

# Redes MLP distribuídas para classificação de arritmias cardíacas

Lucca Gamballi, Daniel G. Tiglea, Renato Candido e Magno T. M. Silva

**Resumo**— Neste artigo, redes neurais perceptron multicamada são utilizadas para classificar arritmias cardíacas com uma abordagem distribuída. Os dados de treinamento são divididos entre redes que se comunicam por meio de uma determinada topologia, sem que uma rede tenha acesso aos dados de treinamento das outras. Essa abordagem é empregada para garantir a privacidade dos dados dos pacientes. Para obter um resultado clinicamente realista, dados de um mesmo paciente não são considerados concomitantemente nos conjuntos de treinamento e teste. Simulações indicam que o desempenho obtido com o treinamento distribuído e uma topologia adequada é semelhante ao observado com o treinamento clássico.

**Palavras-Chave**— Redes neurais artificiais, perceptron multicamada, processamento distribuído, topologia, arritmias cardíacas.

**Abstract**— In this paper, multilayer perceptron neural networks are used to classify cardiac arrhythmias with a distributed approach. The training data is split between networks that communicate through a certain topology, in which each network does not have access to the training data of the others. This approach is employed to ensure patient data privacy. To obtain a clinically realistic result, data from the same patient are not considered concurrently in the training and test sets. Simulations indicate that the performance obtained with distributed training and a suitable topology is similar to the one observed with the classical training.

**Keywords**— Artificial neural networks, multi-layer perceptron, distributed processing, topology, cardiac arrhythmia.

## I. INTRODUÇÃO

A detecção e a classificação de arritmias cardíacas dependem de uma análise precisa da atividade elétrica do coração, registrada no sinal de eletrocardiograma (ECG) [1]. O sinal de ECG contém informações clínicas que permitem que essa tarefa seja realizada por especialistas, mas pode demandar muito tempo e é inviável quando se necessita de resultados imediatos. Diante disso, soluções baseadas em técnicas de aprendizado de máquina para a classificação automática de arritmias cardíacas têm sido amplamente propostas na literatura [1]–[6].

A principal abordagem em aprendizado de máquina é o treinamento supervisionado, em que um modelo é ajustado para mapear exemplos de entrada a rótulos pertencentes a um banco de dados de treinamento. Em geral, o treinamento é realizado em uma única máquina, o que pode não ser adequado quando se trabalha com uma grande quantidade de dados ou quando a privacidade é importante [7], [8]. Atualmente, a

aplicação de técnicas de aprendizado de máquina na medicina, por exemplo, é dificultada pela disponibilidade limitada de conjuntos de dados para treinamento e teste dos algoritmos. Isso ocorre devido à ausência de registros médicos eletrônicos padronizados e devido aos requisitos legais e éticos estritos para proteger a privacidade do paciente. Para garantir a privacidade do paciente e, ao mesmo tempo, promover pesquisas científicas com grandes conjuntos de dados, a busca por soluções que atendem simultaneamente às demandas de proteção e utilização dos dados tem crescido [7]–[10].

Nesse contexto, o treinamento de redes neurais de forma distribuída tem despertado interesse recentemente na literatura [7], [8]. Na abordagem distribuída, cada máquina treina um modelo local utilizando apenas uma parte dos dados e troca informações acerca de seu modelo com máquinas vizinhas, sem compartilhar os dados de treinamento. Para comunicação entre as máquinas considera-se uma topologia pré-definida. Assim, pode-se obter um único modelo global utilizando todos os dados de treinamento sem que uma única máquina tenha acesso a todos eles [8].

Neste artigo, propõe-se o uso de redes neurais perceptron multicamada (*multilayer perceptron* – MLP) treinadas com uma abordagem distribuída para o diagnóstico automático de arritmias cardíacas. Foram considerados sinais de ECG do banco de dados *MIT-BIH Arrhythmia Database* (MITDB) [11], [12], levando-se em conta a separação dos dados dos pacientes nas fases de treinamento e teste. Conforme recomendações da *Association for the Advancement of Medical Instrumentation* (AAMI) [13], os conjuntos de dados de treinamento e de teste dos classificadores de arritmia não devem conter dados dos mesmos pacientes, já que na prática o sistema será utilizado em pacientes cujos dados não foram usados no treinamento.

O artigo está organizado da seguinte forma. Na Seção II, aborda-se o treinamento distribuído de redes MLP. Na Seção III, descrevem-se o banco de dados e a extração de características utilizados e as classes de arritmia consideradas. As Seções IV e V contêm respectivamente os resultados de simulação e as principais conclusões do trabalho.

