

# Classificação Automática de Modulações usando Redes Convolucionais 1D

Jorge Kysnney Santos Kamassury, Vinícius Felipe de Oliveira da Silva e Danilo Silva

**Resumo**—Neste artigo, investigamos o problema de classificação automática de modulações através do aprendizado de máquina avaliando não somente a acurácia de classificação, mas também o tempo de treinamento das arquiteturas propostas. Diferentemente dos trabalhos recentes que lidam com o problema utilizando redes neurais convolucionais (CNN) com filtros de duas dimensões, propomos arquiteturas utilizando filtros de uma dimensão—uma CNN, uma ResNet e uma rede mista (combinando ResNet e DenseNet). Para as nossas arquiteturas, observamos que, em comparação com os trabalhos relacionados, as acurácias são próximas ou melhores e que os tempos de treinamento são menores para todos os casos.

**Palavras-Chave**—Classificação automática de modulação, aprendizado de máquina, redes convolucionais 1D, tempo de treinamento.

**Abstract**—In this article, we investigate the problem of automatic modulation classification through machine learning evaluating not only the classification accuracy, but also the training time of the proposed architectures. Unlike the recent work dealing with the problem using convolutional neural networks (CNN) with two-dimensional filters, we propose architectures using one-dimensional filters—a CNN, a ResNet and a mixed network (combining ResNet and DenseNet). For our architectures, we observe that, in comparison with related works, accuracy is closer or better and training times are lower for all cases.

**Keywords**—Automatic modulation classification, machine learning, 1D convolutional networks, training time.

## I. INTRODUÇÃO

A classificação automática de modulações (*automatic modulation classification*, AMC), cujo propósito consiste em reconhecer de modo autônomo o tipo de modulação do sinal recebido, tem desempenhado um importante papel nos sistemas modernos de comunicação sem fio. Com aplicações militares em vigilância e guerra eletrônica, a AMC também tem adquirido notabilidade por suas aplicações em rádio cognitivo, rádio definido por *software* [1] e comunicação adaptativa [2]. Nesse contexto, a AMC auxilia tanto os usuários na definição apropriada dos parâmetros de transmissão para assegurar a qualidade da comunicação, como na identificação de transmissores, detecção de anomalias e localização de interferências [3].

As mais variadas abordagens para AMC discutidas na literatura podem ser elencadas em duas categorias, a saber: métodos baseados em verossimilhança (*likelihood-based*, LB) e os métodos baseados em características (*feature-based*, FB)

Jorge Kysnney Santos Kamassury, Vinícius Felipe de Oliveira da Silva e Danilo Silva, Departamento de Engenharia Elétrica e Eletrônica, Universidade Federal de Santa Catarina, Florianópolis-SC, Brasil, E-mails: jorge.kamassury@posgrad.ufsc.br, viniciusfelipeoliveirada-silva@outlook.com, danilo.silva@ufsc.br.

[2], [4]. Em particular, os métodos LB fazem uso da teoria da decisão bayesiana, alcançando o desempenho ótimo com a minimização da probabilidade de erro de classificação. Apesar de teoricamente atingirem a solução ideal, os métodos LB enfrentam elevada complexidade computacional e demandam informações prévias dos transmissores, o que pode ser inviável para várias aplicações. Os métodos FB, por sua vez, focam na extração de características e nos critérios de classificação, sendo constituídos de dois subsistemas, o extrator de características e o classificador. As principais características extraídas no contexto da AMC são aquelas baseadas no domínio de tempo instantâneo, nas transformações e nas estatísticas do sinal [2]. No que se refere aos classificadores, árvore de decisão, máquina de vetores de suporte (*support vector machine*, SVM), K vizinhos mais próximos (*K nearest neighbors*, KNN) e redes neurais artificiais figuram entre aqueles mais utilizados [4].

