

Utilização de Técnicas de Aprendizado de Máquina para Demodulação de Sinais

Thiago Almeida e Edson Mintsu Hung

Resumo— O amadurecimento das tecnologias de *Software Defined Radio* (SDR) associado com a maior disponibilidade de ferramentas de aprendizado de máquina têm favorecido o estudo de novas propostas de sistemas de comunicação. Nesse contexto, este artigo discute a utilização de técnicas de aprendizado de máquina na construção de um demodulador digital *Quadrature Phase Shift Keying* (QPSK) utilizando um conjunto de dados de treinamento e validação gerados a partir do GNU Radio. Diferentemente dos métodos tradicionais, este demodulador não depende de considerações prévias sobre modelos de canal, mas utiliza o seu conjunto de dados para realizar o treinamento supervisionado da constelação de símbolos do demodulador para diversos ambientes, obtendo um desempenho comparável ou superior aos demoduladores tradicionais.

Palavras-Chave— Demodulador, Machine Learning, GNU Radio, Rede Neural, SDR.

Abstract— The maturity of *Software Defined Radio* (SDR) technologies associated with the greater availability of machine learning tools have favored the study of new proposals for communication systems. In this context, this article discusses the use of machine learning techniques in the construction of a digital demodulator *Quadrature Phase Shift Keying* (QPSK) using a training set and validation data generated from GNU Radio. Unlike traditional methods, this demodulator does not depend on previous considerations about channel models, but uses its data set to carry out the supervised training of the demodulator symbol constellation for different environments, obtaining a performance comparable or superior to traditional demodulators.

Keywords— Demodulator, Machine Learning, GNU Radio, Rede Neural, SDR.

I. INTRODUÇÃO

Os equipamentos de rádio tradicionais normalmente apresentam hardware e software desenvolvidos para executarem um conjunto de funções específicas, de forma que dificilmente qualquer outro tipo de forma de onda não dimensionada inicialmente no escopo poderá ser trabalhada pelo equipamento. Assim, alterações no ambiente de comunicação, como ruído não previsto ou sinais interferentes, podem requerer o desenvolvimento de um novo hardware ou um novo software para realizar a comunicação, o que eleva os custos do projeto com modificações e certificações, bem como diminui a flexibilidade do sistema. Nesse contexto, surgiu o *Software Defined Radio* (SDR), termo empregado pela primeira vez por Joe Mitola [1] para definir um rádio em que todas (ou quase todas) as funções da camada física podem ser definidas por software. Assim, o hardware específico é substituído por um hardware

geral que pode ser especializado e configurado via software para atender novos e diversos contextos de emprego, incluindo modificações na frequência portadora, forma de acesso ao canal e modulações [2].

Ao mesmo tempo em que a tecnologia de rádios veio se desenvolvendo, as técnicas de aprendizado de máquina ou *Machine Learning* (ML), como as redes neurais, também obtiveram grande avanço em virtude da maior capacidade de processamento dos hardwares e da evolução de aplicações voltadas para o estudo de redes neurais, como Tensorflow [3] e Scikit-Learn. A questão abordada neste artigo é a utilização de técnicas de ML aplicadas ao processo de demodulação de sinais QPSK em meio a um ambiente ruidoso, utilizando uma plataforma SDR [4][5][6].

No desenvolvimento de sistemas de comunicação, observa-se tradicionalmente a necessidade de um amplo estudo de modelos matemáticos e estatísticos do canal a fim de definir, a priori, a melhor técnica de modulação/demodulação a ser utilizada. Assim, os métodos tradicionais não conseguem administrar com eficiência a comunicação em ambientes que não se comportem conforme as hipóteses inicialmente estabelecidas para aquele modelo [7]. Nesse ponto, destaca-se que o emprego de redes neurais apresenta-se como uma nova forma de demodular o sinal, visto que um demodulador baseado nesse tipo de rede pode ser treinado de forma supervisionada com um conjunto de dados (*dataset*) constituído por amostras dos sinais reais gerados pelo transmissor [8][9][10]. A vantagem desse processo é que essas amostras obtidas pelo demodulador já contêm todas as perturbações reais encontradas no ambiente como ruído, emissões espúrias, desvanecimento e distorções não lineares dos próprios transmissores e receptores, possibilitando à rede neural o estabelecimento de novas fronteiras de decisão para a classificação dos símbolos. Tal método dispensa as considerações matemáticas prévias sobre o comportamento do canal, além de permitir a adaptação do demodulador a novos ambientes não previstos inicialmente, por meio de ajustes dos hiperparâmetros da rede.

