

Classificação automática de arritmia em sinais de ECG utilizando Redes Neurais de Grafos baseado em classificação de grafos

Rafael Francisco de Oliveira

Departamento de Computação

Universidade Federal de Ouro Preto - UFOP

Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq

Email: rafael.fo@aluno.ufop.edu.br

Abstract—O processo de identificação e classificação de doenças cardiovasculares (DCVs), especialmente as arritmias em eletrocardiogramas (ECGs) pode ser muito exaustivo para um ser humano e, sujeito a erros, uma vez que seria necessário uma análise de batimento a batimento em horas ou até mesmo dias de registros de ECG capturados. A automatização desse processo torna-se desejável e atualmente, diversas técnicas de inteligência artificial têm sido aplicadas, principalmente as técnicas baseadas em *deep learning*. Neste estudo, com o objetivo de obter um diagnóstico preciso dos tipos de DCVs, de acordo com a arritmia em batimentos cardíacos de ECG, propõe-se um modelo de classificação automática de arritmias em sinais de ECG baseado em Redes Neurais de Grafos por meio do mapeamento dos sinais de ECGs em uma rede de grafo e classificação nível de grafo. São avaliados três modelos de redes GNN: 02 *Graph Convolutional Network* - (02 GCN), 03 *Graph Convolutional Network* - (03 GCN) e 04 *GraphSAGE*. Os modelos são avaliados no conjunto de dados PhysioNet MIT-BIH de acordo com os padrões da AAMI com o uso do paradigma *inter-patient* no desenvolvimento metodológico e experimental. Dentre as métricas analisadas, a sensibilidade (*Se*) foi considerada a de maior peso de avaliação. O modelo 04 *GraphSAGE* foi o que apresentou o melhor desempenho, obtendo 98%, 19% e 87% de *Se* para as classes N, S e V, respectivamente.

Keywords: Classificação; AAMI; Graph Neural Network; Arritmia; Eletrocardiograma; Inter-Patient; Grafo;

I. INTRODUÇÃO

De acordo com a Organização Mundial de Saúde (OMS), doenças cardíacas são a principal causa de morte no planeta. Embora boa parte das doenças cardíacas possam ser diagnosticadas com antecedência, o número de casos continua a crescer. Estima-se que mais de 75% dos casos acontecem em países pobres e em desenvolvimento e 17,9 milhões de pessoas morreram de doenças cardíacas em 2019, representando 32% das mortes globais [1]. Isso se deve ao fato de que, medidas de prevenção, aparentemente, não estão sendo adequadas, por isso o diagnóstico precoce é de fundamental importância.

O principal exame para diagnóstico de doenças cardíacas é chamado Eletrocardiograma (ECG). Segundo Cohen [2], o ECG foi um dos primeiros sinais estudados pelas técnicas de processamento de sinais biomédicos e até os dias de hoje é a técnica mais utilizada no diagnóstico de doenças do coração.

Isto é possível por causa da simplicidade do exame, o qual é um processo não-invasivo.

O processo de identificação e classificação de arritmias pode ser muito exaustivo para um ser humano e, ainda sujeito a erros, visto que seria necessário uma análise de batimento a batimento em horas ou até mesmo dias de registros de ECG capturados. Portanto, existe a necessidade de estudos por soluções que mitigue esses erros. Uma alternativa consiste no uso de inteligência artificial para classificação automática dos padrões de arritmia. Atualmente, métodos baseados em aprendizagem de máquina, do inglês *machine learning*, tornaram-se bastante populares, obtendo resultados expressivos [3], principalmente Redes Neurais Artificiais. Porém, ainda há muito espaço para melhorias, especialmente quando se considera uma avaliação mais rigorosa [4].

Para que soluções propostas possam ser utilizadas em monitores de sinais vitais ou desfibriladores, o projeto e análise de experimentos deve obedecer normas [4] da *Association for the Advancement of Medical Instrumentation (AAMI)*. Manter a conformidade com todas as normas e boas práticas é ainda um grande obstáculo para a introdução de novas técnicas na indústria. As abordagens publicadas na literatura que não seguem esses requisitos tendem a ser superestimados e assim, sem grande valor prático para fabricantes de equipamentos médicos [5].

Neste projeto, propõe-se um modelo de classificação automática de arritmias em sinais de ECG baseado em Redes Neurais de Grafos (*Graph Neural Network* - GNN) por meio do mapeamento dos sinais de ECGs em uma rede de grafo e classificação nível de grafo (*graph-level classification*). São avaliados três modelos de redes GNN: 02 *Graph Convolutional Network* - (02 GCN), 03 *Graph Convolutional Network* - (03 GCN) e 04 *GraphSAGE*. A proposta é uma análise comparativa entre os desempenhos de cada modelo. Os experimentos foram validados utilizando o conjunto de dados PhysioNet MIT-BIH [6], [7] através do paradigma *inter-patient* proposto por De Chazal et al. [8].

O trabalho é organizado como segue. Seção II mostra uma breve revisão dos trabalhos e métodos da literatura. Seção III define o conjunto de dados utilizado para avaliação do modelo.

A metodologia proposta para a classificação de sinais de ECG é descrita na Seção IV. Além disso, na Seção V, a descrição dos experimentos e resultados são mostrados. Finalmente, as observações finais são reportadas na Seção VI.