Embora os métodos FB sejam subótimos, em razão da sua complexidade ser menor em relação aos métodos LB, eles são mais fáceis de implementar. De fato, vale acentuar que o desempenho dos métodos FB depende diretamente das características extraídas e do tipo de classificador. Assim, na maioria dos casos, as características extraídas devem ser projetadas manualmente para a modulação e canal correspondentes, o que exige considerações altamente específicas que, em geral, resultam em abordagens não flexíveis [2].

Nesse contexto, os métodos de aprendizado profundo (*deep learning*, DL) vêm se manifestando como uma ferramenta alternativa e robusta para AMC. Empregando redes multicamadas para extrair a representação interna dos dados brutos [4], os métodos DL aprendem as características automaticamente.

Dentre os métodos DL mais recentes empregados no contexto da AMC, citam-se: *deep belief network* [5], *autoencoders* [6], redes neurais convolucionais (*convolutional neural network*, CNN) [7] e redes neurais recorrentes (*recurrent neural network*, RNN) [4]. As arquiteturas CNNs, por exemplo, não apenas têm alcançado melhores resultados na classificação como também apresentam maior flexibilidade em comparação com as abordagens atuais dos métodos FB. Inspirada pelo sucesso das redes convolucionais utilizadas em processamento de imagens, boa parte das CNNs projetadas para AMC usam filtros de duas dimensões.

Na prática, um dos grandes desafios enfrentados pelos métodos DL para AMC reside no elevado tempo de treinamento, fato esse que dificulta sua aplicação em tempo real, onde o treinamento *online* é inegavelmente necessário para adaptar as arquiteturas de aprendizado profundo às alterações do ambiente. Para os sistemas de comunicação sem fio

autônomos previstos nas novas tecnologias de redes, o tempo de treinamento se tornará um requisito ainda mais crítico para o sucesso dos métodos DL.

Desse modo, visando alcançar um *trade-off* entre desempenho de classificação e tempo para treinamento, propomos três diferentes arquiteturas de redes neurais convolucionais 1D. A escolha em usar filtros de uma dimensão deve-se ao fato de o sinal modulado ser intrinsecamente unidimensional, cuja única variável independente é o tempo.

Utilizando três conjuntos de dados distintos, avaliamos a robustez das arquiteturas propostas, identificando que além dessas treinarem mais rapidamente, elas apresentam desempenhos similares e/ou melhores às arquiteturas apresentadas em [7], [8] e [9].

## II. TRABALHOS RELACIONADOS

Nos últimos anos, as CNNs vêm sendo aplicadas com sucesso no processamento de imagens, linguagem natural e sinais de voz. Em razão da sua reconhecida capacidade em extrair características, uma arquitetura CNN foi introduzida em [7] para classificar 11 tipos de modulações. Os resultados apresentados em [7] demonstram que a CNN supera os resultados obtidos por técnicas como SVM, KNN e Naive Bayes.

Em [8] e [9], os autores estendem o trabalho [7], ajustando parâmetros da CNN e variando a quantidade de camadas convolucionais, assim como o tamanho dos filtros dessas camadas. Além disso, especificamente em [8], combinando as arquiteturas CNN e LSTM (*long short-term memory*), os autores propõem uma arquitetura CLDNN (*convolutional long short-memory deep neural network*), obtendo um resultado superior em relação à CNN adotada em [7]. Uma arquitetura otimizada combinando CNN e LSTM também é desenvolvida em [4]. Recentemente, novas arquiteturas como ResNet (*residual network*) [10] e DenseNet (*densely connected network*) [11] foram introduzidas para a classificação de modulações e alcançaram relativas melhorias na acurácia de classificação.

## III. MODELO DO SISTEMA

De modo geral, o reconhecimento de modulação pode ser modelado como um problema de classificação de  $N$  classes. Uma representação geral do sinal recebido  $r(t)$  é dada por

$$r(t) = s(t) * c(t) + n(t) \quad (1)$$

onde  $s(t)$  denota o envelope complexo em banda-base do sinal transmitido,  $c(t)$  consiste na resposta ao impulso do canal e  $n(t)$  é o ruído gaussiano aditivo branco (AWGN) com média nula e variância  $\sigma_n^2$ .

