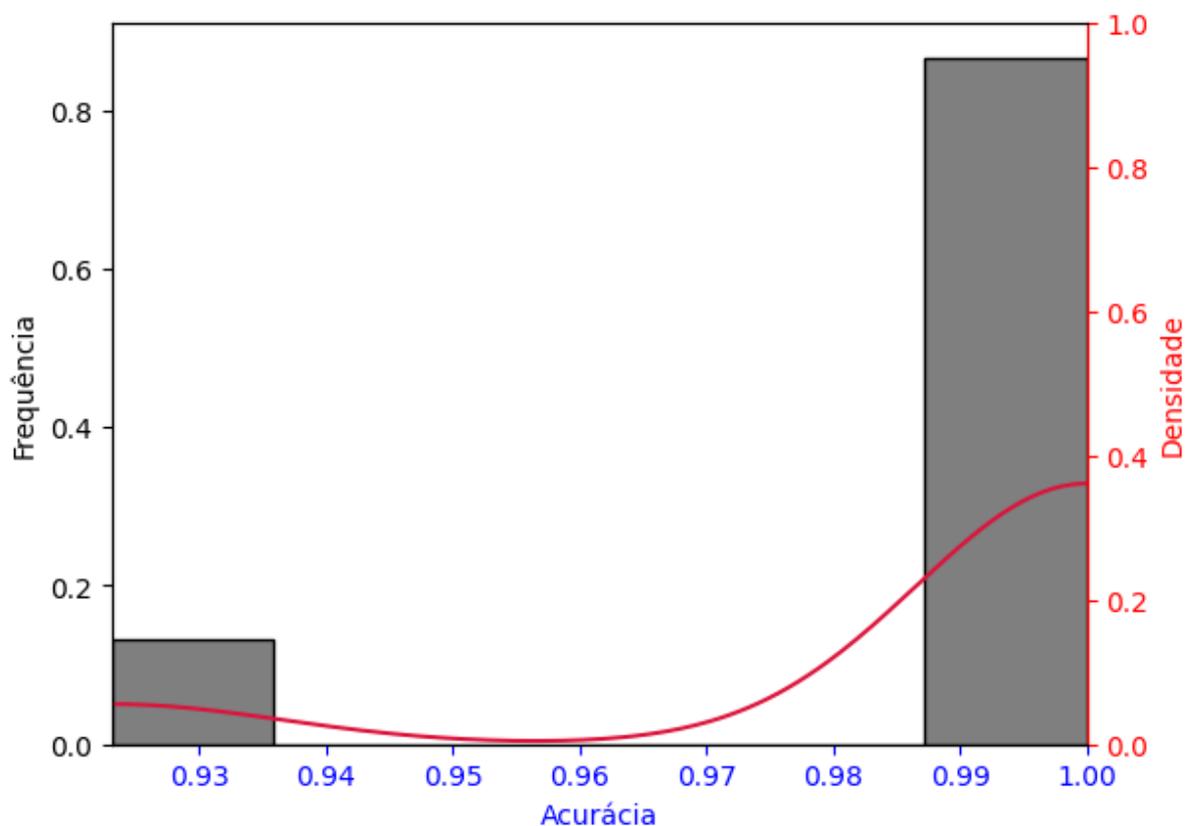
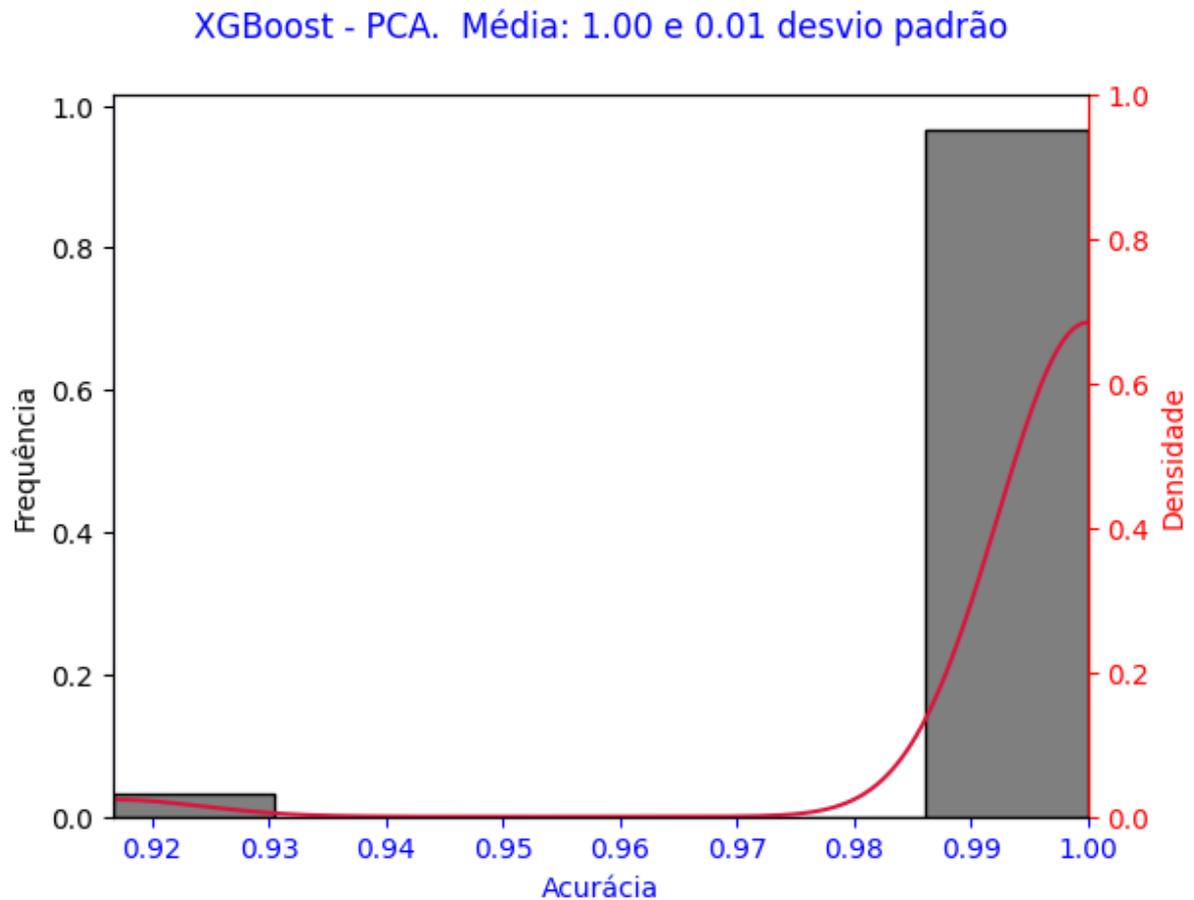


Figura 7.18: XGBoost - Lofar, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo a direita.