II. TRABALHOS RELACIONADOS

A detecção de arritmias com base na classificação de ECG tornou-se um tópico de pesquisa muito popular na literatura nos últimos anos [5]. Os métodos tradicionais utilizam características extraídas do sinal para a classificação dos batimentos cardíacos, como, por exemplo, o intervalo RR, coeficientes *wavelets*, amplitude, complexo QRS, onda P, duração, etc. Os principais pontos e intervalos, denominados de pontos fiduciais, de um sinal ECG são mostrados na Fig. 1. Dentre os pontos apresentados, temos o segmento QRS, o qual normalmente é utilizado durante o processo de segmentação de um sinal de ECG em batimentos menores.

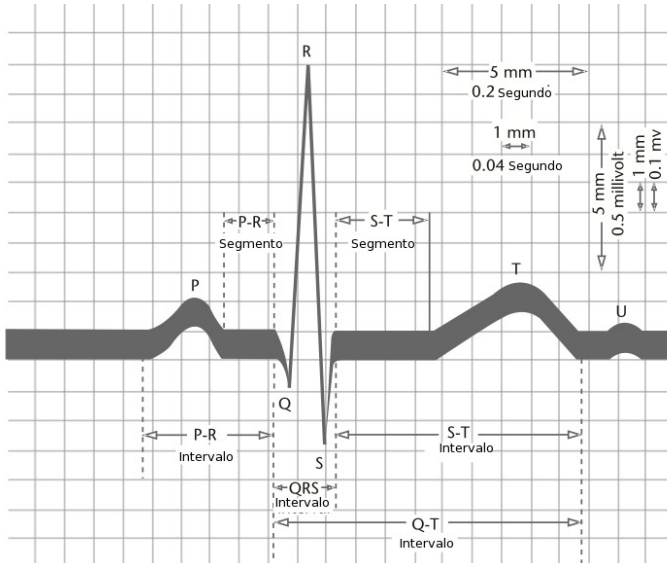


Fig. 1: Batimento normal de ECG da derivação II [9].

A identificação dos sinais ocorre a partir de características extraídas dos sinais de ECG. Com isso, várias características e métodos têm sido utilizados na literatura em diversas propostas. De Chazal *et al* [8] combinaram características rítmicas, duração de onda e características morfológicas. Zheng *et al* [10] utilizaram mais de 200 características no classificador utilizado, tais como idade, gênero, características do canal II do ECG além de características extraídas dos 12 canais presente no banco de dados utilizado. L. Wang *et al*. [11] propuseram um método de seleção de características baseado em transformação *wavelet* com limiar adaptativo, visando reduzir o ruído do sinal de ECG.

Após a extração das características do sinal de ECG, um classificador de aprendizado de máquina é treinado utilizando as características do sinal. A escolha do classificador ocorre de acordo com o problema a ser tratado. Na literatura há vários modelos, tais como SVM [12], random forest [13], AdaBoost [14], XGBoost [15] e redes neurais artificiais [16]. Com isso,

o estudo em novas metodologias de classificação de sinais de ECG torna-se tão importante.

O modelo de redes neurais artificiais apresenta crescente interesse pela comunidade científica, principalmente redes neurais convolucionais, pela sua capacidade de generalização e aplicações. Recentemente, as redes neurais de grafos tem apresentado grande interesse na literatura [17]–[19] em diversos domínios, como estudo social [20], química [21], biologia [22], sistema de recomendação [23], classificação de texto [24], etc. Análise de grafos foca em tarefas como classificação de nós, arestas, grafos e clusterização.

O surgimento das redes neurais de grafos (GNN) deu-se nos anos 90 [18] com as redes neurais recursivas [25] utilizadas em grafos direcionados acíclicos. Posteriormente, redes neurais recorrentes [26] e redes neurais feedforward [27] foram introduzidas para grafos cíclicos. Até esse momento, as redes eram construídas baseadas em sistemas de transição de estado em grafos e iteragiam até a convergência, o que restringia a capacidade de extensão e representação. Com o avanço das redes neurais profundas, em especial pelas redes neurais convolucionais profundas (CNN) [28], as redes neurais de grafos foram descobertas.

Wu *et al* [19] categorizou as GNNs em quatro grupos: redes neurais de grafos recorrentes, redes neurais de grafos convolucionais, grafos *autoencoders* e redes neurais de grafos espaço-temporal. Neste trabalho, é utilizada a rede neural de grafos convolucional.

A GCN [29] é uma rede neural multicamada aplicada diretamente em grafos agregado à informações/propriedades da vizinhança de cada nó. Formalmente [24], considere um grafo como $G(V, E)$, onde $V(|V| = n)$ é o conjunto de nós e E o conjunto de arestas. Cada nó é conectado consigo mesmo (*loop*), isto é, $(v, v) \in E$. Assumindo $X \in \mathbb{R}^{n \times m}$ como a matriz contendo todos os n nós com seus respectivos vetores de atributos, onde m é a dimensão deste vetor, cada linha $x_v \in \mathbb{R}^m$ é o vetor de atributos para v . Considerando ainda a matriz de adjacência A de G e a matriz de graus D , onde $D_{ii} = \sum_j A_{ij}$. A diagonal principal de A contém 1 por causa dos *loops* de cada nó. Com apenas uma camada, a GCN pode capturar informações apenas dos vizinhos imediatos. Com múltiplas camadas, informações de uma “larga” vizinhança é agregada. Para uma camada GCN, a nova k -dimensional matriz de atributos do nó $L^{(1)} \in \mathbb{R}^{n \times k}$ é computada como sendo

$$L^{(1)} = \rho(\tilde{A}XW_0) \quad (1)$$

onde $\tilde{A} = D^{-\frac{1}{2}}AD - \frac{1}{2}$ é uma matriz de adjacência simétrica normalizada, $W_0 \in \mathbb{R}^{m \times k}$ é a matriz de pesos e ρ é a função de ativação, isto é, a ReLU $\rho(x) = \max(0, x)$. Incorporando mais informações da “larga” vizinhança com múltiplas camadas GCN, basta

$$L^{(j+1)} = \rho(\tilde{A}L^{(j)}W_j) \quad (2)$$

onde j denota o número de camadas e $L^{(0)} = X$.