Isto posto, o objetivo basilar do classificador de modulações é fornecer a probabilidade  $P(s(t) \in N_i | r(t))$  onde  $r(t)$  figura como único sinal de referência e  $N_i$  representa a  $i$ -ésima classe. Para esse propósito, podemos implementar os métodos DL, para os quais, processa-se inicialmente um conjunto de dados de treinamento para definir os parâmetros da rede, e em seguida, a acurácia da classificação é calculada para um conjunto de dados de teste [12].

## A. Conjunto de dados

Neste trabalho, utilizamos três conjuntos de dados<sup>1</sup>, a saber: RadioML2016.04c, RadioML2016.10a e RadioML2016.10b. Todos esses conjuntos de dados são gerados sinteticamente usando o GNU Radio [13] com parâmetros de modulação usados comercialmente. Os dados desses conjuntos incluem imperfeições realistas de canal como deslocamento de taxa de amostragem, AWGN, desvanecimento de múltiplos caminhos e desvio de frequência<sup>2</sup>.

Para todos os conjuntos, os exemplos de treinamento (cada um com 128 amostras) são estruturados em vetores  $2 \times 128$  com as partes real e imaginária separadas em amostras de tempo complexas; os rótulos desses exemplos incluem tanto a razão sinal-ruído (*signal-to-noise ratio*, SNR) quanto o tipo de modulação. As SNRs das amostras estão distribuídas uniformemente no intervalo  $[-20, +18]$  com passo de 2 dB.

A Tabela I elenca os parâmetros do conjunto de dados RadioML2016.10a. A diferença entre os conjuntos RadioML2016.04c e RadioML2016.10a reside no fato de que os dados desse último estão normalizados. O conjunto RadioML2016.10b, por sua vez, além de estar normalizado, não possui a modulação AM-SSB e tem mais exemplos (especificamente, 1.200.000).

TABELA I  
PARÂMETROS DO CONJUNTO DE DADOS RADIOML2016.10A [7]

Modulações	Analógicas: WB-FM, AM-SSB, AM-DSB
	Digitais: BPSK, QPSK, 8PSK, 16QAM, 64QAM, BFSK, CPFSK e PAM4
SNR (dB)	-20 a 18
Amostras por símbolo	8
Nº de exemplos	220000
Dimensão do sinal	$2 \times 128$ amostras
Intervalo de amostragem	$1 \mu\text{s}$

## IV. CAMADAS CONVOLUCIONAIS 1D

Em uma CNN, as camadas convolucionais possuem filtros (núcleos de convolução) para processar o sinal recebido, onde para cada filtro, um neurônio está conectado a apenas um subconjunto dos neurônios na camada anterior [12]. Tipicamente, as camadas convolucionais são 1D, 2D ou 3D e suas devidas aplicações dependem essencialmente do tipo de dado a ser processado.

No contexto do aprendizado de máquina, a operação de convolução unidimensional usualmente empregada é definida por

$$y[i] = \sum_{c=1}^{N_c} \sum_{m=0}^{M-1} x_c[i+m] h_c[m] \quad (2)$$

onde  $x$  refere-se à entrada,  $h$  representa o filtro de tamanho  $M$  e  $N_c$  denota a quantidade de canais. Nesse caso,  $i$  corresponde à posição da sequência onde a convolução é realizada,  $m$  indica a posição dos termos da convolução e o subíndice  $c$  refere-se ao canal em que se aplica a referida operação. A Figura 1 ilustra como ocorre a convolução 1D para uma entrada com  $N_c$  canais.

<sup>1</sup>Disponíveis em: <https://www.deepsig.io/datasets>

<sup>2</sup>Para mais detalhes sobre esses conjuntos de dados consulte [13].