7.7.0.2 XgBoost- Resultados PCA

Para um classificador utilizando XgBoost que com sinais *Dataframe-TPSW* como vetores de características PCA os seguintes resultados são calculados para 37 épocas no eixo horizontal e a precisão alcançada é exibida no eixo vertical. A máquina é treinada usando esses hiperparâmetros. O resultado do otimização bayesiana, α de 0,01512, **colsample bytree** de 0,47659, γ a 0,004465, λ de $9.4274 \cdot 10^{-6}$, taxa de aprendizagem de 0,01, **max depth** de 9, *min childweight* de 1, número de estimador 1000 unidades, uma subamostra de 0,93008 gerando uma Acc de 0,99213. Assim treina-se a máquina e a submete a um

conjunto de teste obtendo os resultados abaixo.



One

more step before you proceed...

Figura 7.19: XGBoost - PCA, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo a direita.

7.7.0.3 XgBoost- Resultados NLPCA

Para um classificador utilizando XgBoost que com sinais *Dataframe-TPSW* como vetores de características NLPCA os seguintes resultados são calculados para 37 épocas no eixo horizontal e a precisão alcançada é exibida no eixo vertical. A máquina é treinada usando esses hiperparâmetros. O resultado do otimização bayesiana, α de 0,05489, **colsample** **bytree** de 0,898444, γ a 1,0, λ de 110^{-6} , taxa de aprendizagem de 1, **max depth** de 1, *min childweight* de 1, número de estimador 1000 unidades, uma subamostra de 0,93008 gerando uma Acc de 0,99213. Assim treina-se a máquina e a submete a um conjunto de teste obtendo os resultados abaixo.

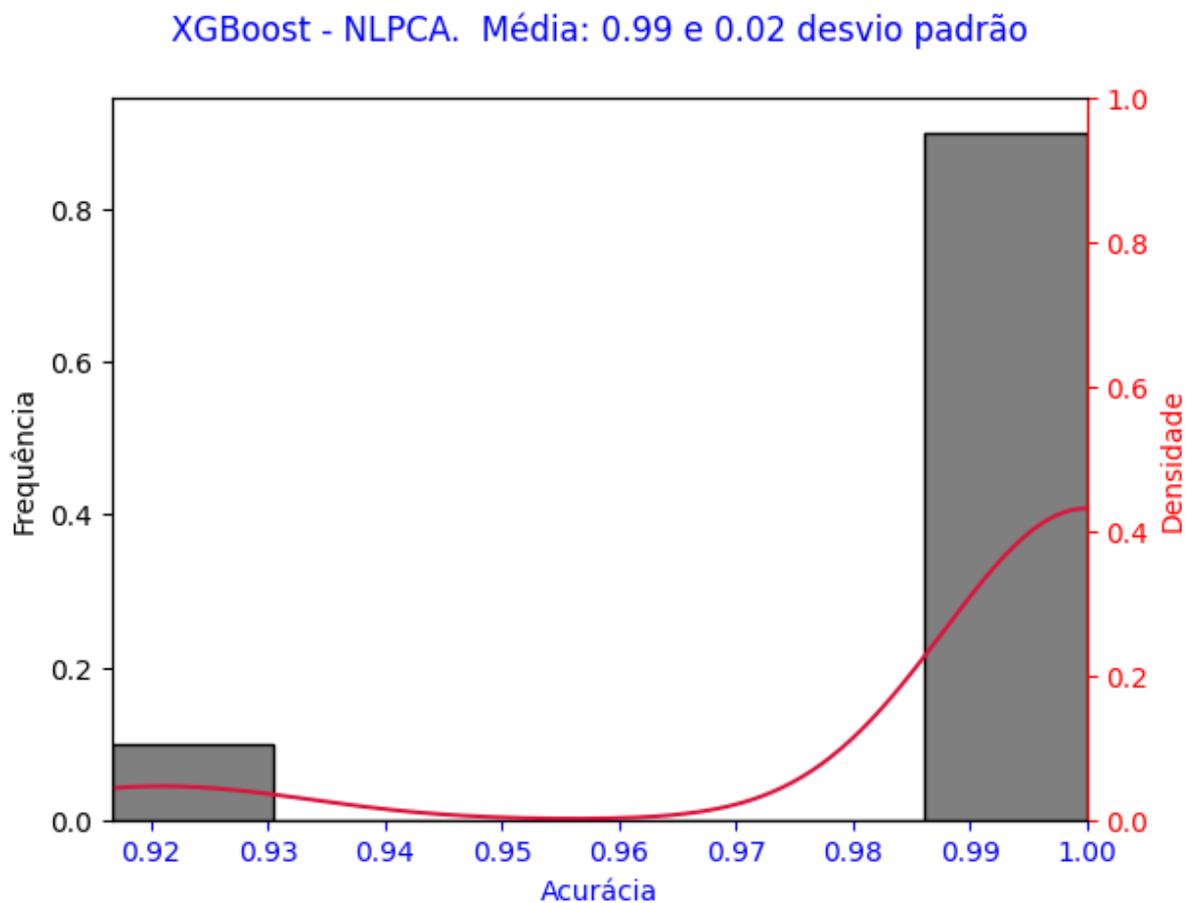


Figura 7.20: XGBoost - NLPCA, histograma com frequência de ocorrência em preto com eixo a esquerda e estimativa de densidade por *Kernel* em vermelho e eixo a direita.

Este foi o melhor resultado resultado 99% com desvio padrão de 0,02%, mostrando o poder desta técnica que trabalha com classificação de árvores em série até que os resíduos sejam classificados. Uma acurácia desta magnitude por certo é adequada ao uso desta máquina.

Alcançar alta precisão em vários testes indica que o modelo provavelmente generalizando bem para dados desconhecidos. O que indica que o modelo aprendeu padrões significativos a partir dos dados de treinamento que se aplicam a exemplos novos e não vistos.