Em complemento às ferramentas de ML, empregou-se o software *GNU Radio Companion* (GRC), o qual possui a capacidade de configurar e simular diversos ambientes de comunicação, além de gerar, se necessário, formas de ondas específicas descritas por funções de processamento de sinais. Por meio desse software foi possível gerar um *dataset* de amostras em fase e quadratura (IQ) de símbolos de um modulador/demodulador QPSK e inserir artificialmente distorções causadas por ruído branco, interferência externa e efeitos de multipercurso. Isso possibilitou, dentro do demodulador, o treinamento de um classificador de símbolos baseado em rede

neural e permitiu comprovar sua adaptabilidade e eficiência em diversos ambientes, comparativamente a um demodulador tradicional.

As seções deste artigo estão divididas em: II – Vantagens da Demodulação com Redes Neurais, III – Demodulação de sinal QPSK e IV – Conclusão.

II. VANTAGENS DA DEMODULAÇÃO COM REDES NEURAIS

Os mecanismos de comunicação digital requerem um amplo estudo estatístico do comportamento do símbolo quando submetido aos ruídos e distorções oriundos do canal de comunicação e de não linearidades dos próprios equipamentos [11]. O estudo tradicional estabelece suposições quanto às características do ruído (branco e colorido) e sobre a probabilidade da emissão de determinado símbolo na fonte para, a partir de então, tecer considerações sobre o limiar de decisão que será utilizado para distinguir um símbolo de outro no receptor [12]. Entretanto, o ponto negativo desse mecanismo é que o sistema fica limitado às considerações estabelecidas antes do seu emprego real, as quais podem não ser integralmente válidas nos diversos ambientes em que o sistema será empregado, fazendo com que este não atue de forma eficiente em determinado cenário.

As técnicas de aprendizado de máquina têm tido empregos diversos na área de telecomunicações. Observa-se o uso de redes neurais convolucionais para a classificação de sinais e modulações [13][14] e também aplicações de *Spatial Transformer Network* para sincronização de variações de tempo frequência e fase entre transmissor e receptor [15]. Nesse sentido, este artigo propõe a criação de um demodulador baseado em rede neural que utiliza um *dataset* com uma sequência símbolos pré-estabelecidos entre o transmissor e o receptor (sequência de rótulos definida a priori entre o transmissor e o receptor), de forma que seja possível, a partir de um treinamento supervisionado, classificar os símbolos transmitidos a partir das amostras IQ do sinal recebido, diferentemente de determinar a modulação empregada como nos trabalhos citados anteriormente. O treinamento pode ser executado periodicamente de forma a adaptar os hiperparâmetros da rede para alterações nas condições reais do ambiente. O mecanismo de decisão de quando a rede deve ser novamente treinada não está no escopo deste trabalho, entretanto pode ser criada uma camada que avalia uma determinada métrica de qualidade do enlace de modo a decidir se é necessário um novo treinamento e assim readequar os hiperparâmetros da rede para aquela condição. Este trabalho mantém o foco em modulações QPSK com simulações envolvendo a presença de ruído e sinais interferentes AM DSB-SC. Trabalhos similares já foram realizados considerando modulações FSK (Frequency Shift Keying) e ASK (Amplitude Shift Keying) [16]. A utilização da modulação QPSK facilita a simulação no GNU Radio, dada a disponibilidade de documentação sobre o assunto. Diversos outros cenários podem ser emulados no GRC, porém este, em específico, é importante para a avaliação de modulações digitais para enlace de dados, utilizando técnicas de ML, em um ambiente de comunicação aeronáutica submetido à

ruídos e interferências típicas da faixa de frequência V/UHF aeronáutica, a qual é tipicamente preenchida por sinais de comunicação por voz AM. Uma possível aplicação prática seria o estabelecimento de uma rede de dados QPSK mais robusta entre aeronaves voando em formação em ambiente de combate.

