UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E ELETRÔNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Unírio Machado dos Santos Júnior

Implementação e análise de classificadores automáticos de modulação usando técnicas de aprendizado de máquina

Florianópolis
2022

Unírio Machado dos Santos Júnior

Implementação e análise de classificadores automáticos de modulação usando técnicas de aprendizado de máquina

Trabalho Conclusão do Curso de Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Bacharel em Engenharia Elétrica.
Orientador: Prof. Fernando Rangel de Sousa, Dr.

Florianópolis
2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Santos Júnior, Unírio Machado dos
Implementação e análise de classificadores automáticos de modulação usando técnicas de aprendizado de máquina / Unírio Machado dos Santos Júnior ; orientador, Fernando Rangel de Sousa, 2022.
84 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia Elétrica, Florianópolis, 2022.

Inclui referências.

1. Engenharia Elétrica. 2. Classificação Automática de Modulação. 3. Aprendizado de Máquina. 4. Modulação Digital. I. Rangel de Sousa, Fernando. II. Universidade Federal de Santa Catarina. Graduação em Engenharia Elétrica. III. Título.

Unirio Machado dos Santos Junior

Implementação e análise de classificadores automáticos de modulação usando técnicas de aprendizado de máquina

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de "Bacharel em Engenharia Elétrica" e aceito, em sua forma final, pelo Curso de Graduação em Engenharia Elétrica.

Florianópolis, 01 de agosto de 2022.

Documento assinado digitalmente
Miguel Moreto
Data: 01/08/2022 10:32:25-0300
CPF: 948.850.100-63
Verifique as assinaturas em https://v.ufsc.br

Prof. Miguel Moreto, Dr.
Coordenador do Curso de Graduação em Engenharia Elétrica

Banca Examinadora:

Documento assinado digitalmente
Fernando Rangel de Sousa
Data: 01/08/2022 10:20:15-0300
CPF: 884.649.114-91
Verifique as assinaturas em https://v.ufsc.br

Prof. Fernando Rangel de Sousa, Dr.
Orientador
Universidade Federal de Santa Catarina

Documento assinado digitalmente
Fabian Leonardo Cabrera Riano
Data: 02/08/2022 10:04:07-0300
CPF: 233.071.638-92
Verifique as assinaturas em https://v.ufsc.br

Prof. Fabian Leonardo Cabrera Riaño, Dr.
Avaliador
Universidade Federal de Santa Catarina

Documento assinado digitalmente
Carlos Raul Morales Hernandez
Data: 01/08/2022 10:37:17-0300
CPF: 801.130.229-99
Verifique as assinaturas em https://v.ufsc.br

Carlos Raúl Morales Hernández, Me.
Avaliador
Universidade Federal de Santa Catarina

Este trabalho é dedicado aos meus amigos e aos meus queridos pais.

AGRADECIMENTOS

Primeiramente, gostaria de agradecer à minha família, principalmente aos meus pais, Unírio Machado dos Santos e Irlene da Silveira Pinto Santos, que tiveram a compreensão do quão difícil foi este trabalho para mim, e nunca mediram esforços para me apoiar ou me fornecer todas as condições necessárias para que eu pudesse concluí-lo. Sem eles eu certamente não teria chegado onde estou. Ao meu irmão, Henrique Pinto Santos que sempre se fez presente, e muito mais do que um irmão, é um verdadeiro amigo o qual sei que sempre terei ao meu lado. E à minha namorada, Júlia Andrade, que me apoiou em absolutamente todos os momentos e me deu forças quando eu achava que já não tinha.

Aos meus amigos que estiveram presentes durante a execução deste trabalho: Alexandre Gomes, Anderson Spessatto, Gabriel Amante, Guilherme Fin, Heitor Medeiros, Jaqueline Clamer, José Brognoli, Juan Yegros, Juliana Fernandes, Lucas Deschamps, Luis Tadeu Nascimento e Nicolas Quadro, que foram companheiros tanto em bons quanto em maus momentos. Sempre mostrando seu apoio e torcida por mim.

E em especial aos meus amigos, Lauro de Lima, Daniel de Senna e Ian Vieira Silveira, que tiveram um papel fundamental na contribuição da construção deste trabalho e mostraram que posso contar com eles, independentemente da situação, do fundo do coração. Ter amigos dispostos a ouvir o problema do outro e fazer o possível para ajudar é uma verdadeira benção.

À Universidade Federal de Santa Catarina, em especial a todos professores e funcionários que fizeram parte desta graduação, sempre mostrando um trabalho de excelência.

E ao professor que orientou este trabalho, Fernando Rangel de Sousa, por ter sido paciente e nunca ter desistido de mim, por todo suporte e empatia apresentados diante de tantas condições adversas, pela disponibilidade quanto a encontros e reuniões, e por ter sido fundamental para fazer este projeto acontecer.

“Aprendizado automático nos permite construir soluções de *software* que superam o entendimento humano e mostram que a inteligência artificial pode inervar qualquer indústria”

(Steve Jurvetson)

RESUMO

O atual projeto tem como objetivo a implementação de classificadores automáticos de modulação, analisando-se modulações digitais identificadas por alterações nos seus parâmetros fundamentais como amplitude, fase e frequência. A proposta foi incentivada principalmente pela alta demanda do espectro de frequências que exige novas técnicas para evitar seu problema de escassez. A área de aprendizado de máquina tem se destacado muito no ambiente científico, uma vez que consegue extrair de um conjunto de dados informações valiosas e importantes muitas vezes não avaliadas por um ser humano. Métodos de aprendizado de máquina tem se mostrado bastante úteis para avaliar conjuntos de dados imensos. Para o desenvolvimento do projeto, foi necessário construir uma base de dados contendo sinais modulados em amplitude, fase e frequência. O método considerado para se aplicar o aprendizado de máquina foi o baseado em características consideradas importantes aos sinais, que pudessem discriminá-los em relação aos parâmetros determinados. O aprendizado foi do tipo supervisionado, seguindo o objetivo do projeto de se classificar dados rotulados. Modelos como KNN, SVM e Floresta Aleatória foram implementados e aplicados sob a base de dados definida. O desempenho de cada modelo foi dado através de métricas de avaliação, as quais têm por característica avaliar as classificações corretas e incorretas de cada um dos modelos.

Palavras-chave: Aprendizado de Máquina. Classificação Automática de Modulação. Modulação Digital.

ABSTRACT

The following project aims to implement automatic modulation classifiers, analyzing digital modulations identified by alterations in their fundamental parameters, such as amplitude, phase and frequency. The proposal was mainly encouraged by the high demand of the frequency spectrum that requires new techniques to avoid its scarcity problem. Machine Learning problems have stood out in the scientific environment, once they are able to extract valuable and important information from a dataset that, by many times, is not well evaluated by humans. Machine Learning methods have proven to be quite useful for evaluating huge data. For the development of this project, it was necessary to build a data set containing modulated signals in amplitude, phase and frequency. The considered method to apply the machine learning task was the one based on key features, which were evaluated as important to the signals, once they are able to discriminate the determined parameters. The learning was supervised, once the aim of the project is to classify labeled data. Models such as KNN, SVM and Random Forest were implemented and applied under the defined database. The performance of each model was given through a few evaluation metrics, which were responsible for evaluating correct and incorrect data of each of the models.

Keywords: Machine Learning. Automatic Modulation Classification. Digital Modulation.

LISTA DE FIGURAS

Figura 1 - Sinal modulado em OOK 26
Figura 2 - Sinal modulado em 2-FSK 28
Figura 3 - Sinal modulado em 2-PSK 31
Figura 4 - Sinal modulado em QPSK ou 4-PSK 32
Figura 5 - Exemplo de modelo de sistema de comunicação com canal de comunicação e ruído 34
Figura 6 - Exemplo de predição de nova instância a partir do conjunto de treinamento 36
Figura 7 - Exemplo de treinamento supervisionado com algoritmo de regressão 37
Figura 8 - Caso de detecção de anomalias através do treinamento não supervisionado 38
Figura 9 - Caso de predição de um dado desconhecido baseado nos 5 vizinhos mais próximos 39
Figura 10 - Modelo comparativo de hiperplanos para SVM 40
Figura 11 - Representação do limiar de decisão de uma árvore com profundidade = 1 42
Figura 12 - Representação do limiar de decisão de uma árvore com profundidade = 2 42
Figura 13 - Floresta aleatória combinada por cinco árvores de decisão 44
Figura 14 - Divisão dos dados em subconjunto de treinamento, validação e teste 45
Figura 15 - Representação do método de k-fold Cross Validation 46
Figura 16 - Acurácia de modelos com underfitting e overfitting 47
Figura 17 - Modelos com Underfitting, ideal e Overfitting, respectivamente 47
Figura 18 - Etapas de uma classificação automática de modulação 50
Figura 19 - Esquema de estruturação da base de dados 60
Figura 20 - Exemplo de sinal ruidoso gerado a partir de um canal AWGN 61
Figura 21 - Sinal modulado em frequência usando o gerador de sinais projetado 62
Figura 22 - Sinal modulado em amplitude usando o gerador de sinais projetado 63
Figura 23 - Sinal modulado em fase usando o gerador de sinais projetado 64
Figura 24 - Representação do funcionamento do bloco extrator de características 65
Figura 25 - Distribuição dos sinais modulados em relação às features σaa e σaf 68
Figura 26 - Distribuição dos sinais modulados em relação às features σap e γmax 69
Figura 27 - Distribuição dos sinais modulados em relação às features σa e σdp 69
Figura 28 - Representação da acurácia do modelo em relação ao número de vizinhos mais próximos 71

LISTA DE QUADROS

Quadro 1 - Métodos utilizados na literatura de AMC 21
Quadro 2 - Contribuição do atual trabalho para a literatura de AMC 77

LISTA DE TABELAS

Tabela 1 - Parâmetros de geração dos sinais avaliados dentro do bloco de configurações 62
Tabela 2 - Parâmetros utilizados para construir o sinal da figura 19 63
Tabela 3 - Parâmetros utilizados para construir o sinal da figura 20 63
Tabela 4 - Parâmetros utilizados para construir o sinal da figura 21 64
Tabela 5 - Parâmetros e features correspondentes a um sinal de frequência 66
Tabela 6 - Parâmetros e features correspondentes a um sinal de amplitude 67
Tabela 7 - Parâmetros e features correspondentes a um sinal de fase 67
Tabela 8 - Matriz de confusão 72
Tabela 9 - Métricas de avaliação 72
Tabela 10 - Hiperparâmetros padrões 73
Tabela 11 - Matriz de confusão 74
Tabela 12 - Métricas de avaliação 74
Tabela 13 - Hiperparâmetros ideais adquiridos após validação cruzada 75
Tabela 14 - Matriz de confusão após classificação sob hiperparâmetros ótimos 75
Tabela 15 - Métricas de avaliação após classificação sob hiperparâmetros ótimos 75
Tabela 16 - Hiperparâmetros padrões usados para estimar o classificador 77
Tabela 17 - Matriz de confusão após classificação inicial 77
Tabela 18 - Métricas de avaliação após classificação inicial 77
Tabela 19 - Hiperparâmetros ótimos 78
Tabela 20 - Matriz de confusão após classificação sob hiperparâmetros ótimos 78
Tabela 21 - Métricas de avaliação após classificação sob hiperparâmetros ótimos 78
Tabela 22 - Acurácia dos classificadores KNN, SVM e Floresta Aleatória 84

LISTA DE ABREVIATURAS E SIGLAS

ALRT *Average Likelihood Ratio Test*

AMC *Automatic Modulation Classification*

ANN *Artificial Neural Network*

ASK *Amplitude-Shift Keying*

AWGN *Additive White Gaussian Noise*

BPSK *Binary Phase-Shift Keying*

CPFSK *Continuous Phase Frequency-Shift Keying*

CNN *Convolutional Neural Network*

DNN *Deep Neural Network*

FB *Feature-Based*

FN *False Negatives*

FSK *Frequency-Shift Keying*

FP *False Positives*

GLRT *Generalized Likelihood Ratio Test*

HLRT *Hybrid Likelihood Ratio Test*

KNN *K-Nearest Neighbor*

LB *Likelihood-Based*

OOK *On Off Keying*

PDF *Probability Density Function*

PSK *Phase-Shift Keying*

QPSK *Quadrature Phase-Shift Keying*

SVM *Support Vector Machine*

TN *True Negatives*

TP *True Positives*

WNN *Wavelet-based Neural Network*

SUMÁRIO

1 INTRODUÇÃO **17**
1.1 MOTIVAÇÃO E RELEVÂNCIA 17
1.2 ESTADO DA ARTE 19
1.3 OBJETIVOS 22
1.3.1 Objetivos gerais **22**
1.3.2 Objetivos específicos **22**
2 FUNDAMENTAÇÃO TEÓRICA **23**
2.1 SINAIS MODULADOS 23
2.1.1 Modulação em amplitude por chaveamento (ASK) **24**
2.1.2 Modulação em frequência por chaveamento (FSK) **26**
2.1.3 Modulação em fase por chaveamento (PSK) **28**
2.2 CANAL DE COMUNICAÇÃO 32
2.3 APRENDIZADO DE MÁQUINA 33
2.3.1 Tipos de aprendizado de máquina **34**
2.3.2 K-Nearest Neighbor **37**
2.3.3 Máquinas de vetores de suporte **39**
2.3.4 Árvore de decisão **40**
2.3.5 Floresta aleatória **41**
2.3.6 Validação cruzada **43**
2.3.7 Métricas de avaliação **47**
3 MATERIAIS E MÉTODOS **49**
3.1 BASE DE DADOS 49
3.1.1 Obtenção da base de dados **49**
3.2 EXTRAÇÃO DE CARACTERÍSTICAS 51
3.2.1 Características baseadas no domínio do tempo **51**
3.2.2 Características baseadas no domínio da transformada **53**
3.2.3 Discriminação das features **54**
3.3 CLASSIFICADORES 55
3.3.1 Ajuste dos Hiperparâmetros dos modelos **56**
4 DESENVOLVIMENTO E RESULTADOS **58**
4.1 ARQUITETURA DOS EXPERIMENTOS 58
4.2 GERAÇÃO DA BASE DE DADOS 58
4.3 IMPLEMENTAÇÃO DO BLOCO EXTRATOR DE CARACTERÍSTICAS 64
4.4 IMPLEMENTAÇÃO DO BLOCO DE CLASSIFICAÇÃO 69
4.4.1 K-Nearest Neighbor (KNN) **69**
4.4.2. Support Vector Machine (SVM) **71**
4.4.3 Floresta Aleatória **75**
5 CONSIDERAÇÕES FINAIS **78**

5.1 CONTEXTUALIZAÇÃO DO PROJETO PROPOSTO E LITERATURA 78
5.2 CONCLUSÃO 80

6 REFERÊNCIAS 84

1 INTRODUÇÃO

1.1 MOTIVAÇÃO E RELEVÂNCIA

Um sistema de comunicação é caracterizado quando há uma mensagem enviada por um transmissor, através de um canal, e entendida por um ou mais receptores. Em se tratando de uma comunicação de longa distância, é necessário que haja o processo da modulação, uma vez que os sinais, como são fornecidos, não podem ser enviados através do canal. A informação enviada será modificada pelo transmissor através da adição de uma onda portadora, cujas propriedades facilitem o processo de comunicação. O transmissor pode escolher a forma de modulação da mensagem a partir de seu interesse, mas para que a mensagem seja recuperada com sucesso pelo receptor, é necessário que este tenha conhecimento prévio do formato utilizado. Para isso, o transmissor adiciona tal informação à mensagem original, possibilitando que o receptor reaja de acordo.

É evidente que não é desejável que o conteúdo presente nesta comunicação seja interceptado por um terceiro. Entretanto, com a guerra eletrônica, surgiu a necessidade da monitoração de comunicações externas. Para tal, a identificação da modulação utilizada seria o primeiro passo para rastrear tropas inimigas ou recuperar o sinal interceptado. Inicialmente, os engenheiros faziam a análise através de observação do sinal e de equipamentos de processamento, o que era considerado uma classificação manual. A partir da década de 80, utilizou-se de tecnologias que estavam surgindo como o reconhecimento de padrões para se automatizar o processo (ZHU; NHANDI, 2015), surgindo assim a classificação automática de modulação.

Portanto, vê-se no cenário militar suas primeiras aplicações. O conhecimento do formato de modulação que tropas inimigas estão usando para se comunicar é útil para atrapalhar ou até mesmo impedir tal comunicação. Uma técnica bastante utilizada é transmitir sinais ruidosos que interfiram no sinal de origem adversário, ou então, sinais falsos que tenham o mesmo formato. Para que essa técnica seja realizada com êxito, o sinal deve ser transmitido com alta potência, o que é sucedido com amplificadores, e na mesma faixa de frequência. Dessa forma, é possível sobrepor os sinais de comunicação inimigos e dificultar a troca de informações. Da mesma maneira, sabendo que os inimigos possuem as mesmas técnicas para impedir a comunicação, deve-se proteger a própria troca de informações. Para tal, é necessário monitorar e identificar os sinais ruidosos que têm como objetivo bloquear a comunicação, e então, mudar o formato da própria modulação para evitar o bloqueio. Por

último, a classificação automática de modulação torna-se muito útil para fazer um mapeamento da posição adversária, e até mesmo, com o auxílio de técnicas mais avançadas, traduzir e decodificar as mensagens envolvidas na comunicação.

Com o avanço tecnológico das comunicações, tem-se uma demanda maior para utilização do espectro de frequências. As políticas atuais de alocação do espectro determinam que uma faixa de frequência será utilizada pelo usuário primário para determinado serviço. Essa faixa fica reservada e não pode ser usada por outros usuários mesmo que não esteja em uso (ALI; HAMOUDA, 2016). Entretanto, o espectro de frequências é um recurso natural e limitado, e com o aumento de usuários, há evidências de que há um congestionamento do mesmo, causando problemas. As políticas de alocação do espectro são ótimas para evitar interferências, mas não oferecem uso flexível da faixa (PATIL; PRASAD; SKOUBY, 2011). Sob esse ângulo, governo e acadêmicos têm trabalhado para melhorar a política para que haja um acesso dinâmico ao espectro, evitando buracos espectrais.

Desenvolveram-se, portanto, métodos para resolver o problema da escassez do espectro de frequências. Refere-se por rádio cognitivo um sistema de comunicação sem fio inteligente que percebe o ambiente ao seu redor e é capaz de adaptar seus parâmetros de operação de forma que se consiga acessar faixas do espectro já utilizadas por usuários primários, sem causar interferências aos mesmos (SUN *et al., 2013)*, mas para isso, necessita de informações a priori. A classificação automática de modulação é importante neste estudo por poder auxiliar o sensoriamento do espectro fornecendo a informação do formato de modulação do usuário. Desta forma, é possível alocar usuários secundários numa mesma faixa de frequência minimizando a interferência com o uso de um esquema de modulação diferente, ou até mesmo usando o espectro quando o mesmo não está sendo utilizado, aumentando assim, a eficiência espectral.

Ainda com o propósito de se usar mais eficientemente o espectro, a modulação adaptativa é um método no qual o transmissor analisa o ambiente de operação, e troca dinamicamente a modulação em função da qualidade do canal. Quando o canal é bom, uma maior constelação pode ser usada na transferência de dados. Essa reconfiguração dinâmica de modulação exige que o receptor identifique com precisão o esquema de modulação de um sinal recebido (HONG, 2006).

Em relação às motivações pessoais do trabalho, decidiu-se por abordar o tema com o intuito de se obter maior entendimento na área de aprendizado de máquina. A popularização do assunto em inúmeras empresas do ramo de engenharia tem auxiliado a expansão das mesmas, buscando mais inovação através de análises de produtos e tendências de mercado. O

conhecimento do tópico se torna bastante necessário e útil para aumentar as chances de contratação após a formação. Além disso, o Laboratório de Pesquisas em Radiofrequências foi um dos trabalhos pioneiros, e se propõe a incentivar novos estudos acerca do tema.

1.2 ESTADO DA ARTE

Em geral, há duas diferentes abordagens quando se trata de classificação automática de modulação, a abordagem baseada em probabilidade (do inglês *Likelihood Based*, ou LB) e a baseada em características do sinal (do inglês *Features Based*, ou FB). A primeira consiste na extração de funções de verossimilhança e modelos estatísticos do sinal de entrada, e pode ser encontrada no trabalho de Orlic e Dukic (2009), e também no trabalho de Madhavan *et al.* (2013). Técnicas de comparação e similaridade serão de essencial importância para o classificador, seja com outros sinais ou até mesmo com um limiar definido, para poder se fundamentar uma decisão (TRIANTARIS *et al.*, 2019). A proposta para resolver o problema é de que a função densidade de probabilidade (do inglês *Probability Density Function,* ou PDF) da onda observada contenha toda informação necessária para se chegar a uma conclusão. Essa abordagem é muito utilizada, uma vez que se houver o conhecimento do modelo do canal e seus parâmetros, ela retorna excelente precisão ao classificador. Entretanto, tamanha precisão exige uma etapa de pré-processamento do sinal bastante intensa, o que remete a uma alta complexidade computacional. De acordo com a literatura, divide-se esta categoria em três conjuntos (DOBRE et al., 2007). Considerando que o número de hipóteses a serem feitas é igual ao número de modulações possíveis no estudo, pode-se calcular a PDF do sinal recebido através da média dos parâmetros, de uma estimativa, ou uma combinação dos dois.

O teste de razão de verossimilhança média (do inglês *Average Likelihood Ratio Test*, ou ALRT) tem por objetivo atingir a PDF de cada hipótese referente a cada modulação se tirando a média delas. Isso só é possível pois os valores desconhecidos (pontos de constelação, parâmetros relevantes do sinal, como amplitude e fase, e potência do ruído) são tratados como variáveis aleatórias. Portanto, faz-se uma distribuição desses parâmetros para se avaliar um possível resultado. Caso a distribuição proposta seja coincidente com a distribuição real, significa que o método retorna uma probabilidade máxima de classificação. O problema da alta complexidade computacional aparece nesse caso quando há muitos parâmetros desconhecidos.

