

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Paulo Henrique de Souza Coelho

Análise comparativa da classificação de séries temporais representadas como grafos

Belo Horizonte
2025

Paulo Henrique de Souza Coelho

Análise comparativa da classificação de séries temporais representadas como grafos

Versão Final

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Mestre em Ciência da Computação.

Orientador: Renato Vimieiro

Belo Horizonte
2025

2025, Paulo Henrique de Souza Coelho.
Todos os direitos reservados

Coelho, Paulo Henrique de Souza Coelho.

C672a Análise comparativa da classificação de séries temporais representadas com grafos [recurso eletrônico] / Paulo Henrique de Souza Coelho – 2025.
1 recurso online (89 f. il., color.) : pdf.

Orientador: Renato Vimeiro.

Dissertação (Mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.

Referências: f. 81-89.

1. Computação – Teses. 2. Análise de séries temporais – Teoria dos grafos – Teses. 3. Universidade Federal de Minas Gerais Exatas, Departamento de Ciência da Computação.
III. Título.

CDU 519.6*(043)

Ficha catalográfica elaborada pela bibliotecária Irénquer Vismeg Lucas Cruz
CRB 6/819 - Universidade Federal de Minas Gerais - ICEx



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

ANÁLISE COMPARATIVA DA CLASSIFICAÇÃO DE SÉRIES TEMPORAIS REPRESENTADAS COMO GRAFOS

PAULO HENRIQUE DE SOUZA COELHO

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores(a):

Prof. Prof. Renato Vimieiro - Orientador
Departamento de Ciência da Computação - UFMG

Prof. Prof. Martín Gómez Ravetti
Departamento de Ciência da Computação - UFMG

Prof. Paulo Salgado Gomes de Mattos Neto
Centro de Informática - UFPE

Belo Horizonte, 29 de janeiro de 2025.



Documento assinado eletronicamente por **Renato Vimieiro, Professor do Magistério Superior**, em 07/07/2025, às 19:23, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

Documento assinado eletronicamente por **Martin Gomez Ravetti, Professor(a)**, em



09/07/2025, às 12:11, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Paulo Salgado Gomes de Mattos Neto, Usuário Externo**, em 14/07/2025, às 15:32, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4360217** e o código CRC **4D21E25C**.

Referência: Processo nº 23072.241915/2025-91

SEI nº 4360217

Agradecimentos

Aos colegas do Speed e a todos os amigos que emprestaram suas casas para que eu confeccionasse o presente texto, aos meus pais, Creuza e Cornélio, aos meus irmãos, David e Mateus e, em especial a Guilherme “Lenda” Rabelo, Pâmela “Pam” Carvalho, Isabela Couto Campos e Eduardo “UNIMED” Brasil, meu mais profundo: *obrigado*.

Um agradecimento especial ao meu orientador, Renato Vimieiro e meus alunos de Iniciação Científica, Luís F. Ramos e Raul Araju, por suas inestimáveis contribuições ao projeto e inesgotável paciência para comigo.

Registro também meu agradecimento ao professor José Nagib Cotrim Árabe por ter me instigado a tentar o Mestrado após a graduação, e aos professores Martin Gómez Ravetti e Paulo Salgado Gomes de Mattos Neto por terem aceitado o convite para integrar a banca de defesa do presente trabalho.

“Pareça fraco quando você for forte, e forte quando você for fraco.”
(Sun Tzu, A Arte da Guerra)

Resumo

A classificação de séries temporais é um problema fundamental em diversas áreas, como medicina, finanças e engenharia. Recentemente, abordagens baseadas em grafos vêm se mostrando promissoras para essa tarefa, permitindo capturar padrões e relações complexas nos dados temporais.

Este trabalho apresenta um estudo comparativo de diferentes modelos para representação de séries temporais como grafos, investigando: (1) os principais modelos disponíveis; (2) como eles se comparam em termos de concordância e desempenho; (3) sua performance em relação a modelos convolucionais; (4) diferenças estatisticamente significativas entre os modelos; (5) arquiteturas mais eficazes; e (6) a influência das características das séries temporais no desempenho dos classificadores.

Os experimentos foram conduzidos em 16 conjuntos de dados, incluindo 10 do UCR Archive e 6 derivados do CODE-15%, um repositório de eletrocardiogramas. A caracterização dessas séries revelou que força de tendência, sazonalidade, ruído e autocorrelação são fatores determinantes no desempenho dos modelos. Os resultados, avaliados através de testes estatísticos de Friedman com post-hoc de Nemenyi e análise de concordância via Kappa de Fleiss, revelaram que os modelos avaliados possuem desempenhos estatisticamente similares e que existe uma concordância leve a moderada entre os modelos ($\kappa = 0.0176$), com maior estabilidade para séries previsíveis e menor concordância em séries ruidosas.

A seleção do modelo ideal depende do tipo da série temporal, sendo que ROCKET foi mais eficaz para ECG e espectros, SimTSC-GNA para séries com alta autocorrelação e VisibilityGraph-3feat para séries sazonais, conforme detalhado na recomendação de modelos.

Este estudo estabelece a primeira análise estatística abrangente da performance de diferentes modelos de CSTG, fornecendo percepções valiosas sobre sua aplicabilidade e orientando futuras pesquisas no uso de aprendizado em grafos para classificação de séries temporais.

Palavras-chave: matemática; computação; séries temporais; grafos.

Abstract

Time series classification is a fundamental problem in various domains, such as medicine, finance, and engineering. Recently, graph-based approaches have shown promise for this task, enabling the capture of complex patterns and relationships in temporal data.

This work presents a comparative study of different models for time series representation as graphs, investigating: (1) the main models available; (2) how they compare in terms of agreement and performance; (3) their effectiveness compared to convolutional models; (4) statistically significant differences among models; (5) the most effective architectures; and (6) the influence of time series characteristics on classifier performance.

Experiments were conducted on 16 datasets, including 10 from the UCR Archive and 6 derived from CODE-15%, a repository of electrocardiograms. The characterization of these datasets revealed that trend strength, seasonality, noise, and autocorrelation are key factors in model performance. The results, evaluated through Friedman statistical tests with Nemenyi post-hoc analysis and agreement assessment via Fleiss' Kappa, showed that the evaluated models exhibit statistically similar performance and that there is a slight to moderate agreement between models ($\kappa = 0.0176$), with greater stability for structured time series and lower agreement in noisy datasets.

The choice of the best model depends on the type of time series: ROCKET performed best for ECG and spectral datasets, SimTSC-GNA was more effective for highly autocorrelated series, and VisibilityGraph-3feat outperformed others in seasonal time series, as detailed in the model recommendations.

This study establishes the first comprehensive statistical analysis of different CSTG (Classification of Time Series as Graphs) models, providing relevant insights into their applicability and guiding future research on graph-based learning for time series classification.

Keywords: math; computing; time-series; graphs.

Lista de Figuras

2.1	Exemplo de um grafo não direcionado e não ponderado com 10 nós e 20 arestas. Cada nó possui um vetor com duas características numéricas acima dos nós.	27
3.1	Ilustração da arquitetura SimTSC. Adaptado de [90].	30
3.2	Histograma de uma série temporal sintética de 20 valores com a projeção de visibilidade determinada pelo critério da equação 3.7. A topologia do GVN resultante preserva a estrutura espaço-temporal de t , sendo não-direcionado, conectado e invariante à escala . Valores da série temporal: {6.0, 2.8, 2.0, 4.2, 5.8, 3.0, 1.8, 4.5, 6.5, 3.0, 1.8, 4.2, 6.2, 2.0, 2.5, 4.8, 3.4, 3.4, 2.0, 5.0}	36
5.1	Arquitetura da GNA utilizada no presente trabalho. A série temporal é representada como um grafo e processada por 4 camadas de GraphSAGE, seguidas de uma MLP modificada para realizar <i>pooling</i> dos <i>embeddings</i> aprendidos. A classificação final é realizada por uma função Log-Softmax. Cada componente da arquitetura é modular, permitindo testar e comparar diferentes métodos de representação e classificação de séries como grafos.	56
5.2	Valores-p corrigidos obtidos pelos testes estatísticos de Friedman e post-hoc de Nemenyi.	62
5.3	Diagrama de Diferença Crítica baseado no Teste de Friedman seguido do teste de Nemenyi. Nessa visualização, os modelos são organizados em um eixo horizontal de acordo com seu ranking médio de desempenho. Modelos que não apresentam diferenças estatisticamente significativas são agrupados em cliques por meio de linhas horizontais.	63
5.4	Mapa de calor com dendrograma representando a concordância Kappa de Fleiss entre os modelos avaliados.	65
5.5	Correlação de Spearman entre características dos dados e F1-score por modelo.	68
5.6	Comparação entre as matrizes de correlação de Pearson (esquerda) e Spearman (direita) para as características estruturais das séries temporais. Observe maior consistência e intensidade das correlações na matriz de Spearman, como entre autocorrelação e tendência ($\rho = 0,83$) e entre autocorrelação e sazonalidade ($\rho = -0,66$).	70
5.7	Gráfico radar com os valores médios de <i>macro F1-score</i> dos modelos por tipo de série temporal. A análise abrange os domínios de cada conjunto de dados, permitindo uma visualização comparativa da performance média de cada modelo.	71

Lista de Tabelas

5.1	Descrição dos Conjuntos de Dados do UCR-Archive	47
5.2	Descrição das Anomalias e distribuição de anotações para cada um dos seis conjuntos de dados derivados do CODE-15%	50
5.3	Características extraídas dos conjuntos de séries temporais, destacando Força da Tendência (trd), Força da Sazonalidade (s), Variância Residual (σ^2), Entropia Residual (H) e Autocorrelação (ρ). Valores em destaque indicam a maior pontuação em cada métrica.	53
5.4	Interpretação dos valores k da métrica Kappa de Fleiss	54
5.5	F1-score médio para cada modelo e conjunto de dados	60
5.6	Kappa Score para cada Conjunto de Dados.	64
5.7	Correlação de Pearson entre Kappa de Fleiss e as Características das Séries Temporais.	66
5.8	Média das características extraídas dos conjuntos de séries temporais, destacando Força da Tendência (trd), Força da Sazonalidade (s), Variância Residual (σ^2), Entropia Residual (H) e Autocorrelação (ρ). Valores em destaque indicam a maior pontuação em cada métrica.	72
5.9	Recomendações de modelos por tipo de série temporal, com base nas características estruturais e no desempenho empírico.	74

Lista de Abreviações e Acrônimos

AM	Aprendizado de Máquina
CNN	Redes Neurais Convolucionais
CSTG	Classificação de Séries Temporais como Grafos
CSST	Classificação Supervisionada de Séries Temporais
DNNs	Redes Neurais Profundas
DTW	<i>Dynamic Time Warping</i> (Alinhamento Dinâmico de Tempo)
ECG	Eletrocardiograma
GAT	<i>Graph Attention Network</i>
GCN	<i>Graph Convolutional Network</i>
GDL	<i>Geometric Deep Learning</i>
GNA	<i>Graph Neural Alchemist</i>
GNNs	Redes Neurais em Grafos
GTPO	Grafo de Transição de Padrões Ordinais
GVN	Grafo de Visibilidade Natural
IoT	Internet das Coisas
LSTM	<i>Long Short-Term Memory</i>
MLP	<i>Multilayer Perceptron</i> (Perceptron Multicamadas)
NIFECG-1	<i>NonInvasiveFetalECGThorax1</i>
NIFECG-2	<i>NonInvasiveFetalECGThorax2</i>
PLN	Processamento de Linguagem Natural
PO	Padrões Ordinais
ReLU	<i>Rectified Linear Unit</i>
ROCKET	<i>RandOm Convolutional KErnel Transform</i>
RSTG	Representação de Séries Temporais como Grafos
UCR	<i>University of California, Riverside</i> (Repositório UCR de Séries Temporais)

Sumário

1	Introdução	14
2	Conceitos Básicos	19
2.1	Séries Temporais	19
2.1.1	Notação	20
2.1.2	Alinhamento e Subsequências	21
2.1.3	Simbolização de Bandt-Pompe	23
2.2	Grafos	25
3	Séries Temporais como Grafos	28
3.1	Similarity-Aware Time-Series Classification	29
3.2	Time2Graph e Time2Graph Plus	30
3.3	Grafo de Transição de Padrões Ordinais	33
3.4	Grafo de Visibilidade	35
4	Aprendizado de Representação em Grafos	38
4.1	DeepWalk	39
4.2	Redes Neurais em Grafos	40
4.2.1	Arquiteturas de GNN	41
4.2.1.1	Modelo Base de GNN	42
4.2.1.2	Graph Convolutional Network (GCN)	42
4.2.1.3	GraphSAGE	43
4.2.1.4	Graph Attention Network (GAT)	43
5	Experimentos e Análises	46
5.1	Conjuntos de Dados	46
5.1.1	Conjuntos de Dados do UCR-Archive	47
5.1.2	Conjuntos de Dados do CODE-15%	49
5.1.3	Extração de Características das Séries Temporais	50
5.2	Métricas de Avaliação	53
5.3	Implementação e Treinamento	55
5.4	Análises Estatísticas	59
5.4.1	Resultados das Avaliações de Desempenho	59
5.4.2	Teste de Friedman e Diferença Crítica	61

5.5	Concordância entre Avaliadores	63
5.5.1	Correlação entre Concordância e Estrutura das Séries Temporais . .	65
5.6	Discussão	67
5.6.1	Recomendação de Modelos por Tipo de Série Temporal	70
6	Conclusão	75
6.1	Resultados Principais	75
6.2	Desafios e Limitações	76
6.3	Considerações Finais e Direções Futuras	78
	Referências	80

Capítulo 1

Introdução

Séries Temporais são uma coleção de observações feitas sequencialmente ao longo do tempo, sendo ubíquas em diversas áreas do conhecimento e da vida cotidiana. Desde o registro de batimentos cardíacos em um eletrocardiograma (ECG), registros sismográficos de terremotos ou até mesmo trajetórias de veículos, séries temporais capturam a evolução de fenômenos ao longo do tempo, fornecendo informações valiosas para análises subsequentes sobre padrões, identificação de tendências e previsões futuras. Conseqüentemente, a análise de séries temporais constitui uma área de pesquisa ativa e multidisciplinar, com aplicações indo desde medicina até meteorologia.

Uma característica essencial desse tipo de dado é que a ordem das observações importa, pois há uma dependência intrínseca entre valores consecutivos. Em específico, no contexto da tarefa de Classificação Supervisionada de Séries Temporais (CSST), a modelagem dessa dependência de maneira eficiente torna-se fundamental para a extração de conhecimento e a realização de previsões precisas.

Métodos estatísticos, como médias móveis, decomposição de séries e suavização exponencial, incluindo modelos como *SARIMA* [5, 26, 75], têm sido amplamente utilizados para previsão e análise de tendências. Além desses, abordagens baseadas na extração de características e no aprendizado de padrões foram propostas para capturar informações relevantes em séries temporais [32, 44], demonstrando sucesso significativo em tarefas de classificação [33].

Além disso, métodos baseados na identificação de subsequências e padrões característicos, como *Shapelets* e *Padrões Ordinais (PO)*, foram propostos para capturar propriedades fundamentais das séries, como propagação, atraso e ordem relativa entre os pontos [3, 87]. Nos últimos anos, técnicas de aprendizado profundo, como Redes Neurais Convolucionais Unidimensionais (CNNs-1D), têm sido exploradas para modelagem de séries temporais, permitindo a extração de características complexas e a captura eficaz de dependências temporais [24, 67]. Paralelamente, técnicas de alinhamento temporal, como *Dynamic Time Warping (DTW)*, são amplamente empregadas para comparar séries temporais de diferentes comprimentos e formas [48].

Embora seja fundamental para a aplicação de modelos de CSST, a extração de

características enfrenta desafios significativos devido à chamada *maldição da dimensionalidade* [50]. Modelos baseados nessa abordagem podem ter dificuldades em capturar padrões complexos devido à alta dimensionalidade e às dependências não lineares presentes nas séries, agravadas pela natureza exponencial da representação em espaços latentes de busca. Dessa forma, desenvolver técnicas eficazes e eficientes para a extração de características torna-se uma etapa crucial para o sucesso desses modelos.

Para superar esse desafio, é necessário adotar representações intermediárias que preservem as informações temporais relevantes e capturem padrões subjacentes de maneira eficiente. Uma abordagem promissora é a Representação de Séries Temporais como Grafos (RSTG), que permite representar relacionamentos de curto e longo alcance ao projetar as séries em uma estrutura não euclidiana de grafos [12, 47, 71]. Essa representação facilita a captura de padrões ocultos e a modelagem das dependências temporais complexas, proporcionando uma visão mais estruturada das relações entre os pontos da série.

A adoção dessa representação viabiliza a aplicação de conceitos de Geometric Deep Learning (GDL), permitindo o aprendizado de representações mais expressivas e robustas para a tarefa de classificação. Ao generalizar princípios fundamentais das redes neurais profundas para domínios sem estrutura regular, o GDL possibilita a aplicação de operações como convoluções e agregações que respeitam a topologia inerente dos dados [7].

Estudos recentes [60, 83] demonstram que a RSTG é capaz de explorar de forma mais eficiente as interdependências temporais por meio de arquiteturas avançadas, como *edge-enhanced dynamic graphs* e *graph tensor networks*. Além disso Chen and Eldardiry [12], Jin et al. [47] oferecem uma visão abrangente de modelos práticos e de código aberto baseados nas abordagens RSTG *series-as-nodes* e *series-as-graphs* – que representam séries temporais como nós ou grafos completos, respectivamente – destacando técnicas avançadas de extração de características e modelos híbridos que combinam Aprendizado de Máquina (AM) e GDL para a classificação de séries temporais.

Dentre as soluções mais representativas, destacam-se o SimTSC[90] e o Time2Graph[15, 16], que demonstram alta eficácia na modelagem de relações temporais complexas. Esses modelos exemplificam como a combinação de representações baseadas em grafos e aprendizado profundo pode superar limitações dos métodos tradicionais, abrindo caminho para avanços na classificação de séries temporais.

Além desses modelos, abordagens estruturais, como *Grafos de Visibilidade* [51, 52, 53, 61] e *Grafos de Transição de Padrões Ordinais* [54, 76], têm sido amplamente estudadas para a construção de modelos de RSTG para classificação de séries temporais [8, 19, 62]. Essas abordagens exploram diferentes princípios estruturais para transformar séries temporais em redes complexas, permitindo a extração de características topológicas ou aplicação de métricas de diversidade que capturam padrões dinâmicos e recorrentes [20].

No entanto, essa diversidade de métodos para extração de características e classificação levanta um desafio fundamental: com tantas abordagens distintas, como podemos

compará-las de maneira justa e objetiva para determinar qual método é mais eficaz para uma tarefa específica? Embora diversos trabalhos proponham novas modelagens e arquiteturas para Classificação de Séries Temporais como Grafos (CSTG), ainda há uma carência de estudos comparativos sistemáticos que avaliem o desempenho dessas abordagens em diferentes domínios e contextos.

Para suprir essa lacuna, esta dissertação propõe e realiza uma análise quantitativa e comparativa do estado da arte de modelos de classificação de séries temporais representadas como grafos. O objetivo principal é investigar a eficácia de diferentes representações e metodologias, considerando tanto aspectos de desempenho preditivo quanto a interpretabilidade dos modelos gerados. Para isso, serão avaliadas diversas abordagens, incluindo métodos estruturais e técnicas baseadas em GDL, em um conjunto diversificado de bases de dados e métricas de avaliação.

Acreditamos que uma comparação sistemática permitirá não apenas identificar as metodologias mais promissoras, mas também oferecer diretrizes para a escolha adequada de modelos em diferentes cenários. Além disso, ao analisar as vantagens e limitações de cada abordagem, buscamos contribuir para o desenvolvimento de futuras pesquisas na interseção entre séries temporais, aprendizado profundo e modelagem baseada em grafos.

A fim de orientar nossa investigação, formulamos as seguintes perguntas de pesquisa:

1. Quais são os principais modelos para CSTG?
2. Como esses modelos se comparam em termos de concordância e desempenho em diferentes conjuntos de dados?
3. Em comparação a um modelo estado da arte baseado em convoluções, como os modelos de CSTG performam?
4. Há diferenças estatisticamente significativas entre os diferentes modelos?
5. A escolha da abordagem de RSTG influencia no desempenho dos classificadores?
6. Os conjuntos de dados e suas características intrínsecas influenciam o desempenho dos classificadores?

Para responder a essas questões, apresentamos, treinamos e testamos nove modelos de CSTG, derivados das abordagens estruturais e de aprendizado profundo apresentadas anteriormente, comparando-os com o modelo **R**and**O**m Convolutional **K**Ernel **T**ransform (ROCKET), um modelo amplamente reconhecido na literatura como eficiente e robusto para CSST [47, 56, 57, 93]. O ROCKET aplica uma série de kernels convolucionais aleatórios, com parâmetros como comprimento, pesos, *bias*, dilatação e preenchimento

definidos aleatoriamente a partir de um pool inicial e sem realizar atualização retroativa (*backpropagation*) dos pesos, acelerando seu treinamento Dempster et al. [24]. As transformações geradas por esses kernels alimentam um classificador linear, formando, na prática, uma CNN-1D de camada única com pesos aleatórios.

Os modelos foram treinados e avaliados em 16 conjuntos de dados de séries temporais univariadas, abrangendo uma diversidade de domínios e características, incluindo séries sintéticas e reais. Para garantir uma comparação rigorosa e objetiva, conduzimos uma análise de concordância utilizando o *Kappa de Fleiss* sobre os *predicts* dos modelos contra os rótulos verdadeiros, além de uma avaliação estatística de desempenho por meio dos testes de hipóteses de Friedman e Nemenyi sobre as métricas de *Macro F1-Score*.

Esses testes permitem identificar não apenas as diferenças de desempenho entre os modelos, mas também a significância estatística dessas variações, oferecendo uma base sólida para recomendações sobre os modelos mais eficazes na tarefa de classificação de séries temporais representadas como grafos.

Com essa abordagem, buscamos não apenas avaliar a eficácia dos modelos existentes, mas também fornecer diretrizes para futuras pesquisas na área, contribuindo para um melhor entendimento sobre os desafios e vantagens do uso de representações baseadas em grafos para a classificação de séries temporais.

Os principais resultados e conclusões deste trabalho incluem:

- O ROCKET obteve o melhor desempenho médio em termos de F1-score, mas modelos baseados em grafos, como SimTSC e Grafos de Visibilidade apresentaram resultados estatisticamente similares em diversas tarefas.
- A análise de Kappa de Fleiss revelou uma concordância geralmente baixa entre os modelos, especialmente para séries ruidosas e com alta variância residual. Em contrapartida, séries altamente estruturadas favorecem maior estabilidade entre as classificações.
- A análise de características estruturais das séries, como tendência e sazonalidade demonstrou que a escolha da representação influencia significativamente o desempenho dos modelos, sendo que séries com forte tendência e alta autocorrelação favoreceram representações baseadas em similaridade, enquanto séries sazonais se beneficiaram de representações em grafos de visibilidade.

Esta dissertação está organizada da seguinte forma:

No Capítulo 2, apresentamos os conceitos fundamentais necessários para a compreensão do trabalho, incluindo definições de séries temporais, técnicas de extração de características e princípios da teoria de grafos.

No Capítulo 3, discutimos em detalhes os diferentes modelos para representação e CSTG. Entre os métodos abordados, destacamos SimTSC, Time2Graph, Grafos de Visibilidade e Grafos de Transição de Padrões Ordinais.

O Capítulo 4 introduz o aprendizado de representações em grafos, explorando desde abordagens clássicas, como *DeepWalk*, até as modernas **Redes Neurais em Grafos (GNNs)**.

No Capítulo 5, realizamos uma análise experimental abrangente, comparando estatisticamente a concordância e o desempenho dos diferentes modelos em diversos conjuntos de dados de séries temporais, recomendando os modelos mais eficazes para diferentes tipos de séries. Além disso, extraímos características estruturais das séries temporais e analisamos a sua influência no desempenho dos modelos.