## II. TREINAMENTO DISTRIBUÍDO DE REDES MLP

Consideremos um problema de aprendizagem supervisionada com um conjunto de treinamento  $\mathcal{D}$  de cardinalidade  $N$ . A ideia do treinamento distribuído de redes MLP consiste em dividir  $\mathcal{D}$  em  $V$  subconjuntos disjuntos de cardinalidade  $N_i$ , denotados por  $\mathcal{D}_i$ ,  $i = 1, 2, \dots, V$ , tal que  $\sum_{i=1}^V N_i = N$ . Os exemplos do subconjunto  $\mathcal{D}_i$  são utilizados no treinamento da MLP da máquina  $i$ . As  $V$  máquinas consideradas se comunicam por meio de uma determinada topologia, como

a da Fig. 1. Nessa figura, cada máquina corresponde a um nó do grafo e as arestas indicam a existência de uma conexão entre determinados pares de máquinas. A vizinhança do nó  $i$ , denotada por  $\mathcal{V}_i$ , é o conjunto de nós com os quais ele consegue se comunicar diretamente (incluindo ele próprio).

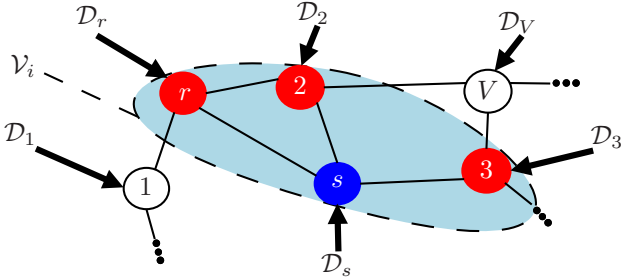


Fig. 1: Exemplo de uma rede de difusão, em que cada nó  $i \in \{1, 2, \dots, V\}$  treina uma rede MLP com base no subconjunto de treinamento  $\mathcal{D}_i$ . Neste exemplo, a vizinhança do nó  $s$  é  $\mathcal{V}_s = \{2, 3, r, s\}$ .

O objetivo de cada nó  $i$  é treinar uma rede MLP utilizando o subconjunto  $\mathcal{D}_i$  a fim de minimizar uma função custo, cuja forma é comum a todos os nós do grafo. Considerando um problema de classificação com  $C$  classes, é comum adotar como função custo a entropia cruzada categórica ponderada. Assim, a MLP da  $i$ -ésima máquina minimiza [14]

$$J_{\text{ECCP}_i} = - \sum_{n_i=1}^{N_i} \sum_{\ell=1}^C p_{i,\ell} d_{i,\ell}(n_i) \ln[y_{i,\ell}(n_i)], \quad (1)$$

em que  $y_{i,\ell}(n_i)$  é a  $\ell$ -ésima saída da rede para o  $n_i$ -ésimo dado de treinamento com rótulo  $d_{i,\ell}(n_i)$ ,  $\ell = 1, 2, \dots, C$  e  $n_i = 1, 2, \dots, N_i$ . Por fim, os pesos  $p_{i,\ell}$  são utilizados em (1) para conferir uma ponderação maior a classes minoritárias [14]. Inspirando-se na heurística de [15], considerou-se neste trabalho o peso  $p_{i,\ell} = \lceil 10N_i / (CB_{i,\ell}) \rceil$ , em que  $B_{i,\ell}$  é o número de amostras da classe  $\ell$  em  $\mathcal{D}_i$ .

A matriz de pesos (incluindo os *biases*) da  $k$ -ésima camada da MLP do nó  $i$ , denotada por  $\mathbf{W}_i^{[k]}$ , pode ser atualizada com o algoritmo de retropropagação do erro (*backpropagation*) [16], [17]. Para garantir a privacidade dos dados, cada nó  $i$  pode trocar informações com a sua vizinhança acerca de seu próprio modelo, mas não sobre seus dados de treinamento. Assim, a matriz  $\mathbf{W}_i^{[k]}$  é compartilhada com os nós vizinhos após cada atualização. A ordem em que essas operações são realizadas dá origem a dois tipos de configuração: adaptação seguida da combinação (ATC – *adapt-then-combine*) e combinação seguida da adaptação (CTA – *combine-then-adapt*) [8]. Por simplicidade, neste artigo será considerada apenas a configuração ATC, embora resultados semelhantes possam ser obtidos com a configuração CTA. Assim, a regra para atualização de  $\mathbf{W}_i^{[k]}$  pode ser descrita por [8]

$$\begin{cases} \Psi_i^{[k]}(m+1) = \mathbf{W}_i^{[k]}(m) - \eta \frac{\partial J_{\text{ECCP}_i}}{\partial \mathbf{W}_i^{[k]}(m)} & (2a) \\ \mathbf{W}_i^{[k]}(m+1) = \sum_{j \in \mathcal{V}_i} a_{ij} \Psi_j^{[k]}(m+1), & (2b) \end{cases}$$