Antes de utilizar uma GCN, é necessário o mapeamento dos dados do problema em estudo para um grafo. Lacasa et. al [30] propuseram o algoritmo *visibility graph* para conversão de uma série temporal em um grafo. Considerando cada dado da série temporal como uma barra vertical, onde a altura corresponde ao valor de cada dado, o grafo gerado terá cada nó correspondente, na mesma ordem, aos dados da série, e dois nós são conectados se houver visibilidade entre os dados correspondentes, ou seja, se houver uma linha reta que conecta os dados da série, desde que esta “linha de visibilidade” não cruze nenhuma altura de dados intermediária. Esta abordagem foi utilizada neste trabalho através da biblioteca *ts2vg* do Python.

III. CONJUNTO DE DADOS

Em razão das propostas de modelos de classificação para o avanço da instrumentação médica [4], foi utilizado o conjunto de dados de arritmias PhysioNet MIT-BIH [6], [7], [31] para treinamento e validação da metodologia proposta, permitindo assim uma possível comparação com outras metodologias da literatura. O conjunto de dados do MIT-BIH é composto por 48 registros de sinais de ECG de 30 minutos de 47 pacientes diferentes amostrados a frequência de 360 Hz. Cada sinal contém duas derivações: derivação V e derivação II. Normalmente, a derivação II é a mais utilizada nas metodologias da literatura. De forma similar, nos experimentos deste trabalho, a derivação II foi utilizada.

Tabela I: Distribuição dos registros do MIT-BIH em conjunto de treinamento e teste.

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

Seguindo recomendações da AAMI, quatro registros (102, 104, 107 e 217) são excluídos do conjunto de dados por conter batimentos artificiais. De Chazal et al. [8] recomenda a divisão do conjunto de dados em dois conjuntos: treinamento (DS1) e teste (DS2) conforme a Tabela I. Essa divisão faz parte do paradigma *inter-patient* utilizado neste trabalho, onde o conjunto de treinamento e teste apresentam conjuntos de registros diferentes.

O conjunto de dados do MIT-BIH contém informações dos tipos de batimentos e a localização do pico R, mediante o consenso de classificação de dois cardiologistas. Dentre os 18 tipos de batimentos, a AAMI também recomenda o agrupamento dos batimentos em cinco grupos principais: *Normal (N)*, *Supraventricular ectopic beat (S)*, *Ventricular ectopic beat (V)*, *Fusion beat (F)* e *Unknown beat (Q)*. A Tabela II resume o agrupamento dos batimentos dos dados do MIT-BIH

após a exclusão dos registros com batimentos artificiais bem como o número de amostras do conjunto de treinamento e teste. Observa-se que as classes F e Q apresentam poucos batimentos, somando menos de 0,81% de presença em relação ao conjunto de dados total. Por esse motivo, as classes F e Q foram removidas, sendo os modelos aplicados apenas às classes N, S e V.

IV. METODOLOGIA

A seguir, é apresentado a metodologia proposta para classificação automática de arritmias em ECG. O diagrama da Fig. 2 mostra as etapas do sistema de classificação proposto: segmentação, balanceamento dos dados, divisão de dados, conversão em grafos e classificação.

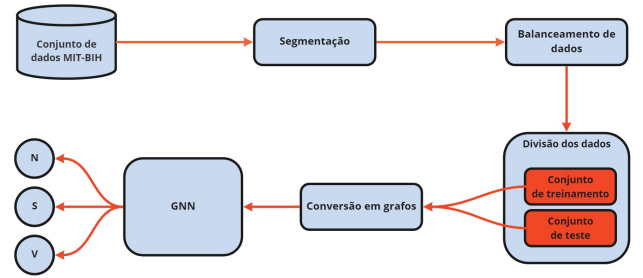


Fig. 2: Diagrama do sistema de classificação de arritmia.

A. Segmentação

Antes da classificação, é necessário a segmentação do sinal ECG em batimentos individuais. Isto requer a identificação da onda QRS e pontos fiduciais do batimento. No entanto, vários métodos são desenvolvidos na literatura para detecção desses pontos fiduciais com altas acurácias. Uma ilustração do processo de segmentação é mostrada na Fig. 3. Neste processo de segmentação, foi utilizado a localização do pico R anotada no conjunto de dados do MIT-BIH, segmentando o sinal em séries de batimentos extraíndo 150 amostras antes e 150 amostras depois do pico R, e realizando o registro do tipo de batimento de acordo com as anotações do MIT-BIH. No final, cada batimento obteve tamanho fixo de 300 amostras do sinal de ECG.

B. Balanceamento de dados

O grande número de batimentos da classe N, conforme mostrado na Tabela II, ocasionava grande tempo de processamento devido ao alto número de grafos gerados. Por esse motivo, os batimentos da classe N foram reduzidos para 3000 batimentos tanto no conjunto de treinamento (DS1) quanto no conjunto de teste (DS2), consequentemente tornando o conjunto de dados mais balanceado em relação às classes S e V. A Tabela III resume o conjunto de dados antes e após o balanceamento.