Por fim, no Capítulo 6, sintetizamos as principais conclusões do trabalho, respondemos às perguntas de pesquisa propostas e discutimos as contribuições, limitações e possíveis direções para pesquisas futuras.

Capítulo 2

Conceitos Básicos

Neste capítulo, apresentamos os fundamentos teóricos essenciais para a compreensão dos métodos e algoritmos utilizados ao longo deste trabalho. Primeiramente, discutimos as Séries Temporais, explorando suas propriedades, desafios e principais técnicas para extração de características, como Shapelets e Simbolização de Bandt-Pompe. Também abordamos métodos de alinhamento, como o DTW, que desempenha um papel fundamental na análise de similaridade entre séries temporais.

Em seguida, apresentamos a teoria de Grafos, uma estrutura matemática fundamental para modelagem de relações complexas. Definimos suas principais propriedades, como nós, arestas, graus e pesos, e exploramos diferentes representações, incluindo matrizes e listas de adjacência. Além disso, discutimos a aplicação de grafos para representar séries temporais, preparando o leitor para os capítulos subsequentes, nos quais abordaremos o uso de GNNs e outras técnicas de aprendizado profundo para análise de séries temporais.

Esse embasamento teórico é fundamental para a compreensão dos modelos apresentados nos próximos capítulos, fornecendo um arcabouço sólido para a aplicação de técnicas de aprendizado de máquina e modelagem em grafos.

2.1 Séries Temporais

Séries temporais são conjuntos de dados obtidos a partir de observações ao longo do tempo, podendo ou não estar uniformemente espaçadas, e caracterizam-se pela dependência serial entre as observações [10]. Essas séries são amplamente utilizadas para modelar e prever comportamentos em diversas áreas do conhecimento, como economia (evolução do PIB), climatologia (variação da temperatura ao longo de um mês), finanças (flutuação do preço de uma ação), e saúde (monitoramento cardíaco por meio de eletro-

cardiogramas).

A análise de séries temporais desempenha um papel crucial em várias aplicações, especialmente em tarefas de CSST. Diferentemente da classificação tradicional, a CSST lida com atributos que possuem ordem e dependência temporal, tornando a extração de características discriminatórias um dos desafios centrais nesse contexto [2]. A identificação desses padrões é essencial para garantir a precisão dos modelos. Por exemplo, na análise de eletrocardiogramas (ECG), a classificação pode indicar se o exame revela uma condição cardíaca saudável ou se há indícios de doenças como arritmia ou insuficiência cardíaca. Nesses casos, a ordem, os padrões e a amplitude dos batimentos cardíacos são determinantes para o diagnóstico.

A extração de informações relevantes de séries temporais requer técnicas capazes de identificar padrões ocultos que não são imediatamente aparentes nos dados brutos. Isso inclui a detecção de sequências altamente discriminativas para uma classe específica, como picos anormais em um ECG ou oscilações características em séries financeiras. Além disso, o uso de medidas de similaridade permite comparar séries temporais distintas considerando variações no tempo ou na amplitude, garantindo uma análise mais robusta [65].

O objetivo final dessas técnicas é transformar as séries temporais em representações matemáticas robustas, como vetores, matrizes ou grafos, que podem ser utilizadas por modelos estatísticos, algoritmos de AM e redes neurais. Essas representações viabilizam tarefas como classificação, previsão de valores futuros e detecção de anomalias.

Dentre as principais técnicas de extração de características amplamente utilizadas na literatura, este trabalho aborda Shapelets e Simbolização de Bandt-Pompe, antes de introduzir os modelos de RSTG que empregam essas abordagens, no Capítulo 3. Além disso, destacamos a DTW, uma técnica de alinhamento temporal amplamente utilizada para medir similaridade entre séries e extrair padrões de forma eficiente, sendo frequentemente empregada como métrica de distância no aprendizado de Shapelets.

2.1.1 Notação

Nesta subseção, apresentamos as notações que serão adotadas ao longo deste e dos próximos capítulos para facilitar a formalização dos conceitos e modelos discutidos.

- **Série Temporal Univariada:** Uma série temporal univariada será denotada por T . Formalmente, uma série temporal é uma sequência de valores $T = \{t_1, t_2, \dots, t_m\}$, onde t_i representa o valor no tempo i e m é o tamanho da série, ou seja, o número

total de observações.

- **Índice Temporal:** O índice que representa o instante no qual uma observação específica ocorre será denotado por i . Assim, t_i refere-se ao valor da série temporal no i -ésimo ponto no tempo.
- **Medidas de Distância/Semelhança:** A similaridade ou distância entre duas séries temporais, digamos T e T' , será denotada por $d(T, T')$. Medidas específicas, como o DTW, serão detalhadas posteriormente quando aplicadas.

2.1.2 Alinhamento e Subsequências

O alinhamento de séries temporais é uma etapa fundamental para medir similaridade e capturar padrões temporais relevantes, especialmente em tarefas de CSST. Uma das técnicas mais utilizadas para esse fim é o DTW, que busca encontrar a correspondência ótima entre duas sequências ao minimizar a distância total entre seus pontos alinhados. O termo “Dynamic” refere-se ao uso de Programação Dinâmica (PD) para garantir uma implementação eficiente do alinhamento entre as séries através de uma matriz de memorização [58].

Formalmente, considere duas séries temporais:

- $T = \{t_1, t_2, \dots, t_m\}$ com m pontos
- $R = \{r_1, r_2, \dots, r_n\}$ com n pontos

O cálculo da distância de alinhamento temporal DTW entre T e R é definido por:

$$\text{DTW}(T, R) = \min_{A \in \mathcal{A}} \left\{ \sum_{(i,j) \in A} \|t_i - r_j\| \right\} \quad (2.1)$$

onde:

- \mathcal{A} é o conjunto de todos os possíveis caminhos de alinhamento entre os pontos de T e R . Esses caminhos representam diferentes formas de mapear os elementos das duas séries temporais, permitindo variações na escala temporal.
- $A = \{(i_1, j_1), (i_2, j_2), \dots, (i_k, j_k)\} \in \mathcal{A}$ é um caminho de alinhamento que respeita três propriedades fundamentais:

- **Monotonicidade:** A sequência de índices não pode retroceder, ou seja, se um ponto t_i está alinhado com um ponto r_j , então qualquer ponto subsequente t_{i+1} só pode ser alinhado com um ponto $r_{j'}$ tal que $j' \geq j$. Essa restrição preserva a ordem temporal dos eventos.
 - **Continuidade:** O alinhamento não pode saltar pontos arbitrariamente. Se um ponto t_i está alinhado com r_j , então os pontos próximos a t_i devem ser alinhados com pontos próximos a r_j . Isso garante uma correspondência suave entre as séries temporais.
 - **Alinhamento completo entre as séries:** Cada elemento da série T deve estar alinhado a pelo menos um elemento da série R , e vice-versa. Esse requisito assegura que não haja trechos das séries ignorados no alinhamento.
- t_i e r_j são pontos correspondentes nas séries temporais T e R , ou seja, representam os valores das séries nos instantes de tempo i e j , respectivamente.
 - $\|t_i - r_j\|$ representa a distância euclidiana entre os pontos t_i e r_j . Essa métrica mede a diferença absoluta entre os valores das séries temporais no alinhamento, sendo usada na minimização da distância total no DTW.

O DTW permite medir similaridade entre séries temporais de diferentes comprimentos e detectar padrões de forma robusta, sendo amplamente utilizado em tarefas de classificação, como a extração de Shapelets, discutida a seguir.

Shapelets

O conceito de Shapelet foi introduzido por Ye and Keogh [86] e define uma subsequência S de uma série T altamente representativa de uma determinada classe, podendo ser usada para classificação e análise de séries temporais. Por exemplo, em um problema de classificação binária, S pode particionar T em dois subconjuntos, S_{pos} e S_{neg} , de forma que a maioria dos elementos da série pertencentes a uma determinada classe e próximos a S sob uma métrica de distância d pertencem a S_{pos} , enquanto aqueles mais distantes pertencem a S_{neg} [45].

A extração de Shapelets pode ser realizada por busca exaustiva em um *pool* de subsequências candidatas de tamanho C , no qual um número razoável de possíveis subsequências de tamanho l em T são geradas. Esse *pool* é obtido ao deslizar uma janela de tamanho l sobre T , gerando o conjunto \mathbf{S}_T de todas as subsequências possíveis S_p^l com

$1 \leq p \leq m - l + 1$. A partir dessas subsequências, calcula-se uma métrica de distância $d = \text{Dist}(T, S')$ entre T e cada $S' \in \mathbf{S}_T$. Usualmente, são extraídos os Shapelets ao selecionar as K subsequências com menor distância.

Uma abordagem alternativa para extração, denominada **Aprendizado de Shapelets Temporais**, propõe a otimização de uma função de classificação diretamente sobre o conjunto de candidatos, utilizando Descida de Gradiente para aprender Shapelets ótimos sem necessidade de busca exaustiva em \mathbf{S}_T [37]. Essa estratégia é particularmente interessante, pois permite analisar a interação entre Shapelets, incluindo sua frequência e as probabilidades de transição entre diferentes subsequências, viabilizando a construção de modelos de classificação baseados nessas características [15].

Apesar das vantagens dessas técnicas para modelagem e classificação, Shapelets e DTW apresentam um alto custo computacional devido à necessidade de comparar múltiplas subsequências e otimizar o alinhamento entre séries temporais. A extração de Shapelets envolve a busca por subsequências discriminativas em grandes volumes de dados, o que pode exigir um tempo considerável, especialmente em conjuntos de dados extensos [28].

Já o DTW, apesar de ser uma das técnicas mais eficazes para medir similaridade temporal, demanda um alto número de cálculos para alinhar sequências de diferentes comprimentos, tornando seu uso impraticável para aplicações em tempo real ou em bases de dados muito grandes [36]. Mesmo com otimizações modernas, esses métodos ainda enfrentam desafios de escalabilidade, tornando sua aplicação limitada em conjuntos de dados extensos ou em tempo real, ressaltando a necessidade de abordagens mais eficientes para a extração e representação de padrões em séries temporais.

2.1.3 Simbolização de Bandt-Pompe

Propostos por Bandt and Pompe [3], os Padrões Ordinais (PO) são uma técnica poderosa para identificar padrões recorrentes em séries temporais e extrair características úteis para tarefas de CSST. Essa abordagem, conhecida como Método de Bandt-Pompe, baseia-se na ordenação relativa dos valores dentro de subsequências da série, gerando uma representação simbólica que preserva a estrutura temporal dos dados.

O método inicia-se com o particionamento da série temporal T em n subconjuntos de dimensão D , com um intervalo τ entre os pontos consecutivos. Formalmente, cada partição é definida como:

$$T_i^{(D,\tau)} = (t_i, t_{i+\tau}, \dots, t_{i+(D-1)\tau}) \quad (2.2)$$

para $i = 1, 2, \dots, n$, onde $n = m - (D - 1)\tau$. Em seguida, para cada partição, são gerados os Padrões Ordinais $\tilde{\pi}_i^D$ a partir da permutação dos índices dos elementos da subsequência em ordem crescente. Finalmente, a ordenação dos índices temporais conforme suas amplitudes resulta na atribuição de uma permutação única a cada partição, estabelecendo assim uma representação simbólica da série.

A partir dessa construção, Chagas et al. [10] definem a Simbolização de Bandt-Pompe como uma abordagem transparente e não paramétrica para representar séries temporais. Cada subsequência $T_i^{(D,\tau)}$ é associada a um símbolo $\pi_i^{(D,\tau)}$ correspondente a uma das $D!$ possíveis permutações ordenadas. Essa representação permite caracterizar a dinâmica da série com base na frequência e na distribuição desses padrões ao longo do tempo.

Por ser um método não paramétrico, a Simbolização de Bandt-Pompe apresenta diversas vantagens:

- **Simplicidade e eficiência:** O método é computacionalmente leve e não requer ajustes complexos de parâmetros.
- **Robustez a ruídos e outliers:** A ordenação relativa dos valores garante que pequenas variações na amplitude dos dados não comprometam a extração dos padrões.
- **Invariância a transformações monotônicas:** Alterações que preservam a ordem relativa dos pontos, como normalizações e escalonamentos, não afetam a representação simbólica.

Essas características fazem do método uma escolha versátil para modelagem de séries temporais em diversas aplicações, incluindo a identificação e classificação de padrões de deslocamento por modais de transporte [8] e a detecção de ataques de botnets em redes sem fio [10], a partir de séries temporais geradas por medições de GPS e sensores interconectados em redes de Internet das Coisas (**IoT**).

Além de sua aplicabilidade direta na análise de séries temporais, a representação simbólica obtida pelo Método de Bandt-Pompe viabiliza o uso de métricas estatísticas e técnicas complementares para classificação e previsão. Entre elas, destacam-se:

- **Probabilidade de Auto-transição:** Mede a recorrência de padrões dentro da série, permitindo a análise da autocorrelação dos símbolos [4].
- **Histogramas de frequência:** Representam a distribuição de probabilidades dos símbolos na série, auxiliando na diferenciação de padrões entre classes.

- **Grafo de Transição de Padrões Ordinais:** Modelo transacional que estrutura os relacionamentos entre os **PO** como um grafo, permitindo a análise de dependências temporais complexas [4, 76].

A Simbolização de Bandt-Pompe, portanto, não apenas reduz a complexidade da série temporal, mas também fornece uma base robusta para análise estrutural e classificação de padrões, tornando-a uma alternativa eficiente a métodos convencionais de extração de características.

2.2 Grafos

Um grafo, denotado por $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, é uma poderosa abstração matemática utilizada para representar dados não euclidianos que possuem relações entre si. Formalmente, um grafo é composto por dois conjuntos: o conjunto \mathcal{V} de nós (ou vértices) e o conjunto \mathcal{E} de arestas, que representam os relacionamentos entre os nós.

A estrutura de um grafo pode ser descrita por uma matriz de adjacência $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, onde $|\mathcal{V}|$ é o número de nós no grafo. Cada elemento da matriz, \mathbf{A}_{ij} , assume o valor 1 se houver uma aresta conectando os nós i e j , e 0 caso contrário. Como essa representação pode ser custosa em termos de memória, uma alternativa eficiente é o uso de uma lista de adjacência L , onde cada entrada $L[u]$ contém os nós vizinhos de u .

A seguir, definimos as principais notações e terminologias utilizadas ao longo deste trabalho para a representação de grafos:

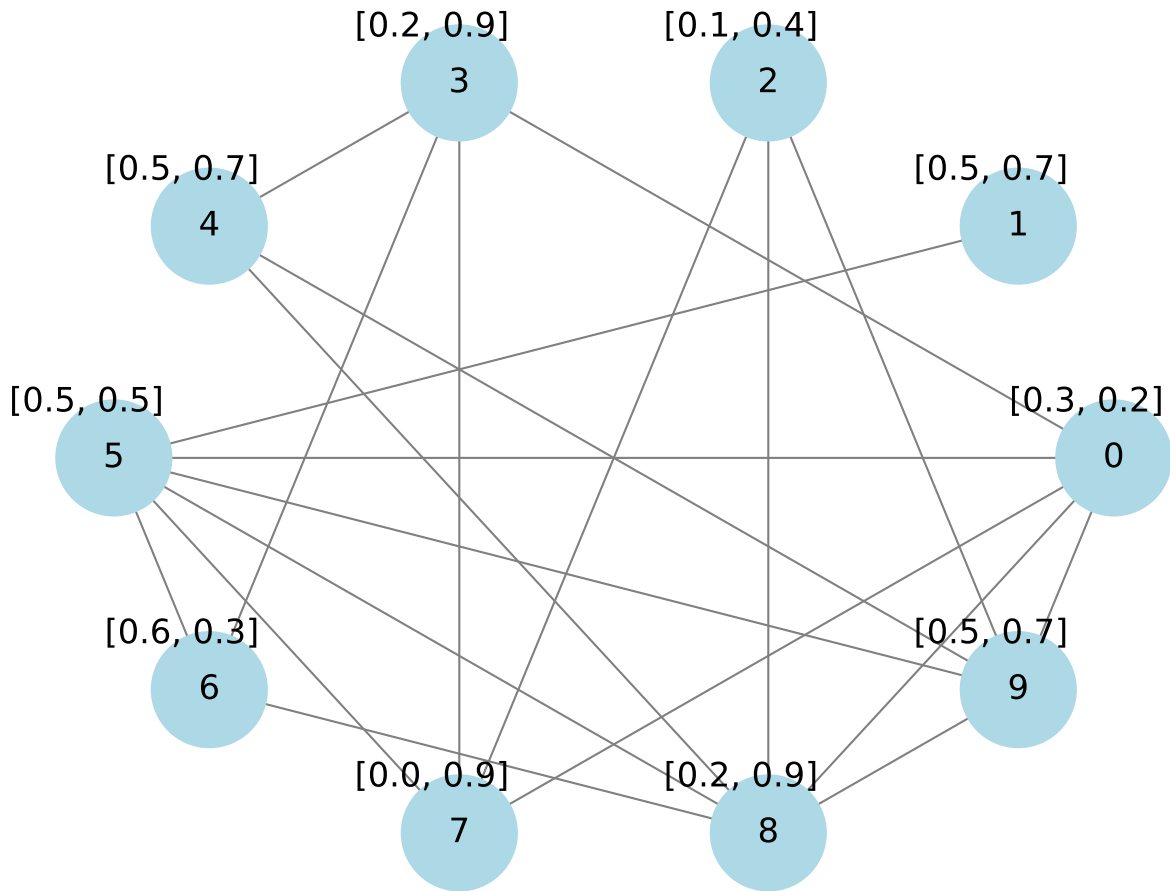
- **Nós e Arestas:** Um nó $v \in \mathcal{V}$ representa uma entidade no grafo, enquanto uma aresta $e \in \mathcal{E}$ é uma ligação entre dois nós. A aresta $e = (u, v)$ conecta os nós u e v .
- **Grau de um Nó:** O grau de um nó v , denotado por $\deg(v)$, é o número de arestas que o conectam a outros nós.
- **Grafo Direcionado:** Em um grafo direcionado $\vec{\mathcal{G}}$, as arestas possuem uma direção específica, ou seja, a aresta (u, v) é distinta da aresta (v, u) . O grau de um nó em um grafo direcionado pode ser subdividido em grau de entrada $\deg_{in}(v)$, que representa o número de arestas que chegam ao nó v , e grau de saída $\deg_{out}(v)$, que representa o número de arestas que partem de v .
- **Grafo Ponderado:** Um grafo ponderado possui arestas associadas a pesos numéricos, os quais podem representar a força, custo ou relevância de uma conexão.

- **Características do Nó:** Cada nó pode ser enriquecido com atributos categóricos ou numéricos, representados por vetores de dimensão d . Esses atributos fornecem informações adicionais que podem ser exploradas em modelos de aprendizado de máquina.
- **Caminho em um Grafo:** Um caminho $c_{(0,k)}$ é uma sequência de nós $\{v_0, v_1, v_2, \dots, v_k\}$, onde cada nó v_i está diretamente conectado ao nó v_{i+1} por uma aresta.
- **Grafo Temporal:** Quando uma série temporal é representada como um grafo, denotamos essa estrutura por G_T , onde os nós correspondem, geralmente, a pontos específicos da série original, e as arestas capturam as relações entre eles.

Grafos são amplamente utilizados na Ciência da Computação devido à sua flexibilidade e capacidade de modelar diferentes tipos de relacionamentos. Eles podem representar redes sociais (onde pessoas são nós e as conexões entre elas são arestas), mapas (onde locais são nós e estradas são arestas), moléculas químicas (com átomos como nós e ligações como arestas), e até mesmo séries temporais, como exploraremos nos capítulos seguintes.

Além disso, os grafos podem incorporar informações adicionais, como direção e peso das arestas, ou características associadas aos nós, permitindo uma modelagem mais expressiva e explicável dos dados. A Figura 2.1 apresenta um exemplo visual de um grafo, onde os nós são representados por círculos numerados e as arestas por linhas que conectam esses círculos. Essa visualização, conhecida como *Graph Drawing*, é amplamente utilizada para representar grafos em aplicações práticas.

Figura 2.1: Exemplo de um grafo não direcionado e não ponderado com 10 nós e 20 arestas. Cada nó possui um vetor com duas características numéricas acima dos nós.



Este capítulo apresentou os conceitos fundamentais necessários para a compreensão dos métodos de representação e classificação de séries temporais abordados ao longo deste trabalho. Discutimos séries temporais e técnicas de extração de características, incluindo abordagens baseadas em Shapelets, DTW e Simbolização de Bandt-Pompe, que serão essenciais para a modelagem dos dados nos próximos capítulos.

No Capítulo 3, exploramos como essas técnicas podem ser utilizadas para transformar séries temporais em representações baseadas em grafos. Abordaremos diferentes modelos de RSTG, analisando como essas representações preservam informações estruturais e temporais para permitir uma classificação mais eficiente dos dados.

Em seguida, no Capítulo 4, investigamos como as GNNs podem ser utilizadas para aprendizado de representações em grafos derivados de séries temporais. Discutiremos abordagens clássicas como *DeepWalk*, que utiliza Caminhos Aleatórios para aprendizado de *embeddings*, e métodos mais avançados baseados em troca de mensagens, como Graph Convolutional Networks (GCNs) e GraphSAGE (Graph Sample and Aggregation), que permitem uma modelagem mais expressiva das relações entre os nós dos grafos.

Capítulo 3

Séries Temporais como Grafos

No capítulo anterior, discutimos séries temporais e exploramos diferentes técnicas de extração de características para CSST. Apresentamos métodos como Shapelets, Simbolização de Bandt-Pompe e DTW, além de introduzirmos a teoria dos grafos e seu potencial para modelar dependências complexas em dados temporais. Neste capítulo, aprofundamos essa discussão ao investigar abordagens que representam séries temporais como grafos, permitindo a extração e modelagem explícita das relações espaço-temporais inerentes a esse tipo de dado.

A **Representação de Séries Temporais como Grafos (RSTG)** tem se consolidado como uma alternativa promissora aos métodos de extração de características, ao estruturar as séries de forma a facilitar a captura de padrões dinâmicos. Diferentes estratégias de modelagem têm sido propostas para RSTG, variando desde abordagens baseadas na transição entre padrões até métodos que preservam a estrutura geométrica da série [12, 71].

A construção de um Grafo Temporal ($G_T = \{V, E\}$) consiste em representar os pontos da série em nós V e definir arestas E que capturem relações entre esses pontos, extraídas através de métodos paramétricos como Shapelets e DTW, ou heurísticas matemáticas, como Distância Euclidiana e Correlação de Pearson [14, 79]. Essa abordagem compacta e interpretável tem demonstrado melhorias significativas no desempenho da CSST, proporcionando uma modelagem mais robusta das dependências temporais e maior capacidade de generalização [47, 57].

Entre as abordagens exploradas neste capítulo, apresentamos o **Similarity-Aware Time-Series Classification (SimTSC)**, que estrutura um conjunto de séries temporais como único grafo e utiliza aprendizado em grafos para classificação de nós, **Time2Graph**, que constrói grafos para cada série temporal a partir da evolução temporal de Shapelets, e o **Grafo de Transição de Padrões Ordinais (GTPO)**, que modela transições entre padrões ordinais extraídos pela **Simbolização de Bandt-Pompe**. Por fim, exploramos o **Grafo de Visibilidade Natural (GVN)**, um modelo de representação baseado em relações geométricas entre todos os pontos da série.

A modelagem dessas representações permite a aplicação de técnicas de aprendizado

em grafos, como **DeepWalk** e **Redes Neurais em Grafos (GNNs)**, para capturar padrões estruturais e temporais que podem ser alimentadas para classificadores como **xGboost** ou **Multilayer Peceptro (MLP)**. No Capítulo 4, aprofundamos a discussão sobre esses modelos de aprendizado, abordando seu funcionamento e como eles podem ser aplicados à **Classificação de Séries Temporais como Grafos (CSTG)**.

3.1 Similarity-Aware Time-Series Classification

O Similarity-Aware Time-Series Classification (SimTSC), proposto por Zha et al. [90], é um modelo paramétrico baseado em métricas de similaridade que reformula o problema de CSST como uma tarefa de classificação de nós em um grafo. Essa abordagem, conhecida como *node-as-graph*, representa cada série temporal como um nó em um grafo ponderado, onde as conexões entre os nós refletem a similaridade entre as séries.