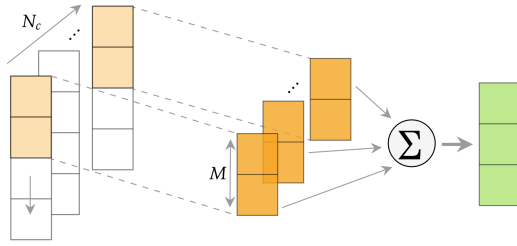


Fig. 1. Representação da ação da convolução 1D com filtros de tamanho  $M$  para uma entrada com  $N_c$  canais.

## V. ARQUITETURAS PROPOSTAS

### A. CNN-1D

A primeira arquitetura proposta refere-se à CNN-1D ilustrada na Figura 2. Em detalhe, essa rede possui três camadas convolucionais 1D e quatro camadas densas. Para todas as camadas convolucionais e as três primeiras camadas densas, usamos a função de ativação ReLU (*rectified linear unit*), enquanto para a última camada, aplicamos a função *softmax* [12]. O emprego da função *softmax* na última camada densa também ocorre para as demais arquiteturas propostas. Para aumentar a capacidade de generalização da CNN-1D incluímos *dropout* [14] de apenas 1% para as camadas densas.

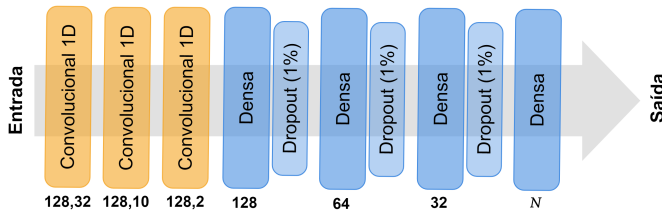


Fig. 2. Arquitetura CNN-1D proposta.

O primeiro parâmetro de cada camada convolucional da Figura 2 representa a quantidade de filtros enquanto o segundo parâmetro denota o tamanho de cada filtro. Para as quatro camadas densas, utilizamos 128, 64, 32 e  $N$  neurônios, em ordem de profundidade da rede, onde  $N$  refere-se à quantidade de classes de modulações do conjunto de dados.

### B. ResNet - 1D

Inspirados pela ResNet apresentada em [10], projetamos uma rede usando somente cinco pilhas residuais seguidas por duas camadas densas conforme ilustra a Figura 3.

Na arquitetura proposta, cada pilha residual é constituída de uma camada convolucional (com função de ativação linear), uma camada de normalização (*BatchNormalization*), uma unidade residual e uma camada de agrupamento (*MaxPooling*) [12]. Em cada unidade residual, há uma conexão de atalho que adiciona a entrada da unidade à saída da segunda camada convolucional da referida unidade. Na primeira camada convolucional da unidade residual, utiliza-se uma função ReLU, enquanto na segunda camada aplica-se a função linear. Ambas as camadas convolucionais da unidade residual são seguidas por camadas de normalização para evitar *overfitting*.

Em detalhe, a penúltima camada densa da ResNet usa uma função de ativação SELU (*scaled exponential linear unit*) e é

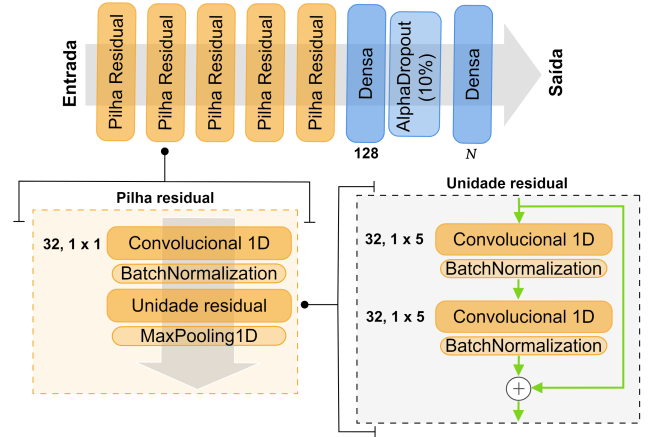


Fig. 3. Arquitetura da ResNet-1D proposta.

acompanhada por uma camada de *Alphadropout* [15] de 10%. Detalhes adicionais como a quantidade e tamanho dos filtros das camadas convolucionais e a quantidade de neurônios das camadas densas podem ser visualizados na Figura 3.

