

Classificação de Séries Temporais Univariadas com Grafos de Visibilidade: Uma Abordagem Baseada em Extração de Características

Raul Araju Korogi Oliveira

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
raularaju@dcc.ufmg.br

Renato Vimieiro

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil

Abstract—O problema de classificação de séries temporais é recorrente em diversas áreas, como medicina, música e sismologia. Uma abordagem recente consiste em transformar séries temporais em grafos, extrair características desses grafos e, a partir delas, realizar a classificação. Nesse contexto, uma estratégia comum é o uso de Redes Neurais em Grafos (GNNs), que aprendem representações diretamente a partir da estrutura do grafo e realizam a classificação. Como alternativa, esta pesquisa propõe uma abordagem mais simples: transformar a série temporal em um grafo de visibilidade, extrair características globais e locais, construir um vetor de atributos com essas informações e utilizá-lo como entrada para modelos clássicos de aprendizado supervisionado. O objetivo geral da pesquisa — iniciada no Projeto Orientado à Computação (POC) I e a ser continuada na POC II — é realizar uma análise comparativa entre essas duas abordagens: (i) GNNs, que realizam aprendizado diretamente sobre os grafos, e (ii) extração de *features* do grafo combinada com classificadores tradicionais, avaliando o desempenho de ambas na tarefa de classificação de séries temporais.

Index Terms—Série Temporal, Grafo de Visibilidade, Classificação

I. INTRODUÇÃO

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

Entre os principais problemas associados a esse tipo de dado, destaca-se a **classificação de séries temporais**, cujo objetivo é atribuir rótulos predefinidos a sequências com base em sua forma ou dinâmica. Diversas abordagens têm sido propostas para essa tarefa, como métodos baseados em extração diretas de *features* da série [1], em distância [9], e em convoluções [6].

Recentemente, uma linha de pesquisa emergente tem investigado a transformação de séries temporais em grafos [13],

com o objetivo de capturar propriedades estruturais que não são evidentes diretamente na série original. Nessa abordagem, características do grafo são extraídas e utilizadas como base para a classificação. A maioria dos trabalhos nessa direção emprega Redes Neurais em Grafos (GNNs) para realizar o aprendizado dessas características e a classificação subsequente.

O objetivo deste projeto é realizar uma análise comparativa entre diferentes estratégias de representação de séries temporais como grafos e sua eficácia na tarefa de classificação. Especificamente, o Projeto Orientado à Computação (POC) I aborda uma abordagem mais simples: transformar as séries em Grafos de Visibilidade (GV) [15], extrair *features* globais (como diâmetro e número de arestas) e locais (como grau e métricas de centralidade), e utilizar essas *features* como entrada para classificadores tradicionais de aprendizado de máquina.

Já o POC II ampliará essa análise, comparando o desempenho dessa abordagem com métodos baseados em GNNs e outras técnicas do estado da arte, especialmente aquelas baseadas em convoluções. O objetivo geral é avaliar, de forma sistemática a eficácia de diferentes abordagens na classificação de séries temporais transformadas em grafos.

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

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

- 6) Qual é o desempenho dos métodos baseados em grafos em comparação com o estado da arte fundamentado em redes convolucionais e em distâncias?

Enquanto o trabalho desenvolvido no POC I focou-se em responder, no mínimo parcialmente, às duas primeiras perguntas de pesquisa, espera-se que as demais questões sejam abordadas e respondidas ao longo do POC II.

Este documento está organizado da seguinte forma:

- **Introdução:** apresenta o contexto, motivação e objetivos da pesquisa.
- **Seção II (Referencial Teórico):** discute os trabalhos relacionados que fundamentam esta pesquisa, abordando diferentes abordagens para o problema de classificação de séries temporais, com ênfase nas técnicas que inspiraram a proposta deste trabalho.
- **Seção III (Conceitos Básicos):** define os principais conceitos utilizados, incluindo a transformação de séries temporais em grafos e a extração de *features*, fundamentais para compreender a metodologia empregada.
- **Seção IV (Desenvolvimento):** descreve detalhadamente o processo de construção do trabalho, incluindo a seleção dos dados, transformação em grafos, extração das *features* e preparação para a classificação.
- **Seção V (Experimentos):** apresenta a metodologia experimental, detalhando os procedimentos de treino, teste e validação dos modelos.
- **Seção VI (Conclusão):** sintetiza os principais resultados obtidos, discute suas implicações e aponta direções para os próximos passos na continuidade do projeto, especialmente no POC II.