O segundo conjunto do estudo baseado em probabilidade é o teste de razão de verossimilhança generalizado (do inglês *Generalized Likelihood Ratio Test*, ou GLRT). Neste caso, computa-se a PDF do sinal tratando seus valores desconhecidos como determinísticas desconhecidas, e atinge-se uma decisão a partir de estimativas de probabilidades máximas.

Enquanto que o terceiro conjunto abordado pela análise de funções de verossimilhança e modelos estatísticos é o teste de razão de verossimilhança híbrido (do inglês *Hybrid Likelihood Ratio Test*, ou HLRT), e possui esse nome pois é uma combinação dos dois métodos apresentados anteriormente.

Para atacar o problema de classificação automática de modulação, como dito anteriormente, além da abordagem baseada na probabilidade, tem-se a abordagem que consiste em extração de *features* chaves pré-determinadas do sinal recebido, para poder se tomar diferentes rumos para a tomada de decisão. Essa técnica apresenta um baixo custo computacional e precisão bastante próxima da ótima, e por esse motivo, prôpos-se trabalhar no seguinte projeto com a abordagem baseada em características do sinal.

Existem diversos tipos de *features* que podem ser utilizadas para o estudo, cada uma com suas devidas peculiaridades. Em Mobasseri (2000), percebe-se o uso de *features* baseadas no formato da constelação, enquanto que em Kim et *al.* (2016), tem-se o uso de *features* estatísticas.

Encontram-se, também, na literatura, diversas formas de executar a classificação. A árvore de decisão binária é um modelo onde se determinam limites de decisão, e a partir da característica identificada no sinal recebido, podem-se tomar decisões nos nós da árvore até chegar na decisão final, como pode ser encontrado no trabalho de Bagga e Tripathi (2013). Além disso, percebe-se o uso de técnicas de aprendizado de máquina para aplicar as características em classificadores específicos para o projeto, considerando o tamanho da base de dados, a categorização das etiquetas, entre outros detalhes. Classificadores que serão utilizados nos capítulos seguintes são também referenciados nos trabalhos de Wu *et al.* (2017), que utiliza do modelo *Support Vector Machine* e Aslam, Zhu e Nandi (2012), que utiliza do modelo K-*Nearest Neighbor*. Por último, encontram-se artigos que utilizam de redes neurais para se fazer a classificação, como os trabalhos de Hassan *et al.* (2010) e Zhao, Ren e Zhong (2006).

Quadro 1 - Métodos utilizados na literatura de AMC

Trabalho	**Abordagem**	**Estratégia**	**Classificador**	**Canal**	**Ano**	**Modulações**
TRIANTARIS *et al.*	FB	Rede Neural	CNN	AWGN	2019	BPSK, QPSK, 8PSK, 16QAM, 64QAM, GFSK, CPFSK, e PAM4. WB-FM e AM-DSB
ORLIC; DUKIC	LB	Cumulantes dos sinais analíticos		AWGN, *Rayleigh*, *Rice*, *Log-normal*	2009	BPSK, QPSK, 16-QAM, 64-QAM
WU *et al.*	FB		SVM	AWGN	2017	M-ASK, M-FSK, M-PSK e 16-QAM
HASSAN *et al.*	FB	*Continuous Wavelet Transform*	ANN	AWGN	2010	M-ASK, M-FSK, M-PSK, M-QAM
MOBASSERI	FB	*Features* baseadas no formato da constelação		AWGN	2000	QPSK, 8-PSK e 16-QAM
ZHAO; REN; ZHONG	FB	Rede neural baseado em *wavelet*	WNN	AWGN	2006	QPSK, OQPSK, GMSK, 16QAM, 4PSK, 2PSK, 4FSK, 2FSK, 4ASK e 2ASK
BAGGA; TRIPATHI	FB	*Features* estatísticas	Árvore de decisão	AWGN e *Rayleigh*	2013	2ASK, 4ASK, 2PSK, 4PSK 2FSK, 4FSK e 16 QAM. GMSK, 64QAM e 256QAM
MADHAVAN *et al.*	LB	Cumulantes cíclicos		AWGN e *Rayleigh*	2013	PSK e QAM
KIM *et al.*	FB	*Features* estatísticas	DNN	AWGN e *Rician*	2016	BPSK, QPSK, 8PSK, 16QAM e 64QAM
ASLAM; ZHU; NANDI	FB	Cumulantes de quarta e sexta ordem	KNN	AWGN	2012	PSK e QAM

Fonte: Elaborado pelo autor (2022).

1.3 OBJETIVOS

A seguir, descrevem-se os objetivos deste trabalho, de forma geral e específica.

1.3.1 Objetivos gerais

A proposta deste trabalho é a implementação e a análise de classificadores automáticos de modulação em um ambiente de modulações digitais conhecidas. Para tal, serão utilizadas técnicas de aprendizado de máquina sob abordagem de características baseadas nos parâmetros fundamentais dos sinais, como amplitude, fase e frequência.

1.3.2 Objetivos específicos

1. Estruturar uma base de dados com esquemas de 9 modulações digitais binárias conhecidas. São elas: "2-ASK", "4-ASK", “8-ASK”, "2-FSK", "4-FSK", “8-FSK”, "2-PSK" e "4-PSK", “8-PSK”.
2. Extrair *features* dos sinais da base de dados, as quais exploram características espectrais únicas de diferentes modulações em três aspectos chaves do sinal: amplitude, frequência e fase.
3. Definir quais classificadores serão empregados no estudo e aplicá-los.
4. Identificar o desempenho de diferentes classificadores utilizados perante diversas condições e investigar seus desempenhos.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados temas importantes abordados no projeto, tais quais os esquemas de modulação utilizados e suas formulações matemáticas, e conceitos de aprendizado de máquina.

2.1 SINAIS MODULADOS

Para que uma comunicação seja efetuada com sucesso, é necessário que um transmissor envie uma mensagem, através de um canal e entendida por um ou mais receptores. No contexto de telecomunicações, as mensagens são enviadas para longas distâncias. Desta forma, os sinais de informação devem ser modificados através do processo da modulação, que é dado através da alteração de algumas características de uma onda portadora de acordo com a informação que se deseja transmitir (LATHI, 1990). A onda portadora tem esse nome pois é responsabilidade dela portar a mensagem original de informação que será enviada, e possui, normalmente, frequência bem mais elevada que a informação a ser transmitida. Além disso, ela é normalmente um sinal cossenoidal e descrito conforme equação (1).

$$c(t) = A_c \cos(2\pi f_c t + \varphi_c) \tag{1}$$

onde A_c representa a magnitude do sinal, f_c representa a frequência da portadora em *Hertz*, e φ_c representa a fase.

Sendo assim, consegue-se deslocar a faixa de frequência em banda base para banda passante, aproveitando melhor o canal de comunicação. Um dos principais motivos para efetuar a modulação é que o tamanho da antena é inversamente proporcional à frequência utilizada, ou seja, para enviar informações centradas em uma baixa frequência, seria necessário ter antenas muito grandes, o que não é prático. Além disso, é possível que vários transmissores trabalhem paralelamente sem interferência, uma vez que se pode ajustar a frequência da onda portadora de acordo com o interesse, referindo assim, uma faixa única no espectro.

As modulações são divididas em duas categorias, analógicas e digitais. Atualmente, as modulações analógicas não são tão utilizadas nos sistemas modernos de telecomunicações,

sendo mais empregadas em emissoras de rádio. Enquanto que as digitais são amplamente utilizadas em vários outros sistemas de comunicação, e há uma forte tendência de padronização do seu uso. Recentemente, houve a mudança do padrão utilizado na transmissão de sinais de TV no Brasil de analógico para digital. Portanto, optou-se em trabalhar neste projeto somente com modulações digitais.

As modulações digitais são utilizadas quando se deseja transmitir um conjunto de valores discretos representados por um código, ou seja, uma informação digital. Desta maneira, cada grupo de *bits* será responsável por alterar a portadora de uma forma específica através de um chaveamento de seus parâmetros. Na literatura de comunicação digital, foi convencionado que a onda portadora teria energia unitária ao longo de um símbolo de duração (HAYKIN; MOHER, 2007), dessa forma tem-se a equação (2) que representa a sua amplitude.

$$A_c = \sqrt{\frac{2E_b}{T_b}} \tag{2}$$

onde T_b representa a duração de 1 bit. Combinando-a com a equação (1), tem-se a equação (3).

$$c(t) = \sqrt{\frac{2E_b}{T_b}} cos(2\pi f_c t + \varphi_c) \tag{3}$$

Além disso, uma técnica bastante utilizada no ambiente digital é o agrupamento de *bits* para se ter uma maior eficiência espectral, ou seja, monta-se um conjunto definido de *bits* denominado símbolo, e este é representado pelo parâmetro referente. A quantidade de *bits* que são agrupados é definida pelo projetista e é chamada de ordem. O sistema fica, deste modo, mais complexo, mas com a vantagem de se aumentar a taxa de transmissão, tornando-o mais eficiente.

2.1.1 Modulação em amplitude por chaveamento (ASK)

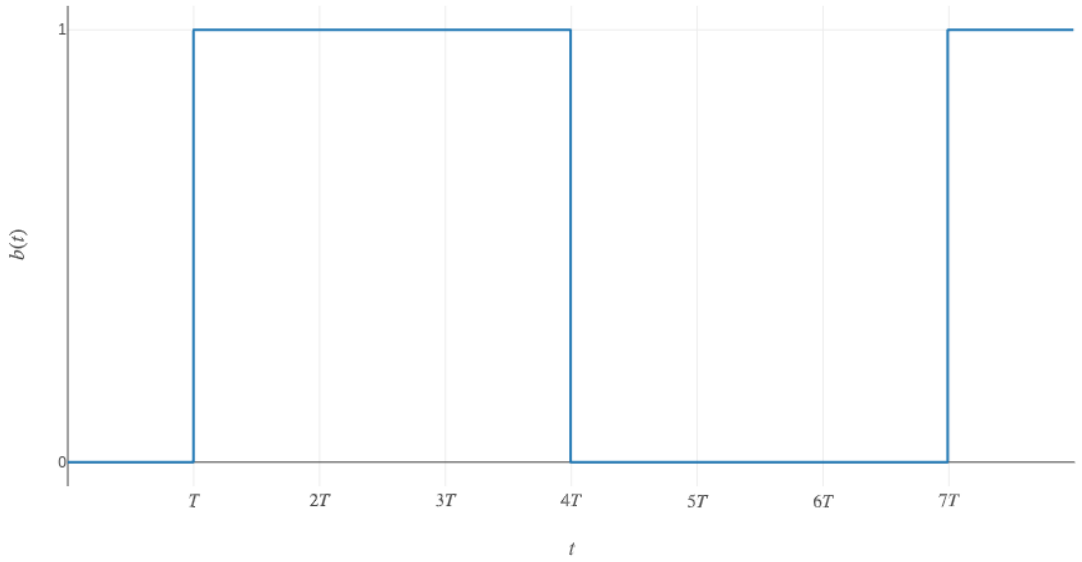
Na modulação em amplitude por chaveamento (do inglês *Amplitude-Shift Keying*, ou ASK*),* cada símbolo que se deseja transmitir é representado por uma amplitude correspondente. O sinal modulante assume um dos níveis discretos de amplitude que

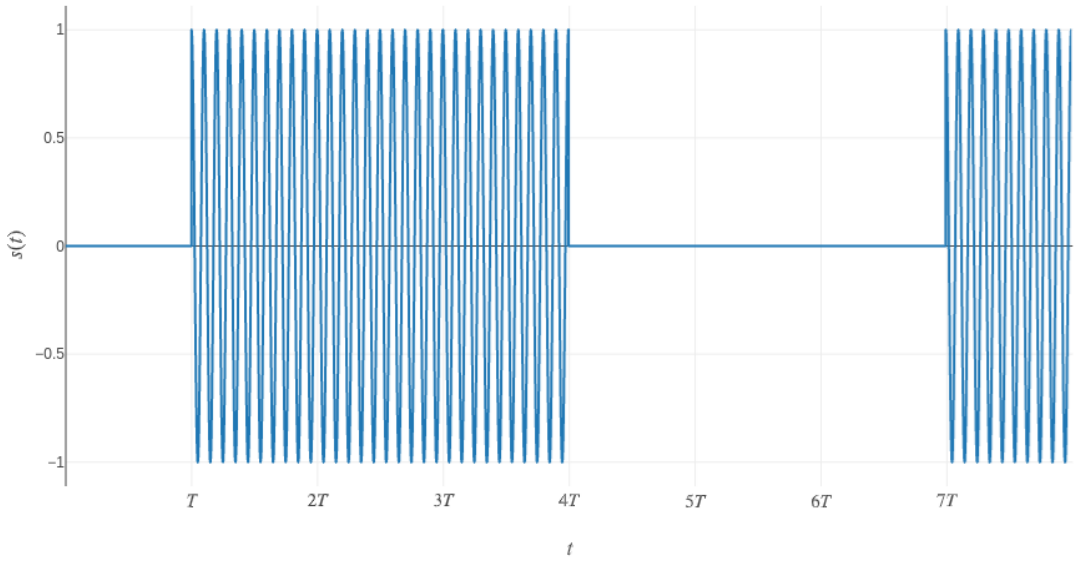
identificam o símbolo definido. Um caso específico da modulação ASK para um sistema com somente dois *bits* transmitidos é usar uma das amplitudes correspondentes sendo zero. Assim, envia-se a onda portadora quando se quer representar um dos *bits*, e quando se deseja representar o outro, não se envia nada, como pode ser visto na figura 1. Este formato é chamado chaveamento *On-Off* (do inglês *On-Off Keying*, ou OOK) e os *bits* enviados são definidos conforme equações (4) e (5).

$$b(t) = \sqrt{E_b},\ para\ o\ símbolo\ binário\ 1 \tag{4}$$

$$0,\ para\ o\ símbolo\ binário\ 0 \tag{5}$$

Figura 1 - Sinal modulado em OOK





Fonte: Elaborado pelo autor (2022).

onde E_b é a energia por bit. Quando o bit é multiplicado pela onda portadora, representa-se pelas equações (5) e (6) o sinal transmitido.

$$s(t) = \sqrt{\frac{2E_b}{T_b}}\ cos(2\pi f_c t + \varphi_c),\ para\ o\ símbolo\ binário\ 1 \quad (5)$$

$$s(t) = 0,\ para\ o\ símbolo\ binário\ 0 \quad (6)$$

Ou seja, quando o bit a ser transmitido é 0, a energia do sinal também é 0. E quando o bit a ser transmitido é 1, a energia do sinal é dada por E_b.

Para os demais casos, cada valor de amplitude identifica um símbolo correspondente e são definidos por M-ASK, sendo M a ordem utilizada.

Este processo é muito interessante devido à sua simplicidade, mas devido a modulação modificar somente a amplitude do sinal, pode gerar erros e confusão quanto à decisão do bit, visto que grande parte dos ruídos afeta a amplitude, refletindo em baixa imunidade ao ruído.

2.1.2 Modulação em frequência por chaveamento (FSK)

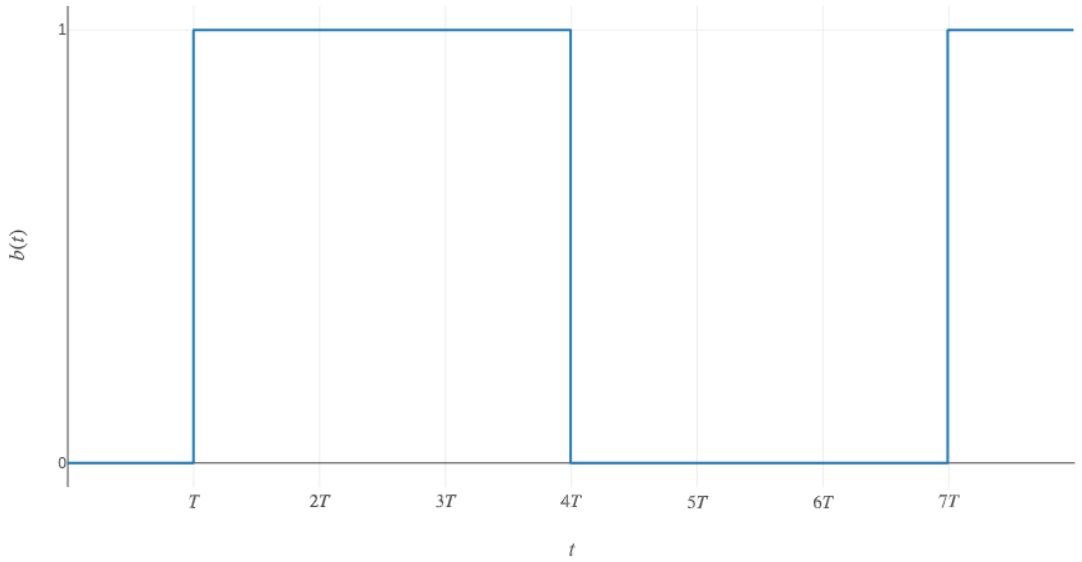
Na modulação em frequência por chaveamento (do inglês *Frequency-Shift Keying*, ou FSK), cada símbolo é representado por uma frequência definida. A modificação das frequências utilizadas é dada, geralmente, por uma variação através de um fator constante da frequência da onda portadora. Dessa forma, o espectro do sinal fica com frequências igualmente distantes da frequência central portadora. Nesse esquema tem-se amplitude e fase constantes, enquanto que a frequência que foi modificada indica o bit a ser transmitido. Para o caso de ordem sendo 2, apresentam-se as equações (7) e (8) que descrevem, respectivamente, o comportamento do sinal modulado com frequência f_1 referente ao bit 1 e frequência f_2 referente ao bit 0.

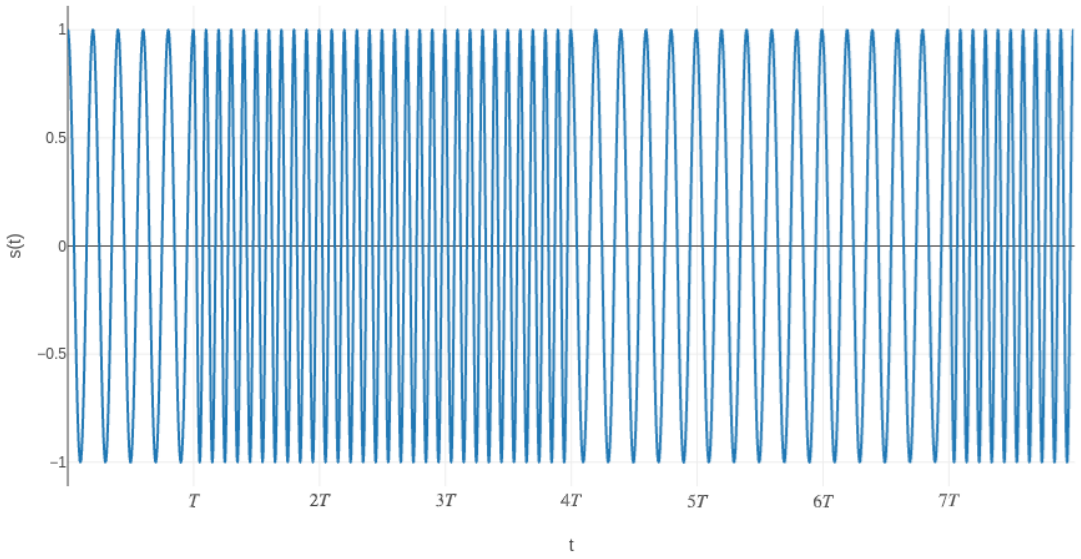
$$s1(t) = \sqrt{\frac{2E_b}{T_b}} cos(2\pi f_1 t),\ para\ o\ símbolo\ binário\ 1 \tag{7}$$

$$s0(t) = \sqrt{\frac{2E_b}{T_b}} cos(2\pi f_2 t),\ para\ o\ símbolo\ binário\ 0 \tag{8}$$

onde E_b é a energia por bit do sinal transmitido, T_b é a duração do bit, f_1 é a frequência referente ao bit 1, enquanto f_2 é a frequência referente ao bit 0 e é ilustrado na figura 2.

Figura 2 - Sinal modulado em 2-FSK





Fonte: Elaborado pelo autor (2022).

É interessante mencionar que dependendo dos valores de f_1 e f_2, pode haver uma descontinuidade no sinal, visto que normalmente não se tem certeza que as duas funções estarão no mesmo ponto em determinado momento. Isso é refletido em lobos secundários maiores no espectro de frequência. Existe uma forma de modulação particular que previne a descontinuidade na transição de *bits*. Essa forma é chamada de modulação em frequência por chaveamento de fase contínua (do inglês *Continuous Phase Frequency-Shift Keying*, ou CPFSK) e obedece uma relação entre as frequências utilizadas. Dessa maneira, tem-se um sistema mais complexo, porém garante-se uma melhor eficiência quanto à densidade espectral.

O uso de frequências para representação dos símbolos traz um problema evidente que é o aumento na largura de banda do sinal transmitido. As múltiplas frequências sucedem a vários componentes no espectrograma, ocupando mais espaço. Costuma-se apontar que a modulação FSK usa o espectro de forma ineficiente por tal motivo. Entretanto, há uma maior robustez ao ruído, visto que como característica principal da modulação, a amplitude é constante, enquanto as frequências variam de acordo com o símbolo.