em que  $\Psi_i^{[k]}$  é uma estimativa intermediária de mesma dimensão que  $\mathbf{W}_i^{[k]}$ ,  $\eta$  é o passo de adaptação,  $m$  é a iteração

do algoritmo e  $a_{ij} \geq 0$  são pesos de combinação, tal que  $a_{ij} = 0$  se as máquinas  $i$  e  $j$  não estão diretamente conectadas e  $\sum_{j=1}^V a_{ij} = 1$  para todo  $i, j$ . Existem diversas regras possíveis para a escolha desses pesos. Neste trabalho, adota-se a regra Metropolis, dada por [7], [18]

$$a_{ij} = \begin{cases} \frac{1}{\max\{|\mathcal{V}_i|, |\mathcal{V}_j|\}}, & \text{se } j \in \mathcal{V}_i \text{ e } j \neq i \\ 1 - \sum_{q \in \mathcal{V}_i} a_{iq}, & \text{se } j = i \\ 0, & \text{caso contrário} \end{cases}, \quad (3)$$

em que  $|\cdot|$  denota cardinalidade. O caso particular em que  $a_{ij} = 1$  se  $i = j$  e  $a_{ij} = 0$  caso contrário corresponde a um cenário de treinamento individual, ou não-cooperativo, em que cada nó treina sua MLP com base em  $\mathcal{D}_i$  sem interagir com os demais. Pode-se demonstrar que, adotando-se a abordagem cooperativa e inicializando-se as redes MLP distribuídas da mesma maneira, o treinamento distribuído leva a um único modelo global. Nesse caso, o desempenho como um todo da redes MLP distribuídas é equivalente ao obtido com uma única rede MLP treinada com o conjunto  $\mathcal{D}$ , sem que nenhuma máquina tenha acesso a todos os elementos desse conjunto simultaneamente [8].

### III. BANCO DE DADOS E ENTRADA DA MLP

O banco de dados de arritmia do MIT contém 48 gravações de 30 min. de dois sinais de derivação de ECG, denotados como derivação A e derivação B [11]. Na maioria das gravações, a derivação A é uma derivação II modificada e a derivação B é a derivação precordial V1. Os dados são filtrados com um filtro passa-faixa 0,1–100 Hz e amostrados em 360 Hz. Seguindo o padrão da AAMI, as quatro gravações de pacientes com marcapasso foram retiradas da análise e as gravações restantes foram divididas em dois conjuntos de dados, denotados como DS1 e DS2 de acordo com a Tab. I. Cada conjunto de dados contém aproximadamente  $5 \times 10^4$  batimentos cardíacos e tem uma mistura de arritmias de rotina (“100 series”) e complexas (“200 series”) [12]. Neste trabalho, o DS1 foi utilizado no treinamento e o DS2 no teste dos classificadores, conforme feito em [6], [19].

TABELA I: Esquema de divisão interpaciente de [6], [19].

|     |     |     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| DS1 | 101 | 106 | 108 | 109 | 112 | 114 | 115 | 116 | 118 | 119 | 122 |
|     | 124 | 201 | 203 | 205 | 207 | 208 | 209 | 215 | 220 | 223 | 230 |
| DS2 | 100 | 103 | 105 | 111 | 113 | 117 | 121 | 123 | 200 | 202 | 210 |
|     | 212 | 213 | 214 | 219 | 221 | 222 | 228 | 231 | 232 | 233 | 234 |

Cada batimento cardíaco do MITDB contém anotações feitas por especialistas com as informações de diagnóstico. No total, existem 15 anotações de diagnóstico diferentes. A AAMI recomenda agrupar essas 15 anotações em cinco classes: batimentos do nó sinoatrial (N), batimentos ectópicos supraventriculares (S), batimentos ectópicos ventriculares (V), fusão de batimentos ectópicos normais e ventriculares (F) e batimentos desconhecidos ou com marcapasso (Q). Devido à quantidade reduzida de dados das classes Q e F, elas não foram consideradas neste trabalho. Portanto, o problema se reduz à classificação em três classes: N, S e V. Clinicamente, as classes S e V contêm batimentos cardíacos das arritmias mais graves. Portanto, muitos trabalhos têm focado na obtenção de classificadores capazes de distinguir essas classes [20]–[22].

O diagnóstico automático de arritmias cardíacas com redes neurais pode ser dividido em duas etapas: segmentação dos batimentos cardíacos e classificação. Em alguns trabalhos, uma etapa de pré-processamento também é incluída para reduzir ruídos e artefatos [1]. A etapa de segmentação fornece o delineamento do sinal de ECG com a determinação dos picos e limites das ondas QRS, P e T [23]. Já o classificador decide a classe do batimento com base em suas características. Para a etapa de segmentação, foi utilizado o algoritmo de Pan-Tompkins [24]. Na etapa de classificação, foram utilizadas redes MLP, implementadas com as bibliotecas Tensorflow e Keras [25]. Seguindo [6], [19], a etapa de pré-processamento para reduzir ruídos e artefatos não foi considerada aqui.