II. REFERENCIAL TEÓRICO

Há diferentes abordagens para tratar o problema de classificação de séries temporais. Uma classe de métodos amplamente utilizada é baseada na extração de características (*feature-based*), que transforma a série original em um vetor fixo de atributos extraídos por meio de funções estatísticas, transformadas no domínio do tempo ou da frequência, ou outras heurísticas [1], [2]. Estudos recentes demonstram que essa estratégia pode alcançar resultados competitivos com técnicas mais complexas [3].

Ainda nesse raciocínio, há métodos que se concentram em subsequências ou padrões discriminativos dentro das séries, como *shapelets* [4] ou ordens de permutação [5], que visam capturar dados locais representativos e úteis para a tarefa de classificação. Essas abordagens assumem que determinadas estruturas temporais curtas são suficientes para discriminar entre classes.

Com o avanço das redes neurais profundas, surgiram soluções baseadas em redes convolucionais unidimensionais (1D CNNs), que operam diretamente sobre os sinais temporais. Essas redes são capazes de aprender automaticamente representações latentes relevantes a partir dos dados, dispensando a extração manual de características. Modelos como o *ROCKET* e suas variantes [6] têm se destacado por aliar

desempenho competitivo a uma elevada eficiência computacional, mesmo em conjuntos de dados extensos.

Apesar de apresentarem desempenho promissor, os métodos baseados em extração de características podem ser sensíveis à chamada *maldição da dimensionalidade* [12]. À medida que o número de atributos extraídos cresce, o espaço de busca se torna exponencialmente maior, o que dificulta a discriminação das classes. Diante desse cenário, torna-se fundamental o desenvolvimento de estratégias eficazes de extração e seleção de características, de modo a garantir que apenas informações discriminativas e relevantes sejam mantidas no vetor final de representação.

Sob esse prisma, uma alternativa promissora é a identificação de estruturas intermediárias capazes de representar informações relevantes das séries temporais de forma mais compacta. Especificamente, grafos têm se mostrado eficazes na captura de relações de curto e longo alcance presentes nas séries temporais [13]. Essa representação permite modelar dependências temporais complexas como conexões estruturadas entre elementos, o que potencialmente facilita a extração de padrões latentes e oferece uma visão mais organizada da dinâmica temporal dos dados.

Entre as representações mais promissoras baseadas em grafos estão os *Grafos de Transição de Padrões Ordinários* (GTPOs) [14], [22] e os *Grafos de Visibilidade* (GVs) [15]. Ambas as abordagens convertem séries temporais em estruturas topológicas que preservam, de maneiras distintas, aspectos temporais e estruturais dos dados. Os GTPOs modelam as transições entre padrões de ordem local, capturando a dinâmica da série por meio de variações relativas entre elementos consecutivos. Já os GVs estabelecem conexões entre pontos da série com base em uma relação de visibilidade geométrica, permitindo representar tanto relações de curto quanto de longo alcance. Essas representações têm se mostrado eficazes na caracterização de complexidade, periodicidade e comportamento dinâmico de séries temporais em diferentes contextos.

Nesse contexto, Coelho realizou uma análise comparativa de diferentes formas de classificar séries temporais representadas como grafos em [8]. O autor avaliou diversas possibilidades de construção da seguinte *pipeline*: receber um conjunto de séries temporais univariadas, transformá-las em grafos, extrair características desses grafos e, por fim, utilizar essas características como entrada para modelos de classificação. O estudo considerou distintas alternativas para cada etapa da *pipeline*: transformação da série em grafo, extração de características e classificação.