2.1.3 Modulação em fase por chaveamento (PSK)

Na modulação em fase por chaveamento (do inglês *Phase-Shift Keying*, ou PSK*),* cada símbolo é representado por uma fase única. Os símbolos são transmitidos através de sinais cossenoidais diferenciados por um fator constante de fase entre si. As formas de onda podem ser representadas pela equação (9).

$$s_i\ (t)\ = \sqrt{\frac{2E_s}{T_s}}\, cos[2\pi f_c t\ + \frac{\pi(2i-1)}{M}],\ \text{i} = 1,2,...,\text{M} \tag{9}$$

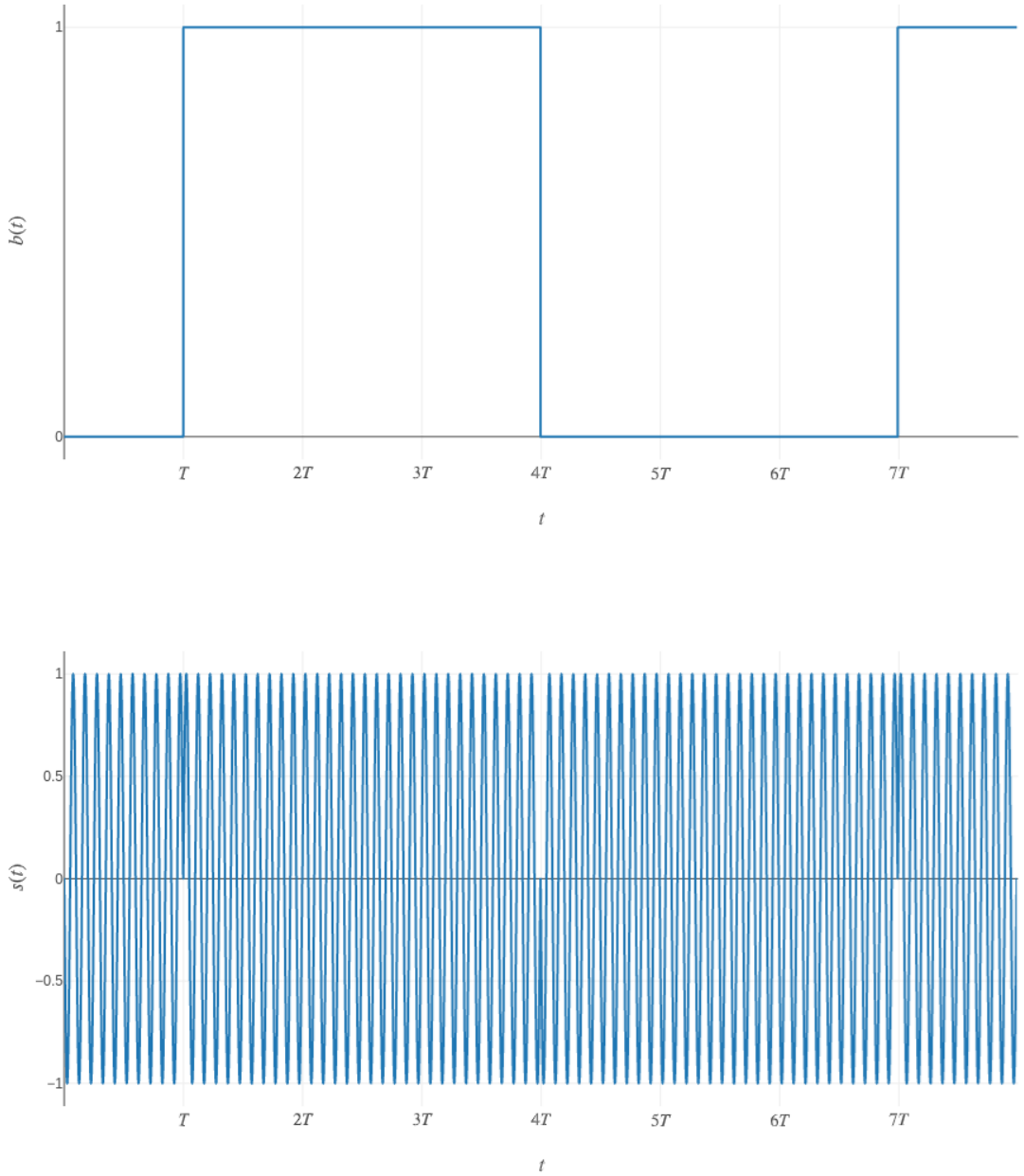
onde Es é a energia do símbolo transmitido, Ts é a duração do símbolo transmitido, fc é a frequência da portadora, e M é a ordem utilizada na modulação.

O caso mais simples é dito quando se deseja transmitir somente os símbolos 0 e 1, ou seja, tem-se a ordem M=2. Este caso é definido como modulação em fase por chaveamento binário (do inglês *Binary Phase-Shift Keying*, ou BPSK*)* ou 2-PSK, e é indicado na figura 3. Nessa situação, os sinais são diferenciados de 180 graus um do outro e o par de sinais fica definido conforme equação (10).

$$s_i\ (t)\ =\ \sqrt{\frac{2E_s}{T_s}}\, cos[2\pi f_c t\ + \frac{\pi}{2}(2i\ -\ 1)],\ i\ =\ 1, 2 \tag{10}$$

onde E_s é a energia do símbolo transmitido, T_s é a duração do símbolo transmitido, e f_c é a frequência da portadora.

Figura 3 - Sinal modulado em 2-PSK

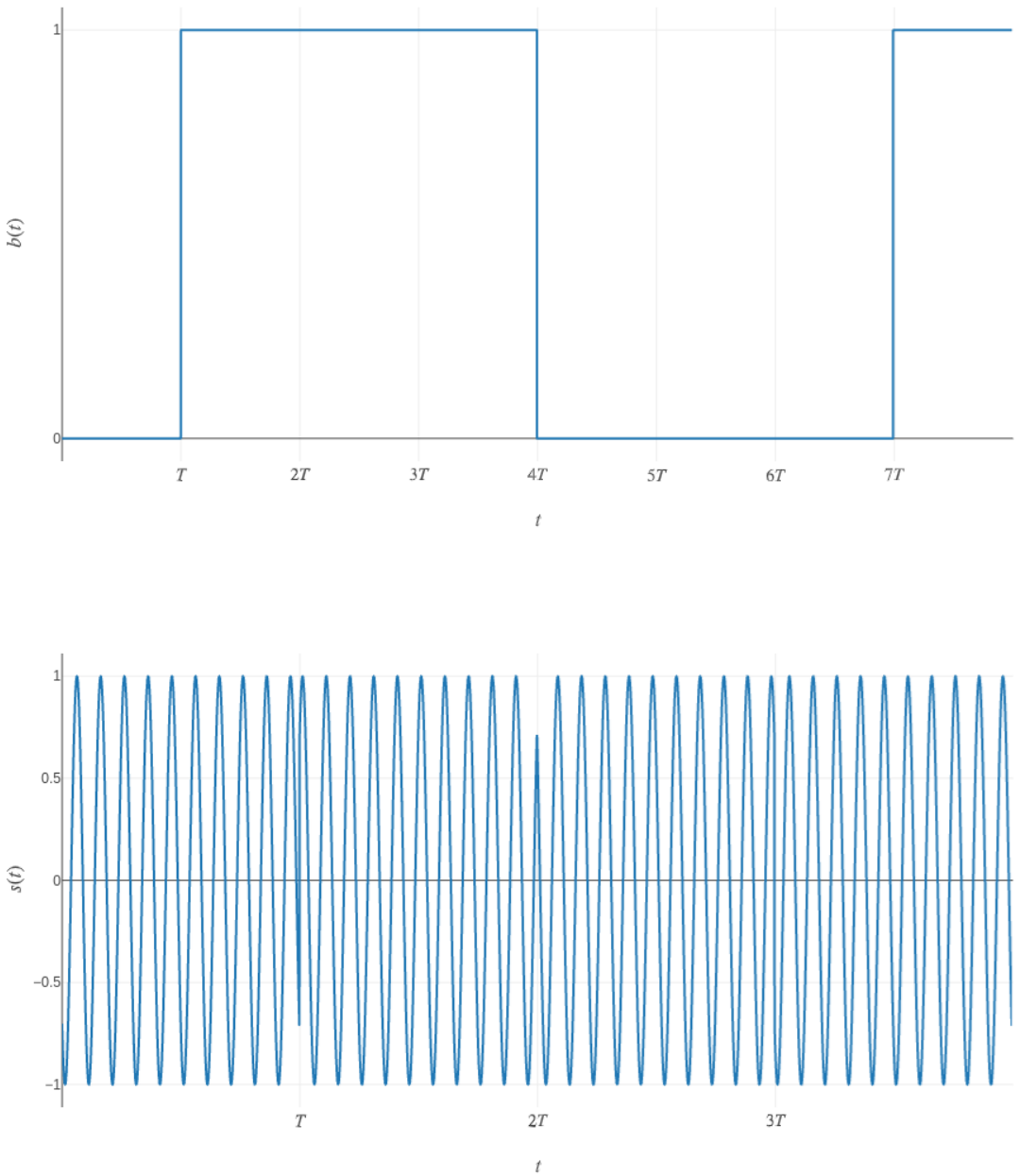


Fonte: Elaborado pelo autor (2022).

Quando se tem M=4, a modulação fica definida por modulação em fase por chaveamento em quadratura (do inglês *Quadrature Phase-Shift Keying*, ou QPSK*)* ou 4-PSK. É relevante mencionar que neste caso a taxa de *bits* enviados é duas vezes menor que no caso de ordem 2. Ou seja, a cada intervalo definido, envia-se dois *bits* ao invés de 1, como pode ser verificado na figura 4. O impacto disso no sistema é que se consegue enviar a mesma

informação em menos tempo, mas em compensação, trabalha-se com um sistema mais complexo.

Figura 4 - Sinal modulado em QPSK ou 4-PSK



Fonte: Elaborado pelo autor (2022).

Os sinais ficam diferenciados de 90 graus um do outro e são expressos matematicamente pela equação (11).

$$s_i\ (t)\ = \sqrt{\frac{2E_s}{T_s}} cos[2\pi f_c t\ + \frac{\pi}{4}(2i\ -\ 1)],\ i\ =\ 1, 2,3 \text{ e } 4 \qquad (11)$$

onde Es é a energia do símbolo transmitido, Ts é a duração do símbolo transmitido, e fc é a frequência da portadora.

A modulação QPSK é amplamente utilizada quando se trata de comunicação digital, estando presente em transmissões de satélite, sistemas móveis de celular e em padrões WLAN/Wi-Fi.

2.2 CANAL DE COMUNICAÇÃO

Para que uma comunicação seja efetuada com sucesso, é necessário que uma mensagem seja enviada por um emissor, através de um canal, e seja compreendida por um receptor.

O canal de comunicação é indicado pelo meio físico pelo qual o sinal está sendo transmitido e atua como se fosse um filtro que o distorce, geralmente atenuando-o (LATHI, 1990). Tal atenuação dependerá de vários fatores, mas principalmente da distância pela qual o sinal tem que percorrer. Além disso, o canal é determinado pela sua capacidade de transmitir informação, ou seja, a qual taxa essa informação pode ser transmitida sem que seja muito prejudicada.

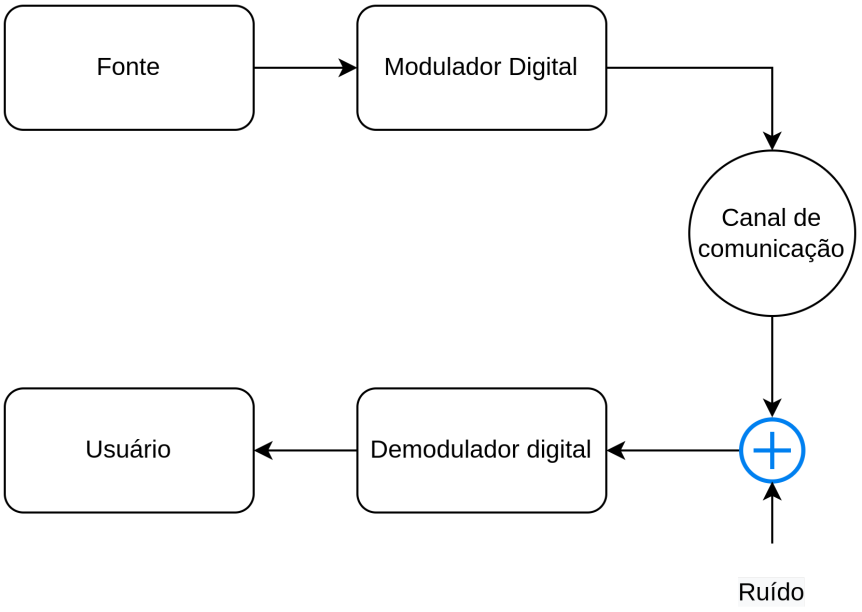
O sinal será, portanto, modificado pelas distorções provenientes do canal e também afetado por interferências presentes ao longo do caminho. No contexto atual, identifica-se tais interações indesejáveis como ruído, e se incluem variados tipos de situações externas, desde sinais transmitidos em canais próximos, até emissões de celulares, ruídos de equipamentos elétricos e radiação solar. Entretanto, o ruído também está presente em casos internos à transmissão, como por exemplo, através da movimentação térmica de partículas carregadas em condutores. Há formas de se reduzir o ruído tanto interno quanto externo, mas nunca é possível eliminá-lo por completo.

A capacidade de canal é definida, no ambiente de comunicação, pela taxa à qual a informação pode ser transmitida de forma confiável em um canal. No ambiente digital, a largura de banda do canal será uma limitante em relação à largura de banda dos sinais que poderão ser transmitidos de forma segura. Define-se como relação sinal ruído (do inglês *Signal Noise Ratio*, ou *SNR*) a comparação de potência de um sinal em relação ao ruído presente no mesmo. Quanto mais alto esse valor for, menor a influência do ruído presente na comunicação.

O modelo que representa o canal com ruído mais comum é conhecido como ruído aditivo branco Gaussiano (do inglês *additive white Gaussian noise*, ou *AWGN*). E é definido como aditivo pois é simplesmente adicionado a quaisquer demais ruídos presentes na comunicação, é considerado branco pois indica potência uniforme ao longo de toda faixa de frequência e Gaussiano por possuir uma distribuição normal no tempo com média zero.

A figura 5 exemplifica um modelo de sistema de comunicação, onde uma fonte gera uma mensagem que se deseja enviar e a envia ao modulador digital que define os parâmetros os quais serão utilizados dependendo do objetivo do projetista. Tal sinal modulado deve passar pelo canal de comunicação, e portanto, será acrescido de ruído. Quando tal sinal chega ao receptor, deve passar por um demodulador digital, e finalmente a mensagem chega ao usuário.

Figura 5 - Exemplo de modelo de sistema de comunicação com canal de comunicação e ruído



Fonte: Elaborado pelo autor (2022).

2.3 APRENDIZADO DE MÁQUINA

Para os humanos, a análise de dados através do reconhecimento de padrões é uma tarefa cotidiana. A experiência adquirida ao longo da vida permite que encontremos tendências ou padrões em determinados assuntos que auxiliam em futuras tomadas de decisão. Quanto mais se experimenta determinada atividade, mais habilidoso nos tornamos.

Isso é fruto da capacidade de aprendizado do ser humano. A repetição frequente de eventos funciona como uma espécie de treinamento (DOUGHERTY, 2012). O aprendizado de máquina (do inglês *Machine Learning*) utiliza, atualmente, um conceito semelhante ao que acontece com os seres humanos. É um sistema que pode avaliar a experiência própria e usar este conhecimento para melhor a tomada de decisões. Tal experiência pode ser adquirida através de uma profunda análise em uma base de dados que irá dar ferramentas para modificar ou não o comportamento do sistema.

Diante da ampla complexidade deste ramo da inteligência artificial, consegue-se abordar o estudo de diversas formas e lidar com as demandas exigidas pelos treinamentos. Os algoritmos são divididos em três grupos referentes a forma como o treinamento é feito: supervisionado, não supervisionado e de reforço. Esses grupos serão detalhados nos capítulos posteriores.

Para o projeto, foram necessários alguns passos para se trabalhar com o reconhecimento de padrões adequadamente. Necessita-se, inicialmente, da estruturação de uma base de dados coerente que funcionará como a experiência da máquina, para fazer futuras comparações com novos dados. Tais comparações serão feitas através da extração de características que sejam relevantes para a identificação dos padrões. Em seguida, é essencial definir um modelo de classificação que utilize as características determinadas previamente, o qual tomará decisões de interesse ao estudo. E por fim, avaliar o desempenho da classificação.

2.3.1 Tipos de aprendizado de máquina

O ramo de aprendizado de máquina se dividiu em diversas áreas com o intuito de lidar com as mais variadas tarefas presentes nos problemas da sociedade. Visto que o aprendizado é uma forma de se transformar experiência em conhecimento, considera-se três áreas referentes à maneira em que o ambiente fornece essa experiência (SHALEV-SHWARTZ; BEN-DAVID, 2014).

O aprendizado é dito supervisionado quando os dados fornecidos contém informações importantes, ou seja, quando para cada amostra da base de dados, tem-se uma resposta correta. Dessa maneira, consegue-se mapear uma regra geral, para poder atribuir um rótulo a novos dados. Também é possível encaixar neste tipo de aprendizado situações em que parte dos dados são rotulados, e outra parte não, através da procura de padrões semelhantes entre

os dados. Classificação e regressão são duas classes de algoritmos que são majoritariamente trabalhadas como aprendizado supervisionado.

Algoritmos baseados em classificação devem ter acesso a amostras corretamente rotuladas, e baseados em tais amostras, poderão receber dados novos em que não se tem resposta, classificando assim, em seus determinados grupos, que são definidos como conjunto de rótulos (do inglês *label set*). É importante notar aqui que a classificação é discreta, visto que as amostras da base de dados serão obrigatoriamente alocadas em algum dos rótulos disponíveis. Além disso, os grupos não precisam necessariamente ser binários, isto é, pode-se ter mais grupos de classificação, que são definidos como classificação de multiclasse. Dessa forma, tem-se um sistema mais complexo mas mais completo. Este tipo de aprendizado retorna uma acurácia, visto que é sabido o resultado em que se quer chegar, e é possível identificar o número das vezes em que a rotulação correta foi atingida.

A figura 6 representa o exemplo de treinamento supervisionado cujo objetivo é classificar novas instâncias quando comparadas com o conjunto de treinamento. Percebe-se que dentro de tal conjunto existem instâncias, caracterizadas pelas cartas, que são atribuídas individualmente a um determinado rótulo. A figura também aponta uma nova instância onde o rótulo não é sabido. O treinamento terá, por sua vez, a função de ensinar o classificador a identificar qual o rótulo referente a novos dados.

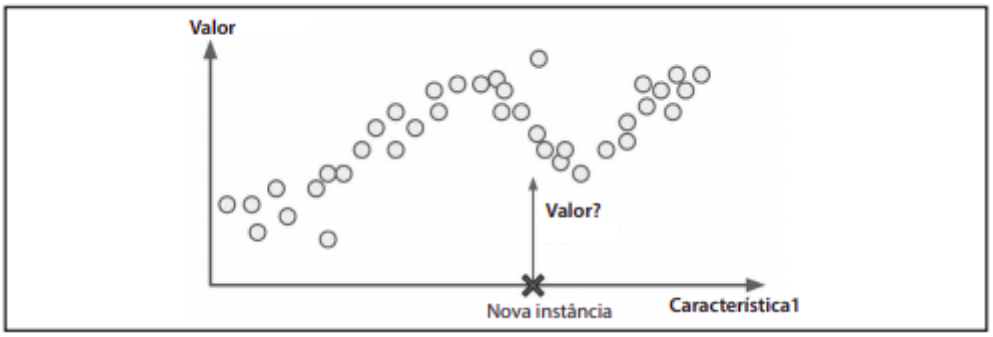
Figura 6 - Exemplo de predição de nova instância a partir do conjunto de treinamento

Fonte: GÉRON (2019).

Em casos em que algoritmos de regressão são utilizados, procura-se encontrar padrões nos dados que possam categorizá-los em uma escala contínua, podendo ser até uma probabilidade. Nesta situação, o modelo retorna uma função que prevê o estudo em contexto. A medida de acurácia nesses tipos de algoritmos é determinada através de parâmetros como o erro quadrático médio entre os valores corretos e os previstos pelo modelo. A figura 7

expressa um exemplo de treinamento supervisionado com algoritmo de regressão onde o objetivo é encontrar um valor numérico que corresponda a novas instâncias.

Figura 7 - Exemplo de treinamento supervisionado com algoritmo de regressão

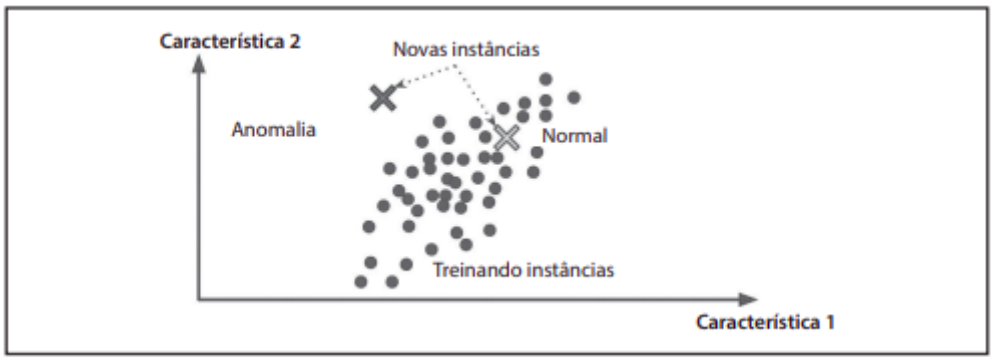


Fonte: GÉRON (2019).

Quando os dados fornecidos não são rotulados, o treinamento é dito não-supervisionado. Neste caso, apesar do objetivo ser definido, não se sabe exatamente a resposta que se deseja ter, uma vez que não se tem um conjunto de rótulos. O modelo se baseia em quaisquer padrões que se consiga encontrar entre os dados e deduz estruturas a partir do problema. Para executar o treinamento, utilizam-se algoritmos de associação ou clusterização de forma que se possa encontrar padrões comuns ou anomalias entre os dados. De forma geral, pode-se dizer que algoritmos de associação funcionam através da procura de correlação entre os dados, enquanto que algoritmos de clusterização procuram segmentar os dados em vários grupos a partir de medidas de similaridade. Por ser um treinamento bastante mais complexo que o supervisionado, visto que não se tem a resposta dos dados, torna-se complicado avaliar a qualidade de um modelo desse tipo.

Uma aplicação bastante comum do treinamento não supervisionado é identificar anomalias em uma base de dados. O sistema é treinado a partir de dados corretos ou normais, e quando recebe um dado que se difere muito da base, aponta-o como um possível erro. Vê-se muito esse tipo de treinamento em transações bancárias, onde pode-se determinar que uma transação foi fraudulenta quando comparada com outras transações. A figura 8 indica justamente um caso de identificação de anomalias. O classificador tem o papel de avaliar novas instâncias e compará-las com a base de dados. Em casos onde a distância do novo dado em relação à base esteja acima de um limiar, identifica-se esta amostra como um erro ou anomalia.