Para determinar a entrada da MLP, cada batimento cardíaco foi considerado como contendo 300 amostras do sinal de ECG, o que representa aproximadamente o número médio de amostras dos batimentos cardíacos contidos no MITDB. Este trecho é então centralizado utilizando a posição do pico R. Utilizando diferentes números de batimentos cardíacos na classificação, observou-se em [6] que as informações dos batimentos anteriores e posteriores contribuem para aumentar as métricas de classificação para a classe S. Além disso, as gravações da derivação A apresentam as métricas de classificação mais altas na maioria dos casos. Portanto, baseando-se nos resultados de [6], como entrada das redes MLP, foram considerados trechos de 900 amostras do sinal de ECG da derivação A, centralizado em torno do pico R do batimento cardíaco cuja classificação se deseja obter. Na Fig. 2, é mostrado um exemplo de entrada da rede MLP.

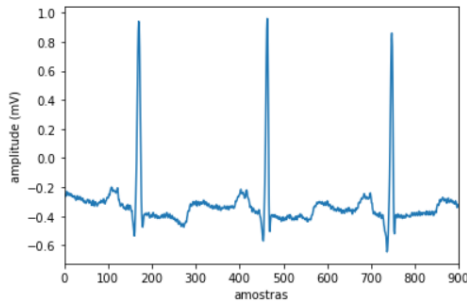


Fig. 2: Exemplo de entrada da rede MLP.

#### IV. RESULTADOS DE SIMULAÇÃO

Cada classe do MITDB possui um número diferente de exemplos, o que torna o uso desse conjunto de dados desafiador. Para lidar com esse desbalanceamento, considerou-se a entropia cruzada categórica ponderada como detalhado na Seção II. Além disso, foram consideradas duas situações: a situação  $S_1$  em que apenas a classe N foi subamostrada e a a situação  $S_2$  em que todas as classes foram subamostradas. Na Tab. II, são mostradas a quantidade de dados disponíveis no MITDB e a quantidade de dados usados neste trabalho em cada situação [6].

TABELA II: Total de batimentos do MITDB e quantidade de batimentos utilizados para os conjuntos de treinamento e teste.

| Classe | Total de batimentos |       | Quantidade de batimentos utilizados |       |       |      |
|--------|---------------------|-------|-------------------------------------|-------|-------|------|
|        |                     |       | $S_1$                               |       | $S_2$ |      |
|        | DS1                 | DS2   | DS1                                 | DS2   | DS1   | DS2  |
| N      | 45784               | 44173 | 9448                                | 10108 | 5940  | 6347 |
| S      | 940                 | 1837  | 940                                 | 1837  | 689   | 1390 |
| V      | 3783                | 3218  | 3783                                | 3218  | 2819  | 2371 |

Foram considerados três tipos de treinamento: o centralizado, o individual e o distribuído, sendo que para os dois últimos foram consideradas as topologias ilustradas na Fig. 3 [7]. Na Fig. 4, é apresentado um esquema de cada tipo de treinamento para a topologia da Fig. 3(a). Cada nó do grafo representa uma máquina onde é treinada uma MLP. As redes MLP consideradas são compostas por duas camadas ocultas com 32 e 16 neurônios cada e função de ativação ReLU. A camada de saída é composta de três neurônios com a função *Softmax*. Por fim, foi considerado o algoritmo de otimização Adam [26] com parâmetros  $\beta_1 = 0,9$ ,  $\beta_2 = 0,99$  e  $\epsilon = 10^{-7}$ . Os pesos foram inicializados de acordo com a inicialização Glorot [27] e foi considerado o *backpropagation* com passo de adaptação  $\eta = 0,001$ , tamanho de mini-batch  $k = 2048$  e 1600 épocas de treinamento. Cabe observar que para comparar o desempenho dos treinamentos centralizado, individual e distribuído, a arquitetura da rede e os parâmetros do algoritmo de treinamento não influenciam, desde que sejam mantidos iguais em cada abordagem. Diante disso, foram escolhidos os parâmetros e as arquiteturas descritas anteriormente por apresentarem bons desempenhos de classificação [6].

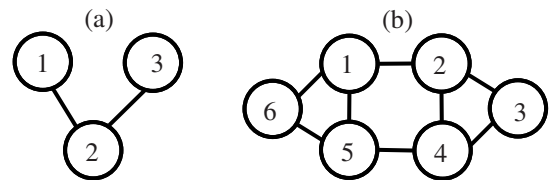


Fig. 3: Topologias com (a) 3 e (b) 6 máquinas.