A ferramenta de aprendizado de máquina que será avaliada será uma rede *Multilayer Perceptron* (MLP) [17], uma vez que normalmente ocupa menos espaço caso portada para um FPGA, proporciona menor consumo de energia/recursos computacionais e apresenta resultados satisfatórios para demonstrar as vantagens do uso de ML em telecomunicações. As sinapses são matematicamente representadas por um conjunto de pesos pelos quais os neurônios multiplicam o valor das entradas, resultado que atuará como argumento para uma função de ativação que determinará o valor da saída, conforme a rede representada de maneira macro na Figura 1. O treinamento utiliza um critério de minimização da entropia dos símbolos recebidos em relação ao conjunto de teste. A rede proposta possui duas entradas, uma para amostra I e outra para amostra Q de cada símbolo, e quatro saídas que representam a probabilidade de cada símbolo (00, 01, 10 e 11). Já o critério de decisão para a escolha do símbolo transmitido está relacionado ao maior valor observado na saída da rede.

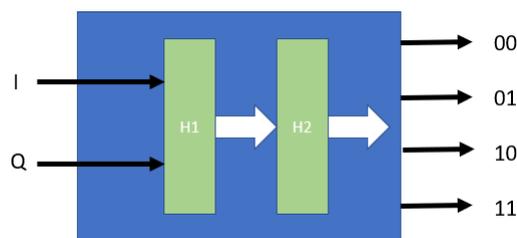


Fig. 1. Estrutura de rede neural com 02 camadas ocultas (H) para demodulação

A utilização de redes neurais é viável e vantajosa para a determinação dos símbolos recebidos por diversos motivos. O primeiro é que as amostras do sinal real já contemplam todas as distorções presentes no sistema, sem a necessidade de criar modelos matemáticos de aproximação para cada efeito físico a que o sinal está submetido, como suposição de ruído, modelos de reflexão, dentre outros [11], exigindo muito menos conhecimento do projetista sobre o assunto [18].

O segundo motivo para a utilização das redes neurais é a grande disponibilidade de ferramentas que, em virtude do aumento da capacidade computacional disponível atualmente, conseguiram viabilizar o treinamento e utilização de redes neurais com resultados bastante satisfatórios para as aplicações de comunicação [19].

O terceiro ponto relevante é que as aplicações de comunicação são tradicionalmente separadas em blocos com funções específicas, embora nem sempre tal separação seja ótima [20]. Essa modularidade permite a inserção de blocos de redes neurais sem ter que repensar completamente o sistema. Adicionalmente, para adaptar o demodulador para outro tipo de modulação digital basta alterar o conjunto de treinamento e redefinir os hiperparâmetros da rede.

Utilizou-se o GRC para criar um ambiente de comunicação próximo ao real e implementar um bloco externo com o objetivo de ler as amostras em fase e quadratura (IQ) de um sinal QPSK para uma demodulação tradicional.

O desenvolvimento dos experimentos de ML foi realizado por meio do software Scikit-Learn de forma conjunta com as demais bibliotecas Python do Anaconda e editor Jupyter, com o objetivo de verificar como a variação de hiperparâmetros de uma rede neural (MLP) influencia no desempenho da classificação de símbolos e demonstrar a viabilidade de aplicações de ML em comunicações.