Figura 8 - Caso de detecção de anomalias através do treinamento não supervisionado



Fonte: GÉRON (2019).

Algoritmos de aprendizado de reforço aprendem através de ações e avaliando as consequências. Neste tipo de aprendizado, o agente realiza uma ação em um ambiente, e esta ação é recompensada ou punida. Dessa forma, o agente consegue distinguir quais ações geraram uma maior recompensa para cada possível cenário no futuro. É comum ver este método de aprendizado em ambientes onde já se conhece as regras, mas não se sabe a melhor forma de executá-las, como em videogames ou em jogos de xadrez.

2.3.2 *K-Nearest Neighbor*

O classificador *K-Nearest Neighbor* (KNN) é um algoritmo baseado em treinamento supervisionado cuja função é determinar o quão semelhante um dado desconhecido é dos demais dados conhecidos. Para tal tarefa, é feito um mapeamento dos dados e suas características, e então verifica-se um número determinado de vizinhos de um novo dado que se deseja classificar. A explicação disso é que as características usadas para descrever as amostras são bastante relevantes no quesito de rotulação, de forma que novas amostras que tenham características semelhantes irão tender a ter o mesmo rótulo.

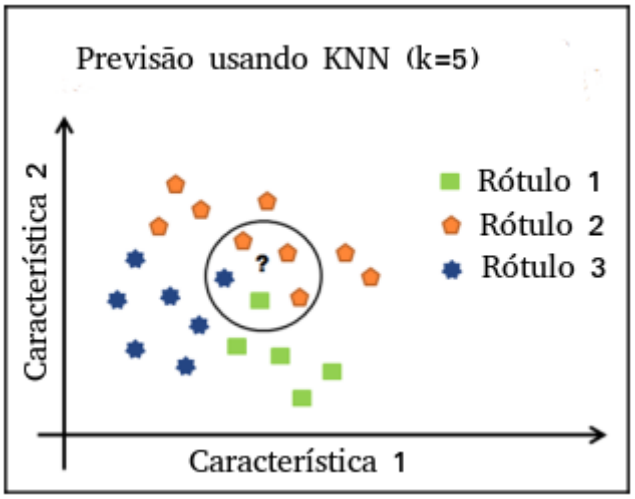
Para efetuar esse tipo de classificação, é necessário definir um espaço de características referência. A função deste espaço é mapear as N características decididas para o projeto em um espaço N-dimensional. Quanto maior o número de características escolhidas, mais preciso se torna a classificação, mas também mais complexa.

Após o recebimento de um dado não classificado, deve-se medir a distância da amostra de teste aos demais vizinhos. A forma que essa distância é calculada é definida pelo projetista, sendo a mais comum a distância euclidiana, mas podendo variar entre, ponderada, Manhattan, entre outras. Sendo assim, obtém-se assim um conjunto de distâncias deste dado

em relação aos demais já conhecidos e rotulados. Por fim, classifica-se este dado desconhecido através de uma verificação das menores distâncias em relação a *k* vizinhos mais próximos.

A figura 9 exemplifica uma situação em que se deseja classificar um novo dado caracterizado pelo ponto de interrogação analisando somente os cinco vizinhos mais próximos. Consegue-se averiguar a distribuição dos demais dados quando avaliados sob a perspectiva das duas características de decisão presentes no problema. Dentre os cinco vizinhos, identifica-se três deles sendo referentes ao rótulo 2, logo, o modelo irá ponderar e classificar o novo dado como sendo pertencente também ao rótulo 2.

Figura 9 - Caso de predição de um dado desconhecido baseado nos 5 vizinhos mais próximos



Fonte: adaptado de DANGETI (2017).

A definição de quantos vizinhos deverão ser considerados precisa seguir algumas regras, a fim de se evitar resultados incorretos. Idealmente, deve-se procurar por um valor ímpar de vizinhos, principalmente para casos binários, com o intuito de se evitar que uma amostra tenha o mesmo número de vizinhos em diferentes classes. Além disso, é necessário que o valor de vizinhos seja suficientemente grande visto que muitas vezes aparecem erros por amostras fora da curva. E certamente, o número de vizinhos verificado deve ser menor do que o número de vizinhos totais no espaço referência. Determina-se que o valor de *k* depende muito do problema e da base de dados. Em algumas situações pode-se usar algoritmos de otimização para escolher o valor, ou até mesmo defini-lo empiricamente.

Conclui-se que o classificador KNN é capaz de realizar classificações em que se tenham várias classes, mas sofre com o aumento do espaço de características referência, uma

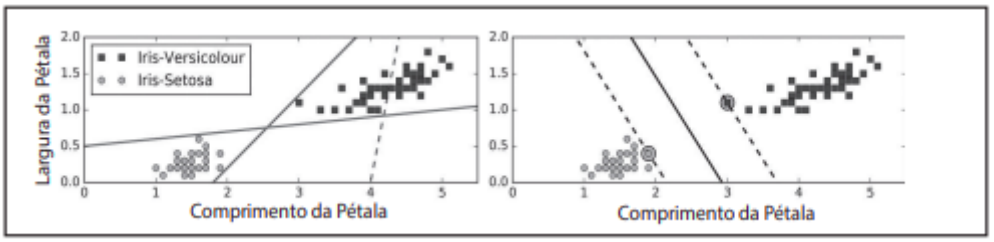
vez que aumenta-se a dimensionalidade do conjunto de características e a complexidade do cálculo da distância. Por muitas vezes, exige-se uma redução de dimensionalidade para tornar o problema mais viável. Além disso, todas as características exercem igual influência sob este classificador, o que pode gerar confusão na decisão final.

2.3.3 Máquinas de vetores de suporte

Algoritmos do tipo máquinas de vetores de suporte (do inglês *Support Vector Machine* ou *SVM*) são uma outra alternativa para modelos de treinamento supervisionado com objetivos de classificação ou regressão. Após o mapeamento dos dados, o modelo determina hiperplanos com o objetivo de fazer uma separação entre os vetores (ou classes). A distância do hiperplano deve ser otimizada de forma que ele fique à máxima distância dos pontos de margem de cada conjunto de rótulos.

Normalmente, algoritmos SVM tendem a separar as classes de forma perfeita através de uma fronteira de decisão, entretanto, isso pode acarretar em uma classificação muito específica, que responderá com alta precisão para o conjunto de dados de teste, mas para novos conjuntos ou dados afetados por ruídos trará resultados modificados. Usa-se o método de margem suave (do inglês *Soft Margin*) com o intuito de evitar esse tipo de problema, que permite classificar de forma incorreta alguns dados do grupo de treinamento, mas o sistema se torna mais generalista. E para isso, busca-se aumentar a distância do hiperplano de decisão em relação aos vetores de suporte, maximizando os pontos que são classificados corretamente.

Figura 10 - Modelo comparativo de hiperplanos para SVM



Fonte: GÉRON (2019).

Observa-se na figura 10 a representação de uma base de dados com dois possíveis rótulos de flores, a Iris Versicolor e a Iris Setosa. Na imagem à esquerda, tem-se hiperplanos que discriminam satisfatoriamente os rótulos, mas não são otimizados, visto que um ponto

fora da curva pode ser classificado com o rótulo errado. Enquanto que na imagem à direita, tem-se o hiperplano otimizado equidistante entre os pontos de margem de ambos os conjuntos.

É importante comentar que esse modelo é computacionalmente menos custoso, uma vez que os dados de teste só são usados para definir o hiperplano. Após a definição, não há necessidade de envolver toda a base de dados para executar o procedimento.

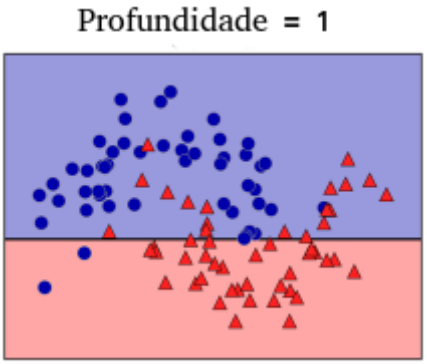
2.3.4 Árvore de decisão

Árvores de decisão são previsores amplamente utilizados no ambiente de aprendizado de máquina em casos de aprendizado supervisionado em tarefas de classificação e regressão. Para executar a previsão, o modelo percorre um caminho através de um conjunto de elementos designados por nós que vão da raiz até a folha da árvore. O nó de partida é chamado de raiz e é seguido por ramificações denominadas de filhos, que consistem em passos condicionais abrangendo as *features* selecionadas.

O aprendizado de uma árvore de decisão se dá, essencialmente, através de uma sequência de perguntas hierárquicas contendo duas possíveis respostas em cada nó, ou seja, os nós são testes referentes a alguma das características, e seus ramos descendentes indicam informações sobre tal atributo.

Para projetar uma árvore, o algoritmo deve procurar dentro de todas as possibilidades de teste aquela que fornece mais informação sobre a variável de interesse (MÜLLER; GUIDO, 2016). O ganho de informação é verificado ao dividir o espaço em sub-regiões de forma que se tenha os dados de diferentes classes separados com maior eficiência. O processo deve ser repetido para os nós seguintes até que um critério previamente determinado seja estabelecido.

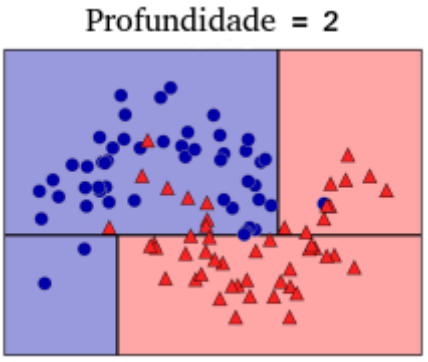
Figura 11 - Representação do limiar de decisão de uma árvore com profundidade = 1



Fonte: Adaptado de MÜLLER; GUIDO (2016).

Conclui-se que a cada nó haverá um teste binário, dividindo parte dos dados em sub-espaços através dos eixos do mapeamento. Ter-se-á regiões de divisão paralelas aos eixos, uma vez que cada teste é definido em relação a alguma *feature* do estudo. Quanto maior a profundidade da árvore, mais sub-regiões se atingem, aumentando assim, a precisão do modelo. A figura 12 demonstra uma árvore de decisão com profundidade 2, ou seja, dividiu-se o espaço dos dados em uma sub-região através de um eixo paralelo ao eixo x, e com o intuito de se obter uma classificação mais precisa, dividiu-se no nó seguinte os dados em mais uma sub-região através de um eixo paralelo ao eixo y.

Figura 12 - Representação do limiar de decisão de uma árvore com profundidade = 2



Fonte: Adaptado de MÜLLER; GUIDO (2016).

2.3.5 Floresta aleatória

Existem métodos para aperfeiçoar modelos que não alcançam a precisão desejada ou os resultados esperados. O método de *ensemble* é a combinação de múltiplos modelos de aprendizado de máquina com objetivo de criar modelos mais poderosos. Quando se trata de classificação e regressão, usa-se a floresta aleatória (do inglês *random forest*).

A floresta aleatória é essencialmente, um conjunto de árvores de decisão, onde os parâmetros avaliados em cada árvore são levemente diferentes uns dos outros. Esse modelo permite resolver problemas existentes na dinâmica de aprendizado de máquina como sobreajuste (do inglês *overfit*) e redução de dimensionalidade.

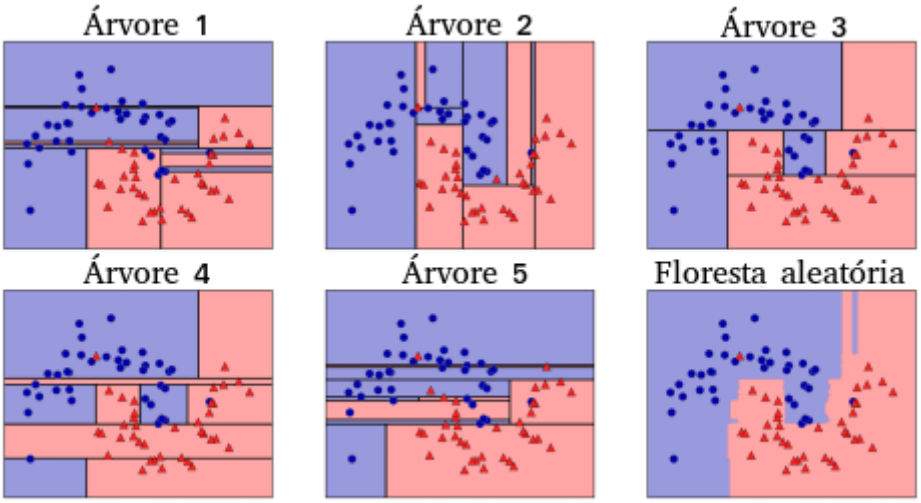
Para implementar esse método em casos de classificação, deve se ter um conjunto de árvores de decisão que irão, individualmente, decidir a classe a partir das características determinadas. A função da floresta aleatória será avaliar a classificação de suas árvores e indicar à qual classe a amostra pertence. Em casos de regressão, dever-se-á avaliar a média das decisões das árvores. A aleatoriedade é dada através de duas maneiras, a primeira consiste em fazer uma seleção dos pontos das amostras que estruturam a árvore, enquanto a segunda é referente às *features* utilizadas em cada nó de teste.

É importante mencionar que cada árvore irá avaliar os testes nos nós através de diferentes conjuntos de *features*, procurando usar o melhor teste possível para cada nó. A escolha dos conjuntos será repetida em cada nó de forma que se tenha diferentes decisões.

Devido a utilização dos mesmos dados em cada árvore contida na floresta aleatória, pode-se perder precisão no classificador. Portanto, é essencial que não se utilize exatamente o mesmo conjunto de dados em cada árvore. A amostragem de *bootstrap* é um método utilizado para diminuir a variância e o erro quadrático médio do classificador, onde em cada nó é utilizado um subconjunto diferente da base de dados, que combinado com a utilização de diferentes conjuntos de *features* aplicados em cada nó, assegura-se uma decisão diferente e otimizada.

Na figura 13, tem-se um exemplo de floresta aleatória que consiste de cinco árvores de decisão que definiram diferentes limites próprios de decisão através de seus algoritmos. Cada uma apresenta seus respectivos erros como pode ser apontado pelos pontos de treino que ficaram fora da região determinada, seja pela combinação de *features* utilizadas nos nós, ou pelos distintos conjuntos de dados definidos pela amostragem de *bootstrap*. A classificação feita pela floresta aleatória é obtida através da média das probabilidades de previsão e apresenta um erro muito menor com regiões mais bem definidas e intuitivas, e supera a classificação feita por cada uma das cinco árvores individualmente.

Figura 13 - Floresta aleatória combinada por cinco árvores de decisão



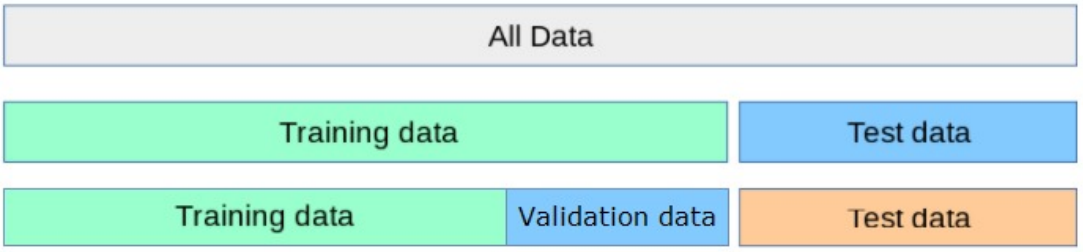
Fonte: Adaptado de MÜLLER; GUIDO (2016).

2.3.6 Validação cruzada

Para determinar o desempenho de um modelo de generalização, deve-se avaliar o modelo quando aplicado a um novo conjunto de dados. Entretanto, em algumas situações, há uma carência na quantidade de dados e não é desejável que os gastem para fazer a validação.

A validação cruzada é um método o qual divide o conjunto de dados em três subconjuntos. O subconjunto de treinamento será responsável, juntamente com o subconjunto de validação, por ajustar os parâmetros do sistema, enquanto que o subconjunto de teste terá a função de simular a entrada de novos dados desconhecidos. Dessa forma, assegura-se que o subconjunto de teste faz parte do modelo, e avalia-se sua acurácia quando verificado em relação a todo o subconjunto de treinamento. Na literatura, existem alguns métodos que se diferenciam em relação à partição da base de dados que será dividida como subconjunto de teste e subconjunto de treinamento.

Figura 14 - Divisão dos dados em subconjunto de treinamento, validação e teste



Fonte: Adaptado de SCIKIT-LEARN (2021).

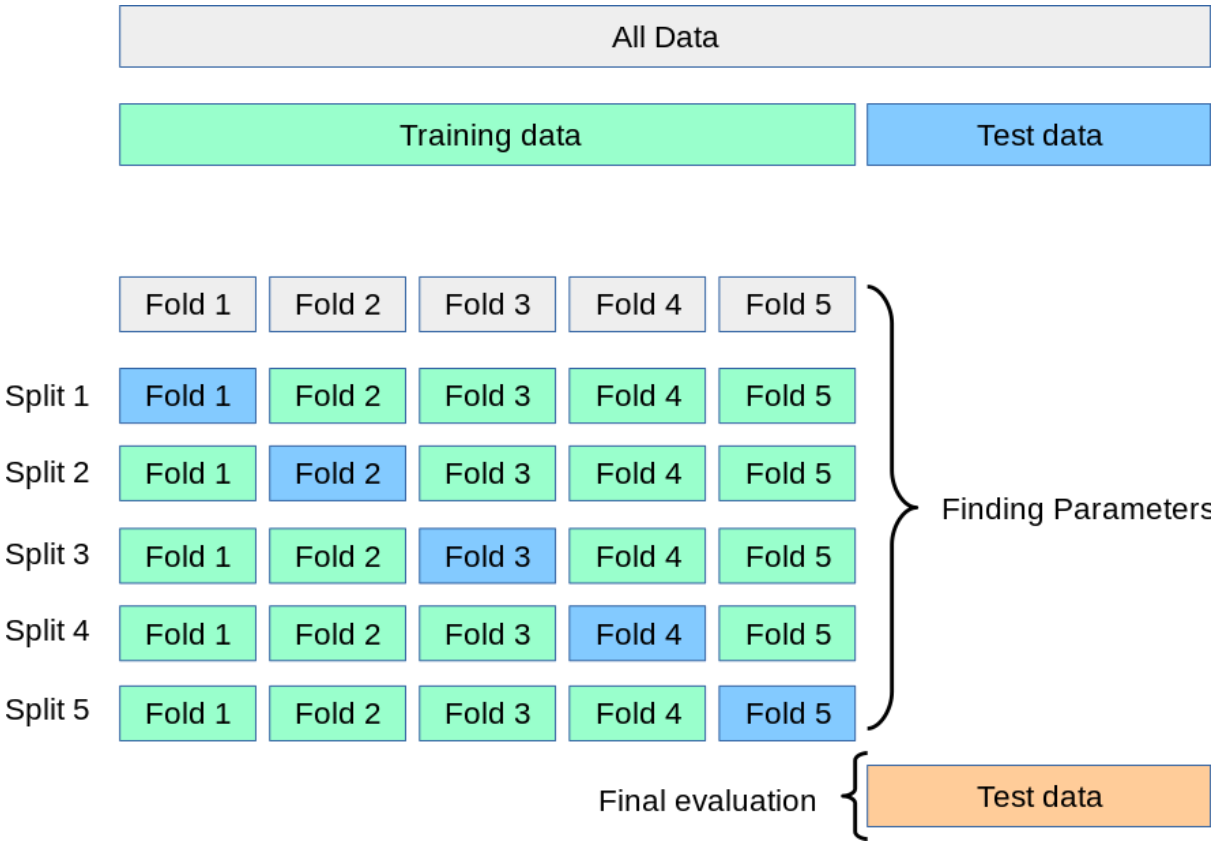
O método *holdout* particiona o conjunto total de dados sendo ⅓ como conjunto de teste, enquanto ⅔ são o conjunto de treinamento. Após a divisão, aplica-se o modelo sobre os dados de treinamento, e verifica-se a acurácia utilizando os dados de teste. Esse método é indicado quando há um grande número de dados disponíveis, mas ainda assim, pode causar erros quando o subconjunto de testes se assemelha muito com o subconjunto de treinamento. Quando se tem poucos dados disponíveis, não é aconselhado usar esse método pois pode causar uma grande variação no erro.

Além disso, tem-se o método *leave-one-out*, ou deixe um de fora. É utilizado normalmente quando se tem um número enorme de dados, onde apenas um dado será usado como subconjunto de treinamento. Dessa maneira, propõe-se um erro estimado para cada dado usado no treinamento. É um método eficiente, porém exige grande custo computacional, uma vez que serão feitos tantos testes quantos dados se tem no modelo.

O método de validação cruzada *k-fold* é utilizado na busca de hiperparâmetros que dêem melhores respostas ao sistema, e sua função é particionar o conjunto de treinamento em *k* subconjuntos de mesmo tamanho. Cada um deles é considerado individualmente como subconjunto de teste, enquanto todos os demais compõem o subconjunto de treinamento, como pode ser verificado na figura 15. Dessa maneira, consegue-se indicar o erro individual de cada um quando comparado com todo o conjunto de treinamento. A média de todos erros aponta o erro verdadeiro do modelo, e é utilizada para decidir o melhor parâmetro que gera menos erros. Uma vez descoberto, aplica-se o algoritmo no conjunto de treinamento particionado, mas agora em combinação com o conjunto de testes que é totalmente novo e ainda não foi verificado pelo modelo. Esse método aponta uma acurácia satisfatória, mas

devido ao grande número de testes em relação à base de dados, tem-se uma alta complexidade computacional, o que certamente deve ser colocado na balança.