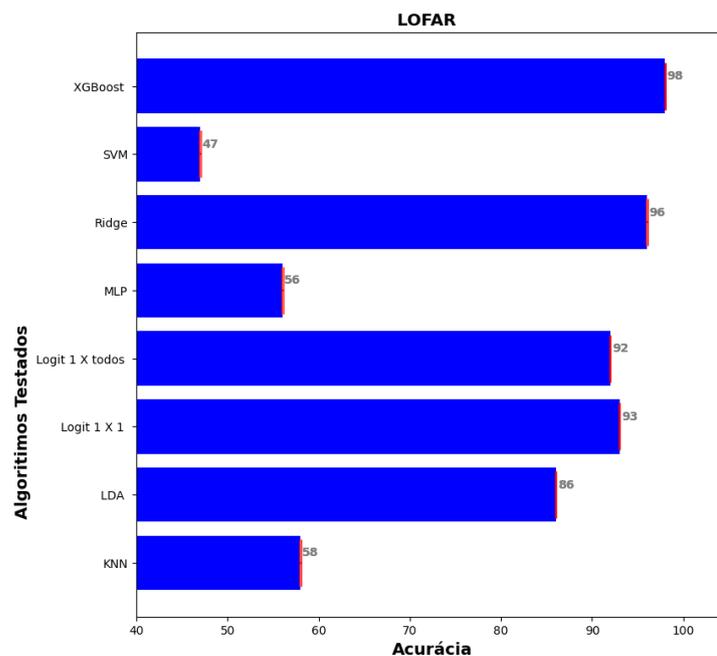
A alta precisão é mantida em diferentes partições de dados. Como o modelo teve um bom desempenho, consistente em vários subconjuntos de dados, adquire-se confiança em suas capacidades de generalização.

7.8 Comparação dos Modelos de Aprendizado de Máquina para Classificação

Nesta subseção, conduzimos uma análise comparando o desempenho de sete modelos diferentes de aprendizado de máquina para a tarefa de classificação. O objetivo é avaliar a eficácia de cada modelo na classificação precisa da variável alvo e identificar o modelo mais adequado.

Neste estudo, comparamos o desempenho de sete modelos populares de aprendizado de máquina: MLP; KNN; Ridge; SVM; Regressão Logística (*One × One*) e (*One × All*); LDA; e XGBoost

A Figura 7.21 apresenta as métricas de desempenho obtidas para cada modelo de aprendizado de máquina.



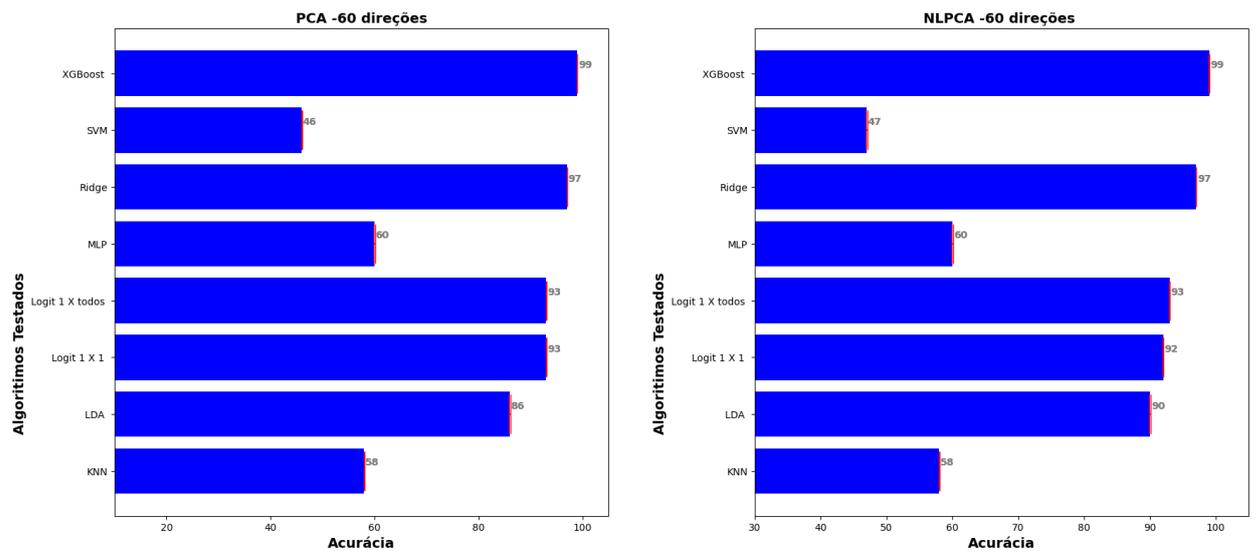


Figura 7.21: Compilação dos resultados de todas as máquinas apresentadas para LOFAR, PCA e NLPCA.

A tabela 7.1 resume os resultados:

Tabela 7.1: Compilação dos resultados apresentados nas seções anteriores, com acurácia e desvio padrão (std).

Máquina	Lofar		PCA		NLPCA	
	acc	std	acc	std	acc	std
MLP	56	0,09	60	0,08	60	0,08
KNN	58	0,08	58	0,08	58	0,08
Ridge	96	0,08	97	0,02	97	0,02
SVM	47	0,09	46	0,09	47	0,09
Regressão Logística (<i>One × One</i>)	93	0,04	93	0,04	92	0,04
Regressão Logística (<i>One × All</i>)	92	0,04	93	0,04	93	0,04
LDA	86	0,09	86	0,09	90	0,07
XGBoost	98	0,03	99	0,02	99	0,03

Em geral os modelos apresentam poucas diferenças na comparação entre os sinais LOFAR, PCA e NLPCA. Portanto, é factível introduzir um vetor de características menor e obter um bom resultado ainda assim. Isso indica no caso de missão naval onde já se sabe quais navios provavelmente estarão na área é possível fazer um classificador mais rápido porém com sem a possibilidade de inserção de novos sinais. Pois caso houvesse um novo sinal teria que se rodar todo o algoritmo de redução de características novamente e não somente incorporar o último sinal.

Os casos mais notórios de aumento de acurácia residem no LDA e MLP. Ambos os algoritmos visam encontrar limites de decisão que separam diferentes classes no espaço de características. No entanto, o MLP pode aprender limites de decisão complexos, incluindo os não lineares, enquanto o LDA assume limites de decisão lineares. E reduzir as características de entrada fez grande diferença para ambos

O XGBoost demonstrou um desempenho consistente, ensejando sua robustez e eficácia para a tarefa de classificação. Superou todos os demais e portanto é entendido frente aos outros classificadores testados como o classificador ideal.