No SimTSC, um conjunto de séries temporais $\mathcal{T} = \{T_1, T_2, \dots, T_N\}$ é modelado como um grafo $G_T = (\mathcal{V}, \mathcal{E}, \mathbb{W})$, no qual os vértices representam as séries $T_i \in \mathcal{T}$ e as arestas são definidas com base na similaridade entre as séries, calculada por meio de uma matriz de similaridade $\mathbf{D} \in \mathbb{R}^{N \times N}$, onde $N = |\mathcal{T}|$.

Cada elemento \mathbf{D}_{ij} da matriz representa a similaridade entre as séries temporais T_i e T_j , obtida por meio de distância entre séries calculada pelo DTW (Seção 2.1.2). A matriz de adjacência \mathbf{A} e os pesos das arestas \mathbb{W} são definidos conforme:

$$A_{ij} = \frac{1}{e^{\alpha \mathbf{D}_{ij}}}, \quad \forall i, j \quad (3.1)$$

onde $\alpha \in [0, \infty]$ é um hiperparâmetro que controla a influência dos vizinhos mais próximos e \mathbf{D}_{ij} denota o elemento (i, j) da matriz de similaridade \mathbf{D} . Os pesos das arestas $w_{ij} \in \mathbb{W}$ são normalizados pela equação:

$$w_{ij} = \frac{A_{ij}}{\sum_{j'} A_{ij'}}, \quad \forall i, j. \quad (3.2)$$

Para evitar um grafo excessivamente denso, os autores definem um hiperparâmetro K que seleciona os *top-K* vizinhos mais relevantes para cada nó, removendo conexões de menor similaridade em \mathbf{A} e resultando em um grafo esparsa G_T .

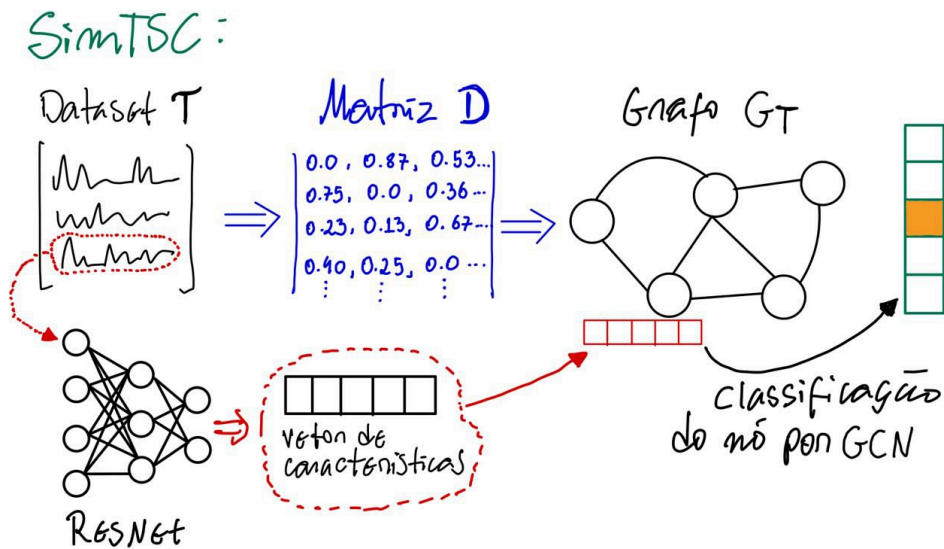
Após a modelagem do grafo, uma DNN auxiliar é treinada para extrair características de cada série temporal, que serão posteriormente incorporadas a G_T como atributos dos nós. O SimTSC utiliza uma versão adaptada da ResNet [43] para dados unidi-

mensionais, embora a metodologia possa ser estendida para outras arquiteturas de redes neurais.

O grafo G_T , enriquecido com as características extraídas, é processado por sucessivas camadas GCN (Seção 4.2.1), responsáveis por propagar e atualizar as informações dos nós. O treinamento ocorre em conjunto com a ResNet, empregando uma estratégia de *negative sampling* [85] para otimizar a aprendizagem das representações dos nós. Os vetores ocultos resultantes são então alimentados a uma função *softmax* para classificar os nós do grafo e, por conseguinte, as séries temporais.

A Figura 3.1 ilustra a arquitetura do SimTSC, adaptada de Zha et al. [90]. Nota-se que a definição da matriz de similaridade \mathbf{D} a partir do conjunto de dados \mathcal{T} e o treinamento da ResNet ocorrem sequencialmente, permitindo a extração das características iniciais das séries temporais. Após a passagem de mensagens no grafo, os nós são classificados em uma das classes de \mathcal{T} .

Figura 3.1: Ilustração da arquitetura SimTSC. Adaptado de [90].



3.2 Time2Graph e Time2Graph Plus

O modelo Time2Graph, proposto por Cheng et al. [15], apresenta uma abordagem para CSST baseada na extração e análise da evolução temporal de Shapelets (Seção 2.1.2). Para isso, a série T é representada como um grafo direcionado e ponderado $\vec{G}_T = (\mathcal{V}, \mathcal{E}, \mathbb{W})$,

onde \mathcal{V} representa um conjunto de K Shapelets, \mathcal{E} define as conexões entre eles com base na sequência de ocorrência na série, e os pesos \mathbb{W} indicam a probabilidade de transição entre dois Shapelets consecutivos, calculada pela Equação 3.5.

Após a modelagem da série como grafo, $\vec{\mathcal{G}}_T$ é alimentado a um modelo de aprendizado em grafos denominado DeepWalk (Seção 4.1), que identifica padrões anômalos a partir de caminhadas aleatórias em $\vec{\mathcal{G}}_T$ e gera um vetor de características que, ao ser alimentado a um classificador xGBoost [13], permite a classificação da série temporal.

Os autores argumentam que abordagens tradicionais de Aprendizado tratam os Shapelets como padrões estáticos, o que não reflete a natureza dinâmica dos dados reais. Em muitas aplicações, a interpretação de um Shapelet depende do contexto temporal. Por exemplo, ao analisar padrões de consumo de energia elétrica ao longo do ano, um baixo consumo na primavera pode ser normal, enquanto no verão pode indicar uma potencial fraude [15].

Para corretamente capturar essa variação temporal, Cheng et al. [15] introduzem o conceito de **Shapelets Cômicos no Tempo** (*Time-aware Shapelets*), incorporando fatores de atenção local e global, \mathbf{w} e \mathbf{u} , na função de distância $d = DTW(S, T)$. Esses fatores quantificam a influência temporal dos Shapelets S em diferentes momentos e escalas na série T , permitindo sua extração via aprendizado de Shapelets [37]. Assim, a Equação 2.1 é adaptada para:

$$\hat{D}(\mathbf{v}, \mathbf{t} \mid \mathbf{w}, \mathbf{u}) = \min_{1 \leq k \leq m} \mathbf{u}_k \cdot \hat{d}(\mathbf{v}, \mathbf{s}_k \mid \mathbf{w}), \quad (3.3)$$

onde a distância $\hat{d}(v, s \mid w)$ é definida como:

$$\hat{d}(\mathbf{v}, \mathbf{s}_k \mid \mathbf{w}) = \left(\sum_{i=1}^p w_{a_1^*(i)} \cdot \|v_{a_1^*(i)} - s_{a_2^*(i)}\|^2 \right)^{\frac{1}{2}}, \quad (3.4)$$

onde a^* representa o melhor alinhamento do DTW, p é o tamanho da sequência de alinhamento, \mathbf{w} denota a importância do i -ésimo elemento de um Shapelet e \mathbf{u} mede a influência cruzada entre segmentos da série.

A probabilidade de transição entre os Shapelets na série, que define os pesos das arestas no grafo \mathbb{W} , é calculada por:

$$w_{ij} = \frac{\max(\hat{d}(v_{i,*}, s_i)) - \hat{d}(v_{i,j}, s_i)}{\max(\hat{d}(v_{i,*}, s_i)) - \min(\hat{d}(v_{i,*}, s_i))}, \quad (3.5)$$

onde $\hat{d}(v_{i,*}, s_i) \leq \delta$. O hiperparâmetro δ define o limiar de similaridade entre os Shapelets S e a série T , devendo ser cuidadosamente ajustado durante o treinamento.

Os autores utilizam a distância euclidiana quadrada no cálculo do DTW, uma escolha estratégica que amplifica diferenças entre subsequências e melhora a sensibilidade na detecção de variações de magnitude. Além disso, essa métrica torna a função de distância

diferenciável, permitindo a otimização via gradiente descendente durante o Aprendizado dos Shapelets.

Time2Graph Plus

Apesar das vantagens do Time2Graph, algumas limitações foram identificadas pelos próprios autores, levando ao desenvolvimento do Time2Graph+ [16], uma versão aprimorada que substitui o DeepWalk por Graph Attention Network (GAT) (Seção 4.2.1), modificando a estratégia de modelagem do grafo.

Nesta nova abordagem, os Shapelets Cômicos no Tempo são estruturados em um grafo parametrizado não ponderado $\vec{\mathcal{G}}_T = (\mathcal{V}, \mathcal{E}, \mathcal{H} \mid \Theta)$, onde:

- \mathcal{V} representa os K Shapelets,
- \mathcal{E} define as probabilidades de transição entre eles,
- $\Theta = \{\mathbf{W}, a\}$ representa os parâmetros do mecanismo de auto-atenção, implementado por uma FNN de uma camada,
- $\mathcal{H} = \{h_1, \dots, h_{|V|}\} \in \mathbb{R}^{|F|}$ contém as características dos nós, concatenando os valores de distância entre o Shapelet v_i e a série temporal T .

A principal inovação do Time2Graph+ em relação à sua versão original está na substituição do DeepWalk por uma Graph Attention Network (GAT) para o aprendizado dos pesos $w_{ij} \in \mathbb{W}$ das arestas. Essa mudança elimina a necessidade de ajuste manual do hiperparâmetro δ , tornando o modelo mais adaptável a diferentes séries temporais e permitindo a captura de padrões mais complexos e sutis na evolução dos Shapelets.

No Time2Graph+, os pesos w_{ij} são aprendidos por meio de um mecanismo de auto-atenção definido pela função escalar $f(i, j \mid \mathcal{H}, \mathcal{E}, \Theta)$, onde os coeficientes de atenção α_{ij} (Eq. (4.5)) ponderam a importância relativa das transições entre Shapelets, considerando tanto suas características individuais quanto sua posição na estrutura do grafo. Esse mecanismo permite que o modelo diferencie automaticamente quais transições são mais relevantes para a representação da série temporal.

Além disso, ao introduzir o hiperparâmetro de percentil p , utilizado para podar as arestas de menor peso e reduzir a densidade do grafo, é possível evitar que transições fracas sejam amplificadas indevidamente, garantindo que apenas as conexões mais significativas sejam preservadas, o que melhora a estabilidade da representação.

Embora a etapa final de classificação ainda seja realizada por meio de um xGBoost, o uso da GAT no lugar do DeepWalk permite capturar com maior precisão as interações dinâmicas entre os Shapelets. Isso torna o modelo mais robusto, pois permite a extração de representações mais expressivas da evolução temporal da série, reduzindo a dependência de heurísticas pré-definidas e aumentando sua capacidade de generalização.

No Capítulo 5, apresentamos uma análise comparativa entre as versões Time2Graph e Time2Graph+, utilizando as implementações originais dos autores. Além disso, investigamos a performance dessa estratégia de RSTG com o uso de uma arquitetura diferente de GNN, o GraphSAGE (Seção 4.2.1) integrado a um classificador MLP, para avaliar a influência da arquitetura de aprendizado no desempenho da classificação.

3.3 Grafo de Transição de Padrões Ordinais

O Grafo de Transição de Padrões Ordinais (GTPO) é uma técnica não paramétrica para representar a dinâmica de séries temporais como grafos, a partir da Simbolização de Bandt-Pompe (Seção 2.1.3). Esse método transforma uma série temporal $T = \{t_1, t_2, \dots, t_m\}$ em uma sequência $\Pi = \{\tilde{\pi}_i^D\}_0^{D!}$ de Padrões Ordinais (PO), que são conectados entre si de acordo com suas transições na série, permitindo a captura de características dinâmicas da evolução temporal dos dados [4, 54, 64, 76].

Nesse *modelo*, T é representada como um grafo direcionado e ponderado $\vec{\mathcal{G}}_T = (\mathcal{V}, \mathcal{E}, \mathbb{W})$, onde:

- \mathcal{V} corresponde ao conjunto de diferentes padrões ordinais $\tilde{\pi}_i^D$,
- \mathcal{E} representa as conexões direcionadas entre dois padrões consecutivos na sequência Π , de modo que cada transição $\tilde{\pi}_i^D \rightarrow \tilde{\pi}_j^D$ origina uma aresta direcionada de v_i para v_j em $\vec{\mathcal{G}}_T$,
- \mathbb{W} armazena os pesos associados às arestas, indicando a probabilidade de transição entre padrões.

Os pesos $w_{ij} \in \mathbb{W}$ são definidos como a frequência relativa de cada transição, conforme:

$$w_{ij} = \frac{|\Pi_{\tilde{\pi}_i^D, \tilde{\pi}_j^D}|}{m - (D - 1)\tau - 1}, \quad (3.6)$$

onde $|\Pi_{\tilde{\pi}_i^D, \tilde{\pi}_j^D}|$ representa o número de transições do padrão $\tilde{\pi}_i^D$ para o padrão $\tilde{\pi}_j^D$, e $m - (D - 1)\tau - 1$ normaliza o total de transições possíveis. Esse cálculo permite capturar a dependência temporal entre os padrões ordinais e inferir relações subjacentes na estrutura da série.

Cardoso-Pereira et al. [8] demonstraram a aplicação do GTPO na representação de trajetórias de GPS de usuários, modelando latitude, longitude e altitude como padrões ordinais e classificando os veículos em diferentes modais de transporte. A análise foi realizada a partir da Probabilidade de Auto-transição, calculada como a soma dos pesos das arestas de autoloop em $\vec{\mathcal{G}}_T$, ou seja, das arestas $w(v_{\pi_i}, v_{\pi_i})$ que representam transições consecutivas do mesmo padrão ordinal.

Apesar de seu potencial para modelar dinâmicas temporais, o GTPO enfrenta desafios relacionados à sua dimensionalidade, pois a densidade do grafo cresce à medida que D aumenta, devido à explosão combinatória do espaço de estados na Simbolização de Bandt-Pompe. Para mitigar esse problema, o intervalo de valores para D e τ costuma ser definido empiricamente [3, 70], o que pode impactar a expressividade da representação. Trabalhos mais recentes, como os de McCullough et al. [54] e Pessa and Ribeiro [64], propõem ajustes baseados na complexidade e na entropia da série para uma definição mais informada desses parâmetros. Neste estudo, optamos por manter $D = 5$ e $\tau = 1$ como configuração padrão, com base em testes preliminares que indicaram um bom equilíbrio entre granularidade e viabilidade computacional.

Até o momento, não há estudos na literatura que explorem o uso de GNNs para aprendizado em GTPOs. Uma possível explicação para essa lacuna é que, na modelagem tradicional, os GTPOs não associam atributos diretamente aos nós do grafo, limitando a informação disponível para o aprendizado. Embora essa abordagem capture a conectividade entre padrões ordinais, ela não incorpora informações temporais adicionais que poderiam enriquecer a representação da série.

Para viabilizar o uso de GNNs em GTPOs e explorar seu potencial para classificação de séries temporais, propomos duas estratégias para injetar informações nos nós:

- **Incorporação de métricas estruturais:** Inspirado por Oliveira et al. [62], adicionamos atributos como o grau de entrada $\text{deg}_{in}(v)$ e valores de centralidade, como PageRank, enriquecendo a representação do grafo com informações sobre a relevância de cada padrão ordinal na dinâmica da série temporal;
- **Codificação vetorial dos padrões ordinais:** Cada padrão ordinal é representado como um vetor de *one-hot encoding*, permitindo que a rede aprenda relações entre diferentes padrões e amplie a expressividade do modelo.

Com base nessas estratégias, desenvolvemos os modelos **GTPO-2feat** e **GTPO-encoded**, que integram a representação de GTPOs a camadas de GraphSAGE e MLP

para a classificação de séries temporais. No Capítulo 5, detalhamos a implementação dessas abordagens e avaliamos seu desempenho em comparação com outras representações de séries como grafos. Esses modelos representam um avanço inédito na aplicação de GNNs para aprendizado em GTPOs, abrindo novas possibilidades para a análise de padrões temporais em dados simbólicos.

3.4 Grafo de Visibilidade

Proposto por Lacasa et al. [51], o Grafo de Visibilidade Natural (GVN) é um método de RSTG que preserva a estrutura e a dinâmica da série original. O GVN é um modelo não paramétrico que define um grafo não ponderado $G_T = (\mathcal{V}, \mathcal{E})$ no qual os vértices representam todos os valores da série $T = \{t_0, t_1, \dots, t_m\}$, e as arestas \mathcal{E} são estabelecidas de acordo com um critério de visibilidade entre esses pontos. A estrutura do grafo resultante é altamente influenciada pela natureza da série original, preservando propriedades como periodicidade, auto-similaridade, fractalidade e aleatoriedade [51].

A relação de visibilidade entre dois pontos $A(t_a, y_a)$ e $B(t_b, y_b)$ é definida pelo critério de visibilidade, que estabelece que um ponto intermediário $C(t_c, y_c)$ não pode bloquear a linha imaginária que conecta A e B . Esse critério é formalizado pela Equação 3.7:

$$y_c < y_b + (y_a - y_b) \frac{t_b - t_c}{t_b - t_a}. \quad (3.7)$$

Para construir o GVN a partir de uma série T , seus valores são representados em um histograma $H(m, n)$ de distribuição ao longo do tempo. Em seguida, para cada par de pontos H_A e $H_B \in H(m, n)$, a visibilidade é verificada utilizando a Equação 3.7. Se o critério for satisfeito, uma aresta é adicionada entre os nós correspondentes a H_A e H_B .

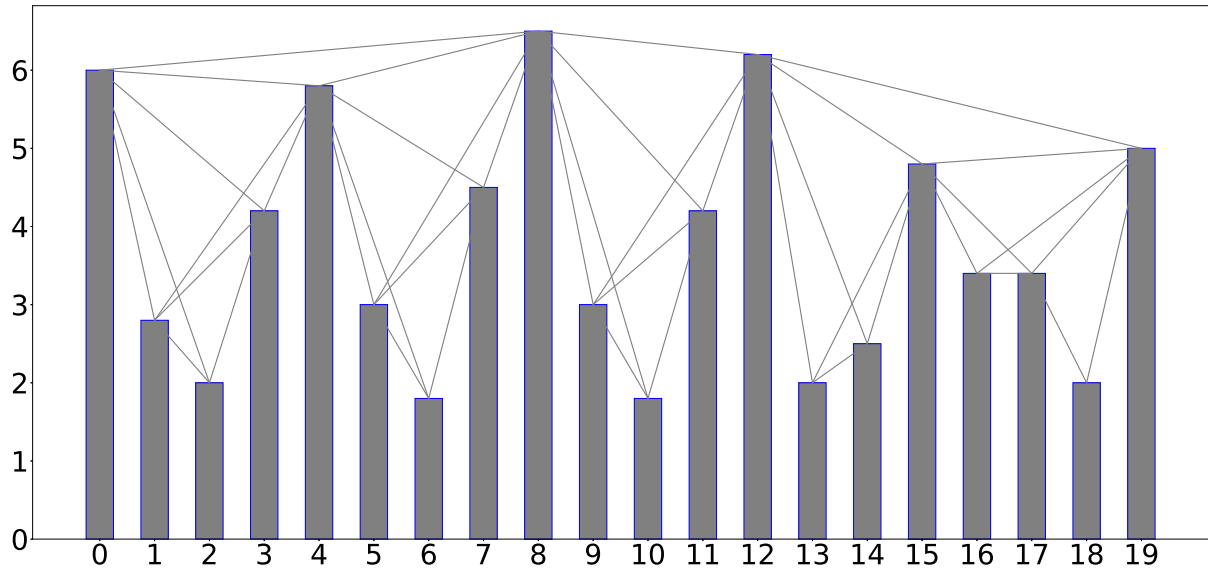
A Figura 3.2 ilustra esse processo. Na Figura 3.2 (a), observa-se a projeção da série como um histograma, onde as conexões de visibilidade são determinadas pelo critério da Equação 3.7. O grafo resultante na Figura 3.2 (b) mantém a estrutura espaço-temporal da série original, apresentando características como ser **não-direcionado**, **conectado** e **invariante à escala**, ou seja, sua topologia permanece consistente independentemente da escala dos valores da série.

Figura 3.2: Histograma de uma série temporal sintética de 20 valores com a projeção de visibilidade determinada pelo critério da equação 3.7.

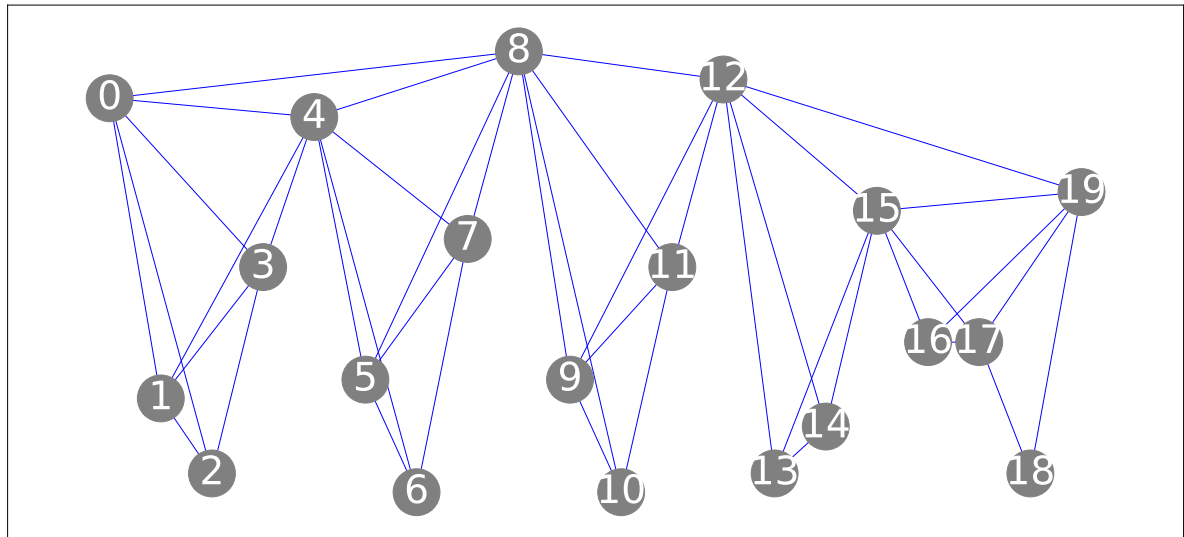
A topologia do GVN resultante preserva a estrutura espaço-temporal de t , sendo **não-direcionado, conectado e invariante à escala**.

Valores da série temporal: {6.0, 2.8, 2.0, 4.2, 5.8, 3.0, 1.8, 4.5, 6.5, 3.0, 1.8, 4.2, 6.2, 2.0, 2.5, 4.8, 3.4, 3.4, 2.0, 5.0}

a)



b)



Modelos para CSST utilizando GVN incluem o trabalho de Oliveira et al. [62], que explorou a representação de exames de eletrocardiograma (ECG) como GVN direcionados para a classificação de arritmias. Os autores compararam o desempenho de diferentes arquiteturas de GNNs, incluindo camadas GCN e GraphSAGE. Como o GVN não incorpora diretamente informação temporal, os autores definiram como atributos dos nós métricas como grau de entrada $\text{deg}_{in}(v)$ e centralidade de PageRank, que capturam a conectividade e a importância relativa dos valores da série temporal. Os resultados

indicaram que arquiteturas baseadas em GraphSAGE superaram as baseadas em GCN, apontando caminhos promissores para o uso do GVN em tarefas de classificação.