Tabela II: Agrupamento de tipos de batimento no conjunto de dados do MIT-BIH.

Grupo AAMI	Batimentos	% do total	# Treinamento (DS1)	# Teste (DS2)	# Batimentos total
Normal (N)	N, e, j, L, R	89,47	45844	44238	90082
Supraventricular ectopic beat (S)	A, a, J, S	2,76	944	1837	2781
Ventricular ectopic beat (V)	V, E	6,96	3788	3220	7008
Fusion beat (F)	F	0,80	414	388	802
Unknown beat (Q)	P, f, U, Q	0,01	8	7	15
Total		100	50998	49690	100688

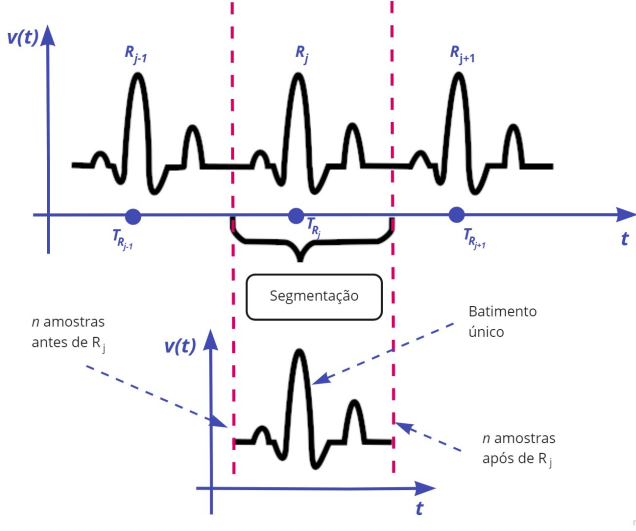


Fig. 3: Exemplo de segmentação de sinal ECG [32].

Tabela III: Descrição do número de amostras antes e depois do balanceamento de dados.

Batimentos	Treinamento		Teste	
	Antes	Depois	Antes	Depois
N	45844	3000	44238	3000
S	944	944	1837	1837
V	3788	3788	3220	3220
Total	50576	7732	49295	8057

C. Divisão do conjunto de dados

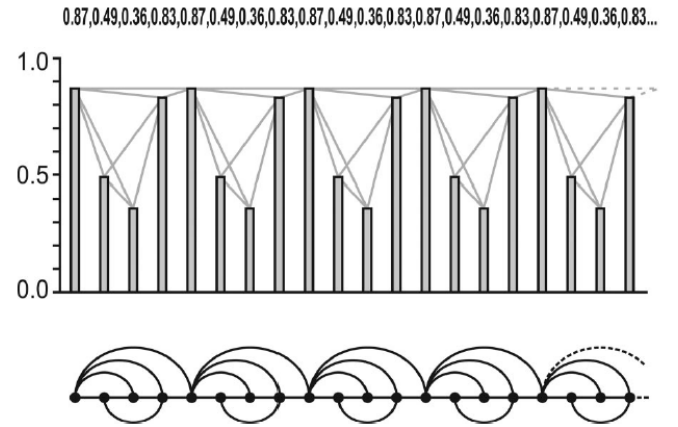
Os batimentos gerados na etapa de segmentação são divididos em conjunto de treinamento (DS1) e conjunto de teste (DS2) conforme mostrado na Tabela I. Essa divisão proposta por De Chazal et al. [8] permite utilizar o paradigma *inter-patient* de classificação, onde os batimentos do conjunto de treinamento são diferentes do conjunto de teste, permitindo assim uma simulação de cenário do mundo real.

D. Conversão dos sinais ECG em grafos

Como mencionado na Seção II, o método *visibility graph* [30] foi utilizado na conversão das séries temporais dos sinais de ECG em grafos. A Figura 4 exemplifica a conversão realizada pelo método *visibility graph*.

A Figura 7 do Apêndice mostra a conversão dos cinco tipos de batimentos presente no conjunto de dados do MIT-BIH. Podemos verificar que cada sinal é convertido em um grafo diferente, possibilitando a classificação pelas GNNs.

O processo realiza a conversão de um batimento S em um grafo $G(V, E)$, onde V corresponde ao conjunto de vértices/nós e E corresponde ao conjunto de arestas/links do grafo G . O processo de classificação de grafos em *deep learning* pode ocorrer em três níveis: nós (*node-level*), arestas (*edge-level*) e grafo (*graph-level*).

Fig. 4: Exemplo de uma série temporal periódica (20 valores) e o grafo derivado pelo método *visibility graph* [30].

Nesse trabalho, utilizou-se a abordagem de grafo (*graph-level*), onde a classificação ocorre considerando o grafo por completo. Informações contidas nos nós e arestas são utilizadas durante o processo de classificação. Com isso, mais informações contidas no grafo tendem a melhorar o desempenho das GNNs. Aqui, apenas duas informações foram configuradas em cada nó p do conjunto de nós V : o grau (número de arestas incidentes no respectivo nó) e *pagerank* (medida da importância do nó no seu respectivo grafo).

E. Classificação

As bibliotecas utilizadas na implementação das redes GNN foram a *Deep Graph Library* - DGL [33] e o *PyTorch* [34] do Python. A vantagem da biblioteca DGL é a possibilidade de criação de diversos modelos de redes de grafos convolucionais. Após a aplicação de todas as etapas anteriormente descritas, os grafos foram classificados por três modelos de redes GNN:

02 *Graph Convolutional Network* - (02 *GCN*), 03 *Graph Convolutional Network* - (03 *GCN*) e 04 *GraphSAGE*.

