

# Análise Comparativa de Abordagens Baseadas em Grafos para Classificação de Séries Temporais

Raul Araju Korogi Oliveira

Departamento de Ciência da Computação

Universidade Federal de Minas Gerais

Belo Horizonte, Brasil

raularaju@dcc.ufmg.br

Renato Vimieiro

Departamento de Ciência da Computação

Universidade Federal de Minas Gerais

Belo Horizonte, Brasil

**Abstract**—O problema de classificação de séries temporais é recorrente em diversas áreas, como medicina, música e sismologia. Uma abordagem recente consiste em transformar séries temporais em grafos, extrair características desses grafos e, a partir delas, realizar a classificação. Nesse contexto, uma estratégia emergente é transformar as séries temporais em um grafo e depois executar a classificação a partir dessa representação intermediária. O objetivo geral da pesquisa é realizar uma análise comparativa e robusta entre diferentes pipelines de classificação.

**Index Terms**—Séries Temporais, Grafos, GNN, Classificação, Comparação Experimental

## I. INTRODUÇÃO

Séries temporais são sequências de dados indexadas temporalmente, representando a evolução de um fenômeno ao longo do tempo. Elas ocorrem naturalmente em diversas áreas, como medicina, finanças, climatologia e sismologia. Por exemplo, um eletrocardiograma (ECG) registra a atividade elétrica do coração em intervalos regulares, enquanto um sismógrafo monitora oscilações sísmicas ao longo do tempo. Devido à sua ubiquidade, compreender e extrair padrões de séries temporais é uma tarefa fundamental em diversas aplicações.

Entre os principais problemas associados a esse tipo de dado, destaca-se a **classificação de séries temporais**, cujo objetivo é atribuir rótulos predefinidos a sequências com base em sua forma ou dinâmica. Diversas abordagens têm sido propostas para essa tarefa, como métodos baseados em extração diretas de *features* da série [1], em distância [2], e em convoluções [3].

Recentemente, uma linha de pesquisa emergente tem investigado a transformação de séries temporais em grafos [4], com o objetivo de capturar propriedades estruturais que não são evidentes diretamente na série original. Nessa abordagem, características do grafo são extraídas e utilizadas como base para a classificação. A maioria dos trabalhos nessa direção emprega Redes Neurais em Grafos (GNNs) para realizar o aprendizado dessas características e a classificação subsequente.

O objetivo deste projeto é realizar uma análise comparativa entre diferentes estratégias de representação de séries temporais como grafos e sua eficácia na tarefa de classificação. Duas abordagens foram usadas neste trabalho, a primeira é mais simples: transformar as séries em Grafos de Visibilidade (GV) [5], extrair *features* globais (como diâmetro e número

de arestas) e locais (como grau e métricas de centralidade), e utilizar essas *features* como entrada para classificadores tradicionais de aprendizado de máquina. Já a segunda envolve métodos baseados em GNNs e outras técnicas do estado da arte, especialmente aquelas baseadas em convoluções. O objetivo geral é avaliar, de forma sistemática a eficácia de diferentes abordagens na classificação de séries temporais transformadas em grafos.

Sendo assim, as perguntas de pesquisa que orientam este trabalho são:

- 1) Qual é o desempenho da abordagem de extração de características combinada com classificadores clássicos em diferentes conjuntos de dados?
- 2) Há alguma relação entre o desempenho dessa abordagem e o tipo de dado sendo classificado?
- 3) Quais são os principais métodos de classificação de séries temporais baseados em grafos (incluindo a abordagem de extração de características com classificadores padrão)?
- 4) Como esses métodos se comportam em diferentes conjuntos de dados?
- 5) Qual é o desempenho dos métodos baseados em grafos em comparação com o estado da arte para classificação de séries temporais?
- 6) Existem diferenças estatísticas significativas entre os métodos avaliados?

O relatório apresenta oito seções: Introdução, Trabalhos Relacionados, Conceitos Básicos, Séries Temporais como Grafos, Metodologia, Discussão e, por fim, Conclusões e Trabalhos Futuros. A Introdução contextualiza o problema da classificação de séries temporais e define o objetivo de comparar abordagens baseadas na transformação em grafos e métodos de estado da arte. A seção de Trabalhos Relacionados revisa as metodologias de classificação, focando em métodos baseados em características, convoluções e diferentes representações em grafos. Em seguida, Conceitos Básicos estabelece as definições formais de séries temporais, shapelets e detalha as *features* globais e locais utilizadas para a representação dos grafos. A seção Séries Temporais como Grafos aprofunda-se nas três estratégias de transformação avaliadas: Grafo de Visibilidade (GV),

SimTSC e Time2graph/Time2graph+. A Metodologia descreve a configuração experimental robusta, as bases de dados e o protocolo de busca de hiperparâmetros e reavaliações para os modelos. Os Resultados apresentam os F1-scores médios e o desvio padrão de todos os modelos, juntamente com a análise estatística de ranking. A Discussão analisa e interpreta esses achados, oferecendo recomendações de modelos conforme o tipo de dado. Finalmente, Conclusões e Trabalhos Futuros resumem as descobertas, destacando o potencial das GNNs e listando sugestões para pesquisas futuras.

## II. TRABALHOS RELACIONADOS

Há diferentes abordagens para tratar o problema de classificação de séries temporais. Uma classe de métodos amplamente utilizada é baseada na extração de características (*feature-based*), que transforma a série original em um vetor fixo de atributos extraídos por meio de funções estatísticas, transformadas no domínio do tempo ou da frequência, ou outras heurísticas [1], [6]. Estudos recentes demonstram que essa estratégia pode alcançar resultados competitivos com técnicas mais complexas [7].

Ainda nesse raciocínio, há métodos que se concentram em subsequências ou padrões discriminativos dentro das séries, como *shapelets* [8] ou ordens de permutação [9], que visam capturar dados locais representativos e úteis para a tarefa de classificação. Essas abordagens assumem que determinadas estruturas temporais curtas são suficientes para discriminar entre classes.

Com o avanço das redes neurais profundas, surgiram soluções baseadas em redes convolucionais unidimensionais (1D CNNs), que operam diretamente sobre os sinais temporais. Essas redes são capazes de aprender automaticamente representações latentes relevantes a partir dos dados, dispensando a extração manual de características. Modelos como o *ROCKET* e suas variantes [3] têm se destacado por aliar desempenho competitivo a uma elevada eficiência computacional, mesmo em conjuntos de dados extensos.

Apesar de apresentarem desempenho promissor, os métodos baseados em extração de características podem ser sensíveis à chamada *maldição da dimensionalidade* [10]. À medida que o número de atributos extraídos cresce, o espaço de busca se torna exponencialmente maior, o que dificulta a discriminação das classes. Diante desse cenário, torna-se fundamental o desenvolvimento de estratégias eficazes de extração e seleção de características, de modo a garantir que apenas informações discriminativas e relevantes sejam mantidas no vetor final de representação.