Coelho et al. [20] investigaram o uso de GVN para representar séries multivariadas de exames de ECG contendo 8 canais anotados com 6 tipos de anomalias. O objetivo do estudo foi explorar se a estrutura do grafo poderia auxiliar na seleção dos canais mais relevantes para a classificação das anomalias. Para isso, os autores aplicaram a métrica de diversidade em grafos proposta por Carpi et al. [9] sobre as representações GVN de cada canal e utilizaram esse critério para ranquear os três canais que mais contribuíam com a diversidade global. O desempenho do classificador ROCKET [24] foi então avaliado utilizando essa seleção, comparando-a com seleções aleatórias de canais.

Os resultados não evidenciaram uma correlação clara entre a escolha baseada na diversidade e a seleção aleatória, sugerindo que a métrica de diversidade, isoladamente, pode não ser um critério determinante para a seleção ótima de canais nesse contexto. No entanto, esse estudo abre caminho para investigações futuras sobre a classificação de ECGs representados como grafos de visibilidade.

Existem diversas variações do Grafo de Visibilidade, incluindo o **Grafo de Visibilidade Horizontal (GVH)** [53], que modifica o critério da Equação 3.7 para considerar apenas conexões horizontais entre pontos, e abordagens mais recentes, como a de Xuan et al. [84], que combina o GVN com camadas de convolução para gerar grafos menos densos a partir de sinais de rádio. Esse modelo foi utilizado em conjunto com uma GNN baseada em DiffPool [88] para melhorar a classificação desses sinais.

Neste trabalho, optamos por focar no Grafo de Visibilidade Natural clássico, devido à sua simplicidade conceitual, consolidação na literatura e disponibilidade de implementações confiáveis. Essa escolha permite aplicar metodologias estabelecidas e de fácil reprodução, facilitando comparações diretas com outros modelos.

Neste capítulo, exploramos diferentes abordagens para RSTG, analisando modelos paramétricos e não paramétricos que estruturam as séries em grafos com base em diferentes estratégias de extração de características e modelagem. Discutimos métodos que capturam a evolução temporal de padrões discriminativos, como o Time2Graph, abordagens baseadas na transição entre padrões ordinais, como o GTPO, e técnicas que preservam a estrutura espaço-temporal da série, como o GVN. Além disso, apresentamos o SimTSC, que modela as séries temporais como nós em um grafo global, permitindo a classificação via aprendizado em grafos.

No próximo capítulo, aprofundamos a discussão ao apresentar os algoritmos de aprendizado em grafos utilizados ao longo deste capítulo. Exploraremos técnicas como DeepWalk, GCN, GraphSAGE e GAT, analisando seu funcionamento e aplicabilidade para a classificação de séries temporais representadas como grafos.

Capítulo 4

Aprendizado de Representação em Grafos

No capítulo anterior, exploramos diferentes abordagens para RSTG, destacando como elas permitem capturar de maneira estruturada as relações espaço-temporais dos dados. No entanto, a efetividade dessas representações depende diretamente da capacidade dos modelos de aprendizado em grafos em extrair padrões significativos e classificá-los corretamente. Neste capítulo, aprofundamos a discussão ao apresentar técnicas de **Aprendizado de Representação em Grafos**, fundamentais para a tarefa de CSTG.

Apesar de amplamente utilizadas em diversos domínios, as Redes Neurais Profundas (DNNs) enfrentam desafios ao lidar com dados não euclidianos, como grafos. Além da já mencionada maldição da dimensionalidade, esses modelos muitas vezes carecem de interpretabilidade e enfrentam dificuldades para capturar relações estruturais explícitas entre elementos interdependentes [47]. Isso ocorre porque essas abordagens em geral representam os dados de forma tabular, não preservando adequadamente as relações espaciais e temporais no *Espaço Latente*, o que compromete a qualidade da representação e o desempenho preditivo.

Em contraste, métodos baseados em aprendizado em grafos oferecem uma alternativa mais eficaz para modelar esses tipos de relações [40, 71]. Como discutido na Seção 2.2, os grafos não apenas armazenam os valores das séries temporais, mas também estruturam suas conexões, permitindo a modelagem explícita das interdependências entre os pontos da série. Essa estruturação viabiliza o uso de técnicas avançadas de aprendizado, como DeepWalk e GNNs, que exploram padrões estruturais e contextuais para realizar classificações mais robustas.

Este capítulo apresenta as principais técnicas de Aprendizado de Representação em Grafos, explorando a geração de *embeddings* para nós e grafos completos, que posteriormente podem ser utilizados em modelos de aprendizado supervisionado para classificação. Primeiramente, discutiremos o DeepWalk, um modelo pioneiro que adapta conceitos do Processamento de Linguagem Natural (PLN) para aprendizado em grafos por meio de Caminhos Aleatórios. Em seguida, introduzimos as GNNs, que utilizam Troca de Mensagens

e otimização por funções de perda, como Entropia Cruzada, para aprender representações estruturais profundas e melhorar a tarefa de classificação.

4.1 DeepWalk

O DeepWalk, proposto por Perozzi et al. [63], adapta conceitos de PLN para modelar dados estruturados como grafos. Inspirado em técnicas de *word embeddings*, o modelo trata os nós de um grafo como palavras e as sequências de nós conectados como sentenças, permitindo a aprendizagem de representações latentes que capturam a estrutura e as relações entre os elementos do grafo.

Para alcançar essa representação, o modelo utiliza **caminhadas aleatórias** para explorar a conectividade do grafo, gerando sequências de nós que podem ser interpretadas como sentenças em um corpus textual. Essas sequências são então processadas pelo modelo *Skip-Gram* [17], que aprende a prever os vizinhos de um nó-alvo, maximizando a seguinte função de probabilidade:

$$\max_f \sum_{v_i \in V} \sum_{v_j \in C(v_i)} \log Pr(v_j | f(v_i)), \quad (4.1)$$

onde $C(v_i)$ representa o contexto de v_i , obtido a partir dos nós visitados durante as caminhadas aleatórias, e $Pr(v_j | f(v_i))$ é a probabilidade de prever o nó v_j a partir do nó v_i . O objetivo é aprender uma função de mapeamento $f : V \rightarrow \mathbb{R}^d$ que projete os nós em um espaço vetorial de d dimensões, preservando as relações estruturais do grafo.

Uma das principais vantagens do DeepWalk é sua capacidade de explorar tanto as conexões locais quanto as relações globais dentro do grafo, permitindo a captura de padrões estruturais ocultos. Ao modelar hierarquias e correlações entre os elementos do grafo, o modelo aprende *embeddings* que refletem tanto a estrutura local quanto a conectividade global do grafo, tornando-se uma ferramenta útil para tarefas como classificação de nós, por exemplo.

No entanto, esse modelo apresenta algumas limitações. Como se trata de um método *transdutivo*, ele gera representações apenas para os nós presentes no treinamento, sem capacidade de generalização para novos nós em grafos desconhecidos. Além disso, ao não incorporar características dos nós, o modelo depende exclusivamente da estrutura do grafo, o que pode levar à perda de informações relevantes sobre os dados.

Para superar essas limitações, surgiram as Redes Neurais em Grafos (GNNs), que combinam aprendizado em grafos com técnicas de aprendizado profundo, permitindo a

generalização para novos nós e a incorporação de atributos dos nós e arestas. Esses modelos utilizam o framework de Troca de Mensagens, onde cada nó compartilha e agrega informações com seus vizinhos de forma iterativa, resultando em representações mais expressivas.

Na próxima seção, exploraremos em detalhes as GNNs, analisando suas arquiteturas clássicas e como a troca de mensagens permite que esses modelos aprendam representações de nós e grafos de maneira eficiente, abrindo novas possibilidades para a descoberta de padrões ocultos em grafos.

4.2 Redes Neurais em Grafos

Os modelos de aprendizado profundo tradicionais, como redes neurais convolucionais (CNNs) e redes recorrentes (RNNs), demonstraram grande eficácia na modelagem de dados estruturados em grades regulares, como imagens e sequências. No entanto, sua aplicabilidade a dados não euclidianos, como grafos, é limitada, pois esses modelos não capturam diretamente a estrutura relacional entre os elementos [40]. A fim de superar essa limitação, vem ganhando espaço técnicas de Geometric Deep Learning (GDL), que buscam generalizar as técnicas de aprendizado profundo para domínios não estruturados, como grafos, utilizando princípios geométricos [6, 7].

As Redes Neurais em Grafos (GNNs) são uma das principais aplicações de GDL, permitindo que modelos de aprendizado de máquina incorporem a topologia inerente a estruturas de grafos. Diferentemente de métodos baseados em caminhadas aleatórias ou extração de padrões estruturais, as GNNs aplicam aprendizado supervisionado diretamente sobre grafos, combinando informações locais e globais por meio de um processo iterativo de agregação de vizinhos. Essa abordagem permite capturar relações complexas entre os nós, preservando informações tanto topológicas quanto atributivas [6, 38].

O modelo de uma GNN pode ser descrito como um sistema codificador/decodificador, onde as informações contidas em um grafo $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ são projetadas em um espaço latente através da geração de *embeddings* de nós. Esses embeddings representam não apenas os atributos individuais de cada nó, mas também as relações estruturais entre eles e seus vizinhos, podendo ser utilizados como entrada para modelos de classificação de nós ou grafos.

O processo de aprendizado em GNNs ocorre por meio de uma arquitetura profunda com K camadas, que aplica iterativamente o mecanismo de **troca de mensagens** (Equation 4.2), conforme proposto por Hamilton [38]. Nesse framework, cada nó $u \in \mathcal{V}$

atualiza sua representação latente agregando informações de seus vizinhos ao longo das arestas $e \in \mathcal{E}$, gerando embeddings ocultos $\mathbf{h}_u^{(K)}$ que incorporam tanto as conexões locais quanto a estrutura global do grafo:

$$\begin{aligned} \mathbf{h}_u^{(k+1)} &= \text{UPDATE}^{(k)} \left(\mathbf{h}_u^{(k)}, \text{AGGREGATE}^{(k)} \left(\{\mathbf{h}_v^{(k)} \mid v \in \mathcal{N}(u)\} \right) \right) \\ &= \text{UPDATE}^{(k)} \left(\mathbf{h}_u^{(k)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)} \right), \end{aligned} \quad (4.2)$$

onde *UPDATE* e *AGGREGATE* são funções diferenciáveis aprendidas por meio de descida de gradiente, $\mathcal{N}(u)$ representa a vizinhança do nó u , e $\mathbf{m}_{\mathcal{N}(u)}$ é a mensagem agregada a partir de seus vizinhos.

No início do processo, para $k = 0$, os embeddings iniciais dos nós são definidos como $\mathbf{h}_u^{(0)} = \mathbf{x}_u$, onde \mathbf{x}_u representa os atributos de entrada do nó u . Em cada iteração $k \in \{0, 1, \dots, K-1\}$, a função *AGGREGATE* recebe os embeddings dos vizinhos $\mathcal{N}(u)$ e gera uma mensagem $\mathbf{m}_{\mathcal{N}(u)}^{(k)}$, que é então combinada pela função *UPDATE* com o embedding do próprio nó u . Após K iterações, os embeddings finais dos nós são obtidos como $\mathbf{z}_u = \mathbf{h}_u^{(K)}$, capturando a estrutura local e global do grafo.

Para a predição de rótulos de nós, uma função *softmax* sobre os embeddings \mathbf{z}_u é suficiente para gerar a probabilidade de classificação de cada nó. Já para a classificação de grafos, é necessário agregar as informações de todos os nós em uma única representação, o que é feito por meio de uma camada adicional de *readout*, que normalmente calcula a média dos embeddings de todos os nós do grafo, gerando um vetor \mathbf{Z}_G que representa toda a estrutura da rede.

Embora existam métodos avançados, como os propostos por Ying et al. [88] e Zhang et al. [91], que aplicam *readouts* hierárquicos e técnicas de *pooling* para capturar informações estruturais em diferentes níveis do grafo, a maioria das abordagens na literatura ainda adota agregações simples, como a média dos *embeddings* dos nós. Essa escolha se deve principalmente à sua eficiência computacional e boa capacidade de generalização, conforme demonstrado por Xu et al. [82], que analisaram o poder expressivo de diferentes funções de agregação, e por Hamilton et al. [39], que mostraram que agregações simples, como soma ou média, são eficazes na preservação da estrutura do grafo e na extração de características relevantes para tarefas de classificação.

4.2.1 Arquiteturas de GNN

As Redes Neurais em Grafos podem ser implementadas a partir do framework geral de troca de mensagens (Equation 4.2). No entanto, a escolha das funções *UPDATE* e *AGGREGATE* impacta diretamente na expressividade e eficiência do modelo. Diferentes arquiteturas de GNN foram propostas para otimizar esse processo e adaptar o aprendizado de representação em grafos a diferentes domínios e tipos de conectividade.

Nesta seção, exploramos três arquiteturas amplamente utilizadas: a **Graph Convolutional Network (GCN)**, que introduziu a convolução em grafos inspirada nas redes convolucionais tradicionais; o **GraphSAGE**, que permite um aprendizado mais escalável e generalizável; e a **Graph Attention Network (GAT)**, que aprimora a troca de mensagens com mecanismos de atenção para capturar a importância relativa dos vizinhos de cada nó.

4.2.1.1 Modelo Base de GNN

O modelo mais básico de GNN, conforme introduzido por Scarselli et al. [73] e Merkwirth and Lengauer [55], é definido pela equação:

$$\mathbf{h}_u^{(k+1)} = \sigma \left(\mathbf{W}_{self}^{(k)} \mathbf{h}_u^{(k)} + \mathbf{W}_{neigh}^{(k)} \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{(k)} \right), \quad (4.3)$$

onde $\mathbf{W}_{self}^{(k)}$ e $\mathbf{W}_{neigh}^{(k)} \in \mathbb{R}^{d^{(k+1)} \times d^{(k)}}$ são matrizes de pesos treináveis, σ denota uma função de ativação não linear, como a ReLU, e $\mathcal{N}(u)$ representa a vizinhança do nó u .

Neste caso, a função *AGGREGATE* corresponde à soma dos *embeddings* dos vizinhos, enquanto *UPDATE* aplica uma transformação linear seguida de uma função de ativação não linear. A cada iteração k , os *embeddings* dos nós são atualizados com base nas informações de seus vizinhos, permitindo ao modelo capturar a estrutura do grafo e aprender representações úteis para a tarefa desejada.

4.2.1.2 Graph Convolutional Network (GCN)

A *Graph Convolutional Network (GCN)*, proposta por Kipf and Welling [49], é uma das arquiteturas mais conhecidas de GNNs. Ela adapta o conceito de convolução, originalmente presente em modelos para dados euclidianos, como CNNs, para o domínio dos grafos.

A operação de convolução em grafos é realizada na matriz de adjacência \mathbf{A} e na matriz de *embeddings* dos nós $\mathbf{H}^{(k)}$, gerando uma nova matriz de *embeddings* $\mathbf{H}^{(k+1)}$:

$$\mathbf{H}^{(k+1)} = \sigma \left(\tilde{\mathbf{D}}^{-1/2} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-1/2} \mathbf{H}^{(k)} \mathbf{W}^{(k)} \right), \quad (4.4)$$

onde $\mathbf{H}^{(k)}$ é a matriz de *embeddings* dos nós na camada k , $\mathbf{W}^{(k)}$ é a matriz de pesos treinável, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ é a matriz de adjacência com auto-loops, $\tilde{\mathbf{D}}$ é a matriz diagonal de graus normalizada e σ é uma função de ativação.

A convolução de grafos define a função *AGGREGATE* como uma média ponderada dos *embeddings* dos vizinhos, enquanto *UPDATE* aplica uma transformação linear seguida da função de ativação σ . A adição de auto-loops permite que cada nó incorpore suas próprias informações durante a atualização, fortalecendo a troca de mensagens.

4.2.1.3 GraphSAGE

O *GraphSAGE – Sample and AGgregate*, introduzido por Hamilton et al. [39], aprimora a GCN ao introduzir um aprendizado indutivo e escalável para a geração de *embeddings* de nós. Em vez de considerar todos os vizinhos durante a agregação, o GraphSAGE utiliza uma estratégia de amostragem de k vizinhos, acelerando a troca de mensagens e permitindo que o modelo escale para grafos grandes.

A função *AGGREGATE* no GraphSAGE é flexível e pode ser implementada como média (*mean pooling*), máxima (*max pooling*) ou mesmo uma combinação de features aprendidas por uma rede Long-Short Term Memory (LSTM). A escolha da função de agregação depende da tarefa e da estrutura do grafo, permitindo que o modelo capture diferentes padrões de conectividade. Além disso, o modelo pode incorporar pesos de grafos ponderados diretamente na agregação, enriquecendo a representação dos nós. Essa adaptabilidade torna o GraphSAGE uma abordagem eficiente para generalizar o aprendizado para nós não vistos durante o treinamento.

4.2.1.4 Graph Attention Network (GAT)

A *Graph Attention Network (GAT)*, proposta por Veličković et al. [80], aprimora a troca de mensagens ao introduzir um mecanismo de atenção, que permite que cada nó atribua pesos diferentes a seus vizinhos com base na relevância de suas conexões.

Diferentemente das GCNs e GraphSAGE, que tratam todos os vizinhos de forma equivalente, o GAT aprende coeficientes de atenção $\alpha_{uv}^{(k)}$ para ponderar a importância relativa de cada vizinho durante a atualização dos nós:

$$\mathbf{h}_u^{(k+1)} = \sigma \left(\sum_{v \in \mathcal{N}(u)} \alpha_{uv}^{(k)} \mathbf{W}^{(k)} \mathbf{h}_v^{(k)} \right), \quad (4.5)$$

onde $\alpha_{uv}^{(k)}$ é um coeficiente de atenção treinável que quantifica a influência do nó v na atualização do nó u . Esses coeficientes são aprendidos a partir das características dos nós e da estrutura do grafo, tornando o modelo mais flexível e eficiente em tarefas que envolvem estruturas complexas.

O GAT se destaca por selecionar automaticamente os vizinhos mais relevantes para cada nó, permitindo que o modelo capture padrões mais refinados e específicos. Esse mecanismo é especialmente útil em aplicações onde a conectividade entre os nós não é uniforme ou onde algumas conexões são mais importantes do que outras.

Arquiteturas baseadas em GAT têm sido utilizadas para modelar séries temporais como grafos, como demonstrado no *Time2Graph+* (Seção 3.2), onde os coeficientes de atenção são empregados para capturar padrões de evolução em séries temporais representadas como grafos.

Neste capítulo, exploramos diferentes arquiteturas de Redes Neurais em Grafos (GNNs), destacando como essas técnicas possibilitam a extração de padrões estruturais e a modelagem de interações complexas em dados não euclidianos. Apesar das vantagens dessas redes no aprendizado de representações para grafos, elas também apresentam desafios que devem ser considerados ao aplicá-las em cenários reais.

Um dos principais desafios das GNNs está na complexidade computacional, especialmente ao lidar com grafos de grande escala, densos ou cujos nós possuem muitas características. O custo da troca de mensagens entre nós tende a crescer conforme o número de vizinhos aumenta, tornando o treinamento e a inferência computacionalmente exigentes [39]. Para mitigar esse problema, estratégias como amostragem de vizinhos, *Mini-batch Gradient Descent* e *Dropout* são frequentemente utilizadas [46, 89], permitindo reduzir o número de operações sem necessidade de processar o grafo inteiro a cada iteração. No entanto, essas estratégias podem comprometer a qualidade das representações aprendidas, especialmente quando a amostragem não captura adequadamente a estrutura do grafo ou em grafos esparsos, onde a conectividade varia ao longo do tempo [81].

Outra limitação relevante é o fenômeno de *oversmoothing*. Como as GNNs utilizam a troca iterativa de mensagens (Eq. 4.2) para atualizar as representações dos nós, cada camada adicional permite que as informações fluam por uma vizinhança maior do grafo. No entanto, quando muitas camadas são empilhadas, os embeddings dos nós tendem a se tornar excessivamente homogêneos, dificultando a distinção entre diferentes regiões do grafo [72]. Esse efeito pode impactar negativamente o desempenho do modelo,

principalmente em redes profundas aplicadas a grafos densamente conectados. Essas limitações tornam-se particularmente relevantes quando trabalhamos com séries temporais representadas como grafos, pois esses dados são altamente dinâmicos e de alta dimensionalidade. No presente trabalho, optamos por adotar redes rasas, em consonância com as recomendações da literatura, a fim de evitar o *oversmoothing* e garantir a eficácia do modelo.

No próximo capítulo, detalharemos a implementação dos modelos de CSTG e conduziremos uma avaliação sistemática do desempenho das diferentes abordagens. Além da análise de concordância utilizando a métrica de Kappa de Fleiss e da comparação estatística baseada no Teste de Friedman com post-hoc de Nemenyi sobre os valores médios de *Macro F1-score*, também realizaremos uma investigação aprofundada sobre a relação entre as características estruturais das séries temporais e o desempenho dos modelos. Para isso, extraímos métricas como força de tendência, sazonalidade, ruído e autocorrelação, correlacionando-as com os resultados obtidos na classificação. Essa abordagem nos permitirá não apenas identificar os modelos e representações mais eficazes para a tarefa de CSST baseada em grafos, mas também estabelecer diretrizes para a escolha do modelo mais adequado conforme o tipo de série analisada.

Capítulo 5

Experimentos e Análises

Neste capítulo delineamos e apresentamos os experimentos e resultados obtidos na avaliação dos modelos de CSTG propostos. Inicialmente, descrevemos os conjuntos de dados utilizados e suas características, destacando a diversidade e complexidade dos cenários abordados. Em seguida, detalhamos as métricas de avaliação empregadas e a metodologia de treinamento dos modelos, incluindo a implementação da Graph Neural Alchemist (GNA) e a definição dos hiper-parâmetros de treinamento. Por fim, apresentamos os resultados obtidos e as análises estatísticas realizadas para comparar e avaliar o desempenho dos modelos em diferentes cenários de CSST.

Para garantir uma avaliação rigorosa, conduzimos testes de concordância para verificar a consistência dos resultados, bem como análises estatísticas para determinar diferenças significativas entre os modelos. Utilizamos a estatística de diferença crítica para identificar quais modelos apresentam desempenhos estatisticamente superiores em relação aos demais. Com base nos resultados e discussões apresentadas, encerramos o capítulo com uma recomendação dos modelos mais adequados para cada tipo de série temporal.

5.1 Conjuntos de Dados

Para realizar uma avaliação abrangente, optamos por utilizar 16 conjuntos de dados de séries temporais univariadas, sendo: 10 provenientes do **UCR-Archive** e 6 derivados do **CODE-15%**. A escolha desses conjuntos foi motivada por suas características complementares e sua relevância no contexto do presente estudo.

5.1.1 Conjuntos de Dados do UCR-Archive

Os 10 conjuntos de dados do **UCR-Archive**[22] foram selecionados a partir de uma amostragem estratificada, considerando diferentes domínios e níveis de complexidade. Esses conjuntos abrangem uma variedade de domínios e tarefas, sendo ideais para investigar a generalização dos modelos aqui comparados.

Na Tabela 5.1 a seguir, apresentamos uma descrição detalhada dos conjuntos utilizados, incluindo o número de séries, tamanho das séries, número de classes e o contexto da tarefa de classificação.

Tabela 5.1: Descrição dos Conjuntos de Dados do UCR-Archive

Conj. de dados	Nº Séries	Tamanho	Classes	Tipo	Descrição
Crop	24.000	46	24	IMAGE	Evolução espectral de coordenadas geográficas em imagens de satélite, anotadas em diferentes tipos de culturas agrícolas.
Earthquakes	460	512	2	SENSOR	Séries sismográficas, onde a tarefa é distinguir entre terremotos e ruídos.
ECG200	200	95	2	ECG	Sinais de ECG, com a tarefa de classificar batimentos cardíacos normais e anormais.
ECGFiveDays	884	136	2	ECG	Sinais de ECG de um paciente de 67 anos. As duas classes correspondem às datas de registro, com cinco dias de diferença.
ElectricDevices	16.637	96	7	DEVICE	Séries de sinais elétricos de dispositivos eletrônicos, com o objetivo de classificar o tipo de dispositivo.
NIFEKG-1	3.765	750	42	ECG	Séries temporais obtidas de medições não invasivas do eletrocardiograma fetal do lado esquerdo do tórax.
NIFEKG-2	3.765	750	42	ECG	Mesma série do Thorax1, porém com medições do lado direito.
Phoneme	2.110	1.024	39	AUDIO	Séries temporais de sinais acústicos, com a tarefa de classificar fonemas em diferentes categorias.
Strawberry	983	235	2	SPECTRO	Espectrogramas de Fourier com características de frutas, usados para distinguir entre morangos e não-morangos.
TwoLeadECG	1.162	82	2	ECG	Séries de ECG obtidas de dois canais de eletrodos. O objetivo é classificar entre os dois canais.