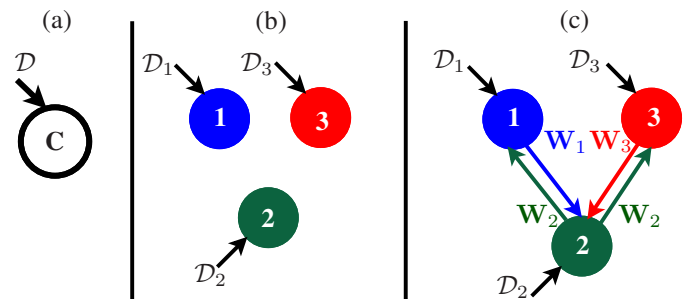


Fig. 4: Tipos de treinamento: (a) Centralizado, (b) Individual e (c) Distribuído, considerando a topologia da Fig. 3(a).

Na Tab. III, são mostrados para o cenário  $S_1$  as seguintes métricas:

sensibilidade

$$Se = VP/(VP + FN),$$

precisão

$$P = VP/(VP + FP),$$

F1-score

$$F1 = 2 \times P \times Se/(P + Se) \quad e$$

coeficiente de correlação de Matthews

$$MCC = \frac{VP \times VN - FP \times FN}{\sqrt{(VP + FP)(VP + FN)(VN + FP)(VN + FN)}}$$



alcançados com cada abordagem para os dados de teste, sendo  $VP$ ,  $VN$ ,  $FP$  e  $FN$  as quantidades de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos, respectivamente. Cabe observar que  $-1 \leq MCC \leq 1$ , sendo que são desejáveis resultados próximos de 1, em que a classificação é perfeita. Na Tab. IV, as mesmas medidas são mostradas para o cenário  $S_2$ . Em cada linha das tabelas, destaca-se em negrito a maior métrica alcançada, sendo que no caso individual, considerou-se a média das máquinas.

Analisando-se as Tabs. III e IV, nota-se que o treinamento individual resultou nas piores métricas tanto no caso  $S_1$  como no caso  $S_2$ . Apesar disso, os resultados do treinamento individual são comparáveis aos dos treinamentos centralizado e distribuído no caso  $S_1$  para a sensibilidade da classe N e para precisão, F1-score e MCC da classe V. Já no cenário  $S_2$ , o treinamento individual produziu desempenhos semelhantes aos observados para os treinamentos centralizado e distribuído apenas para a sensibilidade das classes N e V. Ainda assim, a sensibilidade para a classe N só é comparável aos demais tipos de treinamento considerando-se o treinamento individual com a topologia da Fig. 3(b). Em contrapartida, para a sensibilidade da classe V, apenas a topologia da Fig. 3(a) leva a um desempenho por parte do treinamento individual comparável aos treinamentos centralizado e distribuído. Cabe ressaltar que em ambos os cenários as métricas obtidas com o treinamento individual para a classe S foram significativamente inferiores àquelas alcançadas com os demais tipos de treinamento, qualquer que seja a topologia considerada.

Ainda com relação à Tab. III, nota-se que a diferença entre as métricas alcançadas com a média das máquinas no treinamento distribuído e o treinamento centralizado no cenário  $S_1$  é inferior em módulo a 1,5% para a sensibilidade, precisão e F1-score de todas as classes, exceto para a precisão e o F1-score da classe V. Nesses últimos casos, as métricas obtidas com o treinamento centralizado foram superiores, sendo que o treinamento distribuído considerando-se a topologia da Fig. 3(a) apresentou desempenho mais próximo do treinamento centralizado do o caso em que se considera a topologia da Fig. 3(b). Analogamente, a diferença entre o MCC alcançado com o treinamento distribuído e aquele atingido com o treinamento centralizado foi de no máximo 0,02 em módulo, exceto para a classe V considerando-se o treinamento distribuído com a topologia da Fig. 3(b). Nesse caso, a diferença foi de 0,04, com o treinamento distribuído apresentando o melhor desempenho.