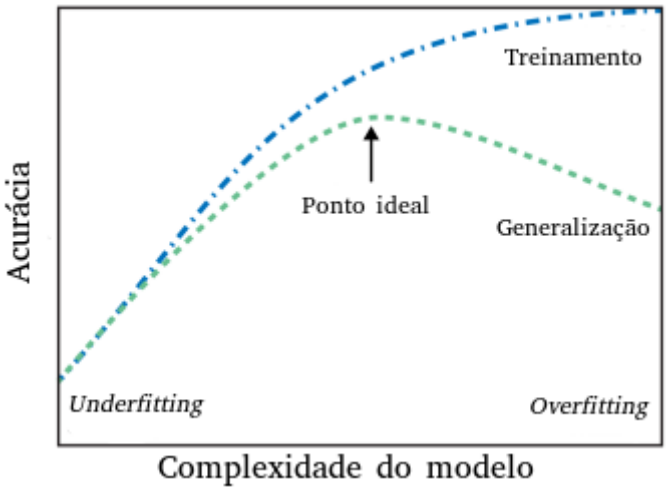
Figura 15 - Representação do método de *k-fold Cross Validation*



Fonte: SCIKIT-LEARN (2021).

Conclui-se, portanto, que se deseja implementar um modelo capaz de fazer generalizações o melhor possível. Entretanto, a única medida que garante se o modelo terá uma boa performance em novos dados é o subconjunto de testes utilizado. Em alguns casos, o modelo se tornará muito assertivo em relação aos dados já existentes, fazendo com que as particularidades do subconjunto de teste se tornem muito relevantes, mas com novos dados poderá apresentar uma taxa elevada de erro. Diz-se que o modelo perde parte da capacidade de generalização devido a um fenômeno chamado de *overfitting*. Do mesmo modo, quando o modelo é treinado a partir de um algoritmo muito simples, ele não é capaz de fazer boas previsões, seja por um baixo número de dados de treino ou *features* irrelevantes ao processo. Diz-se, então, que o modelo sofre de *underfitting*.

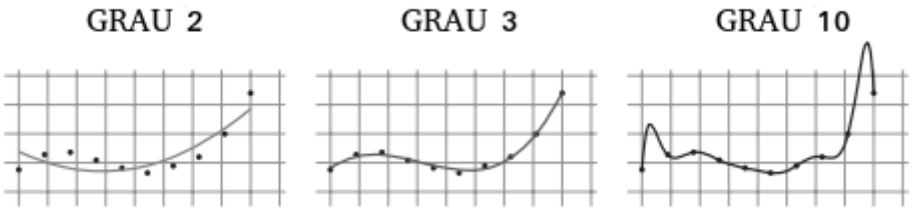
Figura 16 - Acurácia de modelos com *underfitting* e *overfitting*



Fonte: adaptado de MÜLLER; GUIDO (2016).

Como pode ser verificado na figura 16, a acurácia de um modelo com *underfitting* é bastante baixa, tanto no subconjunto que o treinou quanto no subconjunto de teste ou de generalização, devido à simplicidade do modelo proposto. Caso o modelo se torne muito complexo, ele fará ótimas previsões para os dados os quais foi treinado, como pode ser visto pela alta acurácia do subconjunto de treinamento, mas apresenta resultado não satisfatório quando analisado em relação a um subconjunto de teste novo. Por este motivo, deve-se procurar encontrar o ponto ideal que apresente a maior acurácia possível para cada modelo.

Figura 17 - Modelos com *Underfitting*, ideal e *Overfitting*, respectivamente



Fonte: Adaptado de SHALEV-SHWARTZ; BEN-DAVID (2014).

Na figura 17, identificam-se três modelos com o objetivo de se obter uma representação polinomial adequada aos pontos presentes no gráfico. No modelo à esquerda, foi utilizado um polinômio de segundo grau para fazer a previsão. Entretanto, percebe-se que o resultado foi bastante simples e não representa os pontos de forma satisfatória. O modelo à

direita, por sua vez, utiliza um polinômio de décimo grau, que apesar da excelente previsão perante aos dados presentes possui uma baixa capacidade de generalização em relação a dados futuros. O modelo central foi justificado como o melhor modelo dentre os três, uma vez que apresentou uma representação adequada aos dados existentes e é capaz de generalizar com certa precisão os dados futuros.

2.3.7 Métricas de avaliação

Existem algumas formas de avaliar o desempenho dos modelos. Quando se trata de classificação, seja ela binária ou de multiclasses, há algumas que são interessantes de serem mencionadas. Em geral, a métrica é calculada para cada classe, tratando-a como um problema de classificação binária e depois analisando através de uma perspectiva que englobe todas as classes. Avalia-se a eficácia de um modelo analisando a resposta dada por ele em relação à resposta correta. Os meios mais comuns de se verificar isso são: acurácia, precisão, *recall* e F1 *Score*.

Por muitas vezes a métrica da acurácia se confunde com a precisão, até porque ambas têm traduções semelhantes quando passadas pro inglês. A acurácia tem a função de indicar quantos acertos o modelo apresentou em relação ao total. Para o caso de classificação binária, analisa-se o número de verdadeiros positivos (do inglês *True Positives*, ou TP) e verdadeiros negativos (do inglês *True Negatives*, ou TN), e divide-se pelo número total, considerando os falsos positivos (do inglês *False Positives*, ou FP) e falsos negativos (do inglês *False Negatives*, ou FN). Esse indicador pode enganar o projetista quando há muito mais respostas rotuladas de uma forma do que de outra, porque até mesmo em casos que o modelo classifique todos os dados como sendo da mesma classe, em muitas vezes ele vai acertar, acusando uma alta acurácia quando na verdade o modelo é ruim. Esse critério funciona melhor em situações onde a base de avaliação tem quantidades semelhantes de dados rotulados.

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

Por outro lado, a precisão indica as previsões que determinam uma certa classe corretamente, em relação às previsões que determinam essa mesma classe de maneira errada.

Essa métrica permite avaliar um resultado individualmente de maneira mais precisa, visto que indica a quantidade de falsos positivos.

$$Precisão = \frac{TP}{TP + FP} \tag{13}$$

A métrica *recall* tem a função de comparar a quantidade de dados que foram rotulados em uma certa classe com a quantidade total de dados que deveriam ter sido rotulados dessa classe. Ou seja, analisa o total de verdadeiros positivos em comparação com a soma de verdadeiros positivos e falsos negativos. Usa-se essa métrica com o intuito de identificar e minimizar os falsos negativos.

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

A métrica F1 *Score* indica uma combinação entre precisão e *recall*. A média aritmética apresenta um valor coerente para expressar as duas métricas, mas em algumas situações onde um dos dois valores é muito baixo, o resultado não entrega muita informação. Portanto usa-se a média harmônica entre os dois critérios. Sendo assim, consegue-se avaliar casos em que falsos negativos e falsos positivos trazem consequências distintas para o modelo.

$$F1\ Score = \frac{2*Precisão*Recall}{Precisão + Recall} \tag{15}$$

3 MATERIAIS E MÉTODOS

Na literatura de classificação automática de modulações, há diferentes abordagens para o problema. Em geral, divide-se o estudo em três etapas. Primeiro tem-se a geração dos sinais que irão compor a base de dados, então a extração de características relevantes para o estudo, e por último, a classificação.

Figura 18 - Etapas de uma classificação automática de modulação

Fonte: Elaborado pelo autor (2022).

3.1 BASE DE DADOS

Para se ter uma classificação correta, deve-se ter uma base de dados adequada e suficientemente grande. A base de dados foi construída de forma que se obtivessem dados inter relacionados cujo objetivo principal fosse prover um ambiente eficiente para uso e armazenamento das informações. Para isso, deve ser composta dos mais variados tipos de sinais dentro das modulações citadas neste projeto, de forma que se tenha um grande gama de amostras que possam englobar muitas das possibilidades, tornando o estudo mais genérico e eficaz.

3.1.1 Obtenção da base de dados

Para o seguinte projeto, foi necessário projetar uma base própria de dados que pudesse ser utilizada para análise. Optou-se por estruturar a base de dados através de sinais modulados em amplitude, fase e frequência, uma vez que são os parâmetros fundamentais no contexto de modulação. Cada sinal contém uma mensagem, que para ser enviada, tem suas

características modificadas de forma a facilitar a comunicação. No ambiente digital, esta mensagem é representada por um conjunto finito de valores discretos.

Portanto, projetou-se uma fonte binária aleatória a qual pudesse fornecer um conjunto de *bits* de tamanho também aleatório que representasse a mensagem presente na comunicação digital, estruturando assim, a base de dados. De acordo com a literatura, encontram-se estudos com os mais variados tamanhos da base de dados. Em Triantaris et al. (2019), foi usado uma versão extendida da base de dados RadioML 2016, que é muito utilizada em experimentos relacionados à detecção de modulação. Contendo 1.2 milhões de amostras separadas em conjuntos de validação, teste e treinamento. Já em Wu et al. (2017), foram avaliadas 400 amostras em cada um dos 21 diferentes níveis de razão sinal-ruído presentes no trabalho, constituindo a base de dados com 8400 amostras. Para o presente projeto, decidiu-se avaliar o desempenho de diferentes modelos sob a óptica de uma base de dados contendo 3000 amostras.

Ainda dentro do bloco gerador de sinais, tem-se o modulador, cuja função é aplicar os parâmetros de modulação à mensagem digital, de forma a transformar o sinal de maneira adequada para permitir uma comunicação eficiente. Há vários parâmetros relevantes para o bloco do modulador que foram projetados com o intuito de agregar a base de dados. O primeiro passo consiste em aplicar uma das três diferentes modulações utilizadas no projeto através de uma onda portadora, de acordo com o objetivo da mensagem, atentando-se às vantagens e desvantagens presentes em cada uma delas. Além disso, é importante também definir a forma em que os *bits* serão agrupados em cada sinal, indicando a ordem do sinal em questão. Quanto maior a ordem, mais complexos são os sistemas, mas em contrapartida transmitem a mensagem em um intervalo menor de tempo. O bloco modulador em questão será responsável por aleatorizar esses dois atributos para cada mensagem. Ou seja, ter-se-á uma base de dados com mensagens de diferentes tamanhos, sendo atuadas por modulações de amplitude, fase e frequência, e com agrupamentos de *bits* individuais, duplas e trios caracterizando ordens com M = 2, 4 e 8.

Dentro do processo de estruturação da base de dados, também foi necessário modelar um canal de comunicação que representasse o meio pelo qual os sinais projetados percorreriam. O modelo mais recorrente na literatura e que aparece em todos os trabalhos é o canal AWGN, enquanto que outros canais também são citados como o Rayleigh, Rice e Log-Normal. A partir desse fato, decidiu-se implementar um gerador de ruídos cujo objetivo seria fornecer um sinal aditivo branco e gaussiano.

Cada conjunto de parâmetros foi definido e atribuído a uma variável representando o sinal, enquanto que o ruído implementado foi adicionado após a formação do sinal modulado. Todo esse processo foi executado para se montar uma base de dados que fosse suficientemente adequada ao problema.

3.2 EXTRAÇÃO DE CARACTERÍSTICAS

Em geral, quando se trata de classificação automática de modulação, há duas abordagens possíveis para atacar o problema. Uma baseada em funções de probabilidade e outra baseada em características definidas para reconhecimento de padrões. Para se trabalhar apropriadamente com a abordagem baseada em características é fundamental que a extração das características seja executada com sucesso. Para isso, é essencial que se escolha características relevantes para o estudo, de forma que possam auxiliar futuramente o modelo para tomar uma decisão mais correta possível. As características mais relevantes para o estudo foram separadas em cinco categorias: domínio instantâneo do tempo, domínio da transformada, estatísticas, formato da constelação e *zero-crossing* (HAZZA, 2013).

3.2.1 Características baseadas no domínio do tempo

As características baseadas no domínio do tempo estão relacionadas com os parâmetros fundamentais quando se trata de modulação, e representam todas as variações nos sinais modulados. As variações instantâneas da fase, frequência e amplitude da onda portadora são exploradas e usadas para discriminar as modulações.

A primeira característica utilizada, σ_{ap}, é o desvio padrão do valor absoluto da componente não linear da fase instantânea, e é dada pela equação (16).

$$\sigma_{ap} = \sqrt{\frac{1}{N_c}\left(\sum_{A_n[n]>A_t} \Phi_{NL}^2[n]\right) - \frac{1}{N_c}\left(\sum_{A_n[n]>A_t} \left|\Phi_{NL}[n]\right|\right)^2} \tag{16}$$

onde Nc é o número de amostras que satisfaz a condição $A_{n[n]} > A_t$. A variável A_t é um valor limiar que filtra as amostras do sinal que tem baixa amplitude devido à alta sensitividade ao

ruído. Enquanto que o termo Φ_{NL} representa a componente não linear da fase instantânea da n-ésima amostra do sinal.

A segunda característica, σ_{dp}, é o desvio padrão da componente não linear da fase instantânea direta, e é dada pela equação (17).

$$\sigma_{dp} = \sqrt{\frac{1}{N_c}\left(\sum_{A_n[n]>A_t} \Phi_{NL}^2[n]\right) - \frac{1}{N_c}\left(\sum_{A_n[n]>A_t} \Phi_{NL}[n]\right)^2} \tag{17}$$

cujo todos parâmetros são os mesmos expressos na equação ACIMA. Com a única diferença que o valor absoluto da componente da fase instantânea foi removido.

A terceira caraterística, σ_{aa}, é o desvio padrão do valor absoluto da amplitude instantânea centrada e normalizada, e é dada pela equação (18).

$$\sigma_{aa} = \sqrt{\frac{1}{N}\left(\sum_{n=1}^{N} A_{cn}^2[n]\right) - \frac{1}{N}\left(\sum_{n=1}^{N} \left|A_{cn}[n]\right|\right)^2} \tag{18}$$

onde A_{cn} é a amplitude instantânea centrada e normalizada e é dada através da equação (19).

$$A_{cn}[n] = A_n[n] - 1, \text{ onde } A_n[n] = \frac{A[n]}{\mu_A} \tag{19}$$

onde μ_A é a média da amplitude instantânea em um determinado segmento, e é dada através da equação (20).

$$\mu_A = \frac{1}{N}\sum_{n=1}^{N} a[n] \tag{20}$$

A normalização é utilizada para compensar a atenuação desconhecida do canal.

A quarta característica, σ_a, é o desvio padrão da amplitude centrada e normalizada e é dada pela equação (21).

$$\sigma_a = \sqrt{\frac{1}{N_c}\left(\sum_{n=1}^{N} a_{cn}^2[n]\right) - \frac{1}{N_c}\left(\sum_{n=1}^{N} a_{cn}[n]\right)^2} \tag{21}$$

A quinta caraterística, σ_{af}, é o desvio padrão do valor absoluto da frequência instantânea centrada e normalizada, e é dada pela equação (22).

$$\sigma_{af} = \sqrt{\frac{1}{N}\left(\sum_{A_n[n]>A_t} f_N^2[n]\right) - \frac{1}{N}\left(\sum_{A_n[n]>A_t} \left|f_N[n]\right|\right)^2} \tag{22}$$

onde f_N é a frequência normalizada e é dada pela equação (23).

$$f_N[n] = \frac{f_m[n]}{f_s} \tag{23}$$

sendo f_s a frequência de amostragem do sinal e $f_m[n]$ a frequência centrada dada através das equações (24) e (25).

$$f_m[n] = f\ [n] - \ \mu_f \tag{24}$$

$$\mu_f = \frac{1}{N}\sum_{n=1}^{N} f[n] \tag{25}$$

dessa forma, obtém-se a frequência centrada utilizando a frequência média μ_f.

As características apresentadas são, em geral, sensíveis a ruído. Uma vez que são definidas a partir de modificações nos parâmetros fundamentais de um sinal, fase, frequência e amplitude. Apesar de sua fácil implementação, devem ser usadas em ambientes com relação sinal ruído controlada para otimizar sua performance.

3.2.2 Características baseadas no domínio da transformada

Quando as características são definidas através do uso de uma transformada, define-se-ás como baseadas em tal domínio. As transformadas mais comuns para o estudo em AMC são a transformada wavelet e Fourier, mas também existem características baseadas na transformada S. Para o projeto, optou-se por usar apenas uma característica baseada na transformada de Fourier, de forma que pudesse agregar à classificação das modulações abordando conceitos já apresentados anteriormente.

O valor máximo da densidade espectral de potência da amplitude instantânea centrada e normalizada, definido por γ_{max}, é uma característica que auxilia na discriminação de modulações com alteração ou não da amplitude. Em um ambiente digital, tal característica permite a diferenciação de modulações ASK de modulações FSK ou PSK. E pode-se determinar tal *feature* através da equação (26).

$$\gamma_{max} = \frac{1}{N} max \left| DFT(A_{cn}) \right|^2 \quad (26)$$

onde DFT faz referência à transformada discreta de Fourier, A_{cn} é o valor da amplitude centrada e normalizada e N o número total de amostras avaliadas.

3.2.3 Discriminação das *features*

A escolha das características citadas no capítulo anterior é dada através de um estudo minucioso relacionado a aspectos chave dos sinais (AZZOUZ; NANDI, 1998). Deve-se verificar se o sinal transmitido possui informação de fase, frequência ou amplitude. Caso em um sinal não haja informação de fase, ele é considerado um sinal de amplitude. Nesta situação, a fase instantânea é dada a partir de uma função linear. Se da mesma forma em um sinal não houver informação de amplitude, ele é considerado um sinal de fase, e neste caso possui amplitude instantânea constante. É possível, também, que em um sinal haja informações tanto de fase quanto de amplitude, nesta situação, denomina-se por sinal combinado. Mas por motivos de complexidade, optou-se por não trabalhar com tais sinais.

A característica σ_{ap} será responsável por discriminar sinais que tenham informação absoluta de fase de sinais que não tenham. Os sinais modulados em 2ASK, 4ASK e 8ASK, por serem sinais de amplitude, não possuem informação absoluta de fase. Além disso, a fase de sinais modulados em 2PSK assume valores defasados de π graus, e após centralizar, seu valor absoluto será zero. Sendo assim, é importante mencionar que ao tirar o desvio padrão da componente não linear da fase instantânea, deve-se aplicar o módulo. Desta maneira, tem-se o primeiro subconjunto de discriminação da feature σ_{ap} que é dada por sinais os quais não tem informação absoluta de fase. Por outro lado, os sinais modulados em 4PSK e 8PSK, possuem informação acerca de sua fase instantânea e são definidos como o segundo subconjunto neste estudo.

A característica σ_{dp} será bastante semelhante à característica citada acima. Porém, desta vez, não se aplica o módulo após a centralização e normalização dos sinais. Ou seja, os sinais modulados em 2PSK apresentarão informação direta de fase, uma vez que sua fase instantânea assume valores de 0 e π (ou de $\frac{\pi}{2}$ e $\frac{3\pi}{2}$). Fazendo com que esse seja o primeiro subconjunto de discriminação relacionado a tal característica. Enquanto que o segundo

conjunto será dado por sinais de amplitude 2ASK, 4ASK e 8ASK que não possuem nenhuma informação de fase.

A característica σ_{aa} será responsável por discriminar sinais modulados em 2ASK dos sinais modulados em 4ASK e 8ASK. Nesta situação, todos os sinais apresentam informação na amplitude instantânea. Mas é importante ressaltar que os sinais 2ASK possuem amplitude instantânea variando entre dois níveis, que após a centralização e normalização serão iguais e de sinais opostos. Como a feature σ_{aa} aplica o desvio padrão onde o módulo do valor deve ser utilizado, conclui-se que 2ASK não possui informação absoluta de amplitude, enquanto que os sinais 4ASK e 8ASK possuem informação de amplitude absoluta e direta. Desta forma, tem-se dois subconjuntos que são importantes para a classificação.

A característica σ_{a} não avalia o valor absoluto quando se calcula o desvio padrão, logo, ela será importante para distinguir quaisquer sinais com informação na amplitude instantânea de sinais que não tenham informação. O primeiro subconjunto será dado por 2ASK, 4ASK e 8ASK, enquanto que as demais farão parte do segundo subconjunto.

A característica σ_{af} é útil para discriminar sinais que tenham informação na frequência instantânea. Sinais modulados em 2FSK tem sua frequência instantânea variando apenas entre dois níveis que são iguais em magnitude e com sinais opostos, e portanto o valor absoluto é constante, indicando que não há informação na frequência absoluta, enquanto que sinais modulados em 4FSK e 8FSK possuem informação na frequência direta e absoluta. Tornando assim, os dois subconjuntos discrimináveis.

A última característica utilizada neste trabalho γ_{max} é a única baseada no domínio do transformada, e é importante pois discrimina modulações em frequência das demais. Apesar de ser uma característica cujo principal parâmetro é a amplitude, essa característica consegue determinar que as modulações M-FSK e M-PSK não possuem informação em sua amplitude instantânea, implicando em uma densidade espectral de potência zero. Já os sinais 2ASK, 4ASK e 8ASK possuem informação em sua amplitude instantânea por definição. Conclui-se assim, que é uma feature essencial para discriminar modulações M-FSK e M-PSK como primeiro subconjunto, de modulações M-ASK como segundo subconjunto.

3.3 CLASSIFICADORES

A etapa de classificação é essencial no ambiente de aprendizado de máquina. Após a coleta de características convenientes para o estudo, é necessário usá-las para treinar algoritmos que executarão a tomada de decisão referente a qual classe o objeto pertence.

Quando se trata da abordagem por características, tal tomada de decisão é dada de duas maneiras. A primeira consiste em uma árvore de decisões que funciona a partir da criação de vários pontos de decisão denominados nós. O algoritmo define um valor limite para cada nó, e funciona fazendo uma comparação da feature calculada com o limiar. Desta maneira, toma-se a decisão por um dos dois ramos possíveis. É um processo de simples implementação que pode ter a agregação de novas categorias de classificação através de adições de novos ramos. Outra maneira de se trabalhar o problema é através de algoritmos baseados em reconhecimento de padrões, como SVM, que fornece uma generalização maior através da procura da máxima separação entre hiperplanos; ou como KNN, que faz a análise de um número determinado de vizinhos de um certo objeto, e classifica-o de acordo com a maioria.

3.3.1 Ajuste dos Hiperparâmetros dos modelos

Para se ter mais eficiência durante o aprendizado, cada classificador deve ser parametrizado de acordo com suas peculiaridades. A configuração dos parâmetros do sistema, definidos aqui como hiperparâmetros, é ajustada pelo projetista e altera aspectos do algoritmo de aprendizagem apresentando efeitos sobre o modelo final e seu desempenho (CLAESEN; DE MOOR, 2015). Usualmente, passa-se os hiperparâmetros como argumento para o construtor das classes, justamente por não serem parâmetros aprendidos diretamente nos estimadores.