Capítulo 8

Conclusão

A classificação passiva de sinais acústicos SoNaR desempenha um papel crucial na vigilância subaquática e na exploração marinha. Nesta dissertação, tivemos como foco explorar a eficácia de algoritmos de aprendizado de máquina para tarefas de classificação passiva de SoNaR. Comparou-se o desempenho de diversos algoritmos para determinar qual deles seria mais adequado para a classificação precisa de objetos subaquáticos com base em sinais de SoNaR.

Neste estudo, investigamos a eficácia dos seguintes algoritmos de Aprendizado de Máquina para classificação passiva de sinais de SoNaR: Análise de Discriminantes Lineares (LDA), Conjunto de Árvores de Decisão (Xgboost), K-Vizinhos Mais Próximos (KNN), Máquinas de Vetores de Suporte (SVM), Multi-Layer Perceptron (MLP), Regressão Logística e Regressão Ridge. O desempenho desses algoritmos foi avaliado com base em suas capacidades para classificar precisamente os sinais adquiridos do SoNaR e distinguir entre diferentes classes de assinaturas acústicas. Nossos resultados indicam que o XGBoost supera os demais em termos de precisão de classificação e desempenho do modelo.

Para alcançar os objetivos, a revisão da literatura destacou a importância da classificação passiva do SoNaR em diversas aplicações, como defesa naval, pesquisa marinha e robótica subaquática. E, ainda, os diferentes tipos de SoNaR na atualidade que, em outros estudos, mostraram a eficácia dos algoritmos de aprendizado de máquina na classificação de sinais de SoNaR sendo a regressão XGBoost e Ridge escolhas adequadas devido às suas robustez e eficiência.

O capítulo de metodologia descreveu o processo de coleta de dados, o pré-processamento, transformações para conformar o banco de corridas, técnicas de extração de recursos e a avaliação de algoritmos de classificação. O banco de corridas utilizado neste estudo

consistia em sinais de assinatura acústica rotulados, representando diferentes classes de navios. Foram realizados pré-processamento de dados, determinação de características e treinamento de modelos usando os sete algoritmos de aprendizado de máquina.

Apresentou-se uma análise detalhada das métricas de desempenhos do sistemas classificadores, incluindo exatidão de classificação, e R^2 score. A análise revela que o XGBoost alcança maior precisão e melhor desempenho geral em comparação com os outros algoritmos, seguido pelo Ridge, Regressão Logística, Análise de Discriminantes Linear, sendo os algoritmos restantes considerados não-adequados. Discutimos as razões para o desempenho superior do XGBoost, como sua capacidade de lidar adequadamente com padrões complexos e relacionamentos não-lineares nos dados. A abordagem de aprendizagem em conjunto do XGBoost, que combina vários classificadores fracos para formar um classificador robusto, contribuiu para seu bom desempenho.

A identificação do XGBoost como o melhor modelo para a classificação de registros de SoNaR passivo tem implicações tanto para a compreensão teórica quanto para aplicações práticas no campo. Ela sugere a importância de aproveitar técnicas de aprendizagem por conjunto, com o reforço de gradiente, para lidar com relações intrincadas inerentes aos sinais de SoNaR. Além disso, a escalabilidade e a eficiência computacional do XGBoost tornam-no adequado para aplicações em tempo real ou quase em tempo real, aumentando a sua utilidade prática em vários domínios marítimos.

A regressão Ridge, que é uma técnica de regressão linear regularizada com regularização ℓ_2 também demonstrou desempenho apropriado na classificação do sinal de SoNaR. Embora ela possa não capturar interações complexas entre recursos de forma tão eficaz quanto o XGBoost, sua simplicidade e interpretabilidade a tornam uma alternativa valiosa, especialmente em cenários onde os recursos computacionais são limitados ou poder entender o que o modelo faz (interpretabilidade) é crucial.

A regressão Ridge demonstrou precisão de classificação competitiva, sugerindo uma robustez de sua abordagem de regularização na mitigação do *overfitting* e na melhoria da capacidade de generalização. Tal fato ressalta a importância de considerar um espectro de algoritmos de aprendizado de máquina, desde métodos de conjuntos complexos até modelos lineares mais simples, e suas respectivas compensações em complexidade computacional, interpretabilidade e desempenho preditivo. Além disso, o aprendizado obtido com a regressão de Ridge pode facilitar o desenvolvimento de abordagens híbridas ou de conjunto que combinem os pontos fortes de diferentes algoritmos para alcançar uma eficácia (e eficiência) de classificação ainda maior.

A regressão logística também demonstrou precisão de classificação competitiva, sugerindo a eficácia de sua estrutura probabilística na modelagem da relação entre características e classes/rótulos de destino. Isso sublinha o valor de modelos interpretáveis e transparentes, como a regressão logística, em cenários onde a interpretabilidade do modelo é fundamental; isto é, entender como o modelo explica a realização dos mapeamentos dos dados em suas saídas ou conformidade regulamentar.

Seguindo de perto os modelos acima, a Análise Discriminante Linear (LDA) demonstrou bom desempenho notável na modelagem da estrutura subjacente dos sinais de SoNaR e na separação de diferentes classes com alto poder discriminativo. O sucesso do LDA decorre da sua capacidade de capturar as propriedades estatísticas subjacentes dos sinais de SoNaR e explorar a informação discriminante linear para maximizar a separabilidade entre classes. Apesar da sua simplicidade, a dependência do LDA em transformações lineares torna-o particularmente eficaz quando a distribuição de dados subjacente é bem modelada por distribuições Gaussianas ou quando as classes exibem padrões linearmente separáveis..

O desempenho abaixo da média do KNN destaca os desafios associados a abordagens de classificação simplistas no contexto do SoNaR passivo. A dependência do KNN em medidas de proximidade local para decisões de classificação pode ser inadequada para capturar as relações e padrões diferenciados presentes nos dados do SoNaR, resultando em precisão de classificação inferior. As limitações do KNN sublinham a importância de considerar as características inerentes dos sinais de SoNaR e de selecionar algoritmos de aprendizagem de máquina apropriados, capazes de modelar eficazmente tais complexidades.

Observou-se que a separação usando Vetores De Suporte (SVM) é insuficiente no contexto da classificação de registros de embarcações por SoNaR passivo. Apesar dos seus fundamentos teóricos e do uso generalizado em várias aplicações de aprendizagem de máquina, o desempenho do SVM neste domínio revela-se inadequado em comparação com outros modelos e especialmente como XGBoost.