Sob esse prisma, uma alternativa promissora é a identificação de estruturas intermediárias capazes de representar informações relevantes das séries temporais de forma mais compacta. Especificamente, grafos têm se mostrado eficazes na captura de relações de curto e longo alcance presentes nas séries temporais [4]. Essa representação permite modelar dependências temporais complexas como conexões estruturadas entre elementos, o que potencialmente facilita a extração

de padrões latentes e oferece uma visão mais organizada da dinâmica temporal dos dados.

Entre as representações mais promissoras baseadas em grafos estão os *Grafos de Transição de Padrões Ordinais* (GTPOs) [11], [12] e os *Grafos de Visibilidade* (GVs) [5]. Ambas as abordagens convertem séries temporais em estruturas topológicas que preservam, de maneiras distintas, aspectos temporais e estruturais dos dados. Os GTPOs modelam as transições entre padrões de ordem local, capturando a dinâmica da série por meio de variações relativas entre elementos consecutivos. Já os GVs estabelecem conexões entre pontos da série com base em uma relação de visibilidade geométrica, permitindo representar tanto relações de curto quanto de longo alcance. Essas representações têm se mostrado eficazes na caracterização de complexidade, periodicidade e comportamento dinâmico de séries temporais em diferentes contextos.

Nesse contexto, Coelho [13] realizou uma análise comparativa de diferentes formas de classificar séries temporais representadas como grafos. O autor avaliou diversas possibilidades de construção da seguinte *pipeline*: receber um conjunto de séries temporais univariadas, transformá-las em grafos, extrair características desses grafos e, por fim, utilizar essas características como entrada para modelos de classificação. O estudo considerou distintas alternativas para cada etapa da *pipeline*: transformação da série em grafo, extração de características e classificação.

Apesar de abrangente, o trabalho apresenta algumas limitações: não houve busca de hiperparâmetros nos modelos, comprometendo uma avaliação justa do desempenho; não foram realizadas reavaliações sobre o mesmo conjunto de dados, o que pode gerar viés treino-teste; e apenas um modelo estado da arte foi utilizado para comparação.

Nesse contexto, o objetivo deste projeto é expandir o estudo de Coelho [13], aumentando a robustez da análise. Para tanto, o foco é realizar busca de hiperparâmetros, reavaliações sobre o mesmo conjunto de dados, comparação com múltiplos modelos estado da arte e, por fim, incluir a comparação com uma nova abordagem baseada na extração de *features* de grafos e classificadores clássicos.

## III. CONCEITOS BÁSICOS

### A. Série Temporal

Uma série temporal é um conjunto de medições realizadas ao longo do tempo. Exemplos comuns incluem eletrocardiogramas, medições de um sismógrafo e valores financeiros de uma empresa ao longo do tempo.

Uma série temporal pode ser classificada como multivariada quando mais de uma variável é medida simultaneamente em cada instante, ou univariada quando apenas uma variável é observada ao longo do tempo. Este trabalho foca exclusivamente em séries temporais univariadas.

Formalmente, uma série temporal univariada  $T$  é representada como  $T = [(t_1, y_1), (t_2, y_2), \dots, (t_m, y_m)]$ , onde  $(t_i, y_i)$  corresponde ao valor da variável  $y$  observado no instante de tempo  $t_i$ .

## B. Shapelets

*Shapelets* são subsequências curtas e contínuas extraídas de séries temporais, concebidas para representar padrões locais característicos presentes nos dados. Cada *shapelet* é definida como um segmento de comprimento  $l < m$ , onde  $m$  corresponde ao número total de observações da série temporal. A escolha desse segmento visa capturar uma forma ou estrutura recorrente que sintetiza propriedades essenciais da dinâmica temporal observada.

De maneira conceitual, uma *shapelet* pode ser vista como uma entidade descritiva da morfologia local de uma série, expressando variações específicas em amplitude, tendência e oscilação. Ao contrário de medidas globais, que resumem o comportamento integral da série, as *shapelets* enfatizam padrões temporais localizados, permitindo a representação de componentes estruturais intrínsecos ao sinal.

Formalmente, seja uma série temporal univariada  $T = [(t_1, y_1), (t_2, y_2), \dots, (t_m, y_m)]$ , onde  $(t_i, y_i)$  representa o valor  $y_i$  observado no instante de tempo  $t_i$ . Uma *shapelet*  $S$  é definida como uma subsequência contínua de  $T$ ,

$$S = [(t_j, y_j), (t_{j+1}, y_{j+1}), \dots, (t_{j+l-1}, y_{j+l-1})],$$

com  $1 \leq j \leq m - l + 1$ . Essa subsequência preserva a ordem temporal dos elementos e reflete uma forma específica da série original, podendo ser utilizada para caracterização e análise estrutural de padrões temporais.

Em termos conceituais, o estudo de *shapelets* fundamenta-se na hipótese de que séries temporais podem ser decompostas em componentes locais representativos, cujas formas contêm informação relevante sobre o comportamento do processo subjacente.

## C. Grafo de atributos

Um grafo de atributos é uma estrutura formada por nós conectados por arestas, em que cada nó possui um conjunto de atributos associados. Esses atributos podem ser valores numéricos ou categóricos e descrevem propriedades relevantes dos elementos do grafo. Em tarefas que envolvem GNN, esses atributos são essenciais, pois alimentam os processos de agregação e atualização das representações, permitindo que o modelo combine informações estruturais e descritivas.

## D. Rede Neurais em Grafos (GNNs)

Redes Neurais em Grafos (GNNs) são modelos destinados a aprender representações a partir de grafos de atributos. Cada nó possui um conjunto de características que serve como ponto de partida para a rede. A GNN atualiza essas representações por meio de um processo iterativo no qual cada nó combina suas próprias informações com as características dos nós vizinhos. Esse mecanismo de agregação e atualização permite capturar dependências locais e, após múltiplas camadas, padrões estruturais mais amplos.

Modelos como GraphSAGE e GAT diferem principalmente na forma de agregação, mas compartilham o mesmo princípio central: usar atributos dos nós e a conectividade do grafo para produzir embeddings informativos. Em cenários onde séries

temporais são transformadas em grafos, as GNNs exploram simultaneamente a estrutura gerada e os atributos dos nós para realizar a tarefa de classificação.