III. DEMODULAÇÃO NEURAL

O experimento realizado consistiu em construir um sistema de comunicação com modulador e demodulador QPSK utilizando blocos do GRC, a fim de gerar amostras em fase e quadratura contaminadas com ruído branco, interferência e reflexão. Como os símbolos transmitidos são definidos a priori, foi possível construir um *dataset* associando as amostras IQ recebidas na saída do sistema com os rótulos de cada símbolo já conhecidos. O sistema foi configurado de tal forma que a representação esperada para a constelação de símbolos transmitidos deveria corresponder aos valores da Tabela I em um meio ideal. A presença de ruídos e efeitos de multipercursos acabam por poluir essa constelação ideal de símbolos.

TABELA I
MAPA DE SÍMBOLOS E AMOSTRAS

símbolo	valor esperado da amostra
0	$0.707+0.707j$
1	$0.707-0.707j$
2	$-0.707+0.707j$
3	$-0.707-0.707j$

O diagrama de blocos desenvolvido no GRC, representado na Figura 2, foi criado a partir de um conjunto de dados de 160 mil amostras (40 mil por símbolo QPSK), das quais 140 mil foram utilizadas para treinamento e 20 mil para validação.

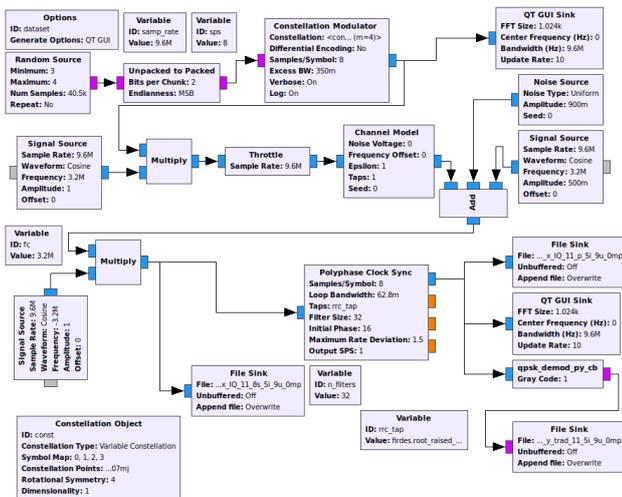


Fig. 2. Diagrama de blocos para construção do *dataset* de símbolos

Dessa forma, as componentes IQ medidas do sinal recebido já contemplam a distorção por ruído do canal. Em um caso não

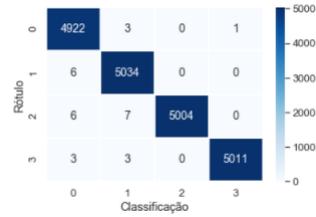
simulado, não haveria a necessidade de realizar modelamentos matemáticos do ruído, uma vez que este já estaria inserido naturalmente nas amostras IQ medidas, de forma que caberia à rede neural se adaptar à melhor maneira de classificar o símbolo recebido, considerando a minimização de uma função de custo que avalia a entropia cruzada entre as previsões realizadas e os rótulos dos símbolos.

Ao fim da simulação as amostras IQ foram salvas em um arquivo específico e, como os símbolos gerados são conhecidos a priori, foi possível determinar a relação entre cada amostra IQ ao rótulo do símbolo transmitido. Os arquivos de rótulo e amostras foram carregados em um notebook Jupyter, a fim de realizar o treinamento adequado das redes.

Utilizando as ferramentas disponíveis no Scikit-Learn foi possível realizar o treinamento das redes utilizando variações de hiperparâmetros para pesquisar melhorias nos resultados. Foram realizados treinamentos utilizando-se redes MLP de 02 camadas ocultas (ou *Hidden Layer* - HL) variando entre 01 ou 05 neurônios cada uma e empregando-se o algoritmo de gradiente descendente estocástico para a otimização dos parâmetros. A variável que controla a taxa de aprendizado foi mantida constante ($\eta = 0.001$), porém foi empregado o método de gradiente acelerado de Nesterov para melhorar a convergência do algoritmo [21] [17]. Além disso, procurou-se modificar a função de ativação, entre função logística e tangente hiperbólica, para verificar se tal mudança ocasiona melhoria significativa nos resultados. A função ReLu também poderia ter sido utilizada, mas as funções escolhidas já apresentaram resultados satisfatórios para os objetivos deste artigo. Observa-se que o termo “melhoria”, no aspecto deste artigo, é a diminuição de erros de símbolos medidos a partir da análise da acurácia dos resultados do treinamento e da análise da matriz de confusão das classificações.