### C. Rede mista - 1D

Conforme mencionado na seção II, as arquiteturas ResNet e DenseNet foram introduzidas com sucesso para AMC. Reconhecendo que tanto a ResNet quanto a DenseNet são derivadas da mesma “topologia densa” [16], diferindo apenas na forma da conexão, projetamos uma arquitetura mista que aproveita os benefícios de ambas arquiteturas.

A arquitetura proposta possui quatro camadas convolucionais e duas camadas densas, onde as duas primeiras camadas convolucionais são seguidas imediatamente por uma camada de agrupamento (*MaxPooling* com *pool size* de tamanho 1) e uma camada dropout de 45%. À terceira camada convolucional, aplica-se somente uma camada *dropout* de 45%. Para as quatro camadas convolucionais e a penúltima camada densa, empregamos a função de ativação ReLU.

As conexões de atalho da arquitetura mista podem ser visualizadas na Figura 4, onde os símbolos “+” e “||” denotam as operações de soma e concatenação, respectivamente.

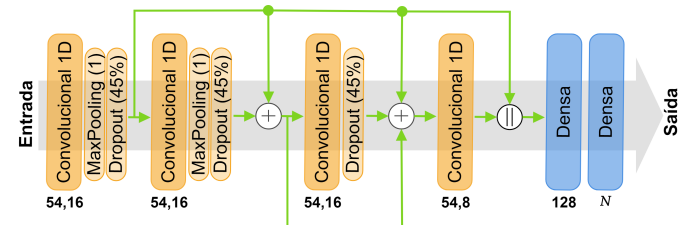


Fig. 4. Arquitetura mista-1D proposta.

## VI. RESULTADOS E DISCUSSÕES

As etapas de treinamento e teste com os conjuntos de dados discutidos na seção III-A foram realizadas usando a biblioteca Keras com Tensorflow como *backend*. Para todas as arquiteturas usamos o otimizador Adam para a otimização dos

TABELA II  
TREINAMENTO DAS ARQUITETURAS PROPOSTAS PARA O CONJUNTO  
RADIOML2016.04C

Arquitetura	Treinamento	
	$N_p$	$T_t$ (s)
CNN-1D	1.641.419	48
ResNet-1D	76.747	88
Mista-1D	1.682.243	55
CNN-2D [7]	2.830.427	560

TABELA III  
TREINAMENTO DAS ARQUITETURAS PROPOSTAS PARA O CONJUNTO  
RADIOML2016.10A

Arquitetura	Treinamento	
	$N_p$	$T_t$ (s)
CNN-1D	1.641.419	130
ResNet-1D	80.235	35
Mista-1D	1.682.243	126
CNN-2D [8]	3.090.033	140

pesos das redes — com uma taxa de aprendizagem de 0,001 — e a entropia cruzada categórica para avaliar a função perda. Além disso, utilizamos um *batch size* de 1024, bem como 100 épocas para o processo de treinamento. As implementações foram feitas na plataforma colab (*google colaboratory*)<sup>3</sup> que nos permitiu o uso da GPU Tesla K80.

Para realizar as simulações, cada conjunto de dados foi dividido em 50% para treinamento e 50% para validação e teste, com exceção do conjunto RadioML2016.04c, no qual, usamos 98060 exemplos para treinamento e 64000 exemplos para teste conforme realizado em [7]. Vale mencionar que a seleção dos dados para treinamento e teste é realizada aleatoriamente.