## E. Features de Grafo

Algumas *features* de grafo foram computadas para servir como base da representação vetorial deles. Essas *features* foram divididas em dois grupos: *features* globais (do grafo como um todo) e *features* locais (baseadas em propriedades dos nós). As *features* globais escolhidas foram: número de arestas, diâmetro, média do caminho mais curto e contagem de *motifs* de tamanho 3, 4 e 5. As *features* locais escolhidas foram: grau, centralidade de grau (*degree centrality*), centralidade de proximidade (*closeness centrality*), centralidade de carga (*load centrality*), centralidade harmônica (*harmonic centrality*), centralidade de intermediação (*betweenness centrality*), PageRank e coeficiente de agrupamento (*clustering coefficient*). A seguir, cada uma dessas *features* será explicada em mais detalhes.

1) *Número de arestas*: Sendo o grafo  $G = (V, E)$ , o número de arestas é dado por  $|E|$ . O número de nós não foi considerado como *feature*, pois na transformação de série temporal para grafos de visibilidade, o número de nós é igual ao tamanho da série original. Como, para os conjuntos de dados utilizados, o comprimento das séries é fixo, todos os grafos associados a um mesmo *dataset* possuem o mesmo número de vértices. Dessa forma, essa medida não fornece poder discriminativo adicional entre as amostras.

2) *Diâmetro*: Dado um grafo  $G = (V, E)$ , a distância entre dois nós  $u, v \in V$  é denotada por  $d(u, v)$  e corresponde ao comprimento do caminho mais curto entre esses dois nós. O *diâmetro* do grafo é definido como a maior dessas distâncias, ou seja:

$$\text{diam}(G) = \max_{u, v \in V} d(u, v) \quad (1)$$

Esse valor representa a maior distância geodésica entre quaisquer dois nós do grafo e fornece uma medida da sua extensão estrutural.

3) *Média do Caminho Mais Curto*: A *média do caminho mais curto* de um grafo  $G = (V, E)$ , onde  $|V| = n$ , é definida como a média das menores distâncias entre todos os pares distintos de nós do grafo. Formalmente:

$$m = \frac{1}{n(n-1)} \sum_{s \neq t \in V} d(s, t) \quad (2)$$

Em que  $d(s, t)$  representa a menor distância entre os nós  $s$  e  $t$ . Esta métrica fornece uma estimativa da eficiência da conectividade global do grafo.

4) *Motifs*: *Motifs* são pequenos subgrafos conexos induzidos e não isomórficos. Cada tipo de *motif* caracteriza um padrão estrutural específico entre os nós que o compõem. Esses *motifs* podem ocorrer múltiplas vezes em um grafo, e a contagem de suas ocorrências permite capturar propriedades topológicas relevantes da estrutura do grafo.

Neste trabalho, todos os *motifs* com 3 nós foram contabilizados exaustivamente. No entanto, como a complexidade da contagem cresce exponencialmente com o número de nós, utilizou-se o algoritmo proposto por Wernicke e Rasche [14] para os *motifs* com 4 e 5 nós. Esse algoritmo utiliza amostragem para reduzir o custo computacional e fornecer uma estimativa eficiente da contagem.

Existem 2 tipos distintos de *motifs* com 3 nós, 6 tipos com 4 nós e 21 tipos com 5 nós. Assim, a extração resultou na contagem de 29 tipos diferentes de *motifs*, sendo cada um deles tratado como uma *feature* individual.



Fig. 1. Motifs não isomórficos com 3 nós em grafos não direcionados.

5) *Centralidade de grau (degree centrality)*: A centralidade de grau é dada pela fórmula:

$$C_D(v) = \frac{d(v)}{n-1} \quad (3)$$

onde  $n$  é o número total de nós do grafo. Essa métrica representa a quantidade de vizinhos de um nó normalizada pelo número máximo possível de vizinhos.

6) *Centralidade de proximidade (closeness centrality)*: A centralidade de proximidade para um nó  $u$  é definida como:

$$C(u) = \frac{n-1}{\sum_{v \neq u} d(v, u)} \quad (4)$$

onde  $n$  é o número total de nós no grafo e  $d(v, u)$  é a distância do caminho mais curto entre os nós  $v$  e  $u$ . Essa medida indica o quão próximo o nó  $u$  está de todos os outros nós no grafo.

7) *Centralidade de carga (load centrality)*: A centralidade de carga mede a importância de um nó com base na quantidade de caminhos mínimos que passam por ele. Formalmente, para um nó  $v$ , a centralidade de carga é a fração de todos os caminhos mínimos entre pares de nós que passam por  $v$ . Essa métrica indica o quanto um nó atua como ponto de passagem na rede.

8) *Centralidade de Intermediação (betweenness centrality)*: A centralidade de intermediação de um nó  $v$  é definida como a soma da fração de todos os caminhos mais curtos entre pares de nós que passam por  $v$ . Formalmente, é dada por:

$$C_B(v) = \sum_{s, t \in V} \frac{\sigma(s, t|v)}{\sigma(s, t)} \quad (5)$$

onde  $V$  é o conjunto de nós,  $\sigma(s, t)$  é o número total de caminhos mais curtos entre os nós  $s$  e  $t$ , e  $\sigma(s, t|v)$  é o número

desses caminhos que passam pelo nó  $v$ . Note-se que se  $v \in \{s, t\}$ , então  $\sigma(s, t|v) = 0$ , e se  $v = t$ ,  $\sigma(s, t) = 1$  [15].

9) *PageRank*: O PageRank é uma medida de centralidade baseada em um modelo de passeio aleatório, que atribui um valor a cada nó proporcional à importância dos nós que o apontam. Embora originalmente desenvolvido para redes direcionadas, o PageRank pode ser aplicado em grafos não direcionados, capturando a relevância estrutural dos nós com base em sua conectividade.

10) *Coefficiente de agrupamento (clustering coefficient)*: O coeficiente de agrupamento de um nó  $v$  mede a tendência dos seus vizinhos em formarem um subgrafo denso, ou seja, a proporção de conexões existentes entre os vizinhos em relação ao total possível. Formalmente, é dado por:

$$C(v) = \frac{2 \times \text{número de arestas entre vizinhos de } v}{d(v) \times (d(v) - 1)} \quad (6)$$

onde  $d(v)$  é o grau do nó  $v$ . Essa métrica captura a presença de comunidades locais dentro do grafo.

#### IV. SÉRIES TEMPORAIS COMO GRAFOS

##### A. Grafo de Visibilidade

A transformação de uma série temporal em um Grafo de Visibilidade Natural (GVN), proposta por Lacasa et al. [5], permite representar a série como um grafo não direcionado e não ponderado, de modo a capturar aspectos topológicos derivados da estrutura temporal dos dados. Existem variações dessa transformação, como o Grafo de Visibilidade Horizontal [16], mas este trabalho se concentrará no GVN. Portanto, o termo Grafo de Visibilidade (GV) será utilizado como sinônimo de Grafo de Visibilidade Natural (GVN).