A definição e escolha dos hiperparâmetros é uma tarefa geralmente feita manualmente ou através de tentativas e testes de conjuntos de parâmetros. O objetivo é avaliar o desempenho de cada um dos hiperparâmetros presentes no conjunto definido quando aplicados ao modelo de treinamento. É evidente que o problema exige um alto custo computacional e se torna impraticável quando demanda um grande conjunto de parâmetros para análise.

Para o atual projeto, aplicou-se o método de pesquisa de grade (do inglês *gridsearch*) para se fazer o ajuste. Esse método consiste em avaliar vários valores de interesse para um determinado hiperparâmetro do projeto, e identificar metodicamente a acurácia para cada combinação de parâmetros do algoritmo especificados na grade perante a base de dados de

estudo. Quando se trata do modelo KNN, o valor do número de vizinhos é um parâmetro essencial para se verificar o desempenho e a robustez do modelo, visto que para cada conjunto de dados, obtém-se um *k* ideal que traz o melhor resultado. A performance foi verificada quando se avaliou a resposta do modelo quando analisado apenas um vizinho da amostra de interesse até uma quantidade de vizinhos igual a 1% do total de amostras presentes na base de dados, o que claramente torna a natureza do algoritmo em relação a complexidade computacional.

Em relação ao modelo SVM, existem parâmetros que auxiliam os hiperplanos a serem posicionados de forma a darem uma ótima resposta, ou seja, que não classifiquem tão corretamente um conjunto de treinamento, mas que quando aplicados a novos conjuntos não apresentem um resultado satisfatório. O hiperparâmetro *C* é responsável por adicionar uma penalidade para cada amostra classificada incorretamente. Ou seja, quanto menor o valor de *C*, menor a cobrança aplicada para amostras incorretas e o hiperplano é capaz de ter uma margem maior, permitindo um maior número de classificações erradas. É importante mencionar que a penalidade é proporcional à distância do dado ao hiperplano.

Algumas vezes, usa-se uma função de Kernel para separar pontos que não são linearmente separáveis, obtendo-se, assim, uma medida de similaridade entre os pontos. Os valores de entrada da função são as amostras iniciais e o valor de saída é essa medida de similaridade entre elas. Dessa forma, o hiperparâmetro *gamma* é capaz de medir a influência de uma amostra individual de treinamento em relação ao espaço. Altos valores de *gamma* indicam que os dados precisam estar altamente agrupados para serem considerados do mesmo grupo. As funções de Kernel também podem ser tratadas como hiperparâmetros, uma vez que indicam de que maneira essa transformação de vetores será feita.

No caso de algoritmos de floresta aleatória, hiperparâmetros configurados pelo projetista previamente ao treinamento são o número de árvores de decisão e o número de *features* consideradas em cada nó.

Quando é executada a validação cruzada, deve-se dividir o subconjunto de treinamento em *k* partes e avaliar o desempenho do modelo quando aplicado cada hiperparâmetro para cada uma das partes. Percebe-se que o número de iterações performadas pelo modelo será tão alta quanto mais hiperparâmetros e quanto mais partes se dividir o subconjunto, elevando consideravelmente a complexidade do processo.

4 DESENVOLVIMENTO E RESULTADOS

O capítulo a seguir tem por objetivo detalhar as três etapas da implementação do projeto e indicar os resultados obtidos através de tais processos. Para isso, aplicar-se-á a metodologia apresentada com o objetivo de percorrer o desenvolvimento do problema de classificação automática de modulação. Os dados provenientes através do desenvolvimento de cada etapa serão apresentados como resultados parciais e serão avaliados nas etapas posteriores.

4.1 ARQUITETURA DOS EXPERIMENTOS

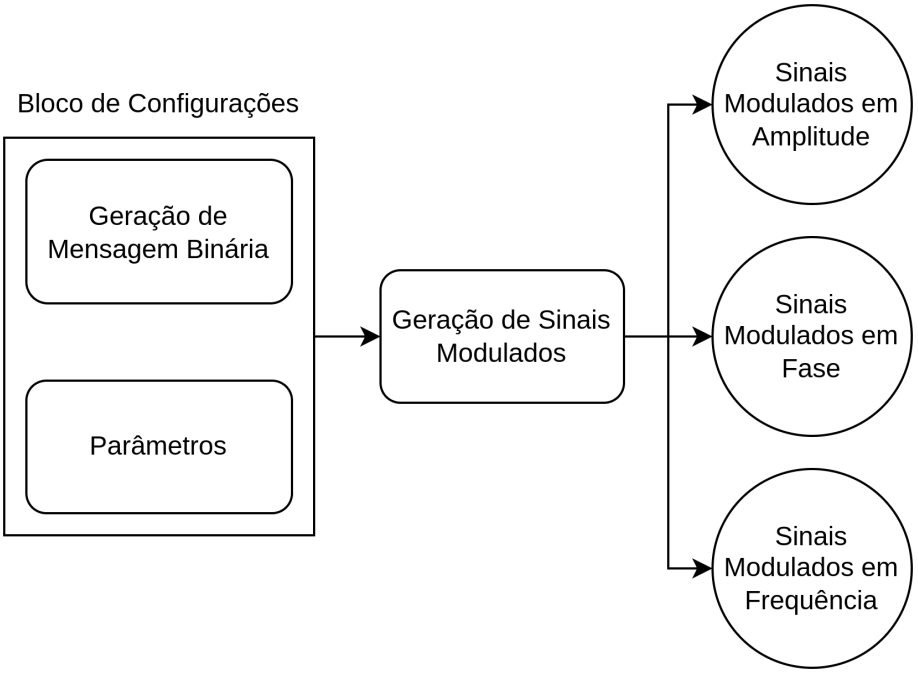
A implementação do projeto foi dada, desde a etapa da geração da base de dados até a análise dos classificadores, através de simulação computacional em *Python* utilizando o *Jupyter Notebook* em ambiente *Linux*, *Fedora* 34 versão 5.17.4.

4.2 GERAÇÃO DA BASE DE DADOS

A primeira etapa do projeto consiste no desenvolvimento de um bloco de geração de sinais cuja entrada é um conjunto de parâmetros determinados pelo projetista, de forma que se obtivesse uma base de dados bastante grande e variada, enquanto que a saída é um conjunto de sinais modulados sob os aspectos determinados e prontos para serem analisados nas etapas posteriores. Uma vez que o propósito inicial do trabalho é analisar o comportamento de diferentes classificadores quando abordados sob a perspectiva de modulações digitais, fugiu-se levemente do contexto real de comunicação para se diminuir a complexidade computacional do projeto, mas que permitisse a obtenção de resultados coerentes e entregasse uma visualização palpável dos dados. Portanto, escolheu-se métodos e parâmetros mais simples que permitissem o desenvolvimento de um algoritmo gerador de baixo custo.

Para cada modelo de formação do sinal, usou-se um bloco de configurações capaz de dispor uma mensagem digital e aleatorizar os parâmetros com o objetivo de se gerar sinais distintos pertencentes a cada uma das três modulações fundamentais apresentadas neste projeto, e organizados de acordo com as ordens 2, 4 e 8, conforme apresentado na figura 19.

Figura 19 - Esquema de estruturação da base de dados



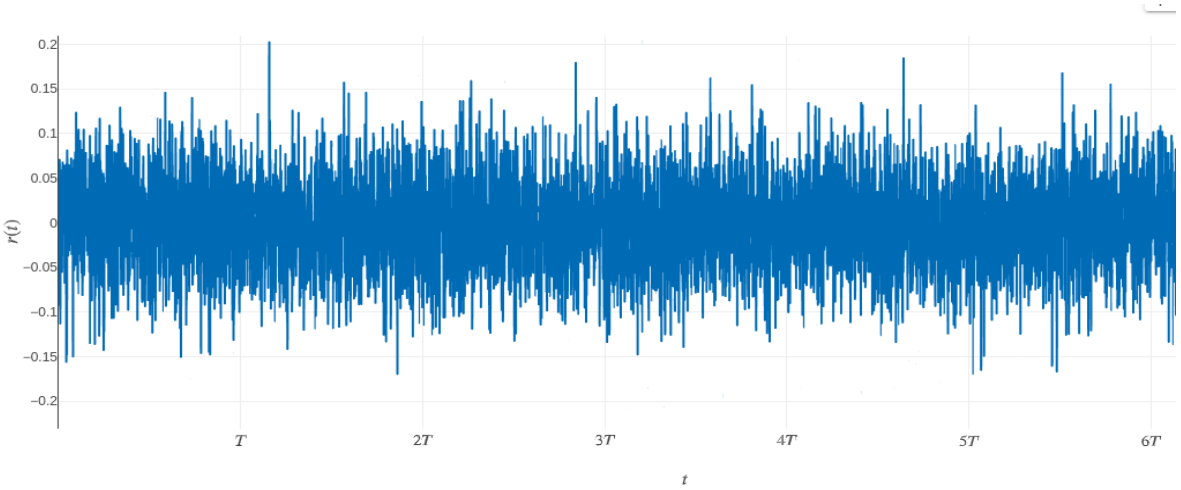
Fonte: Elaborado pelo autor (2022).

O primeiro passo para a geração da base de dados exige, inicialmente, projetar uma fonte geradora de mensagens binárias aleatórias, cuja única função seria entregar uma mensagem com um número aleatório de *bits*, e que no futuro, seriam agrupados conforme a ordem desejada pelo projetista. O algoritmo foi construído dentro do código, de forma que dispusesse de 20 a 40 *bits* individualmente, os quais seriam posteriormente agrupados conforme a ordem aleatorizada pelo próprio algoritmo. Para casos de ordem superior a 2, os agrupamentos são feitos em pares ou trios de bits. Isso gerou problemas quando o número de *bits* não era divisível justamente pela forma em que os *bits* estavam sendo transformados em símbolos. Para resolver este empecilho, foi decidido, pelo projetista, que se complementaria a mensagem com *bits* nulos no final, de forma que se tornasse possível a divisão da mensagem em grupos de *bits* sugeridos pela ordem.

O bloco de configurações também tem influência sob o bloco decisor de parâmetros, cuja função é selecionar todos os parâmetros relevantes para que se tenha um sinal modulado de forma digital. Foi possível tornar aleatória a escolha de alguns parâmetros como tipo e ordem de modulação utilizadas para cada sinal que compõe a base de dados, e selecionar exatamente o quão grande ela seria em conformidade com a literatura apresentada neste trabalho. Nesta etapa, também se definiu o canal de comunicação e o ruído referente que

afetariam o sinal. Implementou-se, de acordo com a literatura, um canal AWGN que pudesse modelar efeitos aleatórios com a mesma intensidade em toda a faixa de frequências. Para tal, propôs-se um modelo gaussiano que tivesse distribuição de 0.05 e média 0 e apontou-se um valor de razão sinal ruído para cada um dos sinais propostos no projeto. Um exemplo de ruído obtido a partir dessa parametrização é apresentado na figura 20 e eventualmente acrescido ao sinal modulado.

Figura 20 - Exemplo de sinal ruidoso gerado a partir de um canal AWGN



Fonte: Elaborado pelo autor (2022).

Desta forma, construiu-se uma base contendo 3000 amostras referentes aos três tipos de modulação, totalizando uma base de dados com 9000 amostras. Demais parâmetros relevantes para a construção de um sinal também foram ajustados aqui neste bloco. A frequência da onda portadora foi ajustada para 100 Hz, enquanto que o número de amostras do sinal em onda passante foi de 1000. Uma modelagem muito importante para a implementação do projeto foi a escolha da taxa de transmissão de símbolos. Percebeu-se que era possível ter um alto valor para tal parâmetro, entretanto, com o intuito de se ter uma melhor visualização e percepção dos testes feitos, decidiu-se finalmente por enviar um símbolo a cada ciclo de portadora, ou seja, 1 Hz, desta maneira seria possível identificar e acompanhar melhor se os sinais estavam funcionando de acordo com o planejado.

Esses valores fogem um pouco do escopo de funcionamento de sinais reais, porém, como dito anteriormente, optou-se por uma complexidade computacional reduzida para evitar análises muito demoradas, mas que ainda servissem ao objetivo de se estudar a classificação de modulação.

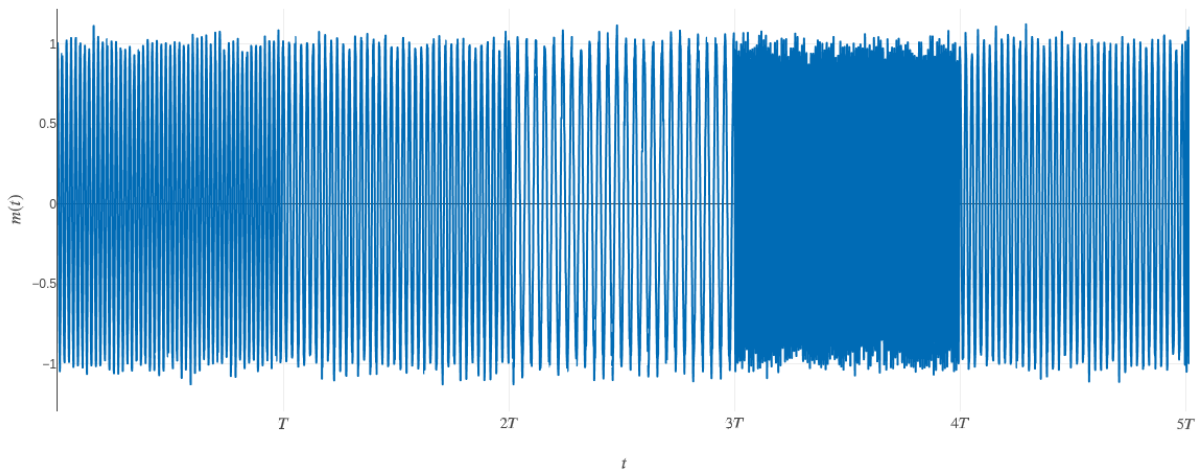
Tabela 1 - Parâmetros de geração dos sinais avaliados dentro do bloco de configurações

Parâmetro	**Valor**
Número de amostras	9000
Tamanho da mensagem	Entre 20 e 40 bits
Ordem	2, 4 ou 8
Frequência da portadora	100 Hz
Taxa de amostragem	1000 Hz

Fonte: Elaborado pelo autor (2022).

Desta maneira, obteve-se sinais modulados que foram representados através de variáveis expressas por *arrays*, ou seja, uma estrutura de dados contendo um conjunto de elementos. Cada elemento deste sinal no tempo é dado através da amostragem que foi feita para gerar o sinal. As figuras 21, 22 e 23 representam exemplos de sinais projetados em frequência, amplitude e fase, respectivamente, quando usado o bloco de geração de sinais modulados. Os parâmetros utilizados para construir os gráficos apresentados nas figuras mencionadas foram expostos nas tabelas 2, 3 e 4, respectivamente, onde pode se identificar o tipo de modulação e ordem utilizados, a mensagem que foi enviada, e a razão da potência do sinal em relação à potência do ruído, expressa pela SNR.

Figura 21 - Sinal modulado em frequência usando o gerador de sinais projetado



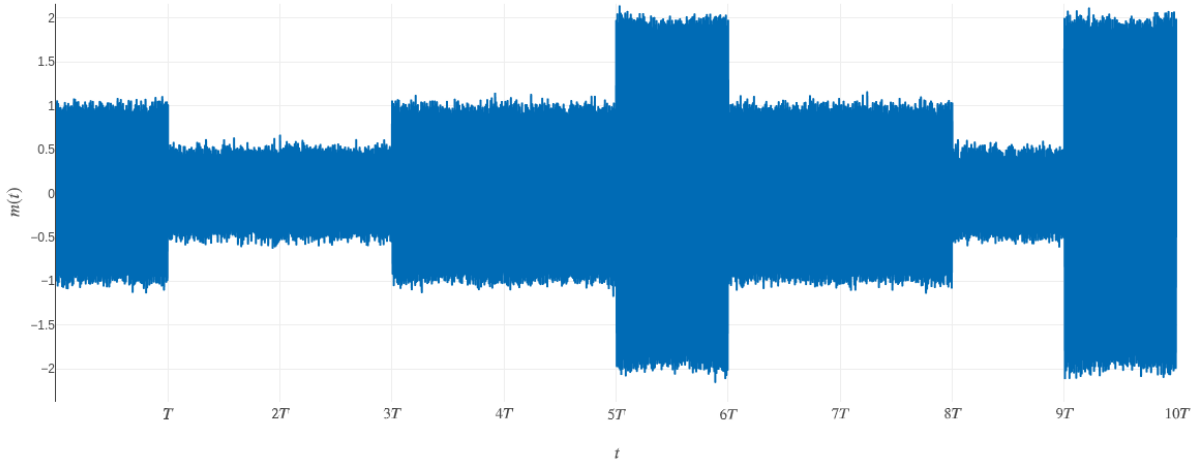
Fonte: Elaborado pelo autor (2022).

Tabela 2 - Parâmetros utilizados para construir o sinal da figura 19

Parâmetro	**Valor**
Tipo de modulação	Frequência
Ordem	8
Mensagem	01101000111101011101010111000
SNR	23 dB

Fonte: Elaborado pelo autor (2022).

Figura 22 - Sinal modulado em amplitude usando o gerador de sinais projetado



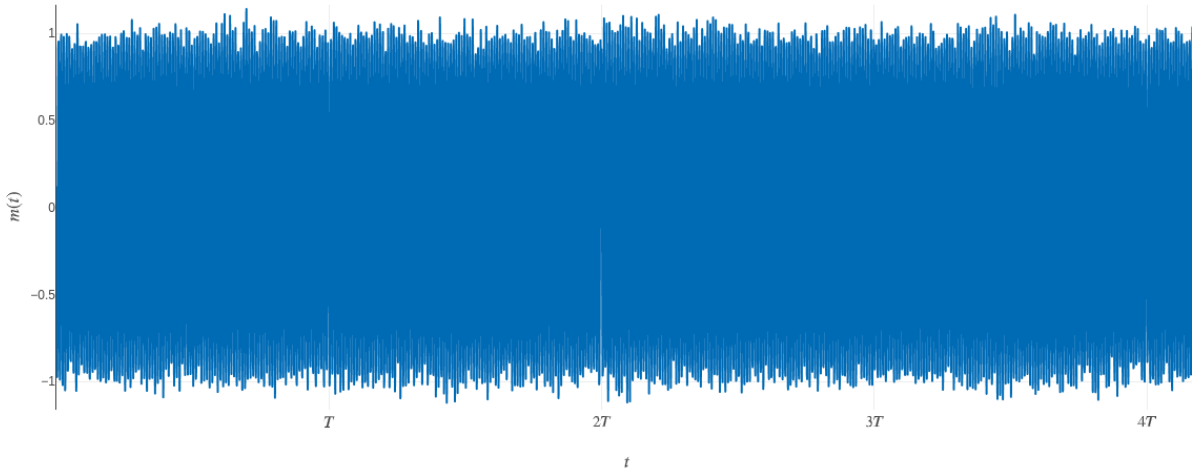
Fonte: Elaborado pelo autor (2022).

Tabela 3 - Parâmetros utilizados para construir o sinal da figura 20

Parâmetro	**Valor**
Tipo de modulação	Amplitude
Ordem	4
Mensagem	0100000101110101001100 10
SNR	24 dB

Fonte: Elaborado pelo autor (2022).

Figura 23 - Sinal modulado em fase usando o gerador de sinais projetado



Fonte: Elaborado pelo autor (2022).

Tabela 4 - Parâmetros utilizados para construir o sinal da figura 21

Parâmetro	**Valor**
Tipo de modulação	Fase
Ordem	2
Mensagem	110111011110111101110 1 000010101010
SNR	23 dB

Fonte: Elaborado pelo autor (2022).

Como mencionado anteriormente, a saída do bloco de geração de sinais será um conjunto de sinais expressos por variáveis, cada qual contendo tantas amostras quanto determinadas pela taxa de amostragem. Tais valores serão armazenados em uma outra variável denominada *base_de_dados* que indicará um dicionário composto de chaves, afirmadas pelos sinais de saída e também parâmetros geradores dos sinais, cada um em correspondência com seu valor numérico, ou conjunto de valores.

Conclui-se, portanto, que a base de dados apresentará um conjunto de estudo bastante variado, uma vez que foi possível aleatorizar alguns parâmetros responsáveis pela geração dos sinais. A proposta de se trabalhar com ordens diferentes teve o intuito de tornar o projeto mais completo e por isso construiu-se uma base com ordens 2, 4 e 8. Foi possível, também, obter mensagens binárias digitais de variados tamanhos, mas sempre se atentando a não fazer com que o projeto ficasse muito complexo computacionalmente.

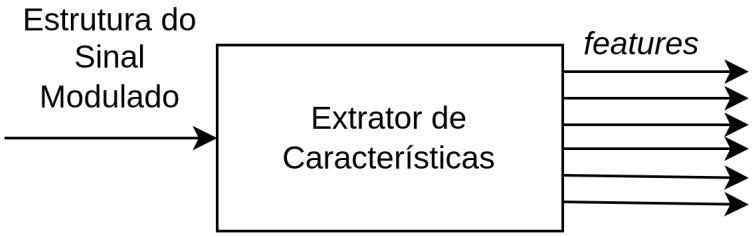
Desta forma, tem-se uma base de dados com 9000 amostras, sendo 3000 correspondente a cada uma das três modulações aqui mencionadas, o que é um número interessante quando comparado com as referências bibliográficas apresentadas.

4.3 IMPLEMENTAÇÃO DO BLOCO EXTRATOR DE CARACTERÍSTICAS

A segunda grande etapa do projeto de classificação automática de modulação é dada através de um extrator de características chaves para o problema, que terá por função aplicar testes dentro do banco de dados e responderá com *features* representativas para cada sinal.

O projeto do extrator de características funciona com o objetivo de converter uma estrutura de símbolos representadas como uma *array* como entrada, que indicam o sinal, em um conjunto de saída com *n* escalares, sendo *n* o número de características extraídas, como indicado na figura 24. Eventualmente, esse conjunto será a entrada do classificador.