Ainda nesse raciocínio, as abordagens apresentadas neste trabalho utilizam Redes Neurais em Grafos (GNNs) após a etapa de transformação, com o objetivo de realizar a classificação. Esse processo envolve a construção de uma nova arquitetura especializada para processar os grafos e capturar suas informações estruturais. Por outro lado, a abordagem alternativa apresentada neste trabalho consiste em extrair *features* diretamente dos grafos gerados — como o diâmetro, número de arestas, medidas de centralidade, entre outras —

e utilizar essas *features* como entrada para classificadores tradicionais de aprendizado de máquina, como *Random Forest* e *Gradient Boosting*.

Sendo assim, o foco deste projeto de pesquisa é incrementar a análise feita por [8], aumentando a robustez dos resultados refazendo a avaliação para o mesmo conjunto de dados e fazendo uso dessa nova abordagem baseada na extração de *features* de grafo e classificadores clássicos.

III. CONCEITOS BÁSICOS

A. Série Temporal

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

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

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

B. Grafo de Visibilidade

Um grafo, denotado por $G = (V, E)$, é composto por dois conjuntos: V e E . O conjunto V contém os nós (ou vértices), enquanto E representa o conjunto de arestas. Dois vértices u e v podem ser conectados por uma aresta $e = (u, v)$.

Neste trabalho, considera-se apenas grafos não direcionados e não ponderados, ou seja, as arestas não possuem direção nem peso associado.

A transformação de uma série temporal em um Grafo de Visibilidade Natural (GVN), proposta por Lacasa et al. [15], permite representar a série como um grafo não direcionado e não ponderado, de modo a capturar aspectos topológicos derivados da estrutura temporal dos dados. Existem variações dessa transformação, como o Grafo de Visibilidade Horizontal [16], mas este trabalho se concentrará no GVN. Portanto, o termo Grafo de Visibilidade (GV) será utilizado como sinônimo de Grafo de Visibilidade Natural (GVN).

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

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

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

$$y_c < y_b + (y_a - y_b) \cdot \frac{t_b - t_c}{t_b - t_a} \quad (1)$$

Essa equação garante que não há obstrução de visibilidade entre os pontos $A(t_a, y_a)$ e $B(t_b, y_b)$ da série. A construção do grafo resulta, por definição, em uma estrutura conectada, o que

é uma vantagem para a análise topológica. A versão utilizada neste trabalho segue a implementação otimizada proposta por Lan et al. [20], com complexidade $O(n \log n)$.

C. Features de Grafo

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

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

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

$$\text{diam}(G) = \max_{u, v \in V} d(u, v) \quad (2)$$

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

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

$$m = \frac{1}{n(n-1)} \sum_{s \neq t \in V} d(s, t) \quad (3)$$

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

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

Neste trabalho, todos os *motifs* com 3 nós foram contabilizados exaustivamente. No entanto, como a complexidade da contagem cresce exponencialmente com o número de nós, utilizou-se o algoritmo proposto por Wernicke e Rasche [21] para os *motifs* com 4 e 5 nós. Esse algoritmo utiliza amostragem para reduzir o custo computacional e fornecer uma estimativa eficiente da contagem.

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



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

5) *Grau*: O grau de um nó v , denotado por $d(v)$, é o número de arestas incidentes a esse nó, ou seja, a quantidade de seus vizinhos. Essa métrica indica a conectividade do nó e é um indicativo das suas características estruturais no grafo.

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

$$C_D(v) = \frac{d(v)}{n-1} \quad (4)$$

onde n é o número total de nós do grafo. Essa métrica representa a quantidade de vizinhos de um nó normalizada pelo número máximo possível de vizinhos. Trata-se de uma métrica altamente correlacionada com o grau, logo, para o POC II, é possível que se mantenha apenas o grau ou a centralidade de grau como uma *feature*.