O modelo 02 *GCN* consiste em duas camadas de convolução de grafos sequenciais introduzidas em [29]. Cada camada é matematicamente definida por

$$h_i^{(l+1)} = \sigma(b^{(l)} + \sum_{j \in \mathcal{N}(i)} \frac{1}{c_{ji}} h_j^{(l)} \mathbf{W}^{(l)}) \quad (3)$$

onde $\mathcal{N}(i)$ é o conjunto de vizinhos do nó i , c_{ji} é o produto das raízes quadradas dos graus do nó, ou seja, $c_{ji} = \sqrt{|\mathcal{N}(j)|} \sqrt{|\mathcal{N}(i)|}$ e σ é a função de ativação de cada camada.

O modelo 03 *GCN* é idêntico ao modelo 02 *GCN* com a diferença da adição de mais uma camada de convolução de grafos. Por último, o modelo 04 *GraphSAGE* consiste em 04 camadas de convolução *GraphSAGE* (SAmple and aggreGatE) sequenciais introduzidas em [35]. Este tipo de estrutura utiliza um conjunto de funções agregadoras que aprendem a agregar informações de recursos da vizinhança local de um nó através das características de cada nós. Isso permite aprender uma função de incorporação que generaliza para nós invisíveis (nunca visto antes). Matematicamente, *GraphSAGE* é definido por

$$h_{\mathcal{N}(v)}^k \leftarrow \mathbf{aggregate}(h_u^{k-1}, \forall u \in \mathcal{N}(v)) \quad (4)$$

$$h_v^k \leftarrow \sigma(\mathcal{W}^k \cdot \mathbf{concat}(h_v^{k-1}, h_{\mathcal{N}(v)}^k)) \quad (5)$$

$$h_k^v \leftarrow \mathbf{norm}(h_k^v, \forall u \in \mathcal{V}) \quad (6)$$

A intuição do *GraphSAGE* é que a informação agregada dos vizinhos locais de cada nó é incrementada cada vez mais, tornando o processo de classificação mais robusto a cada iteração. A cada iteração k , cada nó $v \in \mathcal{V}$ agrega informação dos vizinhos imediatos ($h_u^{k-1}, \forall u \in \mathcal{N}(v)$) em um único vetor $h_{\mathcal{N}(v)}^{k-1}$ (Fórmula 4). Após a agregação, o *GraphSAGE* concatena a representação atual do nó h_v^{k-1} com a do vizinho $h_{\mathcal{N}(v)}^{k-1}$ aplicada a uma função de ativação σ (Fórmula 5), onde as transformações são usadas na próxima iteração por meio de h_v^k . Por fim, as informações são normalizadas (Fórmula 6) para a próxima iteração.

A seguir, a Tabela IV descreve a arquitetura de cada modelo de GCN, a quantidade de neurônios bem como o tipo de cada camada utilizando a biblioteca DGL. É importante observar que nas redes GNN da DGL, a camada de entrada tem como argumento o tamanho do vetor de atributos de cada nó do grafo, sendo para esse trabalho, o tamanho 02 (grau e *pagerank*). A camada de saída sempre irá ter 03 neurônios, cada um para cada classe (N, S, V).

Por ser uma classificação singular considerando cada grafo por completo (*graph-level*) é necessário um tipo de operação que agregue e sume todas as informações contidas nos nós e arestas de cada grafo antes de camada de saída. Este tipo de operação é chamado de *readout*. As operações mais comuns

são soma, mínimo, média e máximo. Aqui, é utilizado a média como operação *readout* conforme a Fórmula 7.

$$h_g = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} h_v \quad (7)$$

onde h_g é a representação de um grafo g , \mathcal{V} o conjunto de nós de g e h_v é o atributo do nó v .

Tabela IV: Arquiteturas de cada modelo de rede GNN.

Modelos	No	Camada	Formato
02 <i>GCN</i>	1	GraphConv	2×20
	2	GraphConv	20×3
	3	Readout	-
	4	Softmax	-
03 <i>GCN</i>	1	GraphConv	2×20
	2	GraphConv	20×20
	3	GraphConv	20×3
	4	Readout	-
	5	Softmax	-
04 <i>GraphSAGE</i>	1	SAGEConv	2×30
	2	SAGEConv	30×20
	3	SAGEConv	20×5
	4	SAGEConv	5×3
	5	Readout	-
	6	Softmax	-

V. RESULTADOS

A. Configurações

Os experimentos foram conduzidos em um ambiente Windows 11 Home e Python 3.9 com Intel®i5-1035G1 com 04 cores físicos (04 threads) de 1.00GHz, 8GB RAM DDR4.

Como medida de avaliação dos modelos propostos, três métricas fora utilizadas: predição positiva ($+P^k$), sensibilidade (Se^k) e F1-score (F_s^k).

As métricas são baseadas nas medidas de verdadeiro positivo (TP^k), falsos negativos (FN^k), verdadeiros negativos (TN^k) e falsos positivos (FP^k), de acordo com as Fórmulas 9, 8 e 10.

$$+P^k = \frac{TP^k}{TP^k + FP^k} \times 100 \quad (8)$$

$$Se^k = \frac{TP^k}{TP^k + FN^k} \times 100 \quad (9)$$

$$F_s^k = 2 \frac{Se^k(+P^k)}{Se^k + (+P^k)} \quad (10)$$

onde $k \in \{N, S, V\}$ indica as classes correspondente de cada métrica. Como a proposta do modelo é a classificação correta de todos os batimentos de acordo com as determinadas classes, a sensibilidade (Se) teve maior peso diante da escolha do melhor modelo de rede, uma vez que a mesma concede uma medida de quão preciso o modelo é capaz de identificar os dados relevantes.