Realizando-se a mesma análise para o cenário  $S_2$ , nota-se que o treinamento distribuído com a topologia da Fig. 3(a) levou a resultados superiores aos do treinamento centralizado para todas as métricas, exceto para a precisão da classe N e para a sensibilidade das classes S e V. Por outro lado, nota-se que a adoção da topologia da Fig. 3(b) em vez daquela da Fig. 3(a) leva a um desempenho consideravelmente inferior para o treinamento distribuído, exceto para a sensibilidade da classe N e para as métricas da classe V. Em particular, nota-se uma degradação significativa para as métricas da classe S, com diferenças superiores a 30% para a sensibilidade, precisão e F1-score entre os treinamentos distribuídos considerando-se a topologia da Fig. 3(a) e aquela da Fig. 3(b). Em comparação

com o treinamento centralizado, o treinamento distribuído com a topologia da Fig. 3(b) levou a desempenhos melhores apenas para a sensibilidade da classe N e para as métricas da classe V. Uma possível interpretação para isso reside no fato de que quando a rede de computadores apresenta muitas máquinas, a divisão do conjunto de treinamento  $\mathcal{D}$  pode levar a subconjuntos  $\mathcal{D}_i$  com cardinalidades muito pequenas. Nesse caso, a quantidade de exemplos a que cada máquina tem acesso pode ser insuficiente para a obtenção de um modelo adequado, o que não é mitigado pela etapa de combinação representada pela Eq. (2b). Cabe notar que a topologia da Fig. 3(b) apresenta o dobro do número de máquinas da topologia da Fig. 3(a). Além disso, a comparação entre os cenários  $S_1$  e  $S_2$  sugere que esse efeito é agravado pela subamostragem de todas as classes, o que implica ainda menos exemplos disponíveis para o treinamento de cada máquina quando se considera a topologia da Fig. 3(b) para o cenário  $S_2$ .

## V. CONCLUSÕES

A classificação automática de arritmias utilizando métodos computadorizados é uma área com muito espaço para desenvolvimento. Por sua vez, o treinamento distribuído de redes neurais tem grande potencial para aplicações práticas, uma vez que visa resolver problemas de privacidade que constituem uma preocupação real em diversos tipos de situações. Neste trabalho, avaliaram-se os efeitos da adoção de treinamentos distribuídos para redes perceptron multicamadas. Constatou-se que com esse tipo de treinamento pode-se alcançar um desempenho comparável com a abordagem centralizada, em que se utiliza uma única máquina para treinar a rede neural com todos os dados do conjunto de treinamento. Nesse caso, pode-se conciliar o desempenho com a proteção à privacidade dos dados. Os resultados mostram que, para isso, é necessário adotar uma topologia adequada. Em particular, deve-se garantir que o número de máquinas não seja excessivamente elevado para que a divisão do conjunto de dados entre as máquinas não leve a subconjuntos de treinamento muito pequenos.

## REFERÊNCIAS

- [1] S. K. Berkaya, A. K. Uysal, E. S. Gunal, S. Ergin, S. Gunal, and M. B. Gulmezoglu, "A survey on ECG analysis," *Biomedical Signal Processing and Control*, vol. 43, pp. 216–235, 2018.
- [2] R. Banerjee, A. Ghose, and S. Khandelwal, "A novel recurrent neural network architecture for classification of atrial fibrillation using single-lead ECG," in *Proc. of 27th European Signal Processing Conference (EUSIPCO)*, Sep. 2019, pp. 1–5.
- [3] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, and A. Y. Ng, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nature Medicine*, vol. 25, pp. 65–69, Jan. 2019.
- [4] U. R. Acharya, H. Fujita, O. S. Lih, Y. Hagiwara, J. H. Tan, and M. Adam, "Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network," *Information Sciences*, vol. 405, pp. 81–90, Sep. 2017.
- [5] U. R. Acharya, H. Fujita, O. S. Lih, M. Adam, J. H. Tan, and C. K. Chua, "Automated detection of coronary artery disease using different durations of ECG segments with convolutional neural network," *Knowledge-Based Systems*, vol. 132, pp. 62–71, Sep. 2017.
- [6] N. Nagata, R. Candido, and M. T. M. Silva, "Combinacões de redes neurais e discriminantes lineares para classificacão de arritmias cardíacas," in *Anais do Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBt'21)*, Fortaleza, CE, 2021.
- [7] B. Liu and Z. Ding, "Distributed heuristic adaptive neural networks with variance reduction in switching graphs," *IEEE Transactions on Cybernetics*, vol. 51, pp. 3836–3844, 2019.

TABELA III: Métricas de desempenho para o caso  $S_1$ .