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

$$C(u) = \frac{n-1}{\sum_{v \neq u} d(v,u)} \quad (5)$$

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

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

9) *Centralidade harmônica (harmonic centrality)*: A centralidade harmônica é uma variação da centralidade de proximidade que utiliza a soma dos inversos das distâncias do nó u para todos os outros nós, mesmo quando esses não são alcançáveis. É definida como:

$$H(u) = \sum_{v \neq u} \frac{1}{d(v,u)} \quad (6)$$

onde $d(v,u)$ é a distância do caminho mais curto entre v e u . Essa medida é altamente correlacionada com a centralidade de proximidade, sendo possível que, na próxima etapa da POC, apenas uma das duas métricas seja utilizada como *feature*.

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

$$C_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)} \quad (7)$$

onde V é o conjunto de nós, $\sigma(s,t)$ é o número total de caminhos mais curtos entre os nós s e t , e $\sigma(s,t|v)$ é o número desses caminhos que passam pelo nó v . Note-se que se $v \in \{s,t\}$, então $\sigma(s,t|v) = 0$, e se $v = t$, $\sigma(s,t) = 1$ [19].

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

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

$$C(v) = \frac{2 \times \text{número de arestas entre vizinhos de } v}{d(v) \times (d(v) - 1)} \quad (8)$$

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

IV. DESENVOLVIMENTO

O fluxo geral do trabalho realizado especificamente nesta POC segue os seguintes passos: primeiramente, foram escolhidas as bases de dados; em seguida, as séries temporais foram transformadas em grafos de visibilidade; posteriormente, foram computadas as *features* desses grafos, construindo-se as representações vetoriais dos grafos com base nessas *features*; por fim, essas representações foram utilizadas como entrada para os classificadores de aprendizado de máquina.

A. Escolha da Base de Dados

Foram utilizadas as mesmas 10 bases de dados de séries temporais univariadas presentes na análise comparativa de [8], oriundas do repositório UCRArchive 2018 [17]. As bases consideradas são: *Crop*, *Earthquakes*, *ECG200*, *ECGFiveDays*, *ElectricDevices*, *NonInvasiveFetalECGThorax1* (NIFECG-1), *NonInvasiveFetalECGThorax2* (NIFECG-2), *Phoneme*, *Strawberry* e *TwoLeadECG*.

A escolha dessas bases se justifica por representarem um conjunto diversificado em termos de desempenho dos classificadores em [8], [11], além de variarem quanto ao tamanho das séries, número de amostras, número de classes e domínio do dado (como sinais ECG, imagens, entre outros).

TABLE I
INFORMAÇÕES GERAIS SOBRE O CONJUNTO DE DADOS

Dataset	Tam. série	Nº séries	Nº classes	Tipo
Crop	46	24000	24	IMAGE
Earthquakes	512	461	2	SENSOR
ECG200	96	200	2	ECG
ECGFiveDays	136	884	2	ECG
ElectricDevices	96	16637	7	DEVICE
NIFECG-1	750	3765	42	ECG
NIFECG-2	750	3765	42	ECG
Phoneme	1024	2110	39	AUDIO
Strawberry	235	983	2	SPECTRO
TwoLeadECG	82	1162	2	ECG

B. Transformação da série em grafo

Todas as séries temporais de cada conjunto de dados são transformadas em grafos de acordo com a definição apresentada na Seção III-B. Essa representação foi escolhida por contar com uma implementação robusta e eficiente já existente, além de já ter sido demonstrado seu potencial para capturar características de dependência temporal das séries temporais, conforme mostrado em [7], [8].

C. Construção da representação do grafo

Usa-se as *features* definidas em III-C para construir uma representação significativa que permita que os modelos de aprendizado de máquina clássico discriminem as classes de maneira eficiente. No caso das *features* de grafo (diâmetro, número de arestas, caminho mínimo médio e contagem de *motifs*), elas são agregadas integralmente na representação do grafo, capturando propriedades estruturais globais relevantes. Como há 2 *motifs* de tamanho 3, 6 *motifs* de tamanho 4 e 21 *motifs* de tamanho 5, além de outras 3 *features* globais do grafo, tem-se um total de $2 + 6 + 21 + 3 = 32$ *features* de grafo.