A transformação de série temporal em GV se dá da seguinte forma: dada uma série temporal

$$T = \{(t_1, y_1), (t_2, y_2), \dots, (t_m, y_m)\},$$

constrói-se o grafo de visibilidade  $G_T(V, E)$ , em que cada ponto  $(t_i, y_i)$  corresponde a um vértice  $v_i \in V$ , e há uma aresta  $(v_a, v_b) \in E$  se, e somente se, para todo ponto intermediário  $(t_c, y_c)$  com  $t_c \in (t_a, t_b)$ , a seguinte condição for satisfeita:

$$y_c < y_b + (y_a - y_b) \cdot \frac{t_b - t_c}{t_b - t_a} \quad (7)$$

Essa equação garante que não há obstrução de visibilidade entre os pontos  $A(t_a, y_a)$  e  $B(t_b, y_b)$  da série. A construção do grafo resulta, por definição, em uma estrutura conectada, o que é uma vantagem para a análise topológica. A versão utilizada neste trabalho segue a implementação otimizada proposta por Lan et al. [17], com complexidade  $O(n \log n)$ .

##### B. Similarity-Aware Time-Series Classification

O *Similarity-Aware Time-Series Classification* (SimTSC) [18] propõe uma reformulação do problema de classificação de séries temporais por meio de grafos. Em vez de representar cada série como um grafo individual, o método transforma cada série temporal em um nó de um grafo global, no qual

as arestas representam o grau de similaridade entre as séries. Dessa forma, a tarefa de classificação passa a ser um problema de *node classification*, em que o objetivo é prever o rótulo de cada nó (série), e não do grafo como um todo. Essa abordagem busca conciliar duas vertentes tradicionais da classificação de séries temporais: (i) métodos baseados em similaridade (por exemplo, 1-NN com DTW) e (ii) modelos *end-to-end* de *deep learning* que aprendem representações diretamente dos dados.

Inicialmente, utiliza-se um *backbone* de aprendizado profundo, tipicamente uma rede convolucional unidimensional (1D-CNN) ou uma *ResNet*, para extrair os *embeddings* iniciais de cada série temporal. Em seguida, constrói-se um grafo de similaridade a partir das distâncias entre as séries, calculadas com base em uma métrica como o *Dynamic Time Warping* (DTW) [19]. As arestas desse grafo podem ser definidas, por exemplo, conectando cada série às suas  $k$  vizinhas mais semelhantes.

O grafo resultante, juntamente com os *embeddings* iniciais, é então utilizado como entrada para uma GNN, que refina as representações de cada nó levando em conta as relações de similaridade. A GNN propaga informação entre séries similares e ajusta os *embeddings* de modo supervisionado, minimizando uma função de perda de entropia cruzada entre as previsões e os rótulos reais das séries. Essa integração entre aprendizado profundo e estrutura de similaridade permite capturar tanto padrões discriminativos locais quanto relações globais entre séries temporais.

### C. Time2graph e Time2graph+

O *framework* Time2graph, de Cheng et al. [20], fundamenta-se no conceito de *shapelets* III-B. Primeiramente, o modelo aprende *time-aware shapelets* a partir de um conjunto de subsequências candidatas, que serão utilizados na construção do *Shapelet Evolution Graph* para capturar a correlação e a evolução temporal entre os *shapelets*. Finalmente, a representação vetorial da série é obtida por meio da agregação dos *embeddings* gerados na fase anterior.

Segundo os autores, a definição tradicional de *shapelets* ignora o fato de que subsequências podem ter significados diferentes em momentos distintos. Por exemplo, baixo consumo de energia na primavera é normal, mas no verão é atípico, pois tende-se a usar mais energia com ar-condicionado, ventiladores etc. Nesse sentido, os autores introduzem o conceito de *time-aware shapelets*, que incorporam fatores temporais locais e globais, permitindo que um mesmo *shapelet* tenha interpretações diferentes ao longo do tempo. A primeira etapa do *framework* consiste, portanto, em aprender os *shapelets* adequados para uma determinada série.

A etapa seguinte consiste na construção do *Shapelet Evolution Graph*, um grafo ponderado  $G(V, E, W)$ , no qual cada vértice representa um *shapelet* aprendido na fase anterior e o peso de cada aresta indica a probabilidade de transição temporal de um *shapelet* para outro. Esse grafo captura como os *shapelets* evoluem ao longo da série.

A etapa final é o aprendizado da representação dos *shapelets* e do grafo. Para isso, é utilizado o algoritmo **DeepWalk**

apresentado por Perozzi et al. [21], que realiza caminhadas aleatórias pelo grafo e obtém como saída *embeddings* representativos de cada *shapelet* (ou nó). Em seguida, cada subsequência da série é associada ao *shapelet* mais similar, o *embedding* correspondente é coletado e a representação final da série é construída por meio da agregação desses *embeddings*.

O Time2graph+ [22] utiliza um *framework* semelhante ao Time2graph original, porém com uma modificação essencial na etapa de aprendizado da representação dos *shapelets*. Em vez de empregar o *DeepWalk* para obter os *embeddings* dos nós do grafo, o modelo adota uma *Graph Attention Network* (GAT), que permite aprender representações mais expressivas ao ponderar, de forma adaptativa, a importância das conexões entre *shapelets*. Dessa forma, a GAT aprende não apenas os vetores de representação de cada nó, mas também os coeficientes de atenção que indicam o grau de relevância de cada vizinho no processo de agregação de informações. Essa abordagem substitui as caminhadas aleatórias usadas no Time2graph original por um mecanismo de atenção, capaz de capturar dependências temporais e estruturais de ordem superior entre os *shapelets*.

## V. METODOLOGIA

### A. Bases de Dados

Foram utilizadas cinco das dez bases de dados de séries temporais univariadas analisadas em [13], todas provenientes do repositório UCR Archive 2018 [23]. A redução no número de bases decorreu de limitações de tempo e recursos computacionais, uma vez que os experimentos envolveram múltiplos modelos, sucessivas reavaliações e buscas extensivas de hiperparâmetros, além da necessidade de execução em GPU.

As bases selecionadas foram: *Earthquakes*, *ECG200*, *ECG-FiveDays*, *Strawberry* e *TwoLeadECG*. A escolha dessas bases fundamenta-se em sua diversidade em termos de desempenho observado nos classificadores em [13], [24], bem como na variação de características como comprimento das séries, número de amostras, e domínio dos dados (por exemplo, sinais fisiológicos e séries derivadas de imagens).