Figura 24 - Representação do funcionamento do bloco extrator de características



Fonte: Elaborado pelo autor (2022).

O sinal modulado proveniente da geração de sinais será a entrada do bloco extrator de características. É importante que o sinal esteja estruturado conforme o planejado para que os cálculos de extração sejam efetuados com sucesso. Conforme citado anteriormente, o sinal estará disposto de acordo com um conjunto de valores, tantos quanto quão grande for a taxa de amostragem e será um dos parâmetros avaliados pelo bloco. Demais parâmetros presentes nas variáveis definidas pela base de dados, como a taxa de amostragem, também serão úteis e necessários para que seja possível a extração das características definidas que serão utilizadas para a classificação.

Para o atual projeto, trabalhou-se com cinco características distintas selecionadas nas categorias de domínio instantâneo do tempo, que proporcionam a análise de propriedades espectrais únicas baseadas em cálculos de desvio padrão de diferentes componentes dos

sinais como amplitude, fase e frequência do sinal. E com uma característica apontada na categoria de domínio da transformada.

A implementação das características foi dada através de uma classe contendo métodos próprios cujo objetivo seria receber como entrada a variável que representa cada sinal, e fazer cálculos definidos que analisassem amplitude, fase e frequência instantânea já que são os parâmetros que representam todas variações nos sinais, e também cálculos para extrair a transformada de Fourier e alguns processamentos, como normalização do sinal.

O resultado proveniente do bloco de extração de características foi agrupado à variável correspondente ao sinal que deu origem às *features*. Desta forma, tem-se uma lista de dicionários bem organizada e com todos os valores adquiridos até então, conforme pode ser visto alguns exemplos nas tabelas 5, 6 e 7.

Tabela 5 - Parâmetros e *features* correspondentes a um sinal de frequência

Parâmetros e *features*	**Valor**
Tipo de modulação	Frequência
Ordem	2
Mensagem	111010011111001110111001
Frequência da portadora	100 Hz
Taxa de amostragem	1000 Hz
γ_{max}	0.6965
σ_{a}	0.2083
σ_{aa}	0.1265
σ_{af}	0.0115
σ_{ap}	3926.55
σ_{dp}	3926.55

Fonte: Elaborado pelo autor (2022).

Tabela 6 - Parâmetros e *features* correspondentes a um sinal de amplitude

Parâmetros e *features*	**Valor**
Tipo de modulação	Amplitude
Ordem	8
Mensagem	10100101011111100101001
Frequência da portadora	100 Hz
Taxa de amostragem	1000 Hz
γ_{max}	537.08
σ_{a}	0.5202
σ_{aa}	0.2002
σ_{af}	0.0091
σ_{ap}	1451.06
σ_{dp}	1451.06

Fonte: Elaborado pelo autor (2022).

Tabela 7 - Parâmetros e *features* correspondentes a um sinal de fase

Parâmetros e *features*	**Valor**
Tipo de modulação	Fase
Ordem	8
Mensagem	00010100110100011000101010111111101
Frequência da portadora	100 Hz
Taxa de amostragem	1000 Hz
γ_{max}	0.0741
σ_{a}	0.0594

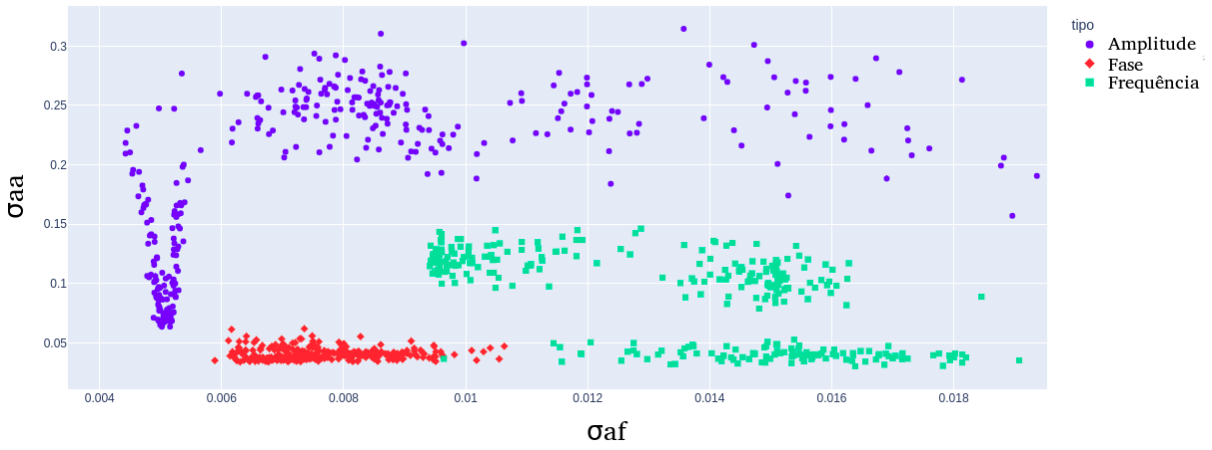
σ_{aa}	0.0383
σ_{af}	0.0090
σ_{ap}	2179.54
σ_{dp}	2179.54

Fonte: Elaborado pelo autor (2022).

Conforme pode ser verificado nas tabelas, cada sinal adquirido por meio dos parâmetros e mensagem indicados, irá trazer consigo um conjunto de características determinadas por meio das formulações apresentadas por AZZOUZ; NANDI, (1998).

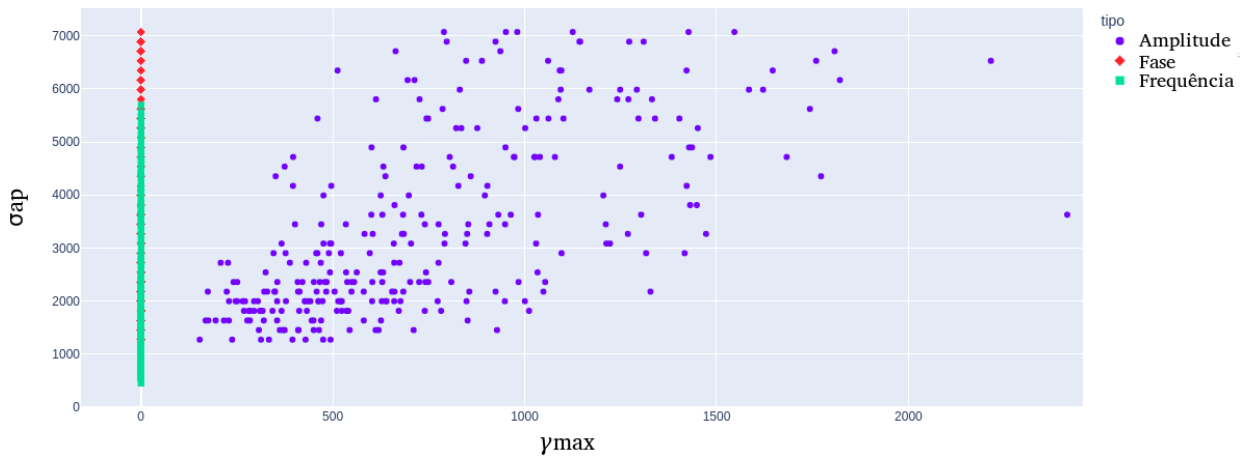
Para uma melhor visualização dos dados adquiridos até o presente momento, decidiu-se plotar alguns dados quando atribuídos à features em um espaço N-dimensional, sendo N o número de *features* abordadas. Entretanto, como foi trabalhado com 6 parâmetros, decidiu-se trabalhar com a análise visual dos dados usando três gráficos distintos, cada um contendo duas *features* apresentadas no projeto. Análises mais profundas e detalhadas serão feitas no capítulo posterior.

Figura 25 - Distribuição dos sinais modulados em relação às features σ_{aa} e σ_{af}



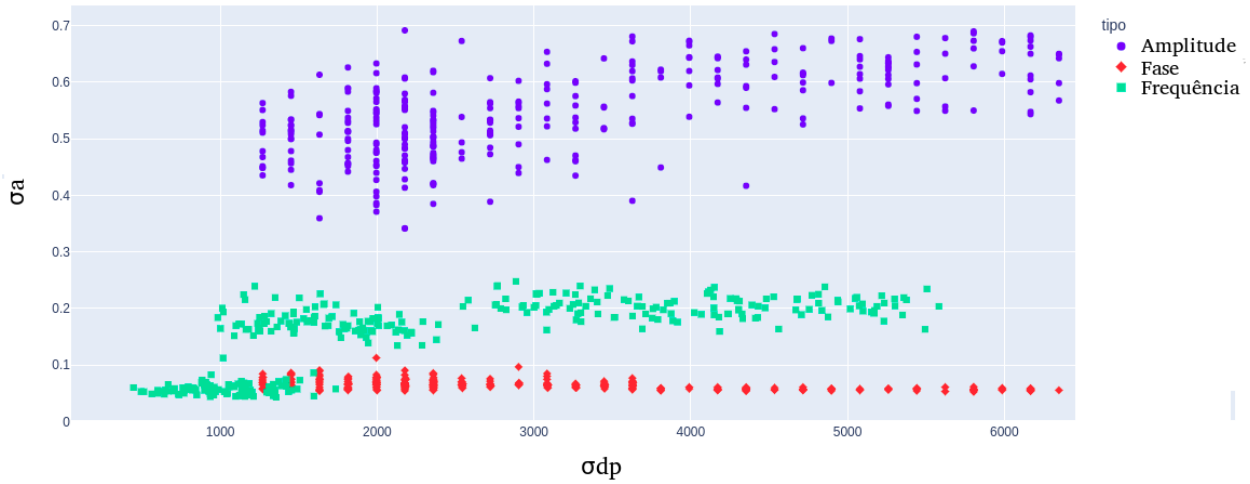
Fonte: Elaborado pelo autor (2022).

Figura 26 - Distribuição dos sinais modulados em relação às *features* σ_{ap} e γ_{max}



Fonte: Elaborado pelo autor (2022).

Figura 27 - Distribuição dos sinais modulados em relação às features σ_a e σ_{dp}



Fonte: Elaborado pelo autor (2022).

Como pode ser interpretado na figura 25, 26 e 27, realizou-se a plotagem de um conjunto de amostras quando avaliados sob um par de *features*. Entende-se através da legenda que os sinais de amplitude são representados pela bola azul, os sinais de fase pelo losango vermelho, e por último, os sinais de frequência pelo quadrado verde.

Efetuando-se uma abordagem e uma classificação mais humana, compreende-se que é possível diferenciar algumas modulações através do gráfico. Dependendo do posicionamento de novos sinais cujas modulações são desconhecidas, decide-se visualmente por qual tipo de modulação esse sinal se refere. Entretanto, essa tarefa se torna extremamente mais precisa e correta quando feita através de aprendizado de máquina, uma vez que se tem uma grande

base de dados para se comparar. Vale mencionar, que para uma interpretação mais correta, todas as *features* devem ser avaliadas em conjunto, e não separadas em pequenos grupos como feito nas figuras acima. Este modelo só foi apresentado dessa maneira para se ter uma visualização e compreendimento maior da base de dados.

4.4 IMPLEMENTAÇÃO DO BLOCO DE CLASSIFICAÇÃO

Diversas ferramentas foram apresentadas no projeto até o presente momento com o intuito de poder chegar à etapa de classificação. Interessamo-nos aqui em analisar a performance de cada classificador quando aplicados os dados adquiridos nas etapas anteriores.

Para que um classificador tenha um desempenho suficientemente bom, é fundamental que haja alguns requisitos básicos (ZHU; NANDI, 2015). Primeiramente, é necessário que o classificador projetado seja capaz de avaliar tantas modulações quanto possível, já que se deseja usá-lo em um grande número de aplicações. Neste projeto, como já foi mencionado, trabalhou-se com um conjunto pequeno de tipos de modulação, uma vez que o intuito principal era implementar e analisar o processo de execução de um classificador automático por completo, desde o início até a classificação de fato. Além disso, é essencial que o classificador projetado responda com alta precisão em relação às suas decisões e que seja robusto diante de diversas adversidades e condições apresentadas pelo canal. Por último, o projeto precisa ser computacionalmente eficiente, uma vez que em muitos casos é necessário que a tomada de decisão seja feita de maneira rápida.

O primeiro passo para se aplicar os classificadores é determinar o tamanho do conjunto de treinamento e conjunto de testes para se ter uma validação da base de dados utilizada. Para o atual trabalho, utilizou-se o método *holdout* apresentado na seção 2.2.6, o qual particiona o conjunto total de dados sendo ⅓ como conjunto de teste, enquanto ⅔ são o conjunto de treinamento. Ou seja, para todas as classificações efetuadas, 6000 dados foram escolhidos aleatoriamente para treinar o sistema, enquanto que os demais 3000 foram usados para se fazer a validação.

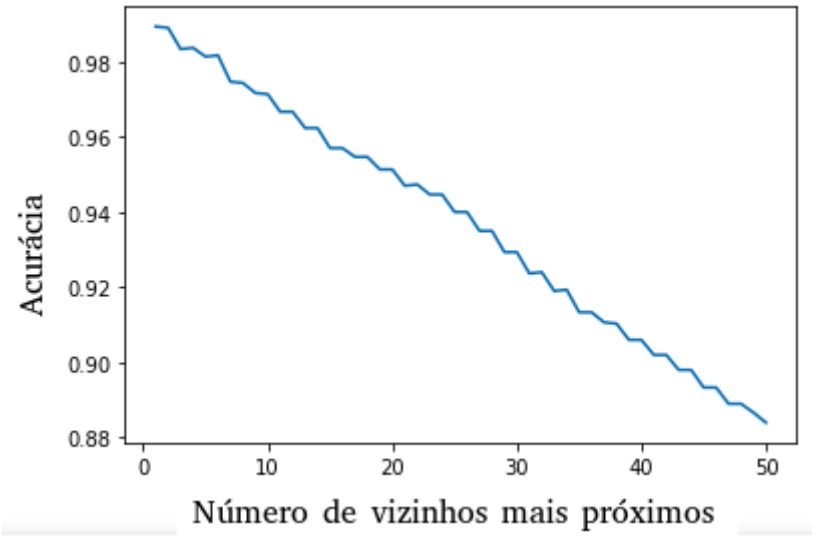
4.4.1 *K-Nearest Neighbor* (KNN)

O primeiro classificador a ser avaliado no projeto foi o *K-Nearest Neighbor* por ser um dos mais simples no ambiente de aprendizado supervisionado e por se encaixar muito bem ao problema de classificação. Tem-se por objetivo determinar o quão semelhante um sinal desconhecido é da base de dados projetada. Apesar de não termos uma representação perfeita do espaço de *features*, quando se analisa as figuras 25, 26 e 27 friamente, já é possível ter uma noção de que cada conjunto de modulações irá ocupar uma determinada área no espaço N-dimensional de *features*.

A análise da semelhança é feita pela distância da amostra desconhecida em relação aos seus vizinhos mais próximos. O número de vizinhos a serem considerados é identificado pela variável k, e também é a grande chave da resolução do problema. A decisão de tal parâmetro depende do quão grande a base de dados é, e deve seguir algumas regras. Em uma primeira proposta, definiu-se tal valor empiricamente com o objetivo de verificar a qualidade do projeto. Portanto, considerou-se os 15 vizinhos mais próximos de cada nova amostra desconhecida. Sob avaliação das 3000 amostras do conjunto de testes, quando comparados em relação ao conjunto de treinamento, verificando o valor citado de vizinhos mais próximos, obteve-se o resultado correto em 2871 vezes, ou seja, uma porcentagem de 95,70%.

Com o propósito de se identificar o número ideal de vizinhos para a determinação de um resultado ótimo, aplicou-se o classificador em cima da mesma base de dados mas variando-se o valor de k, e com isso, obteve-se o resultado apresentado na figura 28.

Figura 28 - Representação da acurácia do modelo em relação ao número de vizinhos mais próximos



Fonte: Elaborado pelo autor (2022).

Percebe-se que o classificador tende a cometer mais erros quanto maior for o número de vizinhos. Isso se deve ao fato de que o espaço de amostras se torna mais complexo com o aumento da dimensionalidade, por consequência do número de *features*. Quando mais vizinhos são considerados, acaba-se confundindo com demais regiões de outros tipos de modulação. Quando considerado apenas um vizinho mais próximo, o modelo apresentou o resultado correto da modulação esperada em 2968 das vezes, ou seja, uma porcentagem de 98,93%. A matriz de confusão para tal caso e a análise das demais métricas de avaliação foram apresentadas na tabela 8 e 9, respectivamente.

Tabela 8 - Matriz de confusão

	Amplitude	**Frequência**	**Fase**
Amplitude	973	0	0
Frequência	0	1022	24
Fase	0	8	973

Fonte: Elaborado pelo autor (2022).

Tabela 9 - Métricas de avaliação

	Precisão	***Recall***	**F1 *Score***
Amplitude	1.00	1.00	1.00
Frequência	0.99	0.98	0.98
Fase	0.98	0.99	0.98

Fonte: Elaborado pelo autor (2022).

4.4.2. *Support Vector Machine* (SVM)

O classificador *Support Vector Machine* funciona com a distinção das classes através da sustentação de um hiperplano que terá o objetivo de separar os dados de classes distintas. Embora seja melhor empregado em classificações de duas classes, optou-se por trabalhar com tal técnica uma vez que se esperava conhecer um pouco mais sobre o tema.

Para determinar a classificação, usou-se a biblioteca *sklearn* presente no *Python* através do método que já continha o classificador. Apesar de que as funções de *Kernel* podem ser utilizadas como hiperparâmetros, o treinamento foi dado através de um único *Kernel* considerado padrão chamado Função de Base Radial (do inglês *Radial Basis Function*, ou RBF). Isso quer dizer que os valores utilizados aqui terão como referência o centro do gráfico, que nesse caso é representado pela origem.

Demais parâmetros foram avaliados sob a perspectiva de buscar resultados com maior acurácia. O hiperparâmetro C funciona como uma espécie de regularizador do classificador. Quanto menor seu valor, menor a exigência para se ter resultados corretos devido a uma função mais simples, o que dá uma margem maior aos hiperplanos e permite classificações erradas. Enquanto que quanto maior seu valor, tem-se um funcionamento de maneira oposta, a exigência para resultados corretos é maior, o que tem por consequência uma função mais complexa, os hiperplanos tem margens menores e classificam os dados de maneira mais precisa. Percebe-se que há um *trade-off* em relação ao valor do hiperparâmetro C.

O hiperparâmetro *gamma* também é avaliado diante deste classificador, uma vez que mede a influência de cada dado de treinamento em relação ao espaço. Valores altos significam que para considerar dados do mesmo grupo, eles precisam estar bastante agrupados.

Para uma primeira ótica, realizou-se a classificação diante de todos os valores padrões apresentados pela biblioteca de acordo com a tabela 10, onde a coluna à esquerda representa os hiperparâmetros calculados, enquanto que a coluna à direita representa seus respectivos valores.

Tabela 10 - Hiperparâmetros padrões

Hiperparâmetro	**Valor**
Kernel	Função de Base Radial
C	1.0
gamma	0.166

Fonte: Elaborado pelo autor (2022).

Efetuou-se, portanto, a predição do método perante o grupo definido como conjunto de testes contendo 66,67% da base de dados, o qual é composto por chaves, representadas aqui pelos sinais, e seus valores, representados para o projeto como o tipo de modulação. A

partir disso, obteve-se um conjunto de predição, que quando comparado com o conjunto de treinamento, respondeu com uma acurácia de 79,77%. Na tabela 11, verifica-se a distribuição das amostras decididas pelo modelo quando comparadas ao resultado esperado representado pelo conjunto de treinamento, ou seja, a matriz de confusão. Desta forma, tem-se os 3000 sinais indicados em como foram classificados, e em como deveriam ter sido classificados. Para uma melhor verificação dos dados, calculou-se as métricas de avaliação determinadas no trabalho e representou-se-as na tabela 12.

Tabela 11 - Matriz de confusão

	Amplitude	**Frequência**	**Fase**
Amplitude	969	1	3
Frequência	0	653	393
Fase	0	210	771

Fonte: Elaborado pelo autor (2022).

Tabela 12 - Métricas de avaliação

	Precisão	***Recall***	**F1 *Score***
Amplitude	1.00	1.00	1.00
Frequência	0.76	0.62	0.68
Fase	0.66	0.79	0.72

Fonte: Elaborado pelo autor (2022).

O baixo valor mediante ambiente e condições totalmente controladas, obrigou o projetista a buscar resultados melhores. Para tal, usou-se a função de pesquisa de grade da própria biblioteca com o foco de se procurar melhores hiperparâmetros que trouxessem um resultado de maior acurácia. Procurou-se dentro de um grupo definido de valores, qual hiperparâmetro referente ao classificador SVM entregaria um resultado mais satisfatório para o problema. Para isso, foi utilizado o método de validação cruzada *k-fold*, o qual divide o conjunto de treinamento em k subconjuntos de mesmo tamanho. Portanto, com um k=5, dividiu-se tal modelo em 5 menores, e aplicou-se em cada um deles as variadas possibilidades de hiperparâmetros. Desta maneira, obteve-se os valores ótimos apresentados na tabela 13, que entregassem resultados com métricas de avaliação melhores.

Tabela 13 - Hiperparâmetros ideais adquiridos após validação cruzada

Hiperparâmetro	**Valor**
Kernel	Função de Base Radial
C	100
gamma	0.001

Fonte: Elaborado pelo autor (2022).

Após aquisição dos hiperparâmetros considerados ótimos para classificação, foi possível efetuar um novo cálculo considerando o mesmo tamanho de conjuntos de treinamento e teste. Os resultados da matriz de confusão e métricas de avaliação são indicados nas tabelas 14 e 15, respectivamente.

Tabela 14 - Matriz de confusão após classificação sob hiperparâmetros ótimos

	Amplitude	**Frequência**	**Fase**
Amplitude	973	0	0
Frequência	1	1002	43
Fase	0	1	980

Fonte: Elaborado pelo autor (2022).