O desempenho abaixo da média do MLP sugere que a sua arquitetura ou abordagem de treinamento pode não ser adequada para capturar os padrões diferenciados presentes nos sinais de SoNaR passivos. Pesquisas futuras poderiam explorar arquiteturas alternativas de redes neurais e estratégias de treinamento para melhorar o desempenho de modelos baseados em neurônios artificiais para a classificação de registros de embarcações por SoNaR passivo.

As implicações das descobertas apresentadas vão além do âmbito desta dissertação, com implicações significativas tanto para a investigação como para aplicações práticas no campo da acústica subaquática e da segurança marítima. Ao demonstrar a eficácia de algoritmos de aprendizado de máquina, como XGBoost e regressão Ridge para classificação de registros de embarcações de SoNaR passivo, este estudo fornece informações para pesquisadores que busquem melhorar a precisão e a eficiência de sistemas de classificação e processamento de sinais subaquáticos.

No domínio da segurança marítima, a classificação precisa dos sinais sonares é fundamental para detectar e identificar ameaças potenciais, tais como submarinos ou veículos subaquáticos. A adoção de técnicas avançadas de aprendizagem automática, conforme demonstrado nesta dissertação, tem o potencial de melhorar as capacidades dos sistemas de SoNaR passivos implantados para fins de vigilância e defesa.

Como fator limitador destes resultados está a dependência de conjuntos de dados que pode não capturar totalmente a complexidade e a variabilidade dos sinais de SoNaR do mundo real. Pesquisas futuras poderiam se concentrar na coleta e análise de dados do mundo real em condições de mar diferentes para validar as descobertas deste estudo e avaliar as capacidades de generalização dos algoritmos propostos.

À medida que o campo da aprendizagem automática continua a evoluir, há amplas oportunidades para mais investigação e inovação na classificação de sonares passivos. Ao abordar as limitações e explorar novos caminhos de exploração, estudos futuros podem se basear nas bases estabelecidas nesta dissertação e continuar a expandir os limites do conhecimento em acústica subaquática e aquisição de novos sinais a serem inseridos no banco de dados.

Embora esta dissertação tenha fornecido informações valiosas sobre a classificação de SoNaR passivo usando regressão XGBoost e Ridge, vários caminhos para pesquisa e desenvolvimento futuros ainda precisam ser explorados. Em primeiro lugar, a integração de recursos ou modalidades de dados adicionais, como variáveis ambientais ou fusão multisensor, poderiam melhorar ainda mais o desempenho e a confiabilidade da classificação em diversos cenários operacionais.

Em segundo lugar, a investigação de outros algoritmos de aprendizagem automática, incluindo arquiteturas de aprendizagem profunda e métodos de conjunto além do XGBoost, pode revelar abordagens alternativas para melhorar a precisão e eficiência da classificação.

A aplicação de técnicas de transferência de aprendizagem ou estratégias de adaptação de domínio poderia facilitar a transferência de conhecimento entre diferentes conjuntos de dados de SoNaR ou ambientes de simulação, melhorando assim a generalização e a adaptabilidade do modelo. Além disso, os esforços para abordar as preocupações de interpretabilidade e explicabilidade associadas aos modelos de aprendizagem automática de caixa preta, como o XGBoost, são essenciais para promover a confiança e a aceitação entre os utilizadores finais e as partes interessadas nos domínios marítimos.

As descobertas apresentadas nesta dissertação de mestrado ressaltam a importância de aproveitar algoritmos de aprendizado de máquina para classificação de SoNaR passivo, com o XGBoost emergindo como o modelo de melhor desempenho e a regressão Ridge como a segunda melhor alternativa. Esses resultados oferecem informações sobre a otimização de algoritmos de classificação e processamento de sinais acústicos subaquáticos, contribuindo assim para avanços na segurança marítima e na eficácia operacional. No futuro, a investigação e o desenvolvimento contínuo neste domínio têm potencial para melhorar ainda mais as capacidades dos sistemas de SoNaR e mitigar ameaças emergentes em ambientes marítimos, garantindo a segurança das operações marítimas e dos ecossistemas.

Referências

- [1] [HTTPS://ESCOTISMOAZUL.ORG/](https://escotismoazul.org/). *Imagem - Amazônia Azul*. 2024. <https://escotismoazul.org/2022/o-que-e-amazonia-azul/> Acesso em: 05 de Janeiro de 2024.
- [2] ABRAHAM, D. A. *Underwater Acoustic Signal Processing*. [S.l.]: Springer Nature Switzerland AG 2019, 2019.
- [3] [HTTPS://ESCOTISMOAZUL.ORG/](https://escotismoazul.org/). *Imagem - Poder Naval*. 2024. <https://www.naval.com.br/> Acesso em: 05 de Janeiro de 2024.
- [4] ISLAND, U. of R.; CENTER, I. S. *Sound Travel in the SOFAR Channel*. 2024. <https://dosits.org/science/movement/sofar-channel/sound-travel-in-the-sofar-channel/> Acesso em: 05 de Janeiro de 2024.
- [5] PENG, R.; MATSUI, E. *The Art of Data Science: A Guide for Anyone who Works with Data*. Lulu.com, 2016. ISBN 9781365061462. Disponível em: <<https://books.google.com.br/books?id=ZDH9DAEACAAJ>>.
- [6] JUNIOR, N. M.; MELLO, V.; CERQUEIRA, L.; SEIXAS, J. Classificação de sinais de sonar passivo utilizando stacked autoencoders. In: . [S.l.: s.n.], 2017.
- [7] CHENG, C. *Principal Component Analysis (PCA) Explained Visually with Zero Math*. 2024. Disponível em: <https://towardsdatascience.com/principal-component-analysis-pca-explained-visually-with-zero-math-1cbf392b9e7d/> Acesso em: 14 de Fevereiro 2024.
- [8] AZANK, F. *Modelos de Predição | KNN*. 2024. Disponível em: <https://medium.com/turing-talks/turing-talks-13-modelo-de-predi%C3%A7%C3%A3o-knn-3be880c9b9d1> Acesso em: 13 de Fevereiro 2024.
- [9] BRASIL, M. do. *Plano Estratégico da Marinha (PEM 2040)*. [S.l.]: Estado-Maior da Armada, Brasília-DF: 2020., 2020.
- [10] CARMONA, R. G. *PODER NACIONAL E GRANDE ESTRATÉGIA: UMA ANÁLISE GEOPOLÍTICA DOS CONCEITOS FUNDAMENTAIS DO PROJETO BRASILEIRO DE POTÊNCIA*. Tese (Doutorado) — Universidade de São Paulo, 2017.
- [11] MAHAN, A. T. *THE INFLUENCE OP SEA POWER UPON HISTORY 1660-1783*. [S.l.]: LITTLE, BROWN, AND COMPANY, 1890.
- [12] SANTOS, P. de. *Porto de Santos fecha 2022 com recorde histórico na movimentação de cargas*. 2023. Disponível em: <https://www.portodesantos.com.br/2023/01/11/portodesantosfecha-2022-com-recorde-historico-na-movimentacao-de-cargas/> Acesso em: 14 de Agosto 2023.