A caracterização dos conjuntos revela variações importantes nas distribuições das classes, com implicações diretas na modelagem e na escolha de estratégias específicas para cada um:

O conjunto Crop apresenta uma distribuição perfeitamente balanceada entre suas 24 classes, cada uma correspondendo a exatamente 4,17% das amostras tanto no treino quanto no teste. Já o ECG200 exhibe predomínio da classe 1 (69% treino, 64% teste),

indicando claro desbalanceamento. De forma semelhante, o ECGFiveDays apresenta uma leve assimetria, com a classe 0 predominando no treino (60,87%) e a classe 1 mais frequente no teste (50,29%). Essa diferença entre treino e teste pode afetar negativamente a capacidade de generalização dos modelos.

Já o Earthquakes possui forte desequilíbrio entre suas duas classes, com a classe majoritária correspondendo a 82% das amostras de treino e 75% das amostras de teste. Esse cenário também ocorre no conjunto ElectricDevices, com sete classes e distribuição heterogênea: a classe 5, mais representada, atinge aproximadamente 27% no treino e 24% no teste, enquanto a classe 6 é significativamente menos frequente, podendo introduzir viés durante o treinamento.

Os conjuntos NIFECG-1 e NIFECG-2, ambos com 42 classes, mostram uma distribuição relativamente uniforme, com proporções entre 1,83% e 3,11%. Já o Phoneme, com 39 classes, possui forte assimetria, com classes majoritárias, como a 23 e a 31, ultrapassando 7%, enquanto outras classes apresentam menos de 1% de representatividade. Esse cenário sugere a necessidade de métodos específicos para lidar com classes raras.

O conjunto Strawberry, com duas classes, tem leve predominância da classe 1 (64%), enquanto o TwoLeadECG, também com duas classes, apresenta distribuição praticamente equilibrada (aproximadamente 50%). Essa diferença nos níveis de balanceamento afeta diretamente a robustez e a performance dos modelos resultantes.

Conceitualmente, uma forma de se lidar com desbalanceamento de classes exige estratégias personalizadas: conjuntos com alto desequilíbrio, como ECG200, Earthquakes e Strawberry, podem demandar abordagens como reamostragem ou ajustes de pesos na função de perda para mitigar impactos negativos na generalização dos modelos. Por outro lado, conjuntos mais equilibrados ou com muitas classes, como Crop ou NonInvasiveFetalECGThorax, requerem técnicas que lidem com dispersão ou classes menos representadas, evitando perda de desempenho em categorias específicas.

Outro fator relevante é a grande variação no tamanho das séries temporais: Enquanto conjuntos como Crop (46 pontos) e TwoLeadECG (82 pontos) possuem séries relativamente curtas, outros como Phoneme (1.024 pontos) e NonInvasiveFetalECGThorax (750 pontos) exigem modelos capazes de capturar padrões mais complexos e dinâmicos ao longo do tempo. Portanto, além das estratégias de balanceamento, é fundamental considerar abordagens que sejam eficazes frente à variabilidade intrínseca no tamanho das séries.

5.1.2 Conjuntos de Dados do CODE-15%

Os 6 conjuntos derivados do **CODE-15%** foram incluídos para avaliar a capacidade dos modelos de classificar séries temporais de alta resolução e complexidade. Este repositório contém dados reais do contexto médico, o que traz desafios práticos, como a presença de ruídos e alta resolução, além de agregar relevância aplicada ao estudo.

As séries que compõem a base foram coletados pela Rede de Telessaúde do Hospital das Clínicas da UFMG [68, 69], consistindo de sinais de ECG de 12 derivações amostrados a $400Hz$, em intervalos de 7 e 10 segundos, resultando em séries com 4096 pontos.

Para facilitar os experimentos, foi selecionado um coorte de 4395 exames, dos quais 752 foram anotados como normais e 3643 correspondem a seis diferentes anomalias cardíacas. Cada anomalia foi tratada como uma tarefa de classificação binária, seguindo a estratégia um-contra-todos: os exames normais foram anotados como 1, enquanto os demais (incluindo outras anomalias) foram anotados como 0.

Dessa forma, foram construídos seis conjuntos de dados distintos, um para cada anomalia, todos com o mesmo conjunto de exames, mas com anotações específicas conforme a condição-alvo. A Tabela 5.2 detalha cada um desses conjuntos, sua descrição e distribuição das classes.

Tabela 5.2: Descrição das Anomalias e distribuição de anotações para cada um dos seis conjuntos de dados derivados do CODE-15%

Anomalia	Descrição	Classe 1	Classe 0	Tipo
1dAVb	Bloqueio Atrioventricular de Primeiro Grau – atraso na condução elétrica entre os átrios e ventrículos, resultando em um prolongamento do intervalo PR no ECG.	658	3737	CODE
RBBB	Bloqueio de Ramo Direito (Right Bundle Branch Block) – interrupção ou atraso na condução elétrica pelo ramo direito do feixe de His, alterando a despolarização ventricular.	682	3713	CODE
LBBB	Bloqueio de Ramo Esquerdo (Left Bundle Branch Block) – interrupção ou atraso na condução elétrica pelo ramo esquerdo do feixe de His, com impacto na ativação ventricular.	736	3659	CODE
SB	Bradicardia Sinusal – ritmo cardíaco anormalmente lento, com uma frequência inferior ao normal (geralmente abaixo de 60 bpm).	691	3704	CODE
AF	Fibrilação Atrial – ritmo cardíaco irregular e, frequentemente, acelerado, com contrações atriais descoordenadas e ineficazes.	651	3744	CODE
ST	Taquicardia Sinusal – ritmo cardíaco mais rápido que o normal, caracterizado por uma frequência superior a 100 bpm.	711	3684	CODE

Essa formulação permite analisar separadamente o desempenho dos modelos de classificação para cada anomalia, mantendo a consistência do conjunto original e refletindo uma prevalência clínica realista das condições cardíacas consideradas.

5.1.3 Extração de Características das Séries Temporais

A caracterização estrutural das séries temporais é um passo essencial para compreender sua complexidade e como suas propriedades impactam o desempenho dos modelos preditivos [41]. Séries temporais podem apresentar diferentes padrões dinâmicos, como tendências, sazonalidades e variações residuais, que influenciam diretamente a escolha do modelo mais adequado para a tarefa de classificação [5, 11]. Portanto, para uma análise quantitativa robusta, é necessário extrair um conjunto de características que descrevam adequadamente a estrutura dessas séries.

Para capturar os componentes fundamentais dos nossos dados, utilizamos o método **Seasonal-Trend Decomposition using Loess (STL)** [18], disponível na biblioteca *statsmodels*¹ [74]. O STL permite a decomposição das séries em três componentes principais: **tendência**, **sazonalidade** e **resíduos**. Essa abordagem possibilita uma análise mais granular da estrutura dos dados, facilitando a extração de características relevantes.

A definição correta do período sazonal das séries é um fator crítico para garantir uma decomposição precisa. No entanto, como os conjuntos de dados analisados possuem diferentes domínios e resoluções temporais, os períodos foram inferidos a partir de *Heurísticas* adaptadas a cada tipo de série, derivadas das descrições presentes nos arquivos de documentação dos conjuntos (*Readme.md*) e considerando fatores como frequência de captura, tamanho das séries e periodicidade dos ciclos subjacentes. A seguir, apresentamos os períodos sazonais definidos para cada tipo de dado:

- **AUDIO (Phoneme)**: $\max(5, \frac{\text{ts_length}}{10})$ – Captura ciclos das ondas sonoras.
- **SPECTRO (Strawberry)**: $\max(10, \frac{\text{ts_length}}{4})$ – Baseado em picos da Transformada de Fourier.
- **CODE (CODE-15%)**: 12 – Os exames desse conjunto contém em média 12 batimentos cardíacos por amostra.
- **ECG (conjuntos ECG)**: $\max(20, \frac{\text{ts_length}}{5})$ – Aproximação de ciclos cardíacos.
- **DEVICE (ElectricDevices)**: $\max(30, \frac{\text{ts_length}}{30})$ – Aproxima periodicidade horária (2 min = 30 leituras/hora).
- **SENSOR (Earthquakes)**: 20 – Ajustado com base na regra de exemplos negativos anotados no conjunto.
- **IMAGE (Crop)**: $\max(5, \frac{\text{ts_length}}{4})$ – Heurística geral para séries derivadas de imagens.

No caso particular do conjunto de dados **Earthquakes**, a periodicidade foi ajustada considerando a metodologia de construção dos dados, que define um evento sísmico positivo como um evento de magnitude superior a 5 na escala Richter que não tenha sido precedido por outro evento maior nas últimas 512 horas. No entanto, a construção dos exemplos negativos segue um critério diferente: uma leitura inferior a 4 é considerada um caso negativo somente se houver pelo menos 20 leituras anteriores não nulas nas últimas 512 horas [22].

Como o conjunto é extremamente desbalanceado (80% negativos e 20% positivos), a escolha de 20 horas como período sazonal é justificada pela necessidade de capturar

¹<https://www.statsmodels.org/stable/index.html>

padrões discriminativos antes de eventos sísmicos, evitando janelas muito grandes que possam obscurecer oscilações locais importantes.

Com base nos valores decompostos, extraímos um conjunto de características que descrevem a complexidade estrutural de cada conjunto de dados, permitindo uma análise quantitativa detalhada sobre a relação entre essas propriedades e o desempenho dos modelos. As métricas extraídas incluem:

- **Força da Tendência** (trd): Mede a presença de um comportamento crescente ou decrescente ao longo do tempo. Séries com alta tendência são mais previsíveis e favorecem modelos que capturam padrões de longo prazo;
- **Força da Sazonalidade** (s): Avalia a presença de ciclos periódicos na série. Quanto maior o valor, mais previsíveis são os padrões sazonais, sendo um fator relevante para modelos que exploram periodicidade explícita;
- **Variância Residual** (σ^2): Representa a fração dos dados não explicada pelos componentes de tendência e sazonalidade. Séries com alta variância residual tendem a ser mais ruidosas e de difícil modelagem;
- **Entropia dos Resíduos** (H): Mede a aleatoriedade dos resíduos após a remoção dos componentes estruturais da série. Séries com alta entropia residual são mais imprevisíveis, dificultando a generalização de modelos preditivos;
- **Autocorrelação** (ρ): Indica o grau de dependência temporal entre observações sucessivas. Séries com alta autocorrelação mantêm padrões consistentes ao longo do tempo, favorecendo abordagens baseadas em aprendizado profundo e redes neurais recorrentes.

A Tabela 5.3 apresenta os valores médios das características extraídas para cada conjunto de dados analisado.

Tabela 5.3: Características extraídas dos conjuntos de séries temporais, destacando Força da Tendência (trd), Força da Sazonalidade (s), Variância Residual (σ^2), Entropia Residual (H) e Autocorrelação (ρ). Valores em destaque indicam a maior pontuação em cada métrica.

Conjunto de dados	trd	s	σ^2	H	ρ
CODE15	0.931485	0.001711	0.006316	7.339803	0.986768
Crop	0.612153	0.182481	0.005636	3.486818	0.784319
ECG200	0.714666	0.177893	0.238436	4.160936	0.934054
ECGFiveDays	0.264419	0.143834	0.593620	4.271018	0.858594
Earthquakes	0.070476	0.039179	0.893438	5.815661	0.047477
ElectricDevices	0.165958	0.301823	0.624670	4.079279	0.303687
NIFEKG-1	0.451022	0.227220	0.447514	6.100526	0.979179
NIFEKG-2	0.463691	0.226040	0.446367	6.086565	0.980768
Phoneme	0.046361	0.100193	0.895269	6.556296	0.815660
Strawberry	0.652759	0.257617	0.489633	5.105436	0.993149
TwoLeadECG	0.790424	0.364308	0.200118	4.101504	0.972313

A análise dos conjuntos de dados permite destacar diferenças fundamentais entre séries temporais altamente estruturadas e aquelas de natureza mais caótica. O conjunto **Earthquakes**, por exemplo, apresenta uma **alta entropia residual (5.81)** e **baixa autocorrelação (0.0475)**, tornando-se um dos mais desafiadores para modelagem preditiva, dada sua elevada imprevisibilidade. Esse comportamento é corroborado pelos testes de concordância e pelos resultados experimentais, que indicam menor desempenho dos modelos nesse cenário.

Por outro lado, o conjunto **Strawberry** apresenta **alta autocorrelação (0.9931)** e **baixa variância residual (0.4896)**, indicando padrões altamente estruturados e previsíveis. Essa característica favorece a concordância entre os modelos e explica o desempenho superior obtido nessa base de dados.

5.2 Métricas de Avaliação

Nos estudos e trabalhos de CSST, a seleção de métricas de avaliação é fundamental para garantir uma análise estatística comparativa justa e robusta do desempenho dos modelos. Essa escolha é particularmente relevante no contexto deste estudo, cujo objetivo central é realizar uma análise estatística para comparar diferentes modelos de series-

as-graphs. Diversas métricas podem ser empregadas, cada qual com características e aplicações específicas. Aqui, adotamos o F1-score como a principal métrica de avaliação, devido à sua capacidade de capturar tanto a precisão quanto a revocação em uma única medida balanceada, como definido na Equação 5.1:

$$F1\text{-score} = 2 \times \frac{\text{Precisão} \times \text{Revocação}}{\text{Precisão} + \text{Revocação}} \quad (5.1)$$

onde Precisão representa a proporção de verdadeiros positivos em relação ao total de predições positivas, e Revocação indica a proporção de verdadeiros positivos em relação ao total de exemplos positivos. O F1-score é pode ser utilizado em problemas de classificação binária e multiclasse, mostrando-se especialmente eficaz em cenários de classes desbalanceadas, onde métricas como acurácia podem fornecer uma visão distorcida do desempenho.

Essa escolha reflete o alinhamento do F1-score médio com os objetivos do estudo, já que a métrica captura nuances importantes no desempenho dos modelos que seriam mascaradas por métricas menos abrangentes. Assim, ela assegura uma análise robusta e alinhada à natureza comparativa da presente pesquisa.

Além da F1-score, utilizamos como métrica secundária o Kappa de Fleiss (κ)[27], amplamente reconhecido por sua capacidade de medir o nível de concordância entre múltiplos avaliadores, sendo particularmente relevante em cenários de classificação multiclasse. Nesse contexto, consideramos os avaliadores como os modelos apresentados na Seção3, acrescidos do ROCKET. O Kappa de Fleiss é calculado sobre os valores reais dos dados e os preditos pelos modelos, conforme a Equação 5.2 abaixo:

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \quad (5.2)$$

onde: \bar{P} é a proporção média de concordância observada entre os avaliadores, e \bar{P}_e é a proporção de concordância esperada ao acaso.

O valor de κ varia entre 0 e 1, sendo interpretado conforme apresentado na Tabela 5.4.

Tabela 5.4: Interpretação dos valores k da métrica Kappa de Fleiss

Valor de Kappa	Interpretação
< 0	Concordância ruim
0.01 - 0.20	Concordância leve
0.21 - 0.40	Concordância razoável
0.41 - 0.60	Concordância moderada
0.61 - 0.80	Concordância substancial
0.81 - 1.00	Concordância quase perfeita

A inclusão do Kappa de Fleiss fortalece a análise comparativa dos resultados, ao avaliar a consistência das previsões entre diferentes modelos, complementando as con-

clusões obtidas com o F1-score. Dessa forma, a metodologia proposta assegura que os experimentos conduzam a compreensões confiáveis e alinhadas às perguntas de pesquisa.

5.3 Implementação e Treinamento

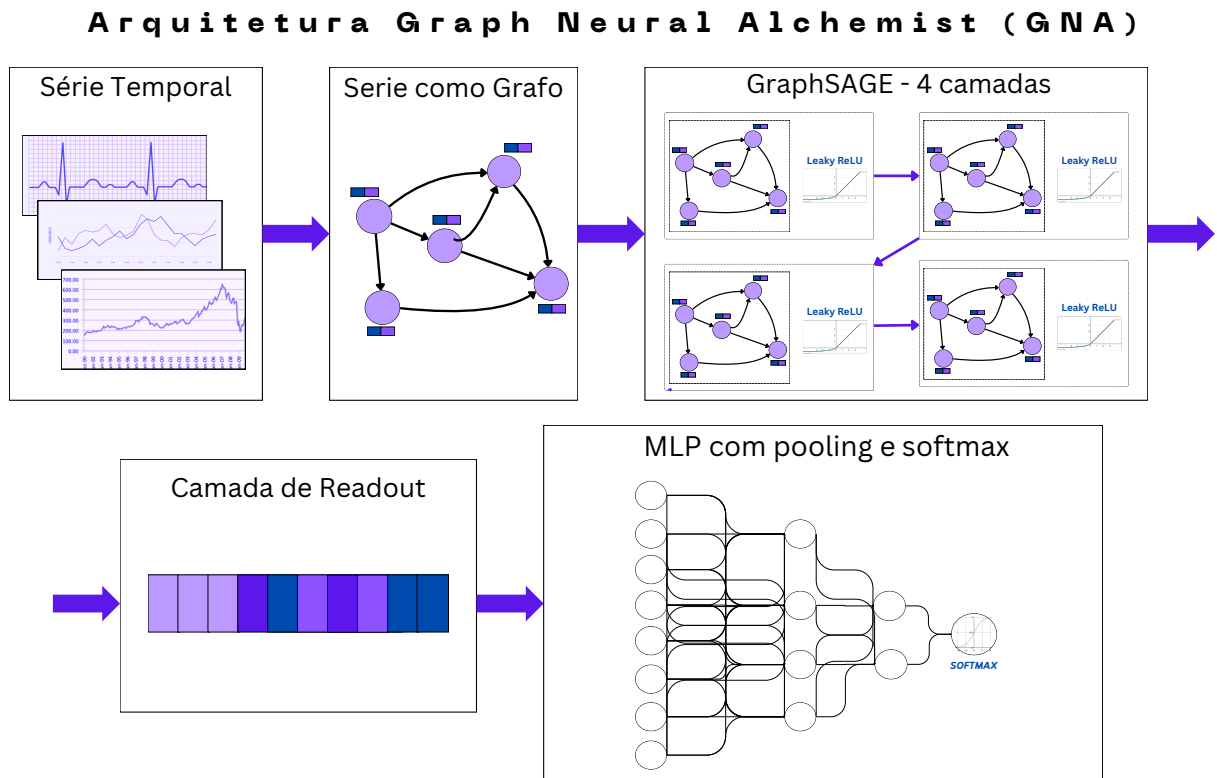
Para a implementação, treino e teste dos modelos, desenvolvemos uma arquitetura modular para CSTG capaz de integrar diferentes representações de séries temporais como grafos e arquiteturas de redes neurais. Assim, propomos a **Graph Neural Alchemist (GNA)**, um *framework* flexível que permite a experimentação com diversas técnicas de RSTG, incluindo, mas não limitado a *SimTSC*, *Time2Graph*, *GVN* e *GTPO*, combinadas com as arquiteturas de GNNs apresentadas na Seção 4.2.1: GCN, GraphSAGE e GAT.

A GNA foi projetada para oferecer modularidade, permitindo a fácil substituição e combinação de diferentes componentes ao longo do pipeline de aprendizado. A estrutura geral do *framework* é composta por quatro módulos principais:

1. **Módulo de Representação:** Converte séries temporais em grafos utilizando abordagens como Grafos de Visibilidade e Grafos de Transição de Padrões Ordinais;
2. **Módulo de Aprendizado em Grafos:** Aplica uma arquitetura de GNN para processar as representações gráficas geradas na etapa anterior;
3. **Camada de Readout:** Condensa as características extraídas pela GNN em um vetor de características representativo da série;
4. **Módulo de Classificação:** Utiliza o vetor aprendido pela GNN como entrada para um classificador, como uma MLP, para realizar a predição das classes.

A GNA é *data-agnostic*, ou seja, pode ser aplicada a diferentes representações de séries temporais, permitindo a comparação objetiva entre abordagens de RSTG. Essa característica a torna uma ferramenta versátil para experimentação, oferecendo um ambiente unificado para avaliação de diferentes estratégias de modelagem de séries temporais como grafos.

Figura 5.1: Arquitetura da GNA utilizada no presente trabalho. A série temporal é representada como um grafo e processada por 4 camadas de GraphSAGE, seguidas de uma MLP modificada para realizar *pooling* dos *embeddings* aprendidos. A classificação final é realizada por uma função Log-Softmax. Cada componente da arquitetura é modular, permitindo testar e comparar diferentes métodos de representação e classificação de séries como grafos.



No presente trabalho, utilizamos a arquitetura ilustrada na Figura 5.1 para implementar e treinar os modelos de RSTG apresentados no Capítulo 3 configurada com uma GraphSAGE de 4 camadas e classificador MLP, resultando nas seguintes variações:

- *Time2Graph-GNA*,
- *GTPO-2feat*, *GTPO-encoded*,
- *VisibilityGraph-2feat*, *VisibilityGraph-3feat*,
- *SimTSC-GNA* (sem camada de readout e MLP, por ser uma tarefa de classificação de nós).

Além dessas variações, também avaliamos as implementações originais do *SimTSC*, *Time2Graph* e *Time2Graph+*, bem como o modelo ROCKET [24], que foi utilizado como *baseline*. Abaixo, detalhamos cada modelo avaliado e suas respectivas implementações:

- **VisibilityGraph-2feat**: GNA com GVN utilizando 2 características de nós: PageRank e $\text{deg}_{in}(v)$;
- **VisibilityGraph-3feat**: GNA com GVN utilizando 3 características de nós: PageRank, $\text{deg}_{in}(v)$ e valor t_i da série;
- **SimTSC-GCN**: Implementação original do SimTSC com GCN [90];
- **SimTSC-GNA**: GNA utilizando estratégia de similaridade do SimTSC, adaptada para classificação de nós;
- **Time2Graph**: Implementação original do Time2Graph com DeepWalk [15];
- **Time2Graph+**: Implementação original do Time2Graph+ com GAT [16];
- **Time2Graph-GNA**: GNA utilizando a representação de grafos de evolução de Shapelets do Time2Graph;
- **GTPO-2feat**: GNA com GTPO utilizando 2 características de nós: PageRank e $\text{deg}_{in}(v)$;
- **GTPO-encoded**: GNA com GTPO codificando os PO nos nós via one-hot encoding;
- **ROCKET**: Baseline baseado em convoluções e *random kernels* [24].

Para os experimentos com o *SimTSC* e *Time2Graph*, utilizamos as implementações disponibilizadas pelos autores², garantindo que os parâmetros e estratégias de treinamento fossem mantidos consistentes. Já os modelos modificados *SimTSC-GNA* e *Time2Graph-GNA*, bem como os demais baseados em GVN e GTPO, foram implementados a partir das versões originais, utilizando:

- **PyTorch** ³ e **PyTorch Lightning** ⁴, com suporte ao TensorBoard ⁵ para visualização dos experimentos;
- **Deep Graph Library (DGL)** ⁶, para operações eficientes em grafos e implementação das GNNs.

Os experimentos foram conduzidos em servidores Ubuntu 22.10, equipados com um processador AMD Ryzen 9 5950X (16 núcleos), 64GB de RAM e uma GPU NVIDIA GeForce RTX 3090 Ti (24GB de VRAM). Utilizamos CUDA (versão 12.4) para otimização

²<https://github.com/petecheng/Time2GraphPlus>, <https://github.com/daochenzha/SimTSC>

³<https://pytorch.org/>

⁴<https://lightning.ai/docs/pytorch/stable/>

⁵<https://www.tensorflow.org/tensorboard>

⁶<https://www.dgl.ai/>

computacional, sendo a versão do Python utilizada a 3.11.3. O código completo está disponível no repositório público do Github: https://github.com/paulohdscoelho/graph_neural_chemist.