B. Resultados

Cada modelo foi treinado com o conjunto de treinamento (DS1) utilizando 100 épocas (*epochs*) e validado no conjunto de teste. Para aumentar a eficiência do tempo de processamento, o conjunto de treinamento quanto o de teste foram divididos em pequenos grupos (*batches*) de tamanho 64. Assim, os modelos eram treinados e atualizados a cada iteração. Os parâmetros para cada modelo foram: otimizador Adam, taxa de aprendizado de 0,01, função de perda *cross entropy* e função de ativação ReLU (*Rectified Linear Unit*).

Na Tabela V é mostrado os resultados das matrizes de confusão de cada modelo de GCN proposto em relação ao conjunto de teste.

Tabela V: Matrizes de confusão do conjunto de teste (DS2) para cada modelo proposto.

Modelos	Matrizes de confusão				
			Predição		
			N	S	V
02 GCN	Verdade	N	1725	5	1270
		S	194	57	1586
		V	388	12	2820
03 GCN	Verdade	N	2813	9	178
		S	170	129	1538
		V	388	40	2792
04 GraphSAGE	Verdade	N	2953	1	46
		S	5	355	1477
		V	33	395	2792

Pode-se notar que apesar dos modelos serem bastante simples, o desempenho na performance foi bastante promissor, apresentando um aumento considerável no modelo *GraphSAGE* em relação aos outros modelos. Em relação às classes N e V, o modelo *GraphSAGE* conseguiu uma taxa de classificação correta em relação aos batimentos corretos de 98,44% e 86,70%, respectivamente. Os modelos *02 GCN* e *03 GCN* tiveram desempenho semelhante para as classes N e V. Porém, na classe S, o modelo *GraphSAGE* não apresentou um bom desempenho, apresentando apenas uma taxa de classificação correta de 19,32%. Em relação ao modelo *02 GCN*, o modelo *GraphSAGE* obteve um aumento de 545,50% de performance para a classe S. Os modelos *02 GCN* e *03 GCN* também não apresentaram grande desempenho para a classe S.

A Tabela VI mostra os resultados das métricas de avaliação para cada modelo proposto. Observa-se que trabalhos que reportam apenas acurácia como principal métrica de avaliação tendem a enviesar os resultados, uma vez que a performance das classes majoritárias influenciam no resultado. Considerando a *Se*, pode-se observar que o modelo *GraphSAGE* apresentou o melhor desempenho nas classes N e S, como *Se* de 98% e 19%, respectivamente. Na classe V, apresentou 87%, 1% abaixo do modelo *02 GCN*.

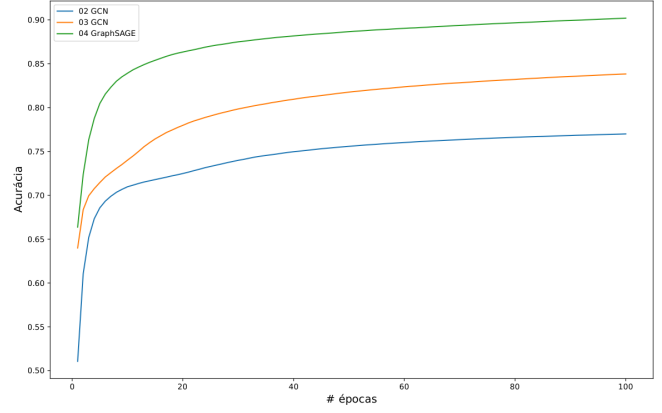


Fig. 5: Curva da acurácia durante o treinamento dos modelos ao longo de 100 épocas.

As Figuras 5 e 6 mostram as curvas de desempenho da acurácia e da função perda (*loss*) ao longo de 100 épocas de treinamento. Novamente, percebe-se que o modelo *GraphSAGE* foi o que apresentou os melhores resultados, tanto na acurácia ($\pm 87\%$) quanto na função de perda. Esse desempenho pode ser justificado pela agregação e concatenação de informações dos nós utilizada pelo modelo a cada iteração.

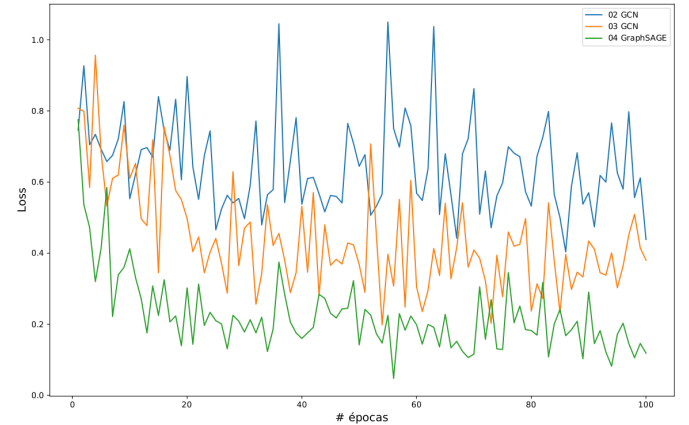


Fig. 6: Curva da função perda (*loss*) durante o treinamento dos modelos ao longo de 100 épocas.