Já as *features* locais (como grau, centralidades e coeficiente de agrupamento) fornecem informações sobre o papel de cada nó na estrutura do grafo. A escolha dessas *features* foi motivada por sua capacidade de capturar diferentes aspectos topológicos e funcionais dos grafos, balanceando simplicidade computacional com poder discriminativo para a tarefa de classificação. Esse grupo de *features* de nó não é agregado integralmente à representação do grafo. Para cada uma dessas

features, são computadas quatro métricas estatísticas: média, desvio padrão, máximo e mínimo. Essas métricas é que são incorporadas ao vetor de representação do grafo. Como há 8 *features* de nó, são agregadas $8 \times 4 = 32$ *features* relativas a esse grupo, totalizando $32 + 32 = 64$ *features* no total.

D. Classificação

O conjunto de dados nesta etapa consiste nas 64 *features* extraídas dos grafos derivados das séries temporais de cada *dataset*. Esses conjuntos de dados são então normalizados e utilizados como entrada para os classificadores: *Gradient Boosting*, *Random Forest*, *Support Vector Classifier (SVC)*, *K-Nearest Neighbors (KNN)*, *Decision Tree* e *Naive Bayes*. Esse conjunto de modelos se justifica pois abrange uma diversidade de paradigmas: árvores de decisão (*Random Forest*, *Gradient Boosting* e *Decision Tree*), distância (*KNN*), probabilísticos (*Naive Bayes*) e classificadores de margem (*SVC*). Além disso, todos esses classificadores já foram amplamente testados em dados tabulares, que é o caso para as 64 *features* de grafo extraídas, e, portanto, há implementação robusta e eficiente para todos esses métodos.

V. EXPERIMENTOS

Todos esses modelos escolhidos apresentam diversos hiperparâmetros, que podem impactar o desempenho do algoritmo. Para fazer uma análise mais robusta e justa entre os modelos, foi feita uma busca de hiperparâmetros para cada modelo em cada *dataset*. A busca se deu da seguinte forma: para cada *dataset*, fez-se uma amostra estratificada usando 30% das amostras. Em seguida, foi realizada uma busca por hiperparâmetros utilizando *Randomized Search* com 30 iterações. Ou seja, foram avaliados 30 conjuntos distintos de hiperparâmetros para cada modelo e para cada *dataset*. Cada conjunto foi treinado e testado por meio de validação cruzada com 5 *folds*. O conjunto de hiperparâmetros que obteve a maior média da métrica *F1-score* macro no conjunto de teste da validação cruzada foi selecionado como o melhor para aquele modelo e *dataset*.

Após definir a escolha ótima de hiperparâmetros, cada modelo foi retreinado em 80% dos dados e testado nos 20% restantes, utilizando esse conjunto de hiperparâmetros ótimos. Esse processo foi repetido 30 vezes para cada modelo em cada *dataset*, com diferentes partições de treino e teste, mantendo sempre a proporção 80-20. A média e o desvio padrão da métrica *F1-score* macro obtidos nessas 30 iterações foram calculados e os resultados agregados em II.

A repetição de treino e teste de um modelo para um mesmo *dataset* foi feita para garantir robustez nos resultados encontrados. Como o objetivo desta pesquisa é fazer uma análise comparativa entre diferentes abordagens, é fundamental que os resultados tenham significado estatístico; a repetição de treinamento e avaliação garante esse requisito.