TABLE I  
INFORMAÇÕES GERAIS SOBRE O CONJUNTO DE DADOS

Dataset	Tam. série	Nº séries	Nº classes	Tipo
Earthquakes	512	461	2	SENSOR
ECG200	96	200	2	ECG
ECGFiveDays	136	884	2	ECG
Strawberry	235	983	2	SPECTRO
TwoLeadECG	82	1162	2	ECG

### B. Modelos Avaliados

Todos os modelos baseados em séries temporais como grafos foram adaptados de Coelho [13], exceto o apresentado na Seção V-B1. No estudo de Coelho, é proposta a arquitetura Graph Neural Alchemist (GNA), composta por quatro módulos principais:

- 1) **Módulo de Representação:** recebe as séries temporais como entrada e as converte em representações baseadas em grafos, como os Grafos de Visibilidade (GVs);
- 2) **Módulo de Aprendizado em Grafos:** aplica uma arquitetura *GraphSAGE* com quatro camadas para aprender os *embeddings* dos nós;
- 3) **Camada de Readout:** agrega as representações dos nós em um vetor único que descreve o grafo como um todo;
- 4) **Módulo de Classificação:** consiste em uma rede *Multi-Layer Perceptron* (MLP) que utiliza o vetor obtido na etapa anterior para realizar a classificação da série.

No trabalho de Coelho [13], diferentes variações dessa arquitetura base foram investigadas, explorando múltiplas formas de conversão das séries em grafos e distintas estratégias de inicialização dos *embeddings* dos nós no segundo módulo. Nesta pesquisa, optou-se por concentrar a análise apenas nos modelos mais promissores, selecionados com base no melhor desempenho médio nos conjuntos de dados avaliados por aquele autor. Além disso, também foram usados 3 outros modelos estado da arte para classificação de séries temporais: RocketClassifier, ShapeletTransformClassifier e ShapeletTransformClassifier.

As subseções a seguir descrevem em mais detalhes os modelos baseados em grafos.

1) *Grafo de Visibilidade + Extração de Características + Classificadores (VG-FE)*: Essa abordagem consiste em transformar as séries temporais em Grafos de Visibilidade (GVs), calcular as *features* definidas em III-E e utilizá-las para compor uma representação vetorial de cada série, empregada como entrada em classificadores tradicionais de aprendizado de máquina.

Após a conversão de todas as séries em GVs, as *features* globais do grafo (referentes à estrutura completa) são concatenadas aos valores máximo, mínimo, médio e ao desvio padrão das *features* locais, resultando em um vetor de características que sintetiza as propriedades estruturais de cada série. Essas representações são normalizadas e, então, utilizadas pelos modelos de classificação considerados: Gradient Boosting, Random Forest, SVC, KNN, Decision Tree e Naive Bayes.

2) *Time2graph+*: Implementação original do Time2graph+ com GAT [22].

3) *Time2graph-GNA*: GNA utilizando a representação de grafos de evolução de Shapelets do Time2graph.

4) *SimTSC-GNA*: GNA utilizando estratégia de similaridade do SimTSC, adaptada para classificação de nós.

5) *VG-GNA*: GNA com GV utilizando as *features* locais de nós computadas em III-E mais o sinal  $t_i$  do ponto da série correspondente ao nó.

### C. Configuração Experimental

Todos os modelos escolhidos apresentam diversos hiperparâmetros, que podem impactar o desempenho do algoritmo. Para fazer uma análise mais robusta e justa entre os modelos, foi feita uma busca de hiperparâmetros para cada modelo em cada *dataset*. A busca se deu da seguinte forma: para cada *dataset*, fez-se uma amostragem estratificada usando

30% das amostras. Em seguida, foi realizada uma busca por hiperparâmetros utilizando *Randomized Search* com 30 iterações. Ou seja, foram avaliados 30 conjuntos distintos de hiperparâmetros para cada modelo e para cada *dataset*. Cada conjunto foi treinado e testado por meio de validação cruzada com 5 *folds*. O conjunto de hiperparâmetros que obteve a maior média da métrica F1-score macro no conjunto de teste da validação cruzada foi selecionado como o melhor para aquele modelo e *dataset*.

Após definir a escolha ótima de hiperparâmetros, cada modelo foi retreinado em 80% dos dados e testado nos 20% restantes, utilizando esse conjunto de hiperparâmetros ótimos. Esse processo foi repetido 30 vezes para cada modelo em cada *dataset*, com diferentes partições de treino e teste, mantendo sempre a proporção 80-20.

## VI. RESULTADOS

### A. Desempenho Quantitativo

Nas tabelas II e III é mostrado o F1-score médio e desvio padrão no conjunto de teste de todos os modelos considerando todas as 30 avaliações.

Os resultados médios de F1-score macro e seus desvios padrão, computados ao longo das 30 reavaliações, mostram variação relevante entre os modelos avaliados. Os modelos estado da arte (InceptionTime, ROCKET e ShapeletTransform) apresentam desempenho superior na maior parte dos conjuntos de dados. Entre os modelos baseados em grafos, o SimTSC-GNA e o VG-GNA apresentam desempenho consistentemente mais alto do que as demais variantes.

Os classificadores baseados na extração de *features* de Grafos de Visibilidade (VG-FE) apresentam desempenho inferior aos modelos estado da arte e às GNNs, mas ainda produzem resultados competitivos em alguns cenários, em especial para dados ECG.

### B. Análise Estatística

Para a análise estatística, empregou-se o teste de Friedman [25]–[27], adotando nível de significância de 0,05 para verificar a existência de diferenças estatisticamente relevantes entre os modelos avaliados. Em seguida, aplicou-se o teste post-hoc de Nemenyi [28], permitindo a comparação par a par dos classificadores e a identificação daqueles cuja diferença de desempenho é estatisticamente significativa. A partir desse procedimento, calculou-se a Diferença Crítica (CD), que determina o valor mínimo de separação entre os rankings médios para que dois modelos sejam considerados distintos. Os resultados foram representados visualmente na Figura 2, na qual os modelos são posicionados em um eixo horizontal segundo seus rankings médios, sendo conectados por linhas quando não apresentam diferenças estatísticas significativas.

Devido ao número limitado de datasets, reconhece-se que o poder discriminativo dos testes estatísticos é reduzido, o que restringe a capacidade de identificar diferenças significativas entre todos os modelos. Ainda assim, o diagrama, aliado às métricas de F1-macro, indica que os métodos baseados em extração de características tendem a apresentar desempenho

TABLE II  
F1-SCORE MÉDIO E DESVIO PADRÃO DOS MODELOS VG-FE

Dataset/Modelo	DecisionTree	GradientBoosting	KNN	NaiveBayes	RandomForest	SVC
ECG200	0.63 ± 0.06	0.72 ± 0.06	0.68 ± 0.06	0.70 ± 0.06	<b>0.75 ± 0.06</b>	0.63 ± 0.07
ECGFiveDays	0.85 ± 0.02	0.94 ± 0.02	0.92 ± 0.02	0.84 ± 0.03	0.93 ± 0.02	<b>0.95 ± 0.02</b>
Earthquakes	0.49 ± 0.05	0.50 ± 0.05	0.45 ± 0.02	0.40 ± 0.03	0.49 ± 0.04	<b>0.53 ± 0.05</b>
Strawberry	0.82 ± 0.02	0.89 ± 0.02	0.89 ± 0.03	0.75 ± 0.03	0.90 ± 0.03	<b>0.92 ± 0.02</b>
TwoLeadECG	0.87 ± 0.02	0.93 ± 0.02	0.89 ± 0.02	0.81 ± 0.02	0.92 ± 0.02	<b>0.95 ± 0.01</b>