VI. CONCLUSÕES

Neste trabalho, foi proposto um modelo de classificação automática de arritmias em sinais de ECG baseado em Redes Neurais de Grafos (*Graph Neural Network - GNN*) por meio do mapeamento dos sinais de ECGs em uma rede de grafo e classificação nível de grafo (*graph-level classification*). Três modelos de redes GNN: *02 Graph Convolutional Network - (02 GCN)*, *03 Graph Convolutional Network - (03 GCN)* e *04 GraphSAGE* foram utilizados e comparados de acordo com suas performances na classificação de três classes de arritmias (N, S, V). Como simulação de aplicabilidade em cenários

Tabela VI: Performance dos modelos em relação ao conjunto de teste (DS2).

Modelos	Acc	N			S			V		
		+P ^N	Se ^N	F _s ^N	+P ^S	Se ^S	F _s ^S	+P ^V	Se ^V	F _s ^V
02 GCN	57	75	57	65	77	3	6	50	88	63
03 GCN	71	83	94	88	72	7	13	62	87	72
04 GraphSAGE	76	99	98	99	47	19	27	65	87	74

reais, os modelos foram treinados de acordo com o paradigma *inter-patient*.

O modelo *04 GraphSAGE* foi o que apresentou a melhor performance entre os modelos analisados, com *Se* de 98%, 19% e 87% para as classes N, S e V, respectivamente. Os resultados ainda mostraram que a acurácia nem sempre descreve o melhor desempenho dos modelos, pois um alto desempenho em apenas uma classe pode ocasionar resultados enviesados.

A utilização de modelos baseados em Redes Neurais de Grafos apresenta potencial como novo método de classificação de arritmias em sinais de ECG. Como trabalhos futuros, propõe-se a construção de redes neurais de grafos mais robustas e a adição de mais informações aos nós de cada grafo, permitindo assim, redes como a *GraphSAGE*, obtenha o máximo de performance através da agregação de informação durante a classificação.

AGRADECIMENTOS

Agradecimentos ao incentivo e suporte da Universidade Federal de Ouro Preto, FAPEMIG e CNPq.

REFERENCES

- [1] W. H. Organization, *Cardiovascular diseases (CVDs)*, WHO, 2021. [Online]. Available: [https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))
- [2] A. Cohen, *Biomedical Signal Processing*. CRC Press., 1988.
- [3] A. Y. Hannun, P. Rajpurkar, M. Haghighpanahi, 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, no. 1, p. 65, 2019.
- [4] ANSI/AAMI, "Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms," American National Standards Institute, Inc. (ANSI), Association for the Advancement of Medical Instrumentation (AAMI), 2008, ANSI/AAMI/ISO EC57, 1998-(R)2008.
- [5] E. J. d. 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.
- [6] 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.
- [7] A. L. Goldberger A., H. J. Glass L., M. R. Ivanov PC., M. G. Mietus JE., and S. H. Peng CK., "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," pp. e215–e220, 2000.
- [8] 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.
- [9] G. D. Clifford, F. Azuaje, P. McSharry *et al.*, *Advanced methods and tools for ECG data analysis*. Artech house Boston, 2006.
- [10] J. Zheng, J. Zhang, S. Danioko, H. Yao, H. Guo, and C. Rakovski, "A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients," *Scientific data*, vol. 7, no. 1, pp. 1–8, 2020.
- [11] L. Wang, W. Sun, Y. Chen, P. Li, and L. Zhao, "Wavelet transform based ecg denoising using adaptive thresholding," in *Proceedings of the 2018 7th International Conference on Bioinformatics and Biomedical Science*, 2018, pp. 35–40.
- [12] V. Mondéjar-Guerra, J. Novo, J. Rouco, M. G. Penedo, and M. Ortega, "Heartbeat classification fusing temporal and morphological information of ecgs via ensemble of classifiers," *Biomedical Signal Processing and Control*, vol. 47, pp. 41–48, 2019.
- [13] P. Yang, D. Wang, W.-B. Zhao, L.-H. Fu, J.-L. Du, and H. Su, "Ensemble of kernel extreme learning machine based random forest classifiers for automatic heartbeat classification," *Biomedical Signal Processing and Control*, vol. 63, p. 102138, 2021.
- [14] K. N. Rajesh and R. Dhuli, "Classification of imbalanced ecg beats using re-sampling techniques and adaboost ensemble classifier," *Biomedical Signal Processing and Control*, vol. 41, pp. 242–254, 2018.
- [15] H. Shi, H. Wang, Y. Huang, L. Zhao, C. Qin, and C. Liu, "A hierarchical method based on weighted extreme gradient boosting in ecg heartbeat classification," *Computer methods and programs in biomedicine*, vol. 171, pp. 1–10, 2019.
- [16] P. Rajpurkar, A. Y. Hannun, M. Haghighpanahi, C. Bourn, and A. Y. Ng, "Cardiologist-level arrhythmia detection with convolutional neural networks," *arXiv preprint arXiv:1707.01836*, 2017.
- [17] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.
- [18] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.
- [19] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.
- [20] Z. Wang, T. Chen, J. Ren, W. Yu, H. Cheng, and L. Lin, "Deep reasoning with knowledge graph for social relationship understanding," *arXiv preprint arXiv:1807.00504*, 2018.
- [21] K. Do, T. Tran, and S. Venkatesh, "Graph transformation policy network for chemical reaction prediction," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 750–760.
- [22] A. M. Fout, "Protein interface prediction using graph convolutional networks," Ph.D. dissertation, Colorado State University, 2017.
- [23] Q. Wu, H. Zhang, X. Gao, P. He, P. Weng, H. Gao, and G. Chen, "Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems," in *The World Wide Web Conference*, 2019, pp. 2091–2102.
- [24] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 7370–7377.
- [25] A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 714–735, 1997.
- [26] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.
- [27] A. Micheli, "Neural network for graphs: A contextual constructive approach," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 498–511, 2009.
- [28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [29] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.
- [30] L. Lacasa, B. Luque, F. Ballesteros, J. Luque, and J. C. Nuno, "From time series to complex networks: The visibility graph," *Proceedings of*

the National Academy of Sciences, vol. 105, no. 13, pp. 4972–4975, 2008.