- [13] SATHEESH, C.; KAMAL, S.; MUJEEB, A.; SUPRIYA, M. H. Passive sonar target classification using deep generative beta -vae. *IEEE signal processing letters*, IEEE, v. 28, p. 808–812, 2021. ISSN 1070-9908.
- [14] FILHO, J. B. S.; SEIXAS, J. M. de. Class-modular multi-layer perceptron networks for supporting passive sonar signal classification. *IET radar, sonar and navigation*, The Institution of Engineering and Technology, v. 10, n. 2, p. 311–317, 2016. ISSN 1751-8784.
- [15] DAS, A.; KUMAR, A.; BAHL, R. Marine vessel classification based on passive sonar data: the cepstrum-based approach. *IET radar, sonar and navigation*, The Institution of Engineering and Technology, HERTFORD, v. 7, n. 1, p. 87–93, 2013. ISSN 1751-8784.
- [16] FILHO, W. S. *Classificação do Ruído Irradiado por Navios Usando Redes Neurais*. Tese (Doutorado) — COPPE/UFRJ, Rio de Janeiro, Brasil, 2001.
- [17] OLIVEIRA, R. L. d. *O Emprego da Análise Multi-modos na Classificação de Contatos Sonar Passiv*. Tese (Doutorado) — UFRJ/COPPE, 2010.
- [18] MEDEIROS, L. J. de. *Hidrofone Ultrassônico com Piezoelreito como Elemento Transdutor*. Tese (Doutorado) — Universidade de São Paulo, 2014.
- [19] CAMARGO, F. S. *Estudo das vocalizações de golfinhos-rotadores, Stenella longirostris (Cetacea, Delphinidae), no arquipélago de Fernando de Noronha*. Tese (Doutorado) — Universidade de São Paulo, 2008.
- [20] FERREIRA, J. Queremos vingança.o afundamento de navios brasileiros pelo submarino alemão u-507 e as reações populares no rio de janeiro. *Crítica Histórica*, v. 08, n. 8, p. 208–231, 2017. ISSN 21779961.
- [21] AINSLIE, M. E. *Principles of sonar performance modelling*. 2010. ed. Berlin, Germany: Springer, 2010. (Geophysical Sciences).
- [22] URICK, R. J. *Principles of Underwater Sound for Engineers*. [S.l.]: McGraw-Hill, 1967.
- [23] LURTON, X. The development of underwater acoustics. In: *An Introduction to Underwater Acoustics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 1–11.
- [24] HODGES, R. P. *Underwater acoustics : analysis, design, and performance of sonar*. [S.l.]: John Wiley and Sons, 2010.
- [25] WANG P.; PENG, Y. Research on feature extraction and recognition method of underwater acoustic target based on deep convolutional network. *Proceedings of the 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA)*, p. 863–868, 2020.
- [26] LATHI, B. P. B. P. *Sinais e Sistemas Lineares*. 2.ed. ed. Porto Alegre: Bookman, 2007. 856p. p. ISBN 85-60031-13-8.
- [27] LORD, R. *The theory of sound: V. 2*. 2. ed. Mineola, NY: Dover Publications, 1945. (Dover Books on Physics).

- [28] LIEBERMANN, L. N. The origin of sound absorption in water and in sea water. *J. Acoust. Soc. Am.*, Acoustical Society of America (ASA), v. 20, n. 6, p. 868–873, nov 1948.
- [29] THORP, W. H. Analytic description of the low-frequency attenuation coefficient. *J. Acoust. Soc. Am.*, Acoustical Society of America (ASA), v. 42, n. 1, p. 270–270, jul 1967.
- [30] FRANCOIS, R. E.; GARRISON, G. R. Sound absorption based on ocean measurements: Part i: Pure water and magnesium sulfate contributions. *J. Acoust. Soc. Am.*, Acoustical Society of America (ASA), v. 72, n. 3, p. 896–907, sep 1982.
- [31] H., M. R.; SCHEIFELE, P. M.; BROWNING, D. G. *Global model for sound absorption in sea water*. [S.l.], 1897.
- [32] MOORE, D. F. *PASSIVE SONAR TARGET RECOGNITION USING A BACK-PROPAGATING NEURAL NETWORK*. Tese (Doutorado) — NAVAL POSTGRADUATE SCHOOL, 1991.
- [33] HOBBS, T.; PLAMENATZ, J. *Leviathan*. London, England: Fontana Press, 1962.
- [34] VINHEDO, C. M. de. *Moção N°1, 2003*. 2003. <https://vinhedo.siscam.com.br/arquivo?Id=68016/> Acesso em: 28 de dezembro 2023.
- [35] GORDON, D. E. *Electronic warfare*. London, England: Pergamon Press, 1981.
- [36] 7 Latest Marine Sonar Technology Applications (2023). <https://www.marinetechmiami.com/latest-marine-sonar-technology-applications>. Accessed: 2010-09-3009/12/2023.
- [37] ZHANG, X.; SUN, H. Recent advances in underwater signal processing. *Sensors*, MDPI AG, v. 23, n. 13, p. 5777, jun 2023. ISSN 1424-8220. Disponível em: <<http://dx.doi.org/10.3390/s23135777>>.
- [38] FOLKESSON, Y. X. . N. B. . J. Bathymetric reconstruction from sidescan sonar with deep neural networks. *IEEE JOURNAL OF OCEANIC ENGINEERING*, v. 48, n. 2, p. 372–383, 2023.
- [39] BIAN1, Y. T. L. W. H. L. e S. Side-scan sonar underwater target segmentation using the bhp-unet. *EURASIP Journal on Advances in Signal Processing*, n. 76, p. 001–023, 2023.
- [40] WANG, J.; LI, H.; HUO, G.; LI, C.; WEI, Y. Multi-mode channel position attention fusion side-scan sonar transfer recognition. *Electronics (Basel)*, MDPI AG, Basel, v. 12, n. 4, p. 791, 2023. ISSN 2079-9292.
- [41] HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. Disponível em: <<http://arxiv.org/abs/1512.03385>>.
- [42] SUN3, X. Z. P. Y. e H. An omega-k algorithm for multireceiver synthetic aperture sonar. *Electronics Letters*, v. 59, n. 13, p. 001–003, 2023.