A quantidade de parâmetros treináveis ( $N_p$ ) e o tempo total ( $T_t$ ), em segundos, que cada arquitetura demanda para treinamento estão apresentados nas Tabelas II, III e IV. Observa-se a partir dessas tabelas que para todos os conjuntos de dados, as arquiteturas propostas levam menos tempo para treinamento quando comparadas com as arquiteturas apresentadas<sup>4</sup> em [7], [8] e [9].

Para o conjunto RadioML2016.04c, os tempos de treinamento da CNN-1D, ResNet-1D e Mista-1D são da ordem de um minuto, tempo este consideravelmente menor quando comparado com aquele que a CNN-2D [7] gasta para treinar. No que concerne à acurácia de classificação (ilustrada na Figura 5), verifica-se que todas as arquiteturas 1D alcançam acurácia superior a 84% para valores de SNR acima de -2 dB. A rede Mista-1D, em geral, tem o melhor desempenho em relação às demais arquiteturas, chegando a alcançar uma acurácia de 98% para 18 dB.

Em relação ao conjunto RadioML2016.10a, para SNRs inferiores a 2 dB, a CNN-2D [8] tem desempenho superior às arquiteturas 1D propostas. Entretanto, para SNRs acima de 2 dB, os resultados da CNN-2D são superados por todos

<sup>3</sup><https://colab.research.google.com/>

<sup>4</sup>Em razão dessas arquiteturas de referência estarem otimizadas em relação aos conjuntos de dados para os quais foram treinadas e testadas, apresentamos os resultados dessas somente para esses respectivos conjuntos de dados.

TABELA IV  
TREINAMENTO DAS ARQUITETURAS PROPOSTAS PARA O CONJUNTO  
RADIOML2016.10B

Arquitetura	Treinamento	
	$N_p$	$T_t$ (s)
CNN-1D	1.641.386	690
ResNet-1D	75.130	713
Mista-1D	1.682.114	663
CNN-2D [9]	324.330	748

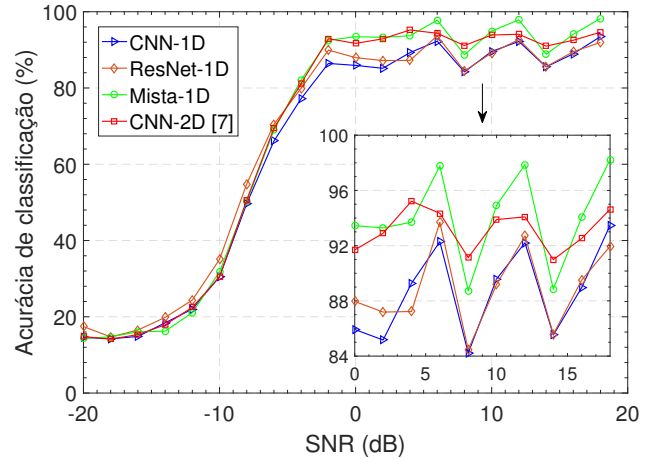


Fig. 5. Comparação das acurácias de classificação das arquiteturas propostas com a CNN-2D usada em [7] para o conjunto de dados RadioML2016.04c.

aqueles das arquiteturas 1D cujas acurácias ultrapassam em aproximadamente 3% os resultados da CNN-2D [8] conforme pode ser visualizado na Figura 6. Nesse caso, é oportuno enfatizar que a arquitetura que apresenta a melhor acurácia (ResNet-1D) para altas SNRs, também é a mais rápida para treinar.

Os resultados do desempenho das arquiteturas 1D para o conjunto RadioML2016.10b estão apresentados na Figura 7. Observa-se que a partir de -2 dB, as acurácias tanto da Resnet-1D quanto da Mista-1D ultrapassam efetivamente os resultados da CNN-2D [9]. Para SNRs elevadas, especificamente as arquiteturas Mista-1D e ResNet, superam em cerca de 3% e 6% a acurácia da CNN-2D [9], respectivamente.