Todos os modelos foram treinados e testados utilizando a divisão original de treino e teste do UCR Archive, enquanto para o conjunto CODE-15%, aplicamos uma divisão 80-20. O treinamento foi realizado utilizando o otimizador Adam com *mini-batches*, sendo a função de perda a entropia cruzada.

Para garantir consistência entre os experimentos e uma escolha equilibrada de hiper-parâmetros, realizamos uma busca aleatória em grade para otimizar os valores de *learning rate*, *batch size*, *hidden dimension* e *number of epochs*. A busca foi realizada em um conjunto diversificado de séries temporais, selecionando os valores que proporcionaram os melhores resultados de F1-score para o modelo visibilidadeGraph-2feat. Os hiper-parâmetros escolhidos foram:

- *número de épocas (epochs)* = 250;
- *taxa de aprendizado (lr)* = 0.001;
- *dimensão dos embeddings ocultos (nhid)* = 128;
- *tamanho do mini-batch (batch_size)* = 32.

A escolha desses hiper-parâmetros segue também o que é amplamente adotado na literatura especializada. A escolha de um número fixo de épocas é consistente com estudos anteriores que apontam que redes neurais em grafos tendem a convergir em poucas centenas de épocas, especialmente quando combinadas com otimização via *Adam*[39, 80].

A definição da também está alinhada com abordagens consagradas na literatura. Modelos de aprendizado profundo frequentemente utilizam valores entre 10^{-4} e 10^{-2} [35], sendo 0.001 um valor amplamente adotado para treinar GNNs[39, 49, 82]. Essa taxa garante estabilidade no aprendizado sem comprometer a velocidade de convergência.

A dimensão dos embeddings ocultos ($nhid = 128$) segue práticas consolidadas para GNNs, onde dimensões típicas variam entre 32 e 256 [39, 80, 82]. O valor 128 é um compromisso entre capacidade expressiva e eficiência computacional, permitindo a captura de estruturas complexas sem inflacionar excessivamente o número de parâmetros do modelo.

O tamanho do mini-batch segue um padrão frequentemente adotado em DNNs [92]. Trabalhos anteriores mostram que tamanhos de mini-batch entre 16 e 64 oferecem um bom compromisso entre estabilidade do gradiente e eficiência computacional em modelos de aprendizado profundo [77]. Em GNNs, essa escolha é particularmente relevante, pois o processamento de grafos pode ser mais intensivo devido à agregação de vizinhos durante a propagação das mensagens [39].

Essas escolhas garantem que os modelos sejam treinados em um regime amplamente validado na literatura, proporcionando resultados comparáveis a estudos prévios e assegurando que a análise experimental seja conduzida em um ambiente padronizado.

5.4 Análises Estatísticas

A avaliação dos modelos foi realizada por meio de testes estatísticos rigorosos para identificar diferenças significativas no desempenho dos modelos. Para isso, utilizamos o Teste de Hipóteses de Friedman [29, 30, 31], um teste não-paramétrico amplamente empregado para comparar classificadores em múltiplos conjuntos de dados. Adicionalmente, aplicamos o teste post-hoc de Nemenyi [59], que permite identificar quais pares de modelos apresentam diferenças estatisticamente significativas.

A aplicação desses testes é essencial para verificar se as diferenças observadas nos resultados dos modelos são reais ou se podem ser atribuídas ao acaso. Essa abordagem segue as diretrizes estabelecidas por Demšar [25] e García and Herrera [34] para avaliação comparativa de modelos de Aprendizado de Máquina.

5.4.1 Resultados das Avaliações de Desempenho

Na Tabela 5.5, apresentamos os valores médios de *F1-score* obtidos pelos diferentes modelos nos experimentos conduzidos. Os melhores desempenhos para cada conjunto de dados estão destacados em negrito.

Tabela 5.5: F1-score médio para cada modelo e conjunto de dados

(a) Primeira parte: modelos de VisibilityGraph a Time2Graph.

Conjunto de dados/Modelo	VisibilityGraph-2feat	VisibilityGraph-3feat	SimTSC-GCN	SimTSC-GNA	Time2Graph
1dAVb	0.54	0.67	0.47	0.73	0.52
AF	0.66	0.87	0.46	0.82	0.50
Crop	0.51	0.67	0.02	0.73	0.09
ECG200	0.77	0.80	0.39	0.84	0.46
ECGFiveDays	0.75	0.80	0.33	0.94	0.57
Earthquakes	0.43	0.48	0.43	0.56	0.56
ElectricDevices	0.61	0.68	0.09	0.62	0.23
LBBB	0.88	0.87	0.46	0.89	0.52
NonInvasiveFetalECGThorax1	0.59	0.74	0.00	0.91	0.05
NonInvasiveFetalECGThorax2	0.66	0.83	0.00	0.90	0.06
Phoneme	0.14	0.17	0.01	0.19	0.03
RBBB	0.85	0.86	0.46	0.93	0.48
SB	0.68	0.83	0.46	0.86	0.48
ST	0.87	0.84	0.46	0.85	0.49
Strawberry	0.92	0.95	0.39	0.96	0.44
TwoLeadECG	0.86	0.87	0.33	1.00	0.53

(b) Segunda parte: modelos de Time2Graph+ a ROCKET.

Conjunto de dados/Modelo	Time2Graph+	Time2Graph-GNA	GTPO-2feat	GTPO-encoded	ROCKET
1dAVb	0.49	0.51	0.47	0.50	0.79
AF	0.52	0.47	0.46	0.53	0.82
Crop	0.67	0.65	0.19	0.39	0.67
ECG200	0.76	0.88	0.39	0.69	0.89
ECGFiveDays	0.65	0.94	0.61	0.51	1.00
Earthquakes	0.45	0.42	0.43	0.57	0.43
ElectricDevices	0.53	0.48	0.46	0.50	0.64
LBBB	0.60	0.54	0.54	0.54	0.91
NonInvasiveFetalECGThorax1	0.66	0.65	0.14	0.09	0.92
NonInvasiveFetalECGThorax2	0.70	0.77	0.13	0.11	0.95
Phoneme	0.06	0.08	0.07	0.07	0.11
RBBB	0.59	0.49	0.46	0.50	0.90
SB	0.69	0.51	0.50	0.54	0.91
ST	0.63	0.49	0.46	0.51	0.94
Strawberry	0.87	0.83	0.68	0.64	0.97
TwoLeadECG	0.61	0.67	0.74	0.65	1.00

A análise preliminar revela que o modelo ROCKET obteve os melhores desempenhos médios em *F1-score*, seguido por SimTSC-GNA e VisibilityGraph-3feat. No entanto, para determinar se essas diferenças são estatisticamente significativas, aplicamos os testes de Friedman e Nemenyi.

5.4.2 Teste de Friedman e Diferença Crítica

O Teste de Friedman é uma alternativa ao teste ANOVA para medidas repetidas, sendo baseado nos rankings de modelos classificadores em diferentes conjuntos de dados. Ele avalia se pelo menos um deles apresenta um desempenho estatisticamente distinto dos demais. A estatística de Friedman é calculada como:

$$\chi_F^2 = \frac{12N}{k(k+1)} \sum_{j=1}^k R_j^2 - 3N(k+1) \quad (5.3)$$

onde: k é o número de modelos, N é o número de conjuntos de dados e R_j representa a soma dos rankings do modelo j ao longo dos conjuntos de dados.

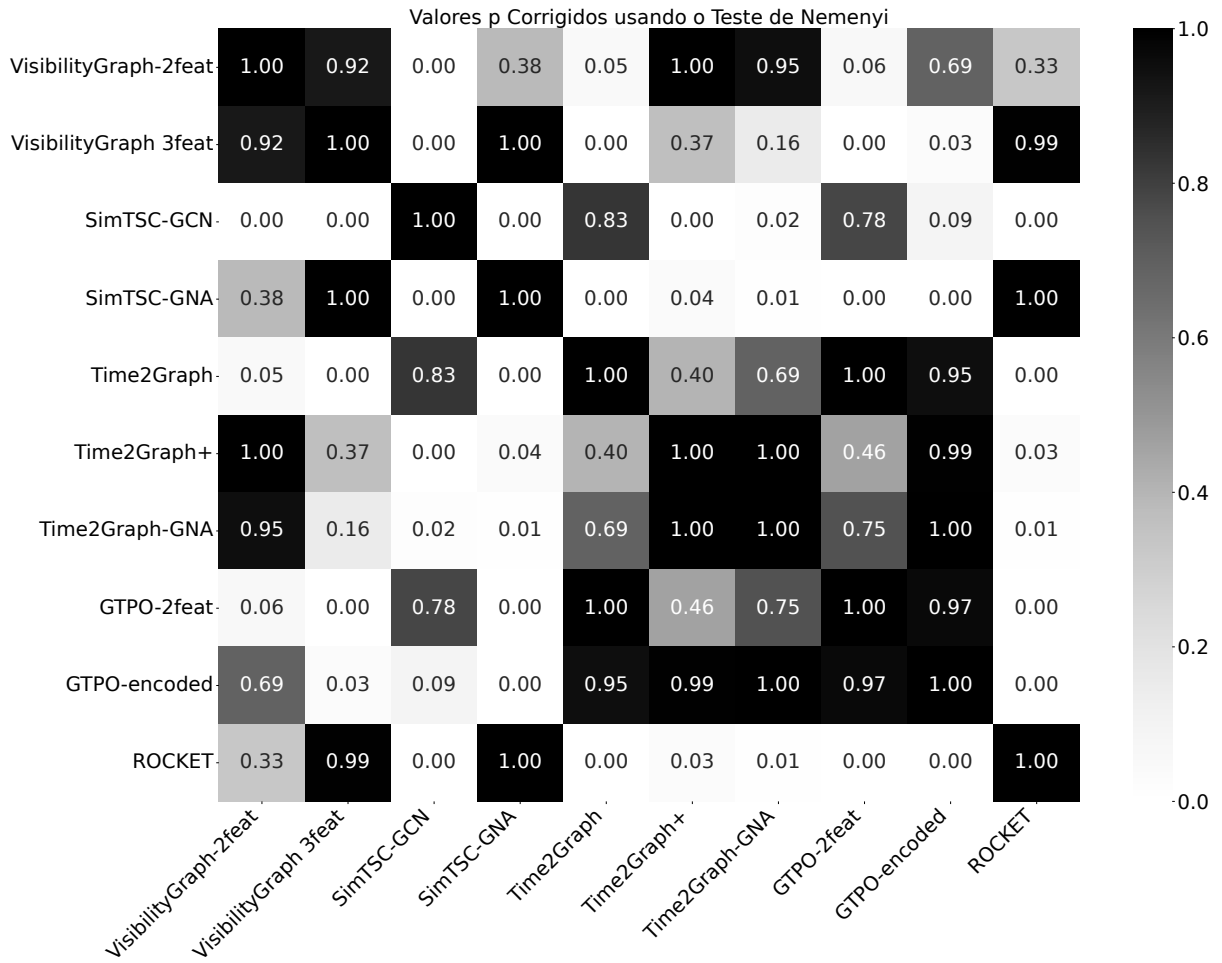
A hipótese nula do teste é:

“Não há diferença estatisticamente significativa entre o desempenho dos modelos analisados”.

Caso o valor-p obtido seja menor que o nível de significância adotado ($\alpha = 0,05$), rejeitamos a hipótese nula e concluímos que existem diferenças estatisticamente significativas entre os modelos. Os valores obtidos pelos nossos testes rejeitam a hipótese nula com valor-p = $1,92 \times 10^{-9}$ indicando que pelo menos um dos modelos apresenta um desempenho distinto dos demais.

No entanto, esse teste apenas indica se há diferenças globais entre os modelos, sem especificar quais diferem entre si. Para aprofundar essa análise, utilizamos o teste post-hoc de Nemenyi [59], que permite comparar os modelos par a par, identificando aqueles que apresentam diferenças estatisticamente significativas. A Figura 5.2 apresenta os valores-p corrigidos obtidos por esse teste.

Figura 5.2: Valores-p corrigidos obtidos pelos testes estatísticos de Friedman e post-hoc de Nemenyi.



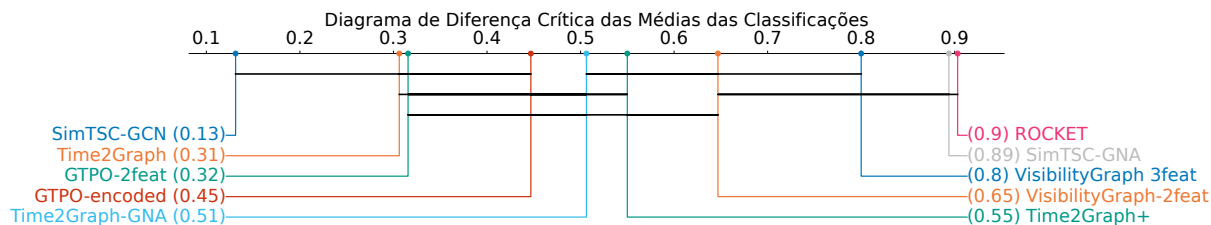
A partir desses valores, calculamos a Diferença Crítica (CD) entre os modelos, que indica a magnitude da diferença nos rankings médios necessária para que dois modelos sejam considerados estatisticamente distintos. A CD é dada por:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \quad (5.4)$$

onde: q_α é o valor crítico da estatística de Nemenyi para um determinado nível de significância α , k é o número de modelos comparados e N é o número de conjuntos de dados. Se a diferença nos rankings médios entre dois modelos for maior que o CD, podemos concluir que eles possuem desempenhos estatisticamente distintos.

Por meio do Diagrama de Diferença Crítica apresentado na Figura 5.3 é possível ter um panorama visual das diferenças entre os modelos avaliados. Esse diagrama ordena os modelos conforme seus rankings médios e agrupa aqueles que não apresentam diferenças estatisticamente significativas entre si. Essa abordagem foi introduzida por Demšar [25] e se tornou um padrão na avaliação comparativa de modelos de aprendizado de máquina.

Figura 5.3: Diagrama de Diferença Crítica baseado no Teste de Friedman seguido do teste de Nemenyi. Nessa visualização, os modelos são organizados em um eixo horizontal de acordo com seu ranking médio de desempenho. Modelos que não apresentam diferenças estatisticamente significativas são agrupados em cliques por meio de linhas horizontais.



A análise do diagrama mostra que o modelo ROCKET apresentou a melhor média de rankings. No entanto, sua diferença em relação ao SimTSC-GNA e ao VisibilityGraph-3feat não é estatisticamente significativa. Isso sugere que, apesar do desempenho superior, o ROCKET não possui uma vantagem substancial sobre essas abordagens baseadas em grafos.

Outro ponto relevante é o desempenho do SimTSC-GNA, que superou sua versão original SimTSC-GCN, evidenciando o impacto positivo do uso do GraphSAGE no aprendizado das representações em comparação com a abordagem de GCN. Além disso, o modelo GTPO-encoded obteve um desempenho superior ao GTPO-2feat, indicando que a codificação dos PO como características de nós pode ser uma estratégia eficaz para melhorar o aprendizado.

Por fim, os resultados indicam que VisibilityGraph-3feat supera VisibilityGraph-2feat, sugerindo que a inclusão da característica temporal t_i nos nós contribuiu para um melhor desempenho. Esses achados reforçam a importância da escolha criteriosa das representações, das características dos nós e da arquitetura de aprendizado ao modelar séries temporais como grafos.

5.5 Concordância entre Avaliadores

Para avaliar a concordância entre os 10 modelos ao longo dos 16 conjuntos de dados, utilizamos a métrica de Kappa de Fleiss, detalhada na Seção 5.2. O valor global obtido foi $\kappa = 0.0176$, indicando uma concordância leve entre os modelos. Embora esse resultado sugira uma concordância mínima, ele reflete tanto as diferenças entre as abordagens utilizadas quanto as características específicas dos conjuntos de dados analisados, não sendo um indicativo direto da eficácia dos modelos.

A Tabela 5.6 apresenta os valores de Kappa de Fleiss para cada conjunto de dados. Observamos uma variabilidade no grau de concordância, de leve a moderada. Essa variação pode estar relacionada à forma como cada modelo representa as séries temporais como grafos, impactando sua concordância ao classificar diferentes conjuntos de dados.

Em conjuntos como ElectricDevices, Strawberry e TwoLeadECG, a concordância foi relativamente maior, enquanto Earthquakes apresentou uma concordância negativa, o que pode indicar inconsistências na estrutura do dado ou desafios específicos para os modelos nesse contexto.

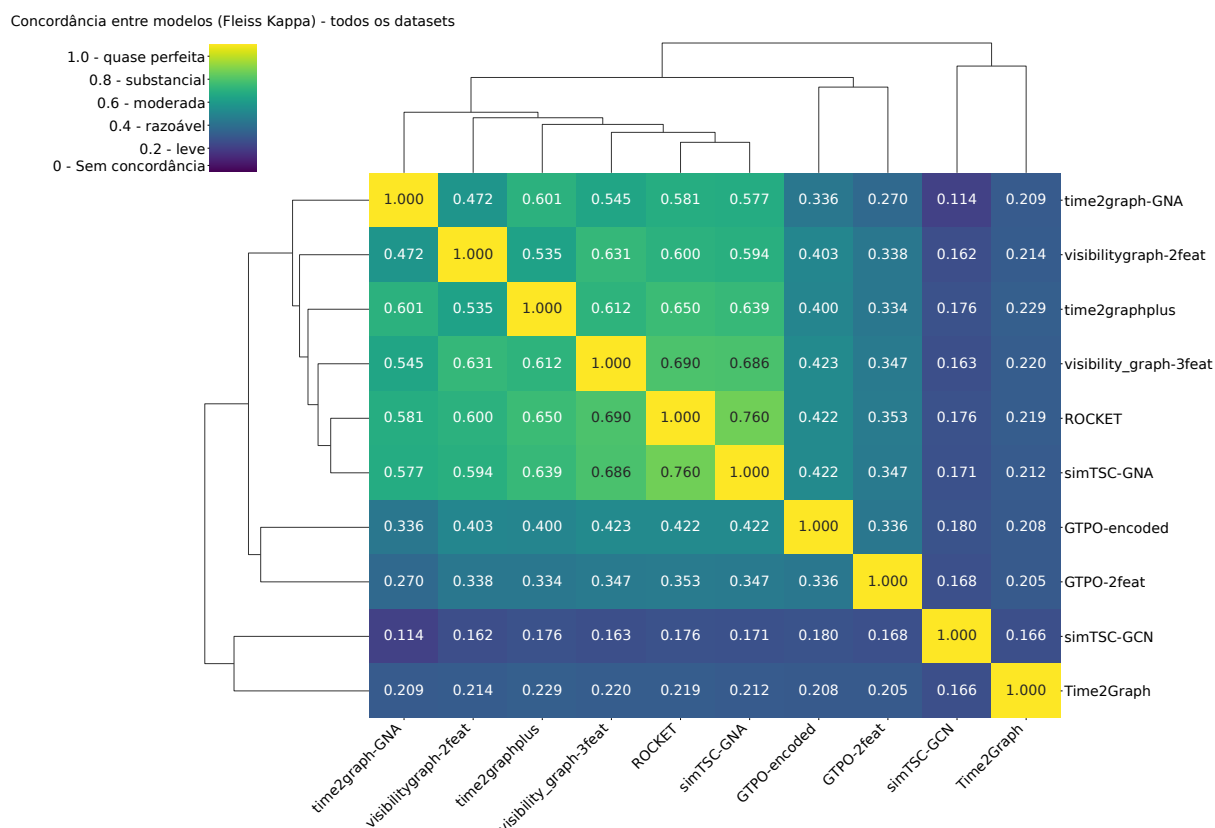
Tabela 5.6: Kappa Score para cada Conjunto de Dados.

Conjunto de Dados	Kappa Score	Interpretação
Crop	0.2697	Concordância razoável
ECG200	0.3260	Concordância razoável
ECGFiveDays	0.2106	Concordância razoável
Earthquakes	-0.0202	Concordância ruim
ElectricDevices	0.3898	Concordância razoável
NonInvasiveFetalECGThorax1	0.2249	Concordância razoável
NonInvasiveFetalECGThorax2	0.2541	Concordância razoável
Phoneme	0.0647	Concordância leve
Strawberry	0.3750	Concordância razoável
TwoLeadECG	0.2178	Concordância razoável
1dAVb	0.0223	Concordância leve
AF	0.0244	Concordância leve
LBBB	0.1731	Concordância leve
RBBB	0.1144	Concordância leve
SB	0.0778	Concordância leve
ST	0.1258	Concordância leve

A Figura 5.4 apresenta um mapa de calor com dendrograma, ilustrando a relação de concordância e os agrupamentos hierárquicos entre os modelos. Observamos um agrupamento entre o ROCKET e SimTSC-GNA, o que é consistente com os resultados do Diagrama de Diferença Crítica (Figura 5.3), sugerindo que suas abordagens de extração de características compartilham similaridades estruturais. Ambos aplicam técnicas baseadas em CNN-1D para capturar padrões nas séries temporais.

Outro agrupamento interessante é o do Time2Graph com o SimTSC-GCN. Embora suas abordagens sejam distintas, essa proximidade sugere que a estratégia de caminhos aleatórios do DeepWalk pode capturar informações estruturais similares às das convoluções em grafos da GCN.

Figura 5.4: Mapa de calor com dendrograma representando a concordância Kappa de Fleiss entre os modelos avaliados.



Outro ponto de interesse é a concordância moderada entre as implementações do GTPO (GTPO-2feat e GTPO-encoded), com $\kappa = 0.336$, sugerindo que a forma como os padrões ordinais são codificados nos nós impacta de maneira consistente os resultados. Além disso, o Time2Graph e sua versão aprimorada Time2GraphPlus também apresentaram concordância considerável, reforçando a robustez da modelagem de Shapelets evolutivos na classificação de séries temporais.

5.5.1 Correlação entre Concordância e Estrutura das Séries Temporais

Para entender melhor a relação entre a estrutura das séries temporais e a concordância entre os modelos, analisamos a correlação r entre os valores de Kappa de Fleiss (Tabela 5.6) e as propriedades estruturais das séries temporais (Tabela 5.3). A Tabela 5.7 apresenta os coeficientes de correlação de Pearson entre essas variáveis.

Tabela 5.7: Correlação de Pearson entre Kappa de Fleiss e as Características das Séries Temporais.

Característica	Correlação com Kappa de Fleiss
Força da Sazonalidade s	0.7086
Força da Tendência trd	0.5640
Autocorrelação ρ	0.3911
Entropia Residual H	-0.5468
Variância Residual σ^2	-0.5701

Os resultados revelam padrões importantes sobre a relação entre a estrutura das séries e a estabilidade dos modelos:

- A **força da sazonalidade** apresentou a maior correlação positiva ($r = 0,71$), indicando que séries com padrões sazonais bem definidos favorecem maior concordância entre os modelos;
- A **força da tendência** também demonstrou correlação positiva ($r = 0,56$), sugerindo que séries com padrões estruturados ao longo do tempo tendem a ser classificadas de forma mais consistente;
- A **autocorrelação** ($r = 0,39$) apresentou uma relação positiva moderada, indicando que séries mais previsíveis ao longo do tempo resultam em menor variabilidade nas classificações entre os modelos;
- A **entropia residual** e a **variância residual** apresentaram correlação negativa com κ ($r = -0,55$ e $r = -0,57$, respectivamente), sugerindo que séries mais ruidosas e imprevisíveis resultam em menor concordância entre os modelos.

Para verificar a robustez dessas relações, também calculamos o coeficiente de correlação de **Spearman** [1], que avalia associações monotônicas entre variáveis, independentemente da linearidade. A análise com Spearman manteve as mesmas tendências observadas com Pearson, embora com intensidades ligeiramente inferiores, como apresentado a seguir: força da sazonalidade ($\rho = 0,62$), força da tendência ($\rho = 0,45$), autocorrelação ($\rho = 0,27$), variância residual ($\rho = -0,37$) e entropia residual ($\rho = -0,52$).