Tabela 15 - Métricas de avaliação após classificação sob hiperparâmetros ótimos

	Precisão	***Recall***	**F1 *Score***
Amplitude	1.00	1.00	1.00
Frequência	1.00	0.96	0.98
Fase	0.96	1.00	0.98

Fonte: Elaborado pelo autor (2022).

Percebe-se que 3000 sinais foram utilizados como conjunto de teste para o cálculo, visto que 33,33% foi o valor determinado para se fazer a validação do modelo. Entretanto, isso não significa que cada um dos três tipos de modulação será representado por 1000 sinais, uma vez que a decisão do conjunto total é feito de forma aleatória, sem saber quais são os verdadeiros rótulos de cada uma das amostras.

A acurácia do classificador subiu para 98% perante novos hiperparâmetros adquiridos através da validação cruzada e cálculos usando a função de *grid search* presente na biblioteca *sklearn*. As demais métricas de avaliação também apresentaram resultados bastante próximos de 100%.

4.4.3 Floresta Aleatória

O último classificador utilizado no projeto é a combinação de um conjunto de árvores de decisão construídas a partir de diferentes aspectos. Árvores de decisão consistem de nós de partida chamados de raiz, seguidos por ramificações de passos condicionais ou decisórios que contém análises de limiares envolvendo as *features* selecionadas. Costuma-se referir a esta união de árvores como floresta aleatória. O propósito do classificador citado é combinar modelos fracos de forma que se tenha como resultado um modelo mais forte.

Para executar a classificação, utilizou-se da classe *RandomForestClassifier* presente na biblioteca *sklearn*. As árvores são consideradas através de subamostras da base de dados definidas pela amostragem de *bootstrap*, a fim de se evitar que haja uma redundância das previsões, ou seja, não é usado todo o conjunto de treinamento para se fazer a estruturação de uma árvore.

Inicialmente, foi proposto avaliar o desempenho do classificador perante parâmetros padrões determinados pela própria biblioteca que o continha. Alguns hiperparâmetros foram analisados e então escolhidos para se estruturar o modelo de decisão. De acordo com o estudo feito, decidiu-se por definir o número de árvores presentes na floresta aleatória, parâmetro representado pela variável *n_estimators*. Além disso, o número mínimo de amostras necessárias para se dividir um nó, representado por *min_samples_split*, o número mínimo de amostras necessárias para estar presentes em uma folha, representado por *min_samples_leaf*, e por último, a profundidade máxima de uma árvore, representado por *max_depth*, que quando não for definido pelo projetista, expande os nós até que todas as folhas sejam puras, ou seja, que contenham menos amostras do que definidas por *min_samples_split*.

Como mencionado anteriormente, tais parâmetros foram utilizados a partir dos valores padrões apresentados pelo classificador e verificados em relação às suas métricas de avaliação. Após este primeiro momento, usou-se o método *gridsearch* com o intuito de descobrir os valores ideais para o estudo.

Tabela 16 - Hiperparâmetros padrões usados para estimar o classificador

Hiperparâmetro	**Valor**
Número de árvores (*n_estimators*)	100
Mínimo de amostras (*min_samples_split*)	2
Mínimo de folhas (*min_samples_leaf*)	1
Profundidade da árvore (*max_depth*)	None

Fonte: Elaborado pelo autor (2022).

Com os hiperparâmetros indicados pela tabela 16, obteve-se uma acurácia de 99,03% para o modelo, e matriz de confusão e demais métricas de avaliação conforme tabelas 17 e 18, respectivamente.

Tabela 17 - Matriz de confusão após classificação inicial

	Amplitude	**Frequência**	**Fase**
Amplitude	996	0	0
Frequência	0	982	20
Fase	0	9	993

Fonte: Elaborado pelo autor (2022).

Tabela 18 - Métricas de avaliação após classificação inicial

	Precisão	***Recall***	**F1 *Score***
Amplitude	1.00	1.00	1.00
Frequência	0.99	0.98	0.99
Fase	0.98	0.99	0.99

Fonte: Elaborado pelo autor (2022).

Como foi feito na seção anterior referente a outro classificador, também foi feita uma procura para obter os hiperparâmetros considerados ótimos para o atual caso. A validação cruzada foi feita com k=3, a fim de se diminuir a complexidade computacional do projeto. Obtiveram-se os resultados indicados na tabela 19, enquanto que a distribuição dos sinais

decididos pelo classificador quando comparados com o conjunto de testes é representada na tabela 20, ou seja, a matriz de confusão, e demais métricas de avaliação na tabela 21.

Tabela 19 - Hiperparâmetros ótimos

Hiperparâmetro	**Valor**
Número de árvores (*n_estimators*)	80
Mínimo de amostras (*min_samples_split*)	3
Mínimo de folhas (*min_samples_leaf*)	8
Profundidade da árvore (*max_depth*)	200

Fonte: Elaborado pelo autor (2022).

Tabela 20 - Matriz de confusão após classificação sob hiperparâmetros ótimos

	Amplitude	**Frequência**	**Fase**
Amplitude	996	0	0
Frequência	0	1002	0
Fase	0	1	1001

Fonte: Elaborado pelo autor (2022).

Tabela 21 - Métricas de avaliação após classificação sob hiperparâmetros ótimos

	Precisão	***Recall***	**F1 *Score***
Amplitude	1.00	1.00	1.00
Frequência	1.00	1.00	1.00
Fase	1.00	1.00	1.00

Fonte: Elaborado pelo autor (2022).

Percebe-se que o classificador floresta aleatória acertou praticamente todos os casos e retornou ótimos resultados em relação às métricas estabelecidas. A tunagem dos hiperparâmetros foi custosa computacionalmente, uma vez que foram avaliados 4 parâmetros e com mais opções para determinação do melhor valor.

5 CONSIDERAÇÕES FINAIS

Neste capítulo serão apresentadas as conclusões tomadas acerca do presente trabalho. Comparar-se-á os métodos definidos e utilizados aqui com a literatura indica na seção da história da arte, e comentar-se-á de que forma o projeto contribuiu no ambiente científico. Também serão comentados os resultados adquiridos e sugestões para projetos futuros.

5.1 CONTEXTUALIZAÇÃO DO PROJETO PROPOSTO E LITERATURA

Diante do atual trabalho apresentado, foi possível fazer um estudo aprofundado em relação à literatura de classificação automática de modulação.

Primeiramente, optou-se por abordar o estudo verificando características chaves de cada sinal. As *features* foram determinadas através da análise de variações dos parâmetros fundamentais nos sinais modulados, com o objetivo de que fosse possível identificar diferenças entre sinais de amplitude, fase e frequência. Para tal, escolheu-se a estratégia de trabalhar com *features* baseadas no domínio do tempo e domínio da transformada.

O canal de comunicação escolhido para representar o ruído apontado para o sinal foi o canal AWGN, uma vez que apresenta uma boa representação do mundo real e gera bons modelos matemáticos, além de que é utilizado em quase todos os trabalhos da área.

Diferentemente dos trabalhos indicados na literatura, cujos objetivos indicam realizar uma classificação automática de modulação de fato, avaliando o efeito de diferentes tipos de ruído com SNRs distintas, ou sob diferentes estratégias, o presente trabalho teve como principal propósito comparar a eficácia de classificadores existentes na área de aprendizado de máquina quando abordados no ambiente de modulações digitais. Foi possível aplicar três classificadores diferentes e calcular a matriz de confusão para cada um deles, quando implementados em cima da base de dados criada pelo projetista, composta de sinais digitais modulados em amplitude, fase e frequência.

Foi possível contribuir no ambiente científico através de mais um estudo na área de aprendizado de máquina, abordado sob um tema muito importante e relevante para a engenharia, que é a área de comunicação digital. Foi um dos trabalhos pioneiros para o Laboratório de Pesquisas em Radiofrequências em relação ao tema, e espera-se que incentive muitos outros a percorrerem esse caminho para ser possível avançar ainda mais no estudo.

Quadro 2 - Contribuição do atual trabalho para a literatura de AMC

Trabalho	**Abordagem**	**Estratégia**	**Classificador**	**Canal**	**Ano**	**Modulações**
TRIANTARIS *et al.*	FB	Rede Neural	CNN	AWGN	2019	BPSK, QPSK, 8PSK, 16QAM, 64QAM, GFSK, CPFSK, e PAM4. WB-FM e AM-DSB
ORLIC; DUKIC	LB	Cumulantes dos sinais analíticos		AWGN, Rayleigh, Rice, Log-normal	2009	BPSK, QPSK, 16-QAM, 64-QAM
WU *et al.*	FB		SVM	AWGN	2017	M-ASK, M-FSK, M-PSK e 16-QAM
HASSAN *et al.*	FB	*Continuous Wavelet Transform*	ANN	AWGN	2010	M-ASK, M-FSK, M-PSK, M-QAM
MOBASSERI	FB	*Features* baseadas no formato da constelação		AWGN	2000	QPSK, 8-PSK e 16-QAM
ZHAO; REN; ZHONG	FB	Rede neural baseado em wavelet	WNN	AWGN	2006	QPSK, OQPSK, GMSK, 16QAM, 4PSK, 2PSK, 4FSK, 2FSK, 4ASK e 2ASK
BAGGA; TRIPATHI	FB	*Features* estatísticas	Árvore de decisão	AWGN e Rayleigh	2013	2ASK, 4ASK, 2PSK, 4PSK 2FSK, 4FSK e 16 QAM. GMSK, 64QAM e 256QAM
MADHAVAN *et al.*	LB	Cumulantes cíclicos		AWGN e Rayleigh	2013	PSK e QAM

Trabalho	Abordagem	Estratégia	Classificador	Canal	Ano	Modulações
KIM *et al.*	FB	*Features* estatísticas	DNN	AWGN e Rician	2016	BPSK, QPSK, 8PSK, 16QAM e 64QAM
ASLAM; ZHU; NANDI	FB	Cumulantes de quarta e sexta ordem	KNN	AWGN	2012	PSK e QAM
SANTOS	FB	*Features* no domínio do tempo e da transforma da	KNN, SVM e Floresta aleatória	AWGN	2022	2ASK, 4ASK, 8ASK, 2FSK, 4FSK, 8FSK, 2PSK, 4PSK e 8PSK

Fonte: Elaborado pelo autor (2022).

No quadro 2, indica-se o mesmo quadro apontado na seção referente ao estado da arte com as principais literaturas acerca do estudo de classificação automática de modulação, mas dessa vez, preencheu-se uma linha com os métodos utilizados no atual trabalho, a fim de que seja feita uma comparação em relação a todos os passos percorridos pelo projetista.

5.2 CONCLUSÃO

Durante este trabalho, foi analisado o comportamento de três diferentes classificadores no ambiente de aprendizado de máquina dentro do ambiente de comunicação digital com a função de classificação automática de modulação. O objetivo foi concluído com sucesso, uma vez que se propôs percorrer todos os passos quando se trata de uma classificação, desde a implementação da base de dados que foi utilizada, até por fim, a classificação efetiva.

Como proposto, construiu-se uma base de dados com 9000 amostras bastante variada em relação aos parâmetros de geração, com três tipos de modulações digitais diferentes baseadas nos parâmetros fundamentais dos sinais, de forma que se obtivesse sinais modulados em amplitude, fase e frequência. Também foi possível avaliar cada uma dessas modulações aplicadas a três ordens diferentes, ou seja, com os bits da mensagem digital agrupados em conjuntos conforme desejado. Adicionou-se um ruído branco gaussiano com distribuição normal de 0.05 e média 0, com o objetivo de simular a presença dos efeitos de um canal AWGN, resultando em diferentes SNRs dentro da faixa de 20 dB e 30 dB.

A base de dados se desviou levemente do ambiente de comunicação digital real, uma vez que no contexto real, costuma-se trabalhar com valores extremamente altos de frequência de portadora e outros parâmetros. Foi necessário se manter o objetivo de analisar o comportamento de classificadores perante diferentes modulações, desde que os parâmetros dos sinais ainda fizessem sentido de acordo com a literatura.

Após a aquisição da base de dados, aplicou-se o método baseado em *features* apresentado na literatura, onde se extraiu *features* definidas pelo projetista de cada um dos 9000 sinais presentes na base de dados. Avaliaram-se 5 *features* definidas como domínio do tempo e 1 *features* definida como domínio da transformada. As características foram determinadas de acordo com a análise de que o desvio padrão dos parâmetros fundamentais dos sinais seria um importante parâmetro a ser considerado para fazer a classificação dos mesmos.

Para a aplicação dos classificadores determinados, primeiro utilizou-se o método *holdout* para partição do conjunto entre treinamento e teste, com o intuito de se fazer a validação do classificador. Para tal, dividiu-se a base de dados em 66% sendo conjunto de treinamento, enquanto que 33% foi determinado como conjunto de teste. Das 9000 amostras, 6000 tiveram o objetivo de treinar os classificadores, enquanto que 3000 foram usadas para avaliação. Como o ambiente de implementação foi totalmente controlado e, mesmo apesar da adição de sinais ruidosos aos sinais modulados, já se esperava ter resultados bastante bons.

O primeiro classificador implementado e avaliado no atual projeto foi o KNN. O modelo apresentou métricas de avaliação bastante próximas a 100%. A acurácia de acerto foi de 98,93% quando considerado o número ideal de vizinhos, ou seja, das 3000 amostras de teste, 2968 foram classificadas corretamente nessa situação. Todos os 973 sinais de amplitude presentes na base de dados foram classificados corretamente como sinais de amplitude, enquanto que dos 1046 sinais de frequência, obteve-se 1022 classificados como sinais de frequência de fato, enquanto que os 24 restantes foram classificados como sinais de fase. E por último, dos 981 sinais de fase, obteve-se 973 classificados corretamente, enquanto que os demais 8 foram classificados como sinais de frequência. Demais métricas de avaliação também foram avaliadas e calculadas.

Em relação ao SVM, que foi o segundo classificador implementado, optou-se por se fazer uma aplicação em duas vias, onde a primeira consistiu de usar os hiperparâmetros definidos pela própria biblioteca, enquanto que a segunda foi responsável por usar os hiperparâmetros considerados ótimos dentro da situação. A primeira via apresentou uma acurácia bastante baixa, principalmente quando considerado o ambiente e as condições ideais

de trabalho presentes no projeto. O valor de 79,77% foi o que motivou o projetista a procurar melhores resultados. Quando os hiperparâmetros foram ajustados para: *Kernel* como Função de Base Radial, C como 100 e *gamma* como 0.001, obteve-se métricas de avaliação muito melhores, em especial, uma acurácia de 98%. Dos 973 sinais de amplitude, todos foram corretamente classificados dentro de sua categoria. Dos 1046 sinais propostos como sinais de frequência, 1002 foram corretamente classificados, enquanto que 1 foi classificado como sinal de amplitude, e 43 como sinais de fase. E por fim, dos 981 sinais de fase, tivemos um número de acerto de 980, enquanto que 1 sinal foi erroneamente classificado como sinal de frequência.

O último classificador foi o que apresentou os melhores resultados, principalmente devido a sua complexidade em relação aos demais. O método de floresta aleatória também teve seu processo separado em duas vias, e os resultados apresentados quando os hiperparâmetros considerados ótimos foram aplicados foi excelente. A acurácia do modelo foi praticamente 100%, uma vez que, dentro de todas as 3000 amostras presentes no conjunto de testes, apenas uma delas foi classificada como errada. O método de se aplicar diversas árvores de decisão utilizando a amostragem de *bootstrap*, para então definir através da classificação da maioria a decisão a ser tomada, se mostrou bastante eficiente, quando considerado a alta complexidade.

Percebe-se que antes de se utilizar o *gridsearch*, alguns classificadores apresentaram resultados bastante ruins. Isso se deve ao fato da não normalização das *features*, que provavelmente gerou um espaço muito distinto e trouxe problemas para a classificação. Após a obtenção dos hiperparâmetros ótimos, obtiveram-se resultados muito próximos de 100%, visto que foram tomadas medidas para se encontrar a melhor situação possível. É evidente que em um um trabalho, onde o ambiente foi totalmente controlado, os sinais deveriam apresentar resultados ótimos para a classificação. As métricas finais de acurácia podem ser verificadas na tabela 22, para se ter uma compreensão melhor dos resultados obtidos.

Tabela 22 - Acurácia dos classificadores KNN, SVM e Floresta Aleatória

Classificadores	**Acurácia**
K-Nearest Neighbor	0.987
Support Vector Machine	0.982
Random Forest	1.000

Fonte: Elaborado pelo autor (2022).

Por fim, conclui-se que o projeto cumpriu o seu objetivo no que foi tangível ao projetista. Foi possível fazer a implementação de um sistema por completo de classificação automática de modulação, analisando cada etapa e suas peculiaridades individualmente. Conseguiu-se integrar duas áreas bastante interessantes no que diz respeito à engenharia. Percorreu-se um grande caminho e aumentou-se o entendimento em relação ao tema de comunicação de sinais digitais. E principalmente, evoluiu-se muito no ambiente de aprendizado de máquina e ciência de dados, tema riquíssimo e presente em inúmeras empresas e indústrias atualmente.

Espera-se que este trabalho incentive o autor a permanecer na área e que seja um passo importante para a inserção do mesmo no mercado de trabalho. Deseja-se, também, que seja um dos primeiros passos dentro do Laboratório de Pesquisas em Radiofrequências no tema, o qual se mostrou muito interessante e curioso, e com muito a se avançar.

6 REFERÊNCIAS

ALI, Abdelmohsen; HAMOUDA, Walaa. Advances on spectrum sensing for cognitive radio networks: Theory and applications. **IEEE communications surveys & tutorials**, v. 19, n. 2, p. 1277-1304, 2016.

ASLAM, Muhammad Waqar; ZHU, Zhechen; NANDI, Asoke Kumar. Automatic modulation classification using combination of genetic programming and KNN. **IEEE Transactions on wireless communications**, v. 11, n. 8, p. 2742-2750, 2012.

AZZOUZ, Elsayed; NANDI, Asoke Kumar. Automatic modulation recognition of communication signals. 1998.

BAGGA, Jaspal; TRIPATHI, Neeta. Automatic modulation classification using statistical features in fading environment. **International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering**, v. 2, n. 8, p. 3701-3709, 2013.

CLAESEN, Marc; DE MOOR, Bart. Hyperparameter search in machine learning. **arXiv preprint arXiv:1502.02127**, 2015.

DANGETI, Pratap. **Statistics for machine learning**. Packt Publishing Ltd, 2017.

DOBRE, Octavia A. et al. Survey of automatic modulation classification techniques: classical approaches and new trends. **IET communications**, v. 1, n. 2, p. 137-156, 2007.

DOUGHERTY, Geoff. **Pattern recognition and classification: an introduction**. Springer Science & Business Media, 2012.

GÉRON, Aurélien. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow**. Alta Books, 2019.

HAMEED, Fahed; DOBRE, Octavia A.; POPESCU, Dimitrie C. On the likelihood-based approach to modulation classification. **IEEE Transactions on Wireless Communications**, v. 8, n. 12, p. 5884-5892, 2009.

HASSAN, Kais et al. Automatic modulation recognition using wavelet transform and neural networks in wireless systems. **EURASIP Journal on Advances in Signal Processing**, v. 2010, p. 1-13, 2010.

HAYKIN, Simon; MOHER, Michael. **Introduction to analog and digital communications**. Wiley, 2007.

HAZZA, Alharbi et al. An overview of feature-based methods for digital modulation classification. In: **2013 1st international conference on communications, signal processing, and their applications (ICCSPA)**. IEEE, 2013. p. 1-6.

HONG, Liang. Maximum likelihood BPSK and QPSK classifier in fading environment using the EM algorithm. In: **2006 Proceeding of the Thirty-Eighth Southeastern Symposium on System Theory**. IEEE, 2006. p. 313-317.

KIM, Byeoungdo et al. Deep neural network-based automatic modulation classification technique. In: **2016 International Conference on Information and Communication Technology Convergence (ICTC).** IEEE, 2016. p. 579-582.

LATHI, Bhagwandas P. **Modern digital and analog communication systems**. Oxford University Press, Inc., 1990.

MADHAVAN, Narendar et al. Spectrum sensing and modulation classification for cognitive radios using cumulants based on fractional lower order statistics. **AEU-International Journal of Electronics and Communications**, v. 67, n. 6, p. 479-490, 2013.

MOBASSERI, Bijan G. Digital modulation classification using constellation shape. **Signal processing**, v. 80, n. 2, p. 251-277, 2000.

MÜLLER, Andreas C.; GUIDO, Sarah. **Introduction to machine learning with Python: a guide for data scientists**. " O'Reilly Media, Inc.", 2016.

ORLIC, Vladimir D.; DUKIC, Miroslav L. Automatic modulation classification algorithm using higher-order cumulants under real-world channel conditions. **IEEE Communications Letters**, v. 13, n. 12, p. 917-919, 2009.

PATIL, Kishor; PRASAD, Ramjee; SKOUBY, Knud. A survey of worldwide spectrum occupancy measurement campaigns for cognitive radio. In: **2011 International conference on devices and communications (ICDeCom)**. IEEE, 2011. p. 1-5.

SCIKIT-LEARN. **Cross-validation: evaluating estimator performance.** scikit-learn.org, 2021. Disponível em: <https://scikit-learn.org/stable/modules/cross_validation.html>. Acesso em: 21/12/2021.

SHALEV-SHWARTZ, Shai; BEN-DAVID, Shai. **Understanding machine learning: From theory to algorithms**. Cambridge university press, 2014.

SUN, Hongjian et al. Wideband spectrum sensing for cognitive radio networks: a survey. **IEEE Wireless Communications**, v. 20, n. 2, p. 74-81, 2013.

TRIANTARIS, Pavlos et al. Automatic modulation classification in the presence of interference. In: **2019 European Conference on Networks and Communications (EuCNC)**. IEEE, 2019. p. 549-553.

WU, Zhilu et al. Robust automatic modulation classification under varying noise conditions. **IEEE Access**, v. 5, p. 19733-19741, 2017.

ZHAO, Yaqin; REN, Guanghui; ZHONG, Zhi. Modulation recognition of SDR receivers based on WNN. In: **2006 IEEE 63rd Vehicular Technology Conference**. IEEE, 2006. p. 2140-2143.

ZHU, Zhechen; NANDI, Asoke K. **Automatic modulation classification: principles, algorithms and applications**. John Wiley & Sons, 2015.