TABLE III  
F1-SCORE MÉDIO E DESVIO PADRÃO DOS MODELOS AVANÇADOS

Dataset/Modelo	VG-GNA	SimTSC-GNA	Time2graph-GNA	Time2graph+	InceptionTimeClassifier	RocketClassifier	ShapeletTransformClassifier
ECG200	0.75 ± 0.07	0.88 ± 0.05	0.75 ± 0.07	0.81 ± 0.05	0.87 ± 0.05	<b>0.91 ± 0.04</b>	0.82 ± 0.05
ECGFiveDays	0.98 ± 0.01	<b>1.00 ± 0.00</b>	0.99 ± 0.03	0.92 ± 0.02	<b>1.00 ± 0.00</b>	<b>1.00 ± 0.00</b>	<b>1.00 ± 0.00</b>
Earthquakes	0.55 ± 0.05	0.53 ± 0.06	0.48 ± 0.04	0.49 ± 0.05	<b>0.56 ± 0.05</b>	0.52 ± 0.05	0.45 ± 0.01
Strawberry	0.91 ± 0.03	0.96 ± 0.03	0.91 ± 0.02	0.89 ± 0.02	0.96 ± 0.03	<b>0.98 ± 0.01</b>	<b>0.98 ± 0.01</b>
TwoLeadECG	0.98 ± 0.03	<b>1.00 ± 0.00</b>	0.99 ± 0.01	0.93 ± 0.01	<b>1.00 ± 0.00</b>	<b>1.00 ± 0.00</b>	<b>1.00 ± 0.00</b>

inferior aos demais, enquanto os modelos de estado da arte demonstram desempenho superior. Observa-se também que o SimTSC-GNA apresenta rankings médios próximos aos modelos mais avançados, sugerindo competitividade em relação a eles. Entre os métodos baseados em extração de características, o SVC parece ser a alternativa mais eficaz, seguido pelo RandomForest.

## VII. DISCUSSÃO

Os resultados evidenciam que o desempenho dos modelos está diretamente relacionado tanto à natureza da representação adotada quanto às características intrínsecas dos dados. Em geral, modelos estado da arte — em especial ROCKET, InceptionTime e ShapeletTransform — mantêm vantagem consistente devido à sua capacidade de explorar padrões morfológicos de alta resolução diretamente na série temporal. Ainda assim, observa-se que abordagens baseadas em grafos, quando bem estruturadas, conseguem atingir desempenho competitivo em uma parcela significativa dos cenários avaliados.

Um outro ponto interessante é que a diferença de desempenho entre os modelos e o desvio-padrão do F1-score parece ser mais significativa no dataset ECG200, que é o dataset com menos instâncias, enquanto os datasets com mais instâncias, como TwoLeadECG e Strawberry parecem ter um desempenho parecido independente do modelo. Isso parece sugerir que os modelos baseados na representação em grafo podem necessitar de um conjunto de dados maiores para aprender os padrões da séries. Isso faz sentido, pois como esse modelos se baseiam em uma representação intermediária da série, parte da informação pode se perder nessa transformação e o modelo pode precisar recuperá-la no treinamento e um conjunto maior de dados favorece isso. Por outro lado, se o modelo opera diretamente na série temporal, como é o caso dos modelos estado da arte escolhidos, um conjunto menor de dados pode ser o suficiente para que se aprenda os padrões dos dados.

A abordagem VG-FE apresenta desempenho aceitável, porém insuficiente para competir com os modelos estado da arte. Apesar disso, o modelo VG-GNA — que utiliza exatamente a mesma representação de grafo — obtém resultados substancialmente superiores e, em alguns casos, competitivos com os melhores modelos avaliados. Esse contraste indica que as GNNs exploram de forma mais eficaz as estruturas locais e relações topológicas do grafo, enquanto o VG-FE perde informação ao reduzir o grafo a estatísticas agregadas, comprometendo a expressividade da representação.

Observa-se também uma diferença consistente de desempenho entre os modelos Time2Graph-GNA / Time2Graph+ e as demais GNNs avaliadas. Os resultados indicam que a transformação em grafo realizada pelo *Shapelet Evolution Graph* é menos eficaz do que as demais representações consideradas. Isso é reforçado pelo fato de que o Time2Graph-GNA apresentou um ranking médio próximo ao obtido pela abordagem VG-FE com SVC, enquanto o Time2Graph+ chegou a ser superado por esse mesmo modelo. Esse comportamento sugere que, mesmo sem GNNs, uma representação simples baseada em grafos de visibilidade pode ser mais informativa do que o *Shapelet Evolution Graph*.

Além disso, o Time2Graph-GNA superou sistematicamente o Time2Graph+. Como a principal diferença entre eles é o uso de GraphSAGE no primeiro e GAT no segundo, os resultados sugerem que o GraphSAGE produz embeddings mais robustos e representativos para esse tipo de grafo, enquanto a atenção empregada pelo GAT não agrega vantagens nesse cenário.

### A. Recomendação de modelos de acordo com o tipo de dado

#### Tipo de dado: ECG

**Modelos recomendados:** RocketClassifier, InceptionTimeClassifier, ShapeletTransformClassifier, SimTSC-GNA, VG-GNA, Time2Graph-GNA.

**Justificativa:** Embora o *dataset* ECG200 apresente maior variabilidade de desempenho entre os modelos, as GNNs exibem performance competitiva nos demais conjuntos (ECG-

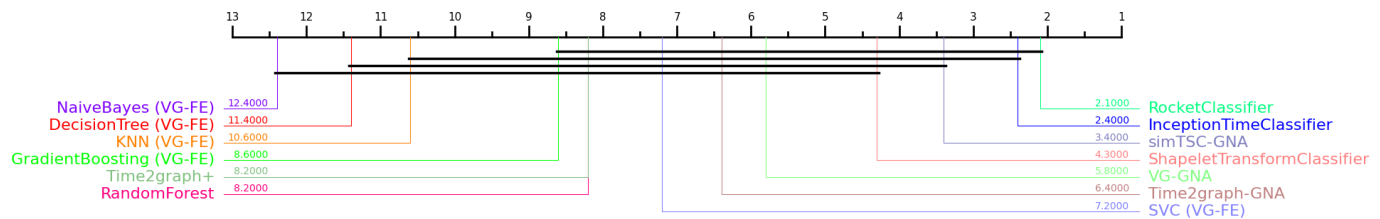


Fig. 2. Diagrama da Diferença Crítica - Ranking dos modelos

FiveDays e TwoLeadECG), nos quais a diferença de F1-macro para o melhor modelo é pequena. Isso é compatível com as propriedades do sinal ECG: trata-se de uma série altamente estruturada, com picos, vales e ciclicidade bem definidas.