Apresentamos na Figura 8, a matriz de confusão (especificamente SNR = 8 dB) para avaliar o quão efetiva é a classificação obtida através da ResNet-1D para cada tipo de modulação do conjunto RadioML2016.10b. Nessa matriz, as linhas e colunas correspondem às modulações corretas (rótulos verdadeiro) e às modulações previstas (rótulos preditos), nessa ordem, enquanto a diagonal principal representa para cada modulação o percentual de classificações corretas da modulação em relação à quantidade total de dados dessa modulação.

É possível observar que as maiores confusões ocorrem entre as modulações digitais 16-QAM 64-QAM e as modulações analógicas AM-DSB e WBFM. A confusão entre as modulações QAM se deve ao fato de elas, eventualmente, ocuparem a mesma região com uma combinação de símbolos específicos, visto que a constelação 16-QAM está contida

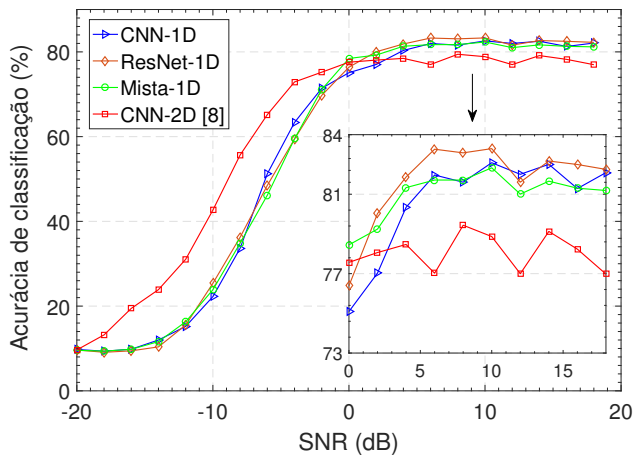


Fig. 6. Comparação das acurácias de classificação das arquiteturas propostas com a CNN-2D usada em [8] para o conjunto de dados RadioML2016.10a.

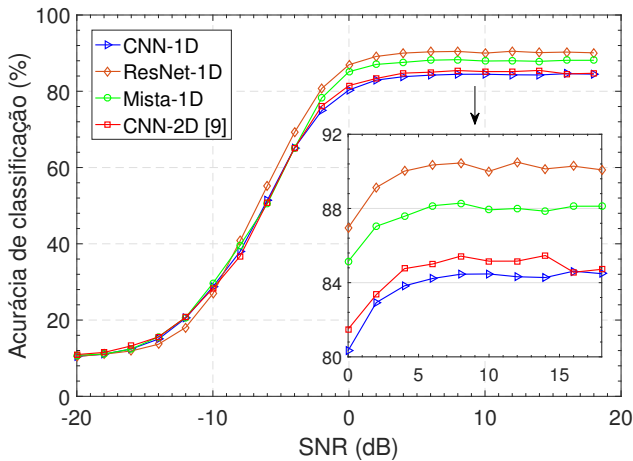


Fig. 7. Comparação das acurácias de classificação das arquiteturas propostas com a CNN-2D usada em [9] para o conjunto de dados RadioML2016.10b.

na constelação 64-QAM. Já a confusão entre as modulações analógicas ocorre pelo fato de a mensagem ser um sinal de voz. Nesse caso, temos períodos de silêncio onde apenas a portadora está sendo enviada.

## VII. CONCLUSÕES

Neste trabalho, diferentemente da abordagem utilizada na literatura que aplica redes convolucionais 2D para AMC, propomos diferentes tipos de arquiteturas (CNN, ResNet e Mista) que usam filtros 1D para tal tarefa. Utilizando diferentes conjuntos de dados para treinamento e teste, verificamos que as arquiteturas 1D propostas são robustas, alcançando desempenhos similares ou superiores aos resultados dos trabalhos relacionados. Particularmente, a ResNet-1D e a Mista-1D, obtiveram acurácias de classificação significativas. Para todos os conjuntos de dados, as arquiteturas 1D exigiram menos tempo para treinamento.