Essas diferenças indicam que, embora a relação entre estrutura das séries e concordância entre modelos não seja estritamente linear, ela é monotônica — ou seja, os modelos tendem a concordar mais conforme aumenta a organização e previsibilidade da série. A consistência entre os dois coeficientes reforça a conclusão de que séries mais estruturadas, com forte sazonalidade e baixa entropia, resultam em maior estabilidade entre as predições dos modelos.

Por outro lado, séries com alta entropia ou variabilidade residual, como o já mencionado conjunto **Earthquakes**, desafiam os modelos de forma distinta, levando a menores

níveis de concordância. Isso evidencia que a estrutura interna dos dados influencia não apenas o desempenho individual dos modelos, mas também a consistência entre eles, um aspecto relevante em cenários de seleção de modelos e construção de ensembles.

5.6 Discussão

Os resultados obtidos indicam que, embora os modelos avaliados apresentem desempenhos estatisticamente similares em muitos cenários, há padrões de agrupamento que sugerem uma influência significativa das estratégias de representação das séries, da escolha das características dos nós e dos modelos de GNN usados. Modelos que compartilham arquiteturas ou abordagens de extração de características semelhantes, como SimTSC-GCN e SimTSC-GNA, ou VisibilityGraph-3feat e VisibilityGraph-2feat, tendem a exibir comportamentos semelhantes.

A análise de concordância revelou que a variação na forma como as séries temporais são convertidas em grafos e as características inerentes do conjunto de dados impactam diretamente a consistência das classificações. Modelos que utilizam representações estruturais similares obtêm maior concordância entre si, enquanto aqueles com abordagens distintas demonstram divergências maiores, especialmente em séries temporais mais complexas ou ruidosas.

Outro ponto relevante é o impacto da GNN no desempenho dos modelos. Os experimentos indicaram que a GraphSAGE, utilizada na GNA, supera consistentemente a GCN, como evidenciado pela performance superior do SimTSC-GNA em comparação ao SimTSC-GCN (Figura 5.3), sugerindo que a indutividade do GraphSAGE e sua estratégia de amostragem de vizinhança contribuem para uma melhor generalização a partir do mesmo grafo.

Para corroborar a importância da representação utilizada para gerar o grafo, e entender como elas respondem individualmente às características dos dados, apresentamos na Figura 5.5 a correlação de Spearman entre as características extraídas das séries temporais (Tabela 5.3) e o F1-score médio obtido por cada modelo (Tabela 5.5).

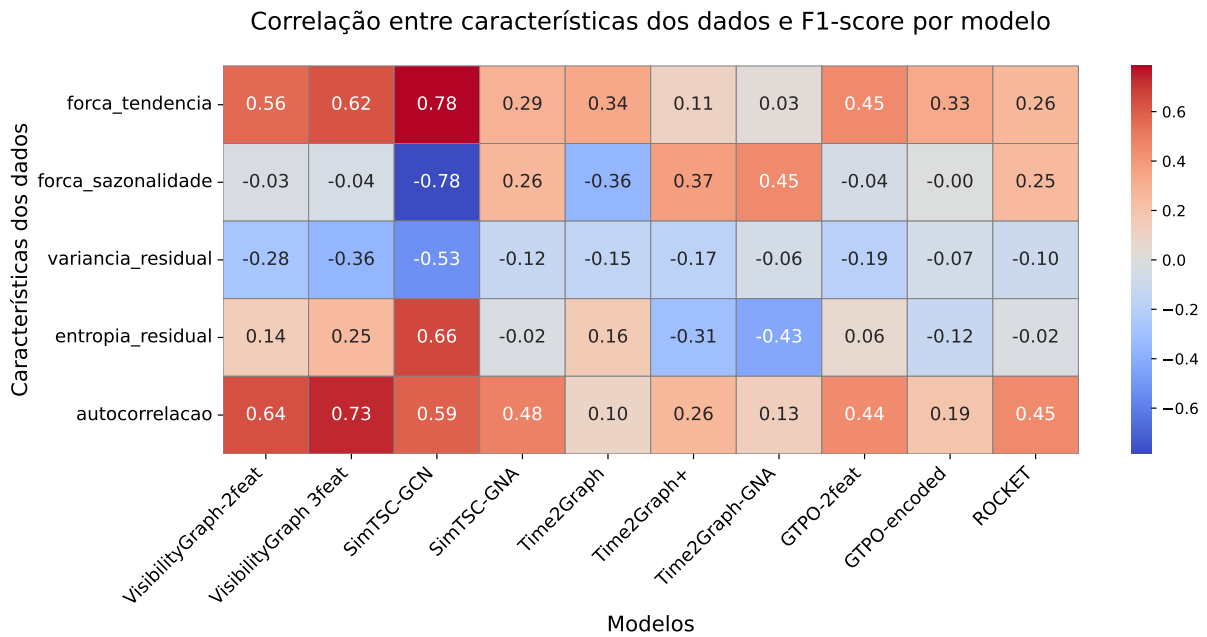


Figura 5.5: Correlação de Spearman entre características dos dados e F1-score por modelo.

A partir dos valores de correlação apresentados acima, podemos interpretar o seguinte sobre as arquiteturas:

- **SimTSC-GCN** é altamente sensível à força da tendência, mas sofre com sazonalidade e entropia;
- **VG-2feat** e **VG-3feat** se beneficiam de autocorrelação elevada e padrões determinísticos;
- **Time2Graph+** e **GNA** respondem bem à sazonalidade, capturando padrões periódicos;
- **GTPO** é mais influenciado pela tendência e autocorrelação, sendo mais evidente em sua versão com 2 características de nó;
- **ROCKET** apresenta baixo grau de sensibilidade, com desempenho robusto e estável.

Estes padrões observados são coerentes com a natureza arquitetural dos modelos:

- *Time2Graph* gera o grafo a partir de extração de Shapelets temporais e tende a ser robusto a ruído;
- *SimTSC* usa distanciamento DTW entre os conjuntos de dados e é sensível à forma global da série;
- *VisibilityGraph* preserva a geometria da série ao representar sua estrutura em grafos, capturando bem padrões como periodicidade;

- *GTPO* ao capturar padrões ordinais numa série, essa RSTG consegue codificar padrões como tendência e relacionamentos entre séries temporais no grafo; ou tendências correlacionadas, de uma série temporal
- *ROCKET* utiliza convoluções unidimensionais randômicas, o que proporciona generalização ampla e robustez.

Dando continuidade à análise da influência da estrutura das séries temporais sobre o comportamento dos modelos, buscamos agora identificar como as características extraídas das séries (Seção 5.1.3) se correlacionam entre si. Para isso, a princípio utilizamos Correlação de Pearson [21] entre os valores de características para cada conjunto de dados.

No entanto, essa métrica mostrou-se insuficiente para derivar uma análise estatisticamente justa, devido à sua sensibilidade a outliers e limitação à detecção de dependências estritamente lineares, o que restringe a capacidade de identificar padrões estruturais mais complexos entre as variáveis [23, 42].

Como alternativa, adotamos o coeficiente de correlação de Spearman, que consiste na aplicação da correlação de Pearson sobre os ranks das variáveis Spearman [78]. Ao operar sobre a ordenação dos dados, o coeficiente avalia associações monotônicas e reduz a influência de valores extremos, ampliando a capacidade de capturar relações estruturais não lineares [66].

Por exemplo, a correlação entre *autocorrelação* e *entropia residual* aumentou de $r = 0,44$ (Pearson) para $\rho = 0,72$ (Spearman), evidenciando uma relação que estava atenuada na análise linear. Esse tipo de reforço na associação indica que Spearman fornece uma base mais apropriada para análise estrutural e menos suscetível a distorções causadas por ruídos ou distribuições assimétricas.

A Figura 5.6 apresenta, para fins comparativos, as matrizes de correlação de **Pearson** (à esquerda) e **Spearman** (à direita), aplicadas às características estruturais das séries temporais, conforme descritas na Tabela 5.3.

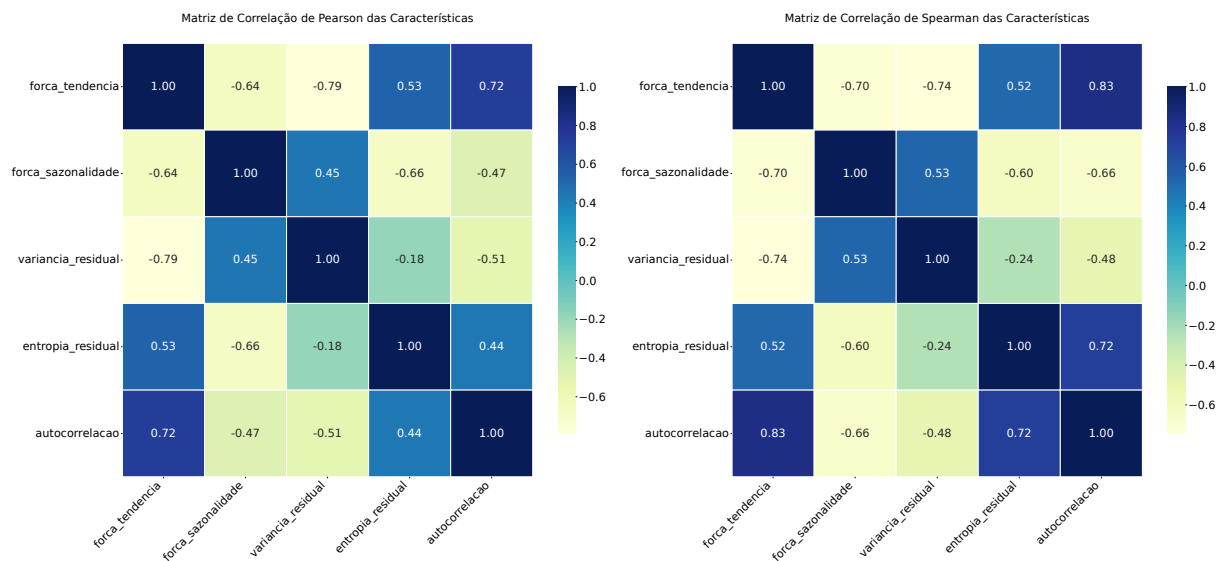


Figura 5.6: Comparação entre as matrizes de correlação de Pearson (esquerda) e Spearman (direita) para as características estruturais das séries temporais. Observa-se maior consistência e intensidade das correlações na matriz de Spearman, como entre autocorrelação e tendência ($\rho = 0,83$) e entre autocorrelação e sazonalidade ($\rho = -0,66$).

A matriz de Spearman demonstra maior robustez na identificação das relações entre as variáveis, sobretudo em contextos com não linearidades ou presença de outliers. Embora haja variações nos valores absolutos, as direções e tendências das correlações não apenas se mantêm, mas se tornam mais pronunciadas em relação à matriz de Pearson, evidenciando de forma mais clara as associações estruturais entre as variáveis.

Esses padrões justificam a adoção do coeficiente de Spearman como ponto de partida para a nossa análise entre estrutura e desempenho, fundamentando as recomendações de modelos por tipo de série temporal que discutimos a seguir.

5.6.1 Recomendação de Modelos por Tipo de Série Temporal

Para recomendar os modelos ideais por tipo de série, organizamos os 16 conjuntos de dados em seis categorias, anotadas no UCRArchive2018: **SENSOR**, **IMAGE**, **ECG**, **DEVICE**, **SPECTRO**, **AUDIO** e **CODE** que identifica os dados do CODE-15%. A figura 5.7 apresenta um gráfico radar com os valores médios de *macro F1-score* dos modelos (Tabela 5.5), agrupados por tipo de série temporal.

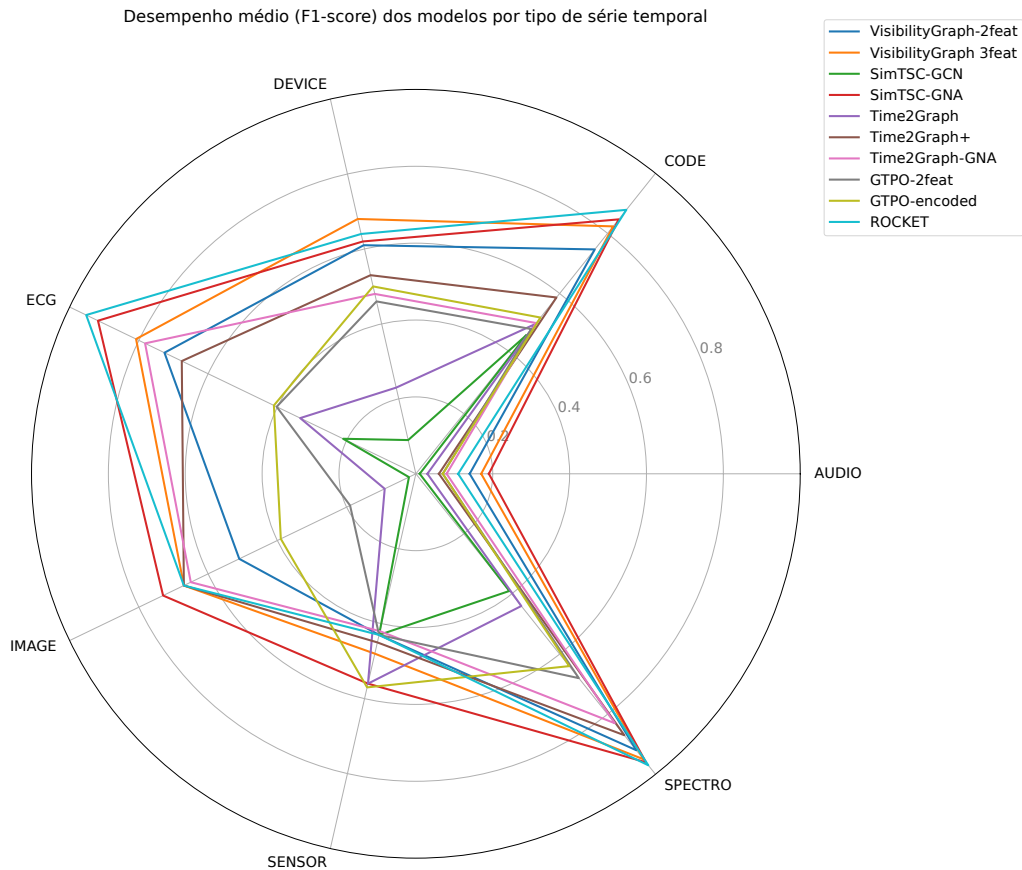


Figura 5.7: Gráfico radar com os valores médios de *macro F1-score* dos modelos por tipo de série temporal. A análise abrange os domínios de cada conjunto de dados, permitindo uma visualização comparativa da performance média de cada modelo.

Essa visualização permite comparar rapidamente o desempenho relativo das abordagens entre as categorias: observa-se que modelos como o **ROCKET** apresentam forte especialização em séries do tipo **ECG**, **CODE** e **SPECTRO**, enquanto o **SimTSC-GNA** demonstra desempenho consistente e competitivo nos mesmos tipos, sendo superior em **AUDIO** e **IMAGE**. Já abordagens baseadas em grafos de visibilidade, como o **VisibilityGraph-3feat**, destacam-se especialmente em **DEVICE** e **SPECTRO**, com destaque para o **GTPO-encoded** que apresenta desempenho ligeiramente superior em **SENSOR**.

Para compreender as razões subjacentes a essas diferenças de desempenho, analisamos as características estruturais médias de cada tipo de série temporal. A Tabela 5.8 apresenta essas métricas — incluindo tendência (trd), sazonalidade (s), variância residual (σ^2), entropia residual (H) e autocorrelação (ρ) — permitindo uma associação direta entre a morfologia dos sinais e os modelos mais adequados para cada domínio.

Tabela 5.8: Média das características extraídas dos conjuntos de séries temporais, destacando Força da Tendência (trd), Força da Sazonalidade (s), Variância Residual (σ^2), Entropia Residual (H) e Autocorrelação (ρ). Valores em destaque indicam a maior pontuação em cada métrica.

Tipo de dado	trd	s	σ^2	H	ρ
AUDIO	0,046361	0,100193	0,895269	6,556296	0,815660
CODE	0,931485	0,001711	0,006316	7,339803	0,986768
DEVICE	0,165958	0,301823	0,624670	4,079279	0,303687
ECG	0,484764	0,234511	0,426600	5,652425	0,967148
IMAGE	0,612153	0,182481	0,005636	3,486818	0,784319
SENSOR	0,070476	0,039179	0,893438	5,815661	0,047477
SPECTRO	0,652759	0,257617	0,489633	5,105436	0,993149

A partir dessa tabela e, voltando à matriz de Spearman da Figura 5.6, destacamos algumas observações importantes:

- **Força da tendência** (trd) e **autocorrelação** (ρ) apresentam alta correlação positiva ($\rho = 0,83$), indicando que séries com tendência bem definida também mantêm forte continuidade temporal. Esse padrão favorece modelos que exploram relações sequenciais, como o *SimTSC-GNA*, que obteve bom desempenho em séries dos tipos **IMAGE** e **SPECTRO**;
- **Força da tendência** (trd) e **variância residual** (σ^2) estão fortemente correlacionadas negativamente ($\rho = -0,74$). Séries com tendência marcada tendem a apresentar menor variabilidade nos resíduos, indicando maior previsibilidade. Esse perfil estrutural está alinhado ao alto desempenho do *ROCKET* em séries do tipo **ECG**, **CODE** e **SPECTRO** caracterizadas por padrões bem definidos e baixa variância residual;
- **Força da sazonalidade** (S) exibe correlação positiva com a **variância residual** (σ^2) ($\rho = 0,53$), sugerindo que séries com forte componente sazonal tendem a gerar resíduos mais dispersos. No entanto, quando essa sazonalidade é acompanhada de baixa entropia, há ganho para modelos baseados em grafos de visibilidade, como o *VisibilityGraph-3feat*, que se destaca em domínios como **DEVICE**, que apresenta alta sazonalidade;
- **Autocorrelação** (ρ) apresenta correlação negativa com a **variância residual** (σ^2) ($\rho = -0,48$), indicando que séries com alta autocorrelação, como o **CODE** tendem a apresentar resíduos mais estáveis. Isso favorece modelos que dependem de continuidade temporal e usam ResNet para aprender características, como o *ROCKET* e *SimTSC-GNA*;

- **Entropia residual** (H) e **autocorrelação** (ρ) possuem correlação positiva significativa ($\rho = 0,72$), revelando que séries com maior autocorrelação produzem resíduos menos caóticos e mais previsíveis. Modelos como o *SimTSC-GNA* se beneficiam dessa estrutura.

Com base na análise conjunta do desempenho empírico dos modelos (Figura 5.7) e das características estruturais médias das séries temporais (Tabela 5.8), foi possível identificar padrões que orientam a escolha dos modelos mais adequados para cada tipo de dado.

A Tabela 5.9 sintetiza essas observações ao apresentar uma recomendação dos modelos mais promissores por categoria de série temporal, acompanhada de justificativas baseadas nas métricas extraídas e nos resultados de classificação. Essa consolidação visa apoiar decisões mais embasadas na seleção de arquiteturas para tarefas de classificação de séries temporais representadas como grafos.

Tabela 5.9: Recomendações de modelos por tipo de série temporal, com base nas características estruturais e no desempenho empírico.

Tipo	Modelos	Justificativa
SENSOR	GTPO-encoded, SimTSC-GNA, Time2Graph	O GTPO-encoded obteve o melhor desempenho no conjunto Earthquakes (F1-score = 0,57), seguido por SimTSC-GNA e Time2Graph (F1-score = 0,56). Essas séries apresentam baixa autocorrelação ($\rho = 0,0475$) e alta variância residual ($\sigma^2 = 0,8934$), exigindo modelos robustos à ausência de padrões temporais regulares.
IMAGE	SimTSC-GNA	Melhor desempenho no conjunto Crop (F1-score = 0,73), superando o ROCKET (F1-score = 0,67). A presença de forte tendência ($trd = 0,6122$) e autocorrelação elevada ($\rho = 0,7843$) favorece modelos baseados em GNNs, como o SimTSC-GNA.
ECG e CODE	ROCKET	O ROCKET foi superior nos conjuntos ST (F1-score = 0,94), ECGFiveDays (F1-score = 1,00) e TwoLeadECG (F1-score = 1,00). Séries do tipo ECG possuem alta autocorrelação ($\rho = 0,9671$) e variância residual moderada ($\sigma^2 = 0,4266$), enquanto séries CODE apresentam tendência extrema ($trd = 0,9315$) e ruído mínimo ($\sigma^2 = 0,0063$), favorecendo arquiteturas convolucionais de alta capacidade.
DEVICE	VisibilityGraph	Apresentou o melhor desempenho no conjunto ElectricDevices (F1-score = 0,68), superando o ROCKET (F1-score = 0,64). A presença de ciclos e sazonalidade ($s = 0,3018$), aliada à variância residual elevada ($\sigma^2 = 0,6247$), beneficia abordagens baseadas em grafos de visibilidade.
SPECTRO	ROCKET, SimTSC-GNA, VisibilityGraph	As séries do tipo SPECTRO apresentam tendência elevada ($trd = 0,6528$) e autocorrelação máxima ($\rho = 0,9931$). No conjunto Strawberry , os três modelos apresentaram desempenho semelhante (F1-scores entre 0,95 e 0,97). Como não houve diferença estatisticamente significativa, entre os modelos recomenda-se o uso intercambiável dessas abordagens, que capturam bem padrões espectrais regulares.
AUDIO	SimTSC-GNA	Melhor desempenho no conjunto Phoneme (F1-score = 0,19), composto por séries acústicas ruidosas e multiclasse. A similaridade por alinhamento da DTW e a extração de características de nó através da ResNet auxiliar da SimTSC-GNA foi mais eficaz do que abordagens baseadas em convoluções ou grafos de visibilidade em um cenário altamente entrópico ($H = 6,5563$), com variância residual elevada ($\sigma^2 = 0,8953$).

Capítulo 6

Conclusão

Nesta dissertação, investigamos diferentes abordagens para a classificação de séries temporais representadas como grafos, realizando uma análise estatística comparativa entre diferentes modelos.

O trabalho foi guiado pelas perguntas de pesquisa levantadas no Capítulo 1, com foco em identificar os modelos mais eficazes e avaliar a sua concordância e desempenho, explorando a influência de diferentes arquiteturas e conjuntos de dados.

6.1 Resultados Principais

Os experimentos detalhados no Capítulo 5 forneceram respostas claras às perguntas de pesquisa e trouxeram revelações relevantes:

- **Quais são os principais modelos para CSTG?** Identificamos e avaliamos diversos modelos para CSTG, incluindo SimTSC, Time2Graph, GTPO e Grafos de Visibilidade, além de variações implementadas na arquitetura Graph Neural Alchemist (GNA). Esses modelos combinam técnicas de representação de séries temporais como grafos com diferentes arquiteturas de GNNs, como GCN, GraphSAGE e GAT.
- **Como esses modelos se comparam em termos de concordância e desempenho em diferentes conjuntos de dados?** A análise de Kappa de Fleiss revelou que a concordância entre os modelos varia de leve a moderada, indicando que diferentes abordagens de aprendizado de grafos produzem previsões independentes. No entanto, foi observado um agrupamento entre modelos que compartilham estratégias de aprendizado similares, como ROCKET e SimTSC-GNA.
- **Em comparação a um modelo estado da arte baseado em convoluções, como os modelos de CSTG performam?** O ROCKET, modelo baseado em

convoluções, obteve o melhor desempenho médio, superando a maioria dos modelos de CSTG em termos de F1-score. No entanto, em casos específicos, como séries altamente autocorrelacionadas ou com padrões sazonais bem definidos, modelos baseados em GNNs e grafos de visibilidade demonstraram desempenho competitivo.