A representação via grafo de visibilidade é adequada para esse tipo de sinal por três razões principais: (i) a periodicidade facilita a criação de grafos estáveis, conectando eventos relevantes de forma consistente; (ii) o GV é resistente a ruído fisiológico ou instrumental, pois captura relações topológicas em vez de valores absolutos; (iii) tanto VG-FE quanto VG-GNA alcançam desempenho elevado, confirmando que a estrutura do ECG é preservada de forma discriminativa no grafo.

Métodos como Time2Graph-GNA e ShapeletTransformClassifier também se beneficiam da presença de padrões morfológicos estáveis, pois facilitam a extração de *shapelets* bem definidos.

Já o desempenho superior do SimTSC-GNA é coerente com o fato de que pequenas variações na forma dos ciclos podem indicar diferenças relevantes, e a combinação entre convoluções e a métrica de distância usada para construir as arestas favorece a discriminação dessas variações.

#### Tipo de dado: SENSOR

**Modelos recomendados:** InceptionTimeClassifier, VG-GNA.

**Justificativa:** O *dataset* Earthquakes contém séries extremamente irregulares, com oscilações abruptas típicas de sinais sismográficos, o que torna a tarefa de classificação substancialmente difícil. Nenhum modelo atinge desempenho elevado nesse cenário. Ainda assim, a VG-GNA aproxima-se do melhor F1-macro obtido (0.56), alcançando 0.55 e superando o métodos estado da arte como RocketClassifier e ShapeletTransformClassifier, evidenciando que representações em grafo conseguem capturar, ao menos parcialmente, padrões morfológicos úteis mesmo em dados altamente incertos. Isso demonstra que abordagens baseadas em grafos podem manter competitividade mesmo em tarefas adversas.

#### Tipo de dado: SPECTRO

**Modelos recomendados:** RocketClassifier, ShapeletTransformClassifier, SimTSC-GNA.

**Justificativa:** Para dados espectrográficos, as GNNs apresentaram desempenho promissor, com destaque para o SimTSC-GNA, cujo F1-macro aproxima-se dos modelos estado da arte. Nesse caso, conjectura-se que a adoção de DTW na definição das arestas contribui significativamente, pois essa métrica

lida bem com variações de velocidade temporal, pequenos deslocamentos no tempo e deformações locais — aspectos comuns em sinais espectrais. Essa robustez explica o bom desempenho dos modelos baseados em similaridade e reforça a adequação do SimTSC-GNA para esse tipo de dado.

#### B. Análise dos modelos baseados em grafo

1) *VG-FE*: A abordagem VG-FE é a solução mais simples dentre os métodos avaliados. Não requer GPU, não utiliza redes neurais e possui baixo custo de inferência. Seu principal custo concentra-se na construção do grafo de visibilidade e na computação das *features* globais e locais, que pode ser demorada em séries longas. Apesar de sua simplicidade, o desempenho obtido é limitado, refletindo a perda de informação decorrente da agregação estatística das propriedades do grafo. Assim, embora seja uma abordagem leve e fácil de implementar, não é competitiva frente aos demais métodos mais expressivos.

2) *VG-GNA*: O modelo VG-GNA utiliza a mesma representação de grafo da abordagem VG-FE, mas com aprendizado profundo sobre a estrutura do grafo. Os resultados mostram desempenho significativamente superior ao VG-FE e, em alguns casos, próximo ao estado da arte, indicando que o grafo de visibilidade é uma representação estruturalmente rica quando processado por uma GNN. Entretanto, o método herda o custo de construção do grafo e de extração das *features* de nós, além de exigir GPU para treinar a GraphSAGE. Assim, representa um compromisso entre custo computacional moderado e desempenho competitivo.

3) *Time2Graph-GNA* e *Time2Graph+*: Esses modelos dependem do *Shapelet Evolution Graph*, cujo desempenho está fortemente ligado à clareza dos padrões morfológicos presentes no conjunto de dados. Em domínios como ECG, onde os padrões são bem definidos e repetitivos, a representação tende a capturar transições relevantes entre *shapelets*. No entanto, no geral, essas abordagens apresentaram desempenho inferior em comparação com outras GNNs. Além disso, a etapa de extração de *shapelets* é computacionalmente onerosa, especialmente sob busca de hiperparâmetros. Entre os dois, o Time2Graph-GNA superou o Time2Graph+, sugerindo que o GraphSAGE gera embeddings mais estáveis que a GAT para esse tipo de grafo.

4) *SimTSC-GNA*: O SimTSC-GNA foi o modelo baseado em grafos mais robusto. Seu desempenho foi consistentemente competitivo com os métodos estado da arte e, em



diversos casos, muito próximo dos melhores valores obtidos. A combinação entre a construção do grafo via DTW e o uso de convoluções permite capturar similaridade morfológica de forma precisa. Entretanto, quando o conjunto de dados é pequeno — como no ECG200 — a sensibilidade do DTW e a dependência da estrutura de vizinhança podem afetar o desempenho. Ainda assim, é a solução baseada em grafos com melhor custo-benefício estrutural e melhor estabilidade global.

## VIII. CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi conduzida uma avaliação comparativa rigorosa de treze modelos de classificação de séries temporais, incluindo três baselines de referência e nove modelos baseados em grafos. A metodologia experimental foi estruturada para garantir estimativas estáveis de desempenho: (i) busca intensiva de hiperparâmetros via *randomized search* com 30 iterações e *cross-validation* em 5 *folds* para cada combinação modelo-dataset; (ii) retreinamento seguido de teste de cada modelo 30 vezes sobre cada *dataset* utilizando o melhor conjunto de hiperparâmetros identificado. Esse protocolo reduz variabilidade, mitiga sobreajuste de configuração e fornece comparações estatisticamente fundamentadas.

Os resultados evidenciam que abordagens baseadas em extração de *features* de Grafos de Visibilidade (VG-FE) apresentam desempenho razoável, porém não competitivo frente aos modelos estado da arte. Em contraste, métodos que utilizam grafos como representação intermediária combinados a GNNs demonstram forte potencial: VG-GNA e SimTSC-GNA se destacam por alcançarem desempenho próximo — e, em alguns casos, superior — aos modelos consolidados da literatura. A análise estatística via teste de Friedman seguida do teste post-hoc de Nemenyi permitiu comparar de forma robusta o ranking médio dos classificadores, fornecendo evidências quantitativas claras sobre a superioridade relativa de cada abordagem. Além disso, a análise qualitativa por tipo de dado permitiu identificar modelos mais adequados para cada domínio, oferecendo recomendações direcionadas para aplicações futuras.