|   | Central | Individual    |               | Distribuído   |              |       |              |               |       |              |       |       |              |             |       |
|---|---------|---------------|---------------|---------------|--------------|-------|--------------|---------------|-------|--------------|-------|-------|--------------|-------------|-------|
|   |         | Topologia (a) | Topologia (b) | Topologia (a) |              |       |              | Topologia (b) |       |              |       |       |              |             |       |
|   |         |               |               | Nó 1          | Nó 2         | Nó 3  | Média        | Nó 1          | Nó 2  | Nó 3         | Nó 4  | Nó 5  | Nó 6         | Média       |       |
| N | Se(%)   | 85,68         | 88,22         | 87,67         | 86,62        | 83,97 | 87,08        | 85,89         | 83,59 | <b>91,95</b> | 83,52 | 83,13 | 83,68        | 83,06       | 84,82 |
|   | P(%)    | 90,99         | 81,43         | 80,96         | 90,45        | 91,18 | 91,42        | 91,02         | 91,94 | 90,53        | 92,03 | 91,20 | <b>92,30</b> | 90,81       | 91,47 |
|   | F1(%)   | 88,26         | 84,68         | 84,18         | 86,90        | 87,43 | <b>89,20</b> | 87,84         | 87,57 | 86,02        | 87,56 | 86,98 | 87,78        | 86,76       | 87,11 |
|   | MCC     | 0,67          | 0,51          | 0,49          | 0,65         | 0,65  | <b>0,69</b>  | 0,66          | 0,67  | 0,64         | 0,65  | 0,65  | 0,65         | 0,66        | 0,65  |
| S | Se(%)   | 66,63         | 9,05          | 5,67          | 66,72        | 65,61 | 67,13        | 66,49         | 71,58 | 72,62        | 67,66 | 71,08 | <b>73,84</b> | 70,20       | 71,16 |
|   | P(%)    | 64,69         | 29,63         | 22,55         | 64,86        | 63,58 | <b>69,89</b> | 66,11         | 61,11 | 64,89        | 58,91 | 64,71 | 63,00        | 66,67       | 63,22 |
|   | F1(%)   | 65,65         | 13,82         | 9,01          | 65,78        | 64,58 | 68,48        | 66,28         | 65,93 | <b>68,54</b> | 62,98 | 67,74 | 67,99        | 68,39       | 66,93 |
|   | MCC     | 0,61          | 0,10          | 0,05          | 0,61         | 0,59  | <b>0,64</b>  | 0,61          | 0,61  | <b>0,64</b>  | 0,58  | 0,63  | <b>0,64</b>  | <b>0,64</b> | 0,62  |
| V | Se(%)   | 90,18         | 86,46         | 86,62         | <b>90,87</b> | 90,32 | 89,62        | 90,27         | 90,37 | 88,10        | 89,44 | 88,63 | 88,13        | 89,38       | 89,01 |
|   | P(%)    | <b>77,34</b>  | 76,72         | 75,21         | 75,24        | 72,23 | 76,61        | 74,76         | 76,79 | 72,66        | 72,12 | 72,33 | 72,13        | 73,73       | 73,29 |
|   | F1(%)   | 83,27         | 81,42         | 80,49         | <b>85,32</b> | 80,27 | 82,72        | 81,77         | 83,02 | 79,64        | 79,85 | 79,65 | 79,33        | 80,80       | 80,38 |
|   | MCC     | <b>0,79</b>   | 0,76          | 0,75          | 0,77         | 0,75  | 0,78         | 0,77          | 0,78  | 0,74         | 0,75  | 0,74  | 0,74         | 0,75        | 0,75  |

 TABELA IV: Métricas de desempenho para o caso  $S_2$ .