- **Há diferenças estatisticamente significativas entre os diferentes modelos?** O Teste de Friedman com post-hoc de Nemenyi indicou que não há diferenças estatisticamente significativas entre muitos dos modelos avaliados, como foco para o ROCKET, SimTSC-GNA e VisibilityGraph que apresentaram um desempenho superior na maioria dos conjuntos de dados.
- **A escolha da abordagem de RSTG influencia no desempenho dos classificadores?** Sim. A análise mostrou que a seleção das características dos nós e a arquitetura da GNN impactam diretamente o desempenho dos classificadores. Modelos como SimTSC-GNA e VisibilityGraph-3feat, que combinam técnicas avançadas de agregação de informações em grafos, demonstraram superioridade em relação a outras abordagens, como GTPO e Time2Graph.
- **Os conjuntos de dados e suas características intrínsecas influenciam o desempenho dos classificadores?** Sim. A análise das características das séries temporais revelou que fatores como força da tendência, autocorrelação e variância residual estão fortemente correlacionados com a performance dos modelos. Modelos baseados em GNNs foram mais eficazes em séries com forte continuidade temporal, enquanto modelos baseados em convoluções se destacaram em séries mais estruturadas, como ECG.

6.2 Desafios e Limitações

O uso de representações de grafos para modelar séries temporais apresentou desafios notáveis, especialmente no que diz respeito à complexidade computacional dos métodos baseados em Shapelets e métricas de distância, bem como à dificuldade de generalização em conjuntos de dados ruidosos. Além disso, a ausência de precedentes para a aplicação de GNNs em GTPO exigiu decisões metodológicas que podem influenciar o desempenho dos modelos. A seguir, discutimos essas limitações em detalhes.

- **Complexidade Computacional:** A implementação de modelos como o *Time2Graph* revelou um alto custo computacional, principalmente devido à necessidade de ajustar

múltiplos hiperparâmetros, como o número de Shapelets (K), os fatores de atenção local w e global u , e a taxa de aprendizado lr para a otimização dos Shapelets. A calibração desses parâmetros varia significativamente entre os conjuntos de dados, exigindo uma busca extensa que tornou o treinamento do modelo particularmente oneroso. Para garantir consistência nos experimentos, optamos por utilizar os valores padrão propostos pelos autores, reduzindo a necessidade de ajustes manuais extensivos.

Outro fator crítico foi o alto custo computacional do aprendizado dos Shapelets Cômicos no Tempo, que tornou o treinamento do Time2Graph consideravelmente mais lento. Para mitigar esse impacto, armazenamos os Shapelets aprendidos e reutilizamos essa informação no Time2Graph+ e Time2Graph-GNA, reduzindo significativamente o tempo de treinamento. No entanto, essa abordagem pode ter limitado a capacidade de generalização do modelo, uma vez que os Shapelets não foram re-otimizados para cada arquitetura.

A escolha do valor de D na Simbolização de Bandt-Pompe para o *GTPO* também representou um desafio. Embora valores maiores de D possam capturar mais informações estruturais das séries e melhorar as métricas de classificação, eles aumentam substancialmente a densidade do grafo e o consumo de memória, tornando a abordagem inviável para conjuntos de dados muito grandes. Optamos por utilizar $D = 5$, um valor intermediário dentro do intervalo recomendado pela literatura [3], balanceando a expressividade do modelo com a viabilidade computacional.

- **Impacto do Ruído nos Dados:** A variabilidade nos conjuntos de dados influenciou diretamente a performance dos modelos. Séries com alta variância residual (σ^2) e entropia residual (H), como *Phoneme* e *Earthquakes*, apresentaram menor concordância entre os modelos e um desempenho global inferior, pois a presença de ruído dificulta a extração de padrões estruturados. Em contrapartida, séries com forte tendência e alta autocorrelação, como *Strawberry*, favoreceram a generalização dos modelos e apresentaram maior consistência entre as abordagens.

Essa discrepância evidencia a sensibilidade das representações baseadas em grafos às características intrínsecas das séries temporais. Modelos como *SimTSC-GNA* e *VisibilityGraph-3feat* demonstraram maior robustez em séries estruturadas, enquanto métodos como *GTPO-encoded* e *Time2Graph* tiveram desempenho mais instável em séries ruidosas, reforçando a necessidade de estratégias que mitiguem a influência do ruído na modelagem.

- **Falta de precedentes para a classificação de GTPO com GNNs:** Até o momento, não encontramos estudos na literatura que explorem a aplicação de GNNs em grafos do tipo GTPO, tornando este um trabalho pioneiro nessa abordagem.

Uma das principais dificuldades enfrentadas foi a ausência de atributos diretamente associados aos nós na modelagem tradicional do GTPO, o que limita as informações disponíveis para o aprendizado da rede neural. A sensibilidade entre as estratégias GTPO-encoded e GTPO-2feat adotadas neste estudo destaca a importância da escolha de características adequadas para a representação dos nós, evidenciando a necessidade de investigações futuras que visem uma seleção mais fundamentada de atributos para esse tipo de grafo.

6.3 Considerações Finais e Direções Futuras

Este trabalho contribuiu para o avanço do estado da arte na classificação de séries temporais representadas como grafos, demonstrando a viabilidade e eficácia das abordagens propostas. Por meio de uma análise comparativa abrangente, exploramos diferentes estratégias de representação e aprendizado para séries temporais em grafos, destacando as vantagens e limitações de cada uma.

A recomendação de modelos por tipo de série temporal apresentada neste estudo tem implicações práticas relevantes, auxiliando pesquisadores e profissionais na escolha das abordagens mais adequadas para diferentes domínios de aplicação. Observamos que modelos como ROCKET, SimTSC-GNA e VisibilityGraph-3feat exibiram desempenhos distintos dependendo das características estruturais dos dados, reforçando a necessidade de considerar fatores como autocorrelação, tendência e sazonalidade na seleção de classificadores. Essa análise direcionada otimiza o desempenho dos modelos e aprimora técnicas de aprendizado baseadas em grafos, tornando sua aplicação mais eficaz em cenários do mundo real.

Apesar dos avanços alcançados, diversas direções podem ser exploradas para aprofundar e expandir os resultados obtidos:

- **Séries temporais multivariadas:** Expandir a análise para séries multivariadas, incorporando múltiplas dimensões temporais em multi-grafos. Essa abordagem poderia capturar interações mais complexas entre variáveis, permitindo uma modelagem mais rica e realista de fenômenos temporais;
- **Representações heterogêneas:** Investigar o uso de grafos heterogêneos, que permitem modelar diferentes tipos de nós e arestas, tornando possível capturar interações mais sofisticadas entre variáveis, eventos ou padrões detectados na série.

Essa abordagem seria particularmente relevante para aplicações multimodais, como análise de séries temporais combinadas com dados espaciais ou textuais;

- **Avaliação em ambientes reais:** Testar os modelos em ambientes de produção, avaliando sua generalização para novos dados e explorando estratégias para mitigar potenciais vieses. A análise da adaptabilidade dos modelos a cenários dinâmicos, como séries temporais em tempo real, também representa uma importante linha de pesquisa;
- **Novas aplicações:** Investigar o uso de representações de séries temporais como grafos em outras tarefas de aprendizado de máquina, como previsão de links, detecção de anomalias e agrupamento. Essas aplicações podem ampliar significativamente o escopo e a utilidade das técnicas exploradas neste trabalho.

Os resultados obtidos nesta pesquisa estabelecem uma base sólida para futuras investigações e aprimoramentos no campo da modelagem de séries temporais como grafos. Com a evolução contínua das técnicas de aprendizado em grafos e o desenvolvimento de arquiteturas mais eficientes, espera-se que novas abordagens ampliem ainda mais a aplicabilidade dessas representações, permitindo avanços significativos em áreas como saúde, finanças e engenharia.

Referências

- [1] K. Ali Abd Al-Hameed. Spearman’s correlation coefficient in statistical analysis. *International Journal of Nonlinear Analysis and Applications*, 13(1):3249–3255, 2022.
- [2] 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*, 31:606–660, 2017. doi: 10.1007/s10618-016-0483-9.
- [3] C. Bandt and B. Pompe. Permutation entropy: A natural complexity measure for time series. *Physical Review Letters*, 88(17):174102, 2002. doi: 10.1103/PhysRevLett.88.174102.
- [4] 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*, 362:124554, 2019. doi: 10.1016/j.amc.2019.124554.
- [5] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015. doi: 10.1002/9781118675021.
- [6] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. doi: 10.1109/MSP.2017.2693418.
- [7] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021. doi: 10.48550/arXiv.2104.13478.
- [8] I. Cardoso-Pereira, J. B. Borges, P. H. Barros, A. A. F. Loureiro, and H. S. Ramos. Classificação de séries temporais através da probabilidade de auto-transição obtida do grafo de transição de padrões ordinais. In *Anais do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 622–635. SBC, 2019.
- [9] L. C. Carpi, T. A. Schieber, P. M. Pardalos, G. Marfany, C. Masoller, A. Diaz-Guilera, and M. G. Ravetti. Assessing diversity in multiplex networks. *Scientific Reports*, 9(1):4511, 2019. doi: 10.1038/s41598-019-40952-3.

- [10] E. Chagas, J. Borges, and H. Ramos. Uso de padrões ordinais na caracterização e análise de ataques de botnets em internet das coisas (iot). In *Anais Estendidos do XXVIII Simpósio Brasileiro de Sistemas Multimídia e Web*, pages 133–137, Porto Alegre, RS, Brasil, 2022. SBC. doi: 10.5753/webmedia_estendido.2022.224372.
- [11] C. Chatfield and H. Xing. *The Analysis of Time Series: An Introduction with R*. Chapman and Hall/CRC, 2019.
- [12] H. Chen and H. Eldardiry. Graph time-series modeling in deep learning: A survey. *ACM Transactions on Knowledge Discovery from Data*, 18(5):1–35, 2024. doi: 10.1145/3570519.
- [13] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794. ACM, 2016. doi: 10.1145/2939672.2939785.
- [14] L. Cheng, P. Zhu, W. Sun, Z. Han, K. Tang, and X. Cui. Time series classification by euclidean distance-based visibility graph. *Physica A: Statistical Mechanics and its Applications*, 625:129010, 2023. doi: 10.1016/j.physa.2022.129010.
- [15] 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*, volume 34, pages 3617–3624, 2020. doi: 10.1609/aaai.v34i04.5769.
- [16] 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*, 35(2):2078–2090, 2023. doi: 10.1109/TKDE.2021.3094908.
- [17] K. W. Church. Word2vec. *Natural Language Engineering*, 23(1):155–162, 2017. doi: 10.1017/S1351324916000334.
- [18] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning. Stl: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1): 3–73, 1990.
- [19] P. Coelho, R. Araujo, L. Ramos, S. Saliba, and R. Vimieiro. Graph neural alchemist: An innovative fully modular architecture for time series-to-graph classification. *arXiv preprint*, 2024.
- [20] P. Coelho, S. Saliba, L. Ramos, and R. Vimieiro. Seleção de features para classificação de ecg: Análise de novo método baseado em diversidade em grafos de visibilidade.

- In *Anais da IX Escola Regional de Computação Aplicada à Saúde (ERCAS 2024)*, Brasil, 2024. SBC. doi: 10.5753/ercas.2024.238705.
- [21] I. Cohen, Y. Huang, J. Chen, J. Benesty, J. Benesty, J. Chen, Y. Huang, and I. Cohen. Pearson correlation coefficient. *Noise reduction in speech processing*, pages 1–4, 2009.
- [22] Dau, H. Anh, Keogh, Eamonn, Kamgar, Kaveh, Yeh, C.-C. Michael, Zhu, Yan, Gharghabi, Shaghayegh, Ratanamahatana, C. Ann, Yanping, Hu, Bing, Begum, Nurjahan, Bagnall, Anthony, Mueen, Abdullah, Batista, Gustavo, and Hexagon-ML. The ucr time series classification archive, October 2018. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.
- [23] J. C. De Winter, S. D. Gosling, and J. Potter. Comparing the pearson and spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. *Psychological methods*, 21(3):273, 2016.
- [24] 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*, 34(5):1454–1495, 2020. doi: 10.1007/s10618-020-00701-z.
- [25] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.
- [26] J. Durbin and S. J. Koopman. *Time Series Analysis by State Space Methods*. Oxford University Press, 2012. ISBN 9780199641178. doi: 10.1093/acprof:oso/9780199641178.001.0001.
- [27] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971. doi: 10.1037/h0031619.
- [28] M. A. Freitas Júnior. Extração e seleção de características para a classificação eficiente de séries temporais. *Dissertação de Mestrado, Universidade Federal de Uberlândia*, 2022. doi: 10.14393/ufu.di.2022.53. URL <https://repositorio.ufu.br/handle/123456789/35084>.
- [29] M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937. doi: 10.1080/01621459.1937.10503522.
- [30] M. Friedman. A correction: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 34(205):109–109, 1939. doi: 10.1080/01621459.1939.10503536.

- [31] M. Friedman. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92, 1940. doi: 10.1214/aoms/1177731944.
- [32] B. D. Fulcher. Feature-based time-series analysis. In G. Zheng, S. Zhu, and V. S. Tseng, editors, *Feature Engineering for Machine Learning and Data Analytics*, pages 87–116. CRC Press, 2018. doi: 10.1201/9781315182787-5.
- [33] B. D. Fulcher and N. S. Jones. Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):3026–3037, 2014. doi: 10.1109/TKDE.2014.2316504.
- [34] S. García and F. Herrera. An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.
- [35] 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*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.
- [36] O. Gold and M. Sharir. Dynamic time warping and geometric edit distance: Breaking the quadratic barrier. *ACM Transactions on Algorithms (TALG)*, 14(4):1–17, 2018. doi: 10.1145/3232334.
- [37] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme. Learning time-series shapelets. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 392–401, 2014. doi: 10.1145/2623330.2623613.
- [38] W. L. Hamilton. *Graph Representation Learning*, volume 14. Morgan and Claypool.
- [39] 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, NIPS'17*, page 1025–1035, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.
- [40] W. L. Hamilton, R. Ying, and J. Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Eng. Bull.*, 40:52–74, 2017. URL <https://api.semanticscholar.org/CorpusID:3215337>.
- [41] J. Han, J. Pei, and H. Tong. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2022.

- [42] J. Hauke and T. Kossowski. Comparison of values of pearson's and spearman's correlation coefficients on the same sets of data. *Quaestiones geographicae*, 30(2): 87–93, 2011.
- [43] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.
- [44] T. Henderson and B. D. Fulcher. An empirical evaluation of time-series feature sets. In *2021 International Conference on Data Mining Workshops (ICDMW)*, pages 1032–1038. IEEE, 2021. doi: 10.1109/ICDMW53433.2021.00135.
- [45] J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall. Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, 28(4): 851–881, 2014. doi: 10.1007/s10618-013-0322-1.
- [46] Z. Hu, Y. Fang, and L. Lin. Training graph neural networks by graphon estimation. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 5153–5162. IEEE, 2021. doi: 10.48550/arXiv.2109.01918.
- [47] 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*, 46(12):10466–10485, 2024. doi: 10.1109/TPAMI.2024.3443141.
- [48] E. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2005. doi: 10.1007/s10115-004-0154-9.
- [49] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint*, 2016.
- [50] M. K"oppen. The curse of dimensionality. In *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*, volume 1, pages 4–8, 2000.
- [51] L. Lacasa, B. Luque, F. Ballesteros, J. Luque, and J. C. Nu"no. From time series to complex networks: The visibility graph. volume 105, pages 4972–4975, 2008. doi: 10.1073/pnas.0709247105. URL <https://www.pnas.org/doi/abs/10.1073/pnas.0709247105>.
- [52] L. Lacasa, V. Nicosia, and V. Latora. Network structure of multivariate time series. *Scientific reports*, 5(1):1–9, 2015. doi: 10.1038/srep15508.

- [53] B. Luque, L. Lacasa, F. Ballesteros, and J. Luque. Horizontal visibility graphs: Exact results for random time series. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 80:046103, 10 2009. doi: 10.1103/PhysRevE.80.046103.
- [54] 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*, 25(5):053101, 2015. doi: 10.1063/1.4919075.
- [55] C. Merkwirth and T. Lengauer. Automatic generation of complementary descriptors with molecular graph networks. *Journal of chemical information and modeling*, 45(5):1159–1168, 2005. doi: 10.1021/ci049613b.
- [56] M. Middlehurst, P. Schäfer, and A. Bagnall. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. *Data Mining and Knowledge Discovery*, 38(4):1958–2031, 2024. doi:10.1007/s10618-024-01022-1.
- [57] N. Mohammadi Foumani, L. Miller, C. W. Tan, G. I. Webb, G. Forestier, and M. Salehi. Deep learning for time series classification and extrinsic regression: A current survey. *ACM Comput. Surv.*, 56(9), Apr. 2024. ISSN 0360-0300. doi: 10.1145/3649448. URL <https://doi.org/10.1145/3649448>.
- [58] M. Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.
- [59] P. B. Nemenyi. *Distribution-free multiple comparisons*. Princeton University, 1963.
- [60] W. T. Ng, K. Siu, A. C. Cheung, and M. K. Ng. Expressing multivariate time series as graphs with time series attention transformer. *arXiv preprint*, 2022. doi: 10.48550/arXiv.2208.0930.
- [61] V. Nicosia, L. Lacasa, and V. Latora. From multivariate time series to multiplex visibility graphs. *arXiv preprint*, 1408:5, 2014. doi: 10.48550/arXiv.1408.0005.
- [62] R. Oliveira, V. Freitas, G. Moreira, and E. Luz. Explorando redes neurais de grafos para classificação de arritmias. In *Proceedings of the 22nd Brazilian Symposium on Computing Applied to Health*, pages 178–189, Porto Alegre, RS, Brasil, 2022. SBC. doi: 10.5753/sbcas.2022.222510.
- [63] 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, page 701–710, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329569. doi: 10.1145/2623330.2623732. URL <https://doi.org/10.1145/2623330.2623732>.

- [64] A. A. B. Pessa and H. V. Ribeiro. ordpy: A python package for data analysis with permutation entropy and ordinal network methods. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(6):063110, 06 2021. ISSN 1054-1500. doi: 10.1063/5.0049901. URL <https://doi.org/10.1063/5.0049901>.
- [65] F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, and E. Keogh. Dynamic time warping averaging of time series allows faster and more accurate classification. In *2014 IEEE International Conference on Data Mining*, pages 470–479, 2014. doi: 10.1109/ICDM.2014.27.
- [66] A. Rebekić, Z. Lončarić, S. Petrović, and S. Marić. Pearson’s or spearman’s correlation coefficient-which one to use? *Poljoprivreda*, 21(2):47–54, 2015.
- [67] A. H. Ribeiro, M. H. Ribeiro, G. M. Paixão, D. M. Oliveira, P. R. Gomes, J. A. Canazart, M. P. Ferreira, C. R. Andersson, P. W. Macfarlane, W. Meira Jr, et al. Automatic diagnosis of the 12-lead ecg using a deep neural network. *Nature communications*, 11(1):1–9, 2020. doi: 10.1038/s41467-020-15432-4.
- [68] A. H. Ribeiro, G. Paixao, E. M. Lima, M. H. Ribeiro, M. M. Pinto Filho, P. R. Gomes, D. M. Oliveira, W. Meira Jr, T. B. Schon, and A. L. P. Ribeiro. Code-15%: A large scale annotated dataset of 12-lead ecgs, 2021.
- [69] A. L. P. Ribeiro, G. M. Paixão, P. R. Gomes, M. H. Ribeiro, A. H. Ribeiro, J. A. Canazart, D. M. Oliveira, M. P. Ferreira, E. M. Lima, J. L. de Moraes, N. Castro, L. B. Ribeiro, and P. W. Macfarlane. Tele-electrocardiography and bigdata: The code (clinical outcomes in digital electrocardiography) study. *Journal of Electrocardiology*, 57:S75–S78, 2019. ISSN 0022-0736. doi: <https://doi.org/10.1016/j.jelectrocard.2019.09.008>.
- [70] M. Riedl, A. Müller, and N. Wessel. Practical considerations of permutation entropy: A tutorial review. *The European Physical Journal Special Topics*, 222(2):249–262, 2013. doi: 10.1140/epjst/e2013-01862-7.
- [71] A. Romanova. Enhancing time series analysis with gnn graph classification models. In H. Cherifi, L. M. Rocha, C. Cherifi, and M. Donduran, editors, *Complex Networks & Their Applications XII*, pages 25–36, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-53468-3.
- [72] T. K. Rusch, M. M. Bronstein, and S. Mishra. A survey on oversmoothing in graph neural networks. *arXiv preprint*, 2023. doi: 10.48550/arXiv.2303.10993.
- [73] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.

- [74] S. Seabold and J. Perktold. *Statsmodels: Econometric and statistical modeling with python*. volume 2010, 01 2010.
- [75] R. H. Shumway and D. S. Stoffer. Arima models. pages 75–163, 2017. doi: 10.1007/978-3-319-52452-8_3. URL https://doi.org/10.1007/978-3-319-52452-8_3.
- [76] M. Small. Complex networks from time series: Capturing dynamics. In *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2509–2512, 2013. doi: 10.1109/ISCAS.2013.6572389.
- [77] L. N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. *arXiv preprint*, 2018. URL <https://api.semanticscholar.org/CorpusID:4714223>.
- [78] C. Spearman. The proof and measurement of association between two things. 1961.
- [79] Y. Sun, X. Yao, X. Bi, X. Huang, X. Zhao, and B. Qiao. Time-series graph network for sea surface temperature prediction. *Big Data Research*, 25:100237, 2021. ISSN 2214-5796. doi: <https://doi.org/10.1016/j.bdr.2021.100237>.
- [80] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. *arXiv preprint*, 2018. URL <https://arxiv.org/abs/1710.10903>.
- [81] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan. Inductive representation learning on temporal graphs. 2020. URL <https://arxiv.org/abs/2002.07962>.
- [82] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? *arXiv preprint*, 2019. URL <https://arxiv.org/abs/1810.00826>.
- [83] Y. L. Xu and D. P. Mandic. Recurrent graph tensor networks: A low-complexity framework for modelling high-dimensional multi-way sequences. *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 1795–1799, 2020. URL <https://api.semanticscholar.org/CorpusID:234365022>.
- [84] Q. Xuan, J. Zhou, K. Qiu, Z. Chen, D. Xu, S. Zheng, and X. Yang. Avgnet: Adaptive visibility graph neural network and its application in modulation classification. *IEEE Transactions on Network Science and Engineering*, 9(3):1516–1526, 2022. doi: 10.1109/TNSE.2022.3146836.
- [85] Z. Yang, M. Ding, C. Zhou, H. Yang, J. Zhou, and J. Tang. Understanding negative sampling in graph representation learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '20*, page 1666–1676, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403218.

- [86] L. Ye and E. Keogh. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, page 947–956, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584959. doi: 10.1145/1557019.1557122. URL <https://doi.org/10.1145/1557019.1557122>.
- [87] L. Ye and E. Keogh. Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. *Data mining and knowledge discovery*, 22:149–182, 2011. doi: 10.1007/s10618-010-0179-5.
- [88] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 4805–4815, Red Hook, NY, USA, 2018. Curran Associates Inc.
- [89] H. Zeng, M. Zhang, Y. Xia, A. Srivastava, A. Malevich, R. Kannan, V. Prasanna, L. Jin, and R. Chen. Decoupling the depth and scope of graph neural networks. *Arxiv preprint*, 2022. URL <https://arxiv.org/abs/2201.07858>.
- [90] D. Zha, K.-H. Lai, Z. Kaixiong, and X. Hu. Towards similarity-aware time-series classification. pages 199–207, 01 2022. doi: 10.1137/1.9781611977172.23.
- [91] Z. Zhang, J. Bu, M. Ester, J. Zhang, Z. Li, C. Yao, H. Dai, Z. Yu, and C. Wang. Hierarchical multi-view graph pooling with structure learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):545–559, 2023. doi: 10.1109/TKDE.2021.3090664.
- [92] P. Zhou, J. Feng, C. Ma, C. Xiong, S. Hoi, and E. Weinan. Towards theoretically understanding why sgd generalizes better than adam in deep learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.
- [93] T. Zhou, P. Niu, X. Wang, L. Sun, and R. Jin. One fits all: power general time series analysis by pretrained lm. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.