Apesar da robustez metodológica, o estudo apresenta limitações. A principal é a pequena quantidade de *datasets* avaliados: cinco conjuntos, ainda que heterogêneos em comprimento das séries, cardinalidade e natureza do sinal. Uma avaliação em escala maior fortaleceria a generalização das conclusões. Ressalta-se que a limitação decorre de custos computacionais elevados — dado o uso intensivo de GPU por quase todos os modelos considerados, a busca intensiva de hiperparâmetros e as múltiplas reavaliações para cada conjunto de dados — e do prazo restrito do projeto.

Para trabalhos futuros, recomenda-se ampliar o conjunto de *datasets* avaliados, incluindo domínios mais variados e *benchmarks* amplamente utilizados na literatura. Outra linha promissora é estender a análise para séries temporais multivariadas, já que este estudo se restringiu ao caso univariado. Além disso, investigar o uso de representações em grafo para outras tarefas além da classificação — como detecção de anomalias, *early classification* e clusterização — pode revelar

propriedades relevantes das abordagens baseadas em grafos e abrir caminhos para novas aplicações.

## REFERENCES

- [1] B. D. Fulcher, “Feature-based time-series analysis,” in *Feature Engineering for Machine Learning and Data Analytics*, G. Zheng, S. Zhu, and V. S. Tseng, Eds. CRC Press, 2018, pp. 87–116. [Online]. Available: <https://doi.org/10.1201/9781315182787-5>
- [2] N. Lucas, C. Hills, J. Lines, and E. Keogh, “Proximity forest: An effective and scalable distance-based classifier,” *Data Mining and Knowledge Discovery*, vol. 33, pp. 607–635, 2019. [Online]. Available: <https://arxiv.org/abs/1808.10594>
- [3] A. Dempster, F. Petitjean, and G. I. Webb, “Rocket: Exceptionally fast and accurate time series classification using random convolutional kernels,” *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1454–1495, 2020. [Online]. Available: <https://doi.org/10.1007/s10618-020-00701-z>
- [4] M. Jin, H. Y. Koh, Q. Wen, D. Zambon, C. Alippi, G. I. Webb, I. King, and S. Pan, “A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10 466–10 485, 2024.
- [5] L. Lacasa, B. Luque, F. Ballesteros, J. Luque, and J. C. Nuño, “From time series to complex networks: The visibility graph,” *Proceedings of the National Academy of Sciences*, vol. 105, no. 13, pp. 4972–4975, 2008.
- [6] B. D. Fulcher and N. S. Jones, “Highly comparative feature-based time-series classification,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 12, pp. 3026–3037, 2014. [Online]. Available: <https://doi.org/10.1109/TKDE.2014.2316504>
- [7] T. Henderson and B. D. Fulcher, “An empirical evaluation of time-series feature sets,” in *2021 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2021, pp. 1032–1038. [Online]. Available: <https://doi.org/10.1109/ICDMW53433.2021.00135>
- [8] L. Ye and E. Keogh, “Time series shapelets: A novel technique that allows accurate, interpretable and fast classification,” *Data Mining and Knowledge Discovery*, vol. 22, pp. 149–182, 2011. [Online]. Available: <https://doi.org/10.1007/s10618-010-0179-5>
- [9] C. Bandt and B. Pompe, “Permutation entropy: A natural complexity measure for time series,” *Physical Review Letters*, vol. 88, no. 17, p. 174102, 2002.
- [10] M. Köppen, “The curse of dimensionality,” in *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*, vol. 1, 2000, pp. 4–8.
- [11] J. B. Borges, H. S. Ramos, R. A. Mini, O. A. Rosso, A. C. Frery, and A. A. Loureiro, “Learning and distinguishing time series dynamics via ordinal patterns transition graphs,” *Applied Mathematics and Computation*, vol. 362, p. 124554, 2019.
- [12] M. McCullough, M. Small, T. Stemler, and H. H.-C. Iu, “Time lagged ordinal partition networks for capturing dynamics of continuous dynamical systems,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 25, no. 5, p. 053101, 2015.
- [13] P. H. S. Coelho, “Análise comparativa da classificação de séries temporais representadas como grafos,” Master’s thesis, Universidade Federal de Minas Gerais, Belo Horizonte, Brasil, 2024.
- [14] S. Wernicke and F. Rasche, “Fanmod: a tool for fast network motif detection,” *Bioinformatics*, vol. 22, no. 9, pp. 1152–1153, 2006.
- [15] U. Brandes, “On variants of shortest-path betweenness centrality and their generic computation,” *Social Networks*, vol. 30, no. 2, pp. 136–145, 2008. [Online]. Available: <https://doi.org/10.1016/j.socnet.2007.11.001>
- [16] B. Luque, L. Lacasa, F. Ballesteros, and J. Luque, “Horizontal visibility graphs: Exact results for random time series,” *Physical Review E*, vol. 80, p. 046103, 2009.
- [17] X. Lan, H. Mo, S. Chen, Q. Liu, and Y. Deng, “Fast transformation from time series to visibility graphs,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 25, no. 8, p. 083105, 2015.
- [18] D. Zha, K.-H. Lai, Z. Kaixiong, and X. Hu, “Towards similarity-aware time-series classification,” in *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 2022, pp. 199–207.
- [19] M. Müller, “Dynamic time warping,” in *Information Retrieval for Music and Motion*. Springer, 2007, pp. 69–84.

- [20] Z. Cheng, Y. Yang, W. Wang, W. Hu, Y. Zhuang, and G. Song, "Time2graph: Revisiting time series modeling with dynamic shapelets," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 3617–3624.
- [21] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*. New York, NY, USA: Association for Computing Machinery, 2014, pp. 701–710. [Online]. Available: <https://doi.org/10.1145/2623330.2623732>
- [22] Z. Cheng, Y. Yang, S. Jiang, W. Hu, Z. Ying, Z. Chai, and C. Wang, "Time2graph+: Bridging time series and graph representation learning via multiple attentions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 2, pp. 2078–2090, 2023.
- [23] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Y. Chen, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, and Hexagon-ML, "The ucr time series classification archive," 2018, october. [Online]. Available: [https://www.cs.ucr.edu/~eamonn/time\\_series\\_data\\_2018/](https://www.cs.ucr.edu/~eamonn/time_series_data_2018/)
- [24] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, 2017.
- [25] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.
- [26] —, "A correction: The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 34, no. 205, pp. 109–109, 1939.
- [27] —, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940.
- [28] P. B. Nemenyi, "Distribution-free multiple comparisons," Ph.D. dissertation, Princeton University, 1963.