Os resultados observados para a matriz de confusão de cada combinação de neurônios na camada oculta e função de ativação estão apresentados na Figura 3, considerando um nível de interferência senoidal AM (*Amplitude Modulation*) DSB-SC (*Double-Sideband Suppressed-Carrier*) de amplitude 0,9V e ruído AWGN (*Additive White Gaussian Noise*) de amplitude 0,3V.

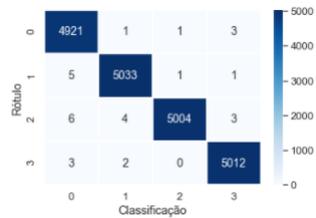
A matriz de confusão para as camadas ocultas com 05 neurônios não apresentou resultados significativamente diferentes, considerando apenas a mudança na função de ativação. Entre as redes com 01 neurônio em cada camada, observou-se uma piora na matriz de confusão e nas acurácias ao passar da função de ativação tangente hiperbólica para a função logística. A configuração de camadas ocultas com 05 neurônios foi a que apresentou melhor resultado dentre todas as configurações testadas. Foi possível validar o modelo por meio da utilização do método K-fold (com $K=5$ subconjuntos) e verificar que a acurácia do modelo se manteve acima de 99% na maioria das configurações, conforme apresentado na Tabela II para os 04 resultados mais relevantes, cujas linhas representam o número de neurônios em cada camada oculta ($H_1 \times H_2$), seguidos da função de ativação utilizada no experimento. Como o K-fold utiliza subconjuntos diferentes de uma mesmo *dataset* para treinamento e validação, a pouca variação da acurácia em todos os testes indica que o modelo se ajustou bem ao conjunto



(a) demodulador tradicional



(b) rede 1 x 1 ativação logística



(c) rede 1 x 1 ativação tangente hiperbólica



(d) rede 5 x 5 ativação tangente hiperbólica

Fig. 3. Matriz de confusão para diferentes estruturas de rede e funções de ativação para 20 mil símbolos na presença de interferência AM DSB-SC de amplitude 0,9V e amplitude de ruído de 0,3V

de dados estando prevenido da ocorrência de *overfitting*.

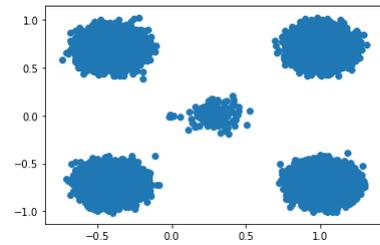
TABELA II

COMPARAÇÃO DE ACURÁCIAS PARA AS CONFIGURAÇÕES DA REDE

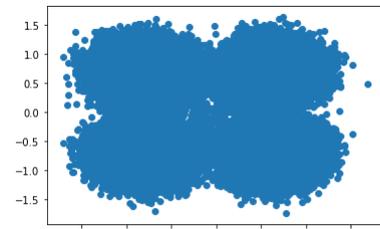
$H_1 \times H_2$ _ativ	Acurácias			
5x5_log	0.999	0.9993	0.99905	0.99885
5x5_tanh	0.9989	0.9994	0.9989	0.9989
1x1_log	0.9985	0.99925	0.9987	0.6023
1x1_tanh	0.9987	0.9989	0.9988	0.9987

Em um segundo experimento, procurou-se realizar uma simulação com o objetivo de avaliar o comportamento do demodulador quanto a variações do ambiente, considerando a