Quanto à métrica de avaliação, foi utilizada a *F1-score* macro. Embora alguns dos *datasets* apresentem alto grau de desbalanceamento entre as classes, essa métrica não favorece

TABLE II
F1-SCORE MACRO MÉDIO E DESVIO PADRÃO PARA CADA MODELO E CONJUNTO DE DADOS

Dataset	GradientBoosting	RandomForest	SVC	KNN	DecisionTree	NaiveBayes
Crop	0.29 ± 0.01	0.42 ± 0.00	0.36 ± 0.00	0.21 ± 0.01	0.41 ± 0.00	0.39 ± 0.00
ECG200	0.64 ± 0.05	0.67 ± 0.04	0.65 ± 0.05	0.70 ± 0.03	0.69 ± 0.04	0.62 ± 0.08
ECGFiveDays	0.80 ± 0.02	0.89 ± 0.01	0.86 ± 0.02	0.81 ± 0.03	0.89 ± 0.01	0.92 ± 0.01
Earthquakes	0.54 ± 0.03	0.53 ± 0.02	0.51 ± 0.02	0.43 ± 0.04	0.49 ± 0.03	0.53 ± 0.03
ElectricDevices	0.59 ± 0.01	0.70 ± 0.00	0.66 ± 0.00	0.38 ± 0.02	0.69 ± 0.00	0.69 ± 0.00
NIFECG-1	0.39 ± 0.01	0.49 ± 0.01	0.46 ± 0.01	0.39 ± 0.02	0.55 ± 0.01	0.57 ± 0.01
NIFECG-2	0.50 ± 0.01	0.61 ± 0.01	0.60 ± 0.01	0.53 ± 0.01	0.68 ± 0.01	0.70 ± 0.01
Phoneme	0.12 ± 0.01	0.12 ± 0.01	0.12 ± 0.01	0.10 ± 0.01	0.15 ± 0.01	0.15 ± 0.01
Strawberry	0.79 ± 0.03	0.85 ± 0.01	0.85 ± 0.01	0.77 ± 0.01	0.84 ± 0.02	0.87 ± 0.01
TwoLeadECG	0.83 ± 0.02	0.90 ± 0.01	0.87 ± 0.02	0.81 ± 0.02	0.90 ± 0.01	0.94 ± 0.01

classes majoritárias, fornecendo uma avaliação mais equilibrada do desempenho dos modelos em todas as classes.

Ao observar os resultados, percebe-se uma grande variação de desempenho entre os diferentes *datasets*. Esse comportamento era esperado, uma vez que um dos critérios de escolha dessas bases foi justamente a diversidade de desempenho observada em [8], [11]. Também é possível notar uma variação menos expressiva entre os modelos dentro de um mesmo *dataset*. Por exemplo, no *dataset Phoneme*, o modelo com melhor desempenho obteve um *F1-score* de 0,15, enquanto o KNN alcançou 0,10 na mesma métrica — uma diferença de 0,05, considerada relativamente pequena. Já no *dataset ElectricDevices*, a diferença entre o maior e o menor *F1-score* observado foi de 0,32, evidenciando um contraste mais acentuado.

O *dataset* com pior desempenho geral foi o *Phoneme*, que apresentou *F1-score* abaixo de 0,20 em todos os modelos. Outros *datasets* também não obtiveram desempenho tão satisfatório, embora sem resultados tão baixos quanto esse. Esse cenário, somado ao fato de vários classificadores apresentarem desempenhos semelhantes nesses casos, sugere que a representação de séries temporais adotada possui limitações. É possível que as *features* extraídas não capturem nuances relevantes das séries ou, ainda, que a própria transformação para grafos não preserve características discriminativas essenciais. Por outro lado, os *datasets TwoLeadECG*, *ECGFiveDays* e *Strawberry* apresentaram desempenho satisfatório, com *F1-score* acima de 0,80. Uma possível explicação é que esses conjuntos sejam intrinsecamente mais simples de classificar, apresentando estruturas mais regulares e menor variabilidade, enquanto os demais são mais complexos ou ruidosos.

De maneira geral, o modelo que apresentou melhor desempenho médio entre os avaliados foi o *Naive Bayes*. Esse resultado é inesperado, dado que o modelo assume independência entre as *features* condicionadas à classe, uma suposição frequentemente violada em conjuntos de dados reais. Ainda assim, seus resultados superaram os dos modelos mais sofisticados nesta tarefa, indicando que as *features* extraídas possuem alto poder discriminativo individual.