Os resultados obtidos evidenciam assim que as arquiteturas 1D são bem-sucedidas em estabelecer um compromisso entre a acurácia de classificação e o tempo gasto para treinamento.

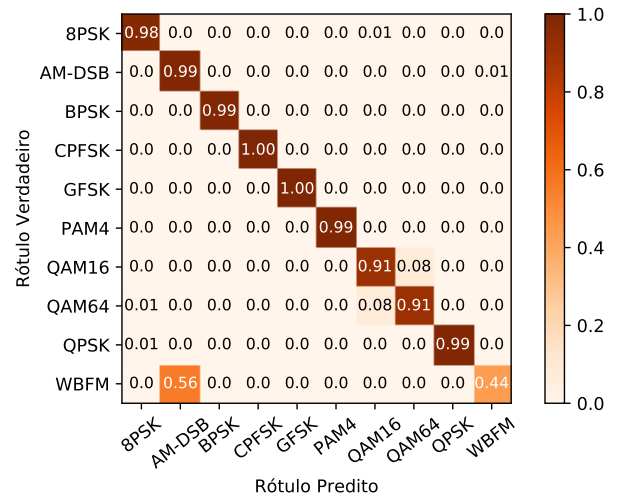


Fig. 8. Matriz de confusão da ResNet-1D em SNR = 8 dB para o conjunto RadioML2016.10b.

## REFERÊNCIAS

- [1] S. Ghodeswar and P. G. Poonacha, "Modulation recognition techniques for improving communication efficiency in SDR networks," in *2013 4th International Conference on Computer and Communication Technology (ICCT)*, Sep. 2013, pp. 233–238.
- [2] S. Zhou, Z. Yin, Z. Wu, Y. Chen, N. Zhao, and Z. Yang, "A robust modulation classification method using convolutional neural networks," *EURASIP Journal on Advances in Signal Processing*, vol. 2019, no. 1, p. 21, Mar 2019.
- [3] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 433–445, Sep. 2018.
- [4] Y. Wu, X. Li, and J. Fang, "A deep learning approach for modulation recognition via exploiting temporal correlations," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, June 2018, pp. 1–5.
- [5] G. J. Mendis, J. Wei, and A. Madanayake, "Deep learning-based automated modulation classification for cognitive radio," in *2016 IEEE International Conference on Communication Systems (ICCS)*, Dec 2016, pp. 1–6.
- [6] A. Dai, H. Zhang, and H. Sun, "Automatic modulation classification using stacked sparse auto-encoders," in *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, Nov 2016, pp. 248–252.
- [7] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Engineering Applications of Neural Networks*, August 2016, pp. 213–226.
- [8] N. E. West and T. O'Shea, "Deep architectures for modulation recognition," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, March 2017, pp. 1–6.
- [9] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, Dec 2017.
- [10] T. J. O'Shea, T. Roy, and T. C. 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, Feb 2018.
- [11] X. Liu, D. Yang, and A. E. Gamal, "Deep neural network architectures for modulation classification," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, Oct 2017, pp. 915–919.
- [12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [13] T. O'Shea and N. West, "Radio machine learning dataset generation with gnu radio," *Proceedings of the GNU Radio Conference*, vol. 1, no. 1, 2016.
- [14] H. Wu and X. Gu, "Towards dropout training for convolutional neural networks," *Neural Networks*, vol. 71, pp. 1–10, 2015.
- [15] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," *CoRR*, vol. abs/1706.02515, 2017.
- [16] W. Wang, X. Li, J. Yang, and T. Lu, "Mixed link networks," *CoRR*, vol. abs/1802.01808, 2018.