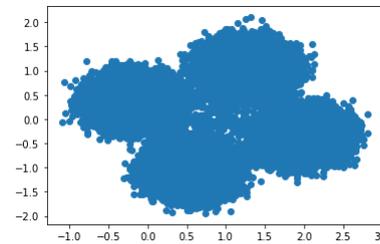
presença de sinal AM DSB-SC interferente em diversas amplitudes (A_{inter}), a presença de ruído (A_{am}) e a presença de sinal refletido decorrente de multipercurso (MulPath) [MulPath=x corresponde a passar o sinal por um filtro $1+(0.1*x)j$]. As Figuras 4 e 5 mostram os resultados comparativos entre um demodulador tradicional e um demodulador baseado em uma rede neural com 02 camadas ocultas de 05 neurônios cada uma (5x5), utilizando função de ativação tangente hiperbólica.



(a) Constelação $A_{am} = 0.9V$ $A_{inter} = 0.3V$



(b) Constelação $A_{am} = 0.9V$ $A_{inter} = 0.9mV$



(c) Constelação $A_{am} = 0.9V$ $A_{inter} = 0.5mV$ MulPath=5

Fig. 4. Constelação de símbolos após ruídos, interferências e reflexões

A partir da análise dos resultados observa-se que para ambientes com baixa interferência e ruído, o demodulador neural apresenta resultados bastante similares ao demodulador tradicional. Porém, em ambientes com forte variação de ruído, interferência severa e presença de sinais espúrios decorrentes de multipercursos, observa-se que o demodulador neural mantém sua capacidade de demodulação enquanto que o demodulador tradicional apresenta um número significativo de erros de símbolo. A diferença se deve à capacidade de adaptação que o demodulador neural possui, uma vez que pode utilizar o *dataset* padrão no novo ambiente e ajustar os hiperparâmetros da rede para operarem de forma ótima na nova condição do canal.

CONCLUSÃO

Concluiu-se que o demodulador neural proposto neste artigo, quando empregado em ambiente de baixa interferência

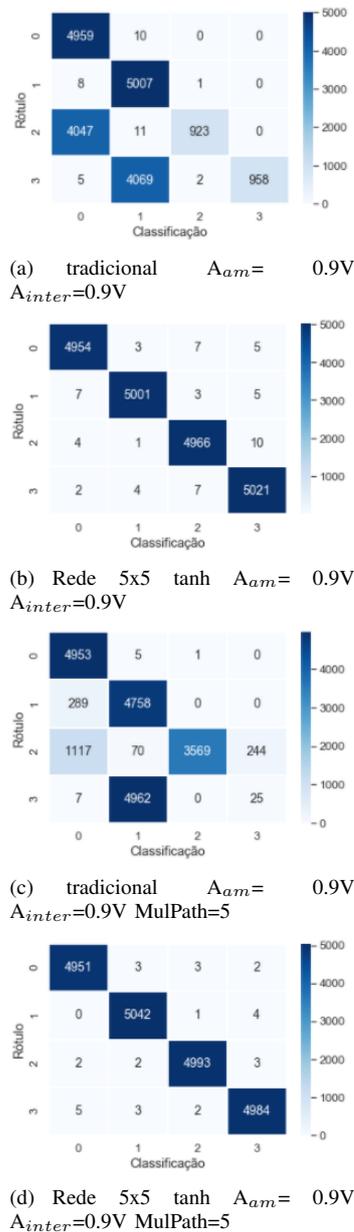


Fig. 5. Matriz de confusão comparativa entre o demodulador tradicional e o demodulador baseado em rede neural

e ruído, apresenta resultados bastante satisfatórios quando comparado com os demoduladores QPSK que empregam os limiares de decisão baseados em modelos matemáticos (tradicionais). Além disso, em ambiente com interferência severa e forte ruído, o demodulador neural apresenta um resultado superior aos demoduladores tradicionais, visto que possui a capacidade de ser treinado durante o emprego em ambiente real, ajustando os seus hiperparâmetros às condições ambientais (eletromagnéticas) daquele momento, diminuindo assim o erro de símbolos.