- [43] ZARE, D. S. K. F. J. ;e A. Image-to-height domain translation for synthetic aperture sonar. *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, v. 61, p. 001–013, 2023.
- [44] MONGA, I. D. G. Deep multi-look sequence processing for synthetic aperture sonar image segmentation. *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, v. 61, n. 2, p. 001–015, 2023.
- [45] SABO, T. O.; HANSEN, R. E. Adaptive phase estimation filter in long range synthetic aperture sonar interferometry. *Electronics letters*, Wiley, v. 59, n. 10, p. n/a, 2023. ISSN 0013-5194.
- [46] HANSEN, R. E. MIMO synthetic aperture sonar: Benefits and limitations. *Electronics Letters*, v. 59, n. 2, 2023.
- [47] WANG, X. L. Q. Z. Y. Y. C. S. J. Z. Y. High-angular- and range-resolution imaging using MIMO sonar with two-dimensional deconvolution. *IET Radar, Sonar and Navigation*, n. 17, p. 991–1001, 2023.
- [48] SUN, K. F. L. C. Multiple-input multiple-output sonar adaptive beamforming using transmission diversity smoothing and backward processing. *IET Radar, Sonar and Navigation*, n. 17, p. 478–489, 2023.
- [49] LIU, X.; SHI, R.; SUN, C.; YANG, Y.; ZHUO, J. Using deconvolution to suppress range sidelobes for MIMO sonar imaging. *Applied acoustics*, Elsevier Ltd, v. 186, p. 108491, 2022. ISSN 0003-682X.
- [50] RICHARDSON, W. H. Bayesian-based iterative method of image restoration*. *Journal of the Optical Society of America*, Optica Publishing Group, v. 62, n. 1, p. 55, jan 1972. ISSN 0030-3941. Disponível em: <<http://dx.doi.org/10.1364/JOSA.62.000055>>.
- [51] LUCY, L. B. An iterative technique for the rectification of observed distributions. *The Astronomical Journal*, American Astronomical Society, v. 79, p. 745, jun 1974. ISSN 0004-6256. Disponível em: <<http://dx.doi.org/10.1086/111605>>.
- [52] FERNANDES, J. d. C. V.; JUNIOR, N. N. de M.; SEIXAS, J. M. de. Deep learning models for passive sonar signal classification of military data. *Remote sensing (Basel, Switzerland)*, MDPI AG, Basel, v. 14, n. 11, p. 2648, 2022. ISSN 2072-4292.
- [53] TUFAN, B. Underwater communication systems. *Koc University*, 2020.
- [54] QIAO, G.; BABAR, Z.; MA, L.; LIU, S.; WU, J. MIMO-OFDM underwater acoustic communication systems—a review. *Physical communication*, Elsevier B.V, v. 23, p. 56–64, 2017. ISSN 1874-4907.
- [55] SANTOS, V. P. A.; SILVA, A. A.; VALE, E. R.; MATOS, L. J.; FERREIRA, T. N. Simulação de técnicas de diversidade espacial e temporal em canais acústicos submarinos. *Encontro de Tecnologia em Acústica Submarina (ETAS)*, 2022.
- [56] CHEN, M.; XU, W.; WANG, D.; WANG, L. Multi-carrier chaotic communication scheme for underwater acoustic communications. *IET communications*, The Institution of Engineering and Technology, v. 13, n. 14, p. 2097–2105, 2019. ISSN 1751-8628.

- [57] KRINSMAN, W. *Tensor Algebra and its Applications to Data Science and Statistics*. arXiv, 2022. Disponível em: <<https://arxiv.org/abs/2210.16182>>.
- [58] JHAWAT, V.; GUPTA, S.; GULIA, M.; NAIR, A. Chapter 5 - artificial intelligence and data science in pharmacogenomics-based drug discovery: Future of medicines. In: TYAGI, A. K.; ABRAHAM, A. (Ed.). *Data Science for Genomics*. Academic Press, 2023. p. 85–97. ISBN 978-0-323-98352-5. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9780323983525000057>>.
- [59] D'AMICO, A.; PITTENGER, R. A brief history of active sonar. *Aquatic Mammals*, Aquatic Mammals Journal, v. 35, n. 4, p. 426–434, dec 2009. ISSN 0167-5427. Disponível em: <<http://dx.doi.org/10.1578/AM.35.4.2009.426>>.
- [60] STERGIOPOULOS, S. Noise normalization technique for beamformed towed array data. *J. Acoust. Soc. Am.*, Acoustical Society of America (ASA), v. 97, n. 4, p. 2334–2345, apr 1995.
- [61] STRUZINSKI, W. A.; LOWE, E. D. A performance comparison of four noise background normalization schemes proposed for signal detection systems. *J. Acoust. Soc. Am.*, Acoustical Society of America (ASA), v. 76, n. 6, p. 1738–1742, dec 1984.
- [62] MCKINNEY, W. Data structures for statistical computing in python. In: AUSTIN, TX. *Proceedings of the 9th Python in Science Conference*. [S.l.], 2010. v. 445, p. 51–56.
- [63] HAYKIN, S. *Neural networks and learning machines / Simon Haykin.—3rd ed.* [S.l.]: Pearson Prentice Hall, 2009.
- [64] M., C. *Pattern Recognition and Machine Learning*. 1. ed. New York, NY: Springer, 2006. (Information Science and Statistics).
- [65] PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.
- [66] BRADLEY, A. H. T. S. *Applied Mathematical Programming*. [S.l.]: Addison-Wesley, Reading, Massachusetts, 1977.
- [67] AMINI, A.; DOLATSHAHI, M.; KERACHIAN, R. Effects of automatic hyperparameter tuning on the performance of multi-variate deep learning-based rainfall nowcasting. *Water resources research*, Blackwell Publishing Ltd, Washington, v. 59, n. 1, p. n/a, 2023. ISSN 0043-1397.
- [68] GARCIA-SANTIAGO, X.; BURGER, S.; ROCKSTUHL, C.; SCHNEIDER, P.-I. Bayesian optimization with improved scalability and derivative information for efficient design of nanophotonic structures. *Journal of lightwave technology*, IEEE, New York, v. 39, n. 1, p. 167–177, 2021. ISSN 0733-8724.
- [69] LEON-GARCIA, A. *Probability, statistics, and random processes for electrical engineering*. 3. ed. Upper Saddle River, NJ: Pearson, 2007.