- [31] G. B. Moody and R. G. Mark, “The mit-bih arrhythmia database on cd-rom and software for use with it,” in *[1990] Proceedings Computers in Cardiology*. IEEE, 1990, pp. 185–188.
- [32] T. Wang, C. Lu, Y. Sun, M. Yang, C. Liu, and C. Ou, “Automatic ecg classification using continuous wavelet transform and convolutional neural network,” *Entropy*, vol. 23, no. 1, p. 119, 2021.
- [33] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, and Z. Zhang, “Deep graph library: A graph-centric, highly-performant package for graph neural networks,” *arXiv preprint arXiv:1909.01315*, 2019.
- [34] *PyTorch*, 2021. [Online]. Available: <https://pytorch.org/>
- [35] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 1025–1035.

APÊNDICE

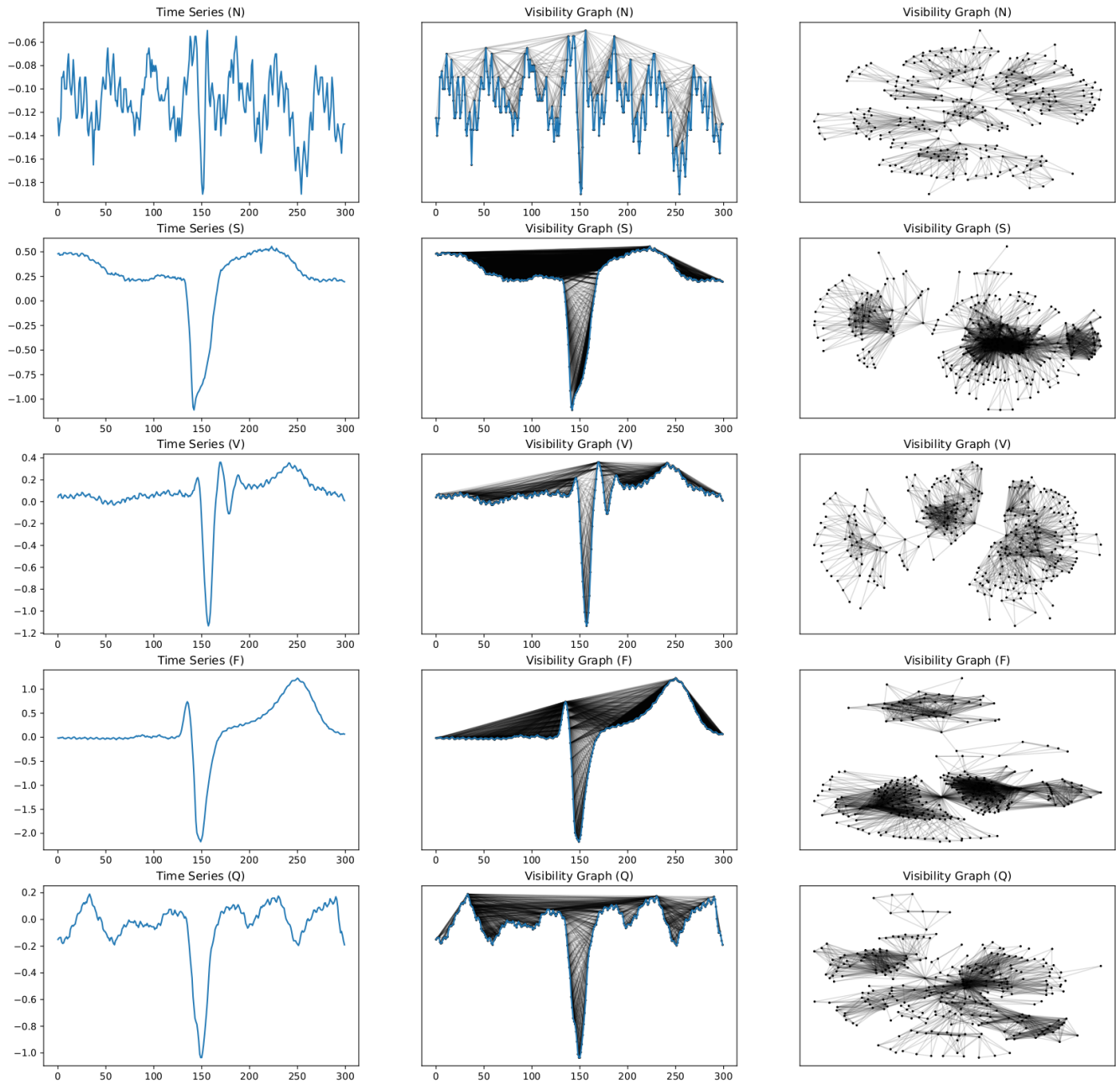


Fig. 7: Mapeamento dos sinais de ECG do conjunto de dados do MIT-BIH utilizando o método *visibility graph*.