Verificou-se também a viabilidade desse tipo de estudo com a utilização de softwares abertos, como GNU Radio, Scikit-Learn e bibliotecas Python, demonstrando a redução de custos no estudo de técnicas de aprendizado de máquinas aplicado à análise de sinais utilizando mecanismos de SDR.

Por fim, sugere-se como trabalhos futuros o emprego de *front-end* de comunicação para gerar amostras de símbolos contaminados como efeitos reais de canal de comunicação, sem a necessidade de simular ruídos ou imperfeições.

REFERÊNCIAS

- [1] Joseph Mitola, “The software radio architecture,” *IEEE Communications magazine*, vol. 33, no. 5, pp. 26–38, 1995.
- [2] Andre Luiz Garcia Reis, Andre Felipe Barros, Karlo Gusso Lenzi, Luis Geraldo Pedroso Meloni, and Silvio Ernesto Barbin, “Introduction to the software-defined radio approach,” *IEEE Latin America Transactions*, vol. 10, no. 1, pp. 1156–1161, 2012.
- [3] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al., “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” *arXiv preprint arXiv:1603.04467*, 2016.
- [4] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.
- [5] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., “Tensorflow: A system for large-scale machine learning,” in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.
- [6] François Chollet et al., “Keras,” 2015.
- [7] Timothy J O’Shea, Tamoghna Roy, and Nathan West, “Approximating the void: Learning stochastic channel models from observation with variational generative adversarial networks,” in *2019 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2019, pp. 681–686.
- [8] Vishnu Pratap Singh Kirar, “Artificial neural networks for cognitive radio network: a survey,” *International Journal of Electronics and Communication Engineering*, vol. 9, no. 1, 2015.
- [9] Tianqi Wang, Chao-Kai Wen, Hanqing Wang, Feifei Gao, Tao Jiang, and Shi Jin, “Deep learning for wireless physical layer: Opportunities and challenges,” *China Communications*, vol. 14, no. 11, pp. 92–111, 2017.
- [10] Mario Bkassiny, Yang Li, and Sudharman K Jayaweera, “A survey on machine-learning techniques in cognitive radios,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1136–1159, 2012.
- [11] Tim Schenk, *RF imperfections in high-rate wireless systems: impact and digital compensation*, Springer Science & Business Media, 2008.
- [12] Simon Haykin and Michael Moher, *Introdução aos sistemas de comunicação*, Bookman Editora, 2009.
- [13] Timothy James O’Shea, Tamoghna Roy, and T Charles Clancy, “Over-the-air deep learning based radio signal classification,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.
- [14] Nathan E West and Tim O’Shea, “Deep architectures for modulation recognition,” in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 2017, pp. 1–6.
- [15] Timothy J O’Shea, Latha Pemula, Dhruv Batra, and T Charles Clancy, “Radio transformer networks: Attention models for learning to synchronize in wireless systems,” in *2016 50th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2016, pp. 662–666.
- [16] M Amini and Einollah Balarastaghi, “Universal neural network demodulator for software defined radio,” *International Journal of Machine Learning and Computing*, vol. 1, no. 3, pp. 305, 2011.
- [17] Ethem Alpaydin, *Introduction to machine learning*, MIT press, 2009.
- [18] Timothy J. O’Shea, Latha Pemula, Dhruv Batra, and T. Charles Clancy, “Radio transformer networks: Attention models for learning to synchronize in wireless systems,” *CoRR*, vol. abs/1605.00716, 2016.
- [19] Mathew McCaskey, Austin Feydt, Robert Corrigan, Kul Bhasin, and David Helms, “Implementation of a machine learning based modulation scheme in gnuradio for over-the-air packet communications,” 2018.
- [20] Ephraim Zehavi, “8-psk trellis codes for a rayleigh channel,” *IEEE Transactions on Communications*, vol. 40, no. 5, pp. 873–884, 1992.
- [21] Aurélien Géron, *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*, “O’Reilly Media, Inc.”, 2017.