Observa-se ainda que o desempenho do *Naive Bayes* foi especialmente superior nos *datasets* relacionados a sinais de ECG (*ECG200*, *ECGFiveDays*, *TwoLeadECG* e *NIFECG-1,2*),

com diferenças significativas em relação aos demais modelos. Isso sugere que, para esse tipo de dado, as distribuições das *features* extraídas a partir dos grafos são bem modeladas pelas suposições do *Naive Bayes*, o que pode ser objeto de análise mais aprofundada trabalhos futuros.

VI. CONCLUSÃO

O objetivo geral deste projeto de pesquisa é fazer uma análise comparativa em termos de desempenho entre as diferentes maneiras de classificar uma série temporal usando um grafo como representação intermediária e extraindo *features* desse grafo para fazer a classificação. Nesse sentido, o trabalho desenvolvido no POC I se concentrou em uma abordagem em específico: transformou-se a série em um *Grafo de Visibilidade* e, para construir uma representação vetorial deste grafo, foram usados atributos dos grafos e dos nós desse grafo e, por fim, foram usados modelos clássicos de aprendizado supervisionado para fazer a classificação.

Para fazer uma análise de desempenho, foram usadas 10 bases de dados retiradas do *UCR Archive* — um conjunto de dados extensivamente usado em *benchmarks* para séries temporais — com diferentes tamanhos de série, número de amostras e número de classes. Foi feita uma busca de hiperparâmetros para evitar que o desempenho do modelo fosse limitado por conta de uma escolha ruim de hiperparâmetros e assim, promover uma análise mais justa dos modelos. Depois, os modelos foram treinados e testados 30 vezes em cada conjunto de dados usando diferentes divisões de treino-teste para que os resultados obtidos fossem robustos e representativos.

Ao final desses experimentos, verificou-se o desempenho da abordagem usada nesses diferentes conjuntos de dados. Além disso, atestaram-se relações entre o tipo de dados e o desempenho do modelo. Essas duas observações contribuíram para responder algumas das perguntas motivadoras desta pesquisa. Nesse sentido, os resultados obtidos foram promissores em alguns *datasets*. No entanto, observaram-se desempenhos insatisfatórios em outros, evidenciando limitações da abordagem atual.

Diante disso, o POC II realizará uma análise comparativa com modelos baseados em GNNs, bem como outros modelos baseados em convoluções e distância que representam o estado da arte. Dessa forma, será possível avaliar, de forma embasada,

qual abordagem apresenta melhor desempenho em diferentes contextos. Para garantir validade estatística nas comparações, será necessária a aplicação de testes como o teste de Friedman seguido da Diferença Crítica. Esses passos permitirão que o restante das perguntas motivadoras sejam respondidas e que uma análise comparativa das diferentes abordagens de classificação de séries temporais baseadas em grafo seja concretizada.

REFERENCES

- [1] B. D. Fulcher. Feature-based time-series analysis. In: G. Zheng, S. Zhu, and V. S. Tseng (eds.), *Feature Engineering for Machine Learning and Data Analytics*, pp. 87–116. CRC Press, 2018. <https://doi.org/10.1201/9781315182787-5>
- [2] 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. <https://doi.org/10.1109/TKDE.2014.2316504>
- [3] T. Henderson and B. D. Fulcher. An empirical evaluation of time-series feature sets. In *2021 International Conference on Data Mining Workshops (ICDMW)*, pp. 1032–1038. IEEE, 2021. <https://doi.org/10.1109/ICDMW53433.2021.00135>
- [4] L. Ye and E. Keogh. Time series shapelets: A novel technique that allows accurate, interpretable and fast classification. *Data Mining and Knowledge Discovery*, 22, pp. 149–182, 2011. <https://doi.org/10.1007/s10618-010-0179-5>
- [5] 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.
- [6] 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), pp. 1454–1495, 2020. <https://doi.org/10.1007