- [70] MURPHY, K. P. *Machine Learning*. London, England: MIT Press, 2012. (Adaptive Computation and Machine Learning series).
- [71] FILHO, J. B. de Oliveira e S. *CLASSIFICAÇÃO NEURAL DE SINAIS DE SONAR PASSIVO*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, COPPE, 2007.
- [72] ADAMI, G.; ORSOLINI, G.; FASSIO, A.; VIAPIANA, O.; SORIO, E.; BENINI, C.; GATTI, D.; BERTELLE, D.; ROSSINI, M. Pos0474 factors associated with erosive rheumatoid arthritis, a multimarker principal component analysis (pca) and principal component regression (pcr) analysis. *Annals of the rheumatic diseases*, BMJ Publishing Group LTD, London, v. 82, n. Suppl 1, p. 497–498, 2023. ISSN 0003-4967.
- [73] GREENACRE, M.; GROENEN, P. J. F.; HASTIE, T.; D'ENZA, A. I.; MARKOS, A.; TUZHILINA, E. Principal component analysis. *Nature Reviews Methods Primers*, 2023.
- [74] KASUYA, E. On the use of r and r squared in correlation and regression. *Ecological research*, John Wiley and Sons, Inc, Hoboken, USA, v. 34, n. 1, p. 235–236, 2019. ISSN 0912-3814.
- [75] SEO, B. Hybrid combiner design for downlink massive mimo systems. *ETRI journal*, Electronics and Telecommunications Research Institute (ETRI), v. 42, n. 3, p. 333–340, 2020. ISSN 1225-6463.
- [76] SAAVEDRA, A.; CAO, R. On the estimation of the marginal density of a moving average process. *Canadian journal of statistics*, Wiley-Blackwell, Hoboken, v. 28, n. 4, p. 799–815, 2000. ISSN 0319-5724.
- [77] NEVITT, J.; HANCOCK, G. Improving the root mean square error of approximation for nonnormal conditions in structural equation modeling. *Journal of Experimental Education - J EXP EDUC*, v. 68, p. 251–268, 03 2000.
- [78] HANCOCK, G. R.; FREEMAN, M. J. Power and sample size for the root mean square error of approximation test of not close fit in structural equation modeling. *Educational and Psychological Measurement*, v. 61, n. 5, p. 741–758, 2001. Disponível em: <<https://doi.org/10.1177/00131640121971491>>.
- [79] APPLGATE, R. A.; BALLENTINE, C.; GROSS, H.; SARVER, E. J.; SARVER, C. A. Visual acuity as a function of zernike mode and level of root mean square error. *Optometry and Vision Science*, Ovid Technologies (Wolters Kluwer Health), v. 80, n. 2, p. 97–105, feb 2003. ISSN 1040-5488. Disponível em: <<http://dx.doi.org/10.1097/00006324-200302000-00005>>.
- [80] EINSTEIN, A.; PODOLSKY, B.; ROSEN, N. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, American Physical Society, v. 47, p. 777–780, May 1935. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRev.47.777>>.
- [81] CHAI, T.; DRAXLER, R. R. Root mean square error (rmse) or mean absolute error (mae)? – arguments against avoiding rmse in the literature. *Geoscientific Model Development*, v. 7, n. 3, p. 1247–1250, 2014. Disponível em: <<https://gmd.copernicus.org/articles/7/1247/2014/>>.

- [82] MYTTENAERE, A. de; GOLDEN, B.; GRAND, B. L.; ROSSI, F. Mean absolute percentage error for regression models. *Neurocomputing (Amsterdam)*, Elsevier B.V, v. 192, p. 38–48, 2016. ISSN 0925-2312.
- [83] KIM, S.; KIM, H. A new metric of absolute percentage error for intermittent demand forecasts. *International journal of forecasting*, Elsevier B.V, AMSTERDAM, v. 32, n. 3, p. 669–679, 2016. ISSN 0169-2070.
- [84] CHICCO, D.; WARRENS, M. J.; JURMAN, G. The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *PeerJ. Computer science*, PeerJ. Ltd, San Diego, v. 7, p. e623–e623, 2021. ISSN 2376-5992.
- [85] FIGUEIREDO, D.; JÚNIOR, S.; ROCHA, E. What is r2 all about? *Leviathan-Cadernos de Pesquisa Política*, v. 3, p. 60–68, 11 2011.
- [86] ROSENBLATT, F. *The perceptron - A perceiving and recognizing automaton*. Ithaca, New York, January 1957.
- [87] CYBENKO, G. Approximation by superpositions of sigmoid function. *Mathematics of Control, Signals and System.*, 1986.
- [88] SCHMIDT, M. Least squares optimization with l1-norm regularization. *CS542B Project Report*, 2005.
- [89] FIX J.L. HODGES, J. E. Discriminatory analysis nonparametric discrimination: Consistency properties. *Wright Patterns in Air Force Base-Dayton Ohio*, 1951.
- [90] ZHANG, S. Challenges in knn classification. *IEEE transactions on knowledge and data engineering*, IEEE, v. 34, n. 10, p. 1–1, 2022. ISSN 1041-4347.
- [91] LAUFER, B.; DOCHERTY, P. D.; MURRAY, R.; KRUEGER-ZIOLEK, S.; JALAL, N. A.; HOEFLINGER, F.; RUPITSCH, S. J.; REINDL, L.; MOELLER, K. Sensor selection for tidal volume determination via linear regression—impact of lasso versus ridge regression. *Sensors (Basel, Switzerland)*, MDPI AG, Basel, v. 23, n. 17, p. 7407, 2023. ISSN 1424-8220.
- [92] VIRGILIO, A. D. D.; ALTUCCI, C.; BAJARDI, F.; BASTI, A.; BEVERINI, N.; CAPOZZIELLO, S.; CARELLI, G.; CIAMPINI, D.; FUSO, F.; GIACOMELLI, U.; MACCIONI, E.; MARSILI, P.; ORTOLAN, A.; PORZIO, A.; SIMONELLI, A.; TERRENI, G.; VELOTTA, R. Sensitivity limit investigation of a sagnac gyroscope through linear regression analysis: Sagnac gyroscope sensitivity limit. *The European physical journal. C, Particles and fields*, Springer Berlin Heidelberg, Berlin-Heidelberg, v. 81, n. 5, 2021. ISSN 1434-6044.