|   | Central | Individual    |               | Distribuído   |              |              |              |               |              |       |       |       |       |       |       |
|---|---------|---------------|---------------|---------------|--------------|--------------|--------------|---------------|--------------|-------|-------|-------|-------|-------|-------|
|   |         | Topologia (a) | Topologia (b) | Topologia (a) |              |              |              | Topologia (b) |              |       |       |       |       |       |       |
|   |         |               |               | Nó 1          | Nó 2         | Nó 3         | Média        | Nó 1          | Nó 2         | Nó 3  | Nó 4  | Nó 5  | Nó 6  | Média |       |
| N | Se(%)   | 84,04         | 83,67         | 85,56         | 88,45        | 88,94        | <b>89,20</b> | 88,86         | 82,93        | 84,01 | 87,96 | 82,51 | 86,44 | 86,75 | 85,10 |
|   | P(%)    | <b>90,01</b>  | 78,39         | 77,72         | 89,67        | 88,77        | 89,80        | 89,41         | 83,09        | 82,99 | 83,08 | 82,12 | 82,78 | 83,79 | 82,97 |
|   | F1(%)   | 86,92         | 80,93         | 81,43         | 89,05        | 88,85        | <b>89,50</b> | 89,13         | 83,01        | 83,5  | 85,45 | 82,31 | 84,57 | 85,25 | 84,02 |
|   | MCC     | 0,67          | 0,46          | 0,46          | 0,71         | 0,70         | <b>0,72</b>  | 0,71          | 0,53         | 0,55  | 0,59  | 0,53  | 0,57  | 0,60  | 0,56  |
| S | Se(%)   | <b>67,27</b>  | 6,84          | 6,28          | 63,56        | 65,01        | 67,17        | 65,25         | 33,91        | 33,33 | 32,20 | 33,04 | 31,69 | 33,18 | 32,89 |
|   | P(%)    | 67,46         | 22,80         | 21,06         | 72,89        | 74,14        | <b>75,79</b> | 74,27         | 49,69        | 52,78 | 58,46 | 44,91 | 55,00 | 51,39 | 52,04 |
|   | F1(%)   | 67,36         | 5,25          | 9,38          | 67,90        | 69,28        | <b>71,22</b> | 69,47         | 40,31        | 40,86 | 41,53 | 38,07 | 40,21 | 40,33 | 40,22 |
|   | MCC     | 0,62          | 0,05          | 0,04          | 0,63         | 0,65         | <b>0,67</b>  | 0,65          | 0,34         | 0,35  | 0,37  | 0,31  | 0,35  | 0,34  | 0,34  |
| V | Se(%)   | 90,30         | 89,34         | 86,02         | 91,03        | 88,25        | 89,51        | 89,60         | <b>91,58</b> | 89,25 | 88,58 | 89,41 | 90,24 | 90,05 | 89,85 |
|   | P(%)    | 76,57         | 72,76         | 74,95         | 81,80        | <b>82,77</b> | 82,52        | 82,36         | 76,32        | 75,80 | 79,68 | 78,74 | 78,62 | 81,72 | 78,48 |
|   | F1(%)   | 82,87         | 80,14         | 80,07         | <b>86,17</b> | 85,42        | 85,87        | 85,82         | 83,25        | 81,97 | 83,89 | 83,74 | 84,03 | 85,68 | 83,76 |
|   | MCC     | 0,78          | 0,74          | 0,73          | <b>0,82</b>  | 0,81         | 0,81         | 0,81          | 0,78         | 0,76  | 0,79  | 0,78  | 0,79  | 0,81  | 0,79  |

- [8] B. Liu, Z. Ding, and C. Lv, "Distributed training for multi-layer neural networks by consensus," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, pp. 1771–1778, May 2020.
- [9] G.A. Kaissis, M.R. Makowski, D. Rückert, and R.F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nat Mach Intell*, vol. 2, pp. 305–311, 2020.
- [10] R. Torkzadehmahani, R. Nasirigerdeh, D. B. Blumenthal, T. Kacprowski, M. List, J. Matschinske, J. Spaeth, N. K. Wenke, and J. Baumbach, "Privacy-preserving artificial intelligence techniques in biomedicine," *Methods of Information in Medicine*, 2022.
- [11] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000 (June 13), *Circulation Electronic Pages*: <http://circ.ahajournals.org/content/101/23/e215.full> PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.
- [12] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.
- [13] Association for the Advancement of Medical Instrumentation *et al.*, "ANSI/AAMI EC57:1998(R)2008 - Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms," *American National Standards Institute, Arlington, VA, USA*, 2008, Association for the Advancement of Medical Instrumentation (AAMI), ANSI/AAMI/ISO EC57:1998-(R)2008, 2008.
- [14] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2020.
- [15] G. King and L. Zeng, "Logistic regression in rare events data," *Political Analysis*, vol. 9, pp. 137–163, 2001.
- [16] S. Haykin, *Neural networks and learning machines*, Pearson Education, Upper Saddle River, 3 edition, 2009.
- [17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
- [18] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.
- [19] P. de Chazal, M. O'Dwyer, and R. B. Reilly, "Automatic classification of heartbeats using ECG morphology and heartbeat interval features," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 7, pp. 1196–1206, 2004.
- [20] E. J. S. Luz, W. R. Schwartz, G. Cámara-Chávez, and D. Menotti, "ECG-based heartbeat classification for arrhythmia detection: A survey," *Computer methods and programs in biomedicine*, vol. 127, pp. 144–164, 2016.
- [21] Z. Luo, L. Liu, J. Yin, Y. Li, and Z. Wu, "Deep learning of graphs with ngram convolutional neural networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 10, pp. 2125–2139, 2017.
- [22] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-time patient-specific ECG classification by 1-D convolutional neural networks," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 3, pp. 664–675, 2015.
- [23] J. P. Martínez, R. Almeida, S. Olmos, A. P. Rocha, and P. Laguna, "A wavelet based ECG delineator: evaluation on standard databases," vol. 51, no. 4, pp. 570–581, 2004.
- [24] P. S. Hamilton and W. J. Tompkins, "Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmia database," *IEEE Transactions on Biomedical Engineering*, vol. BME-33, no. 12, pp. 1157–1165, Dec 1986.
- [25] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.
- [26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [27] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Yee Whye Teh and Mike Titterton, Eds., Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010, vol. 9 of *Proceedings of Machine Learning Research*, pp. 249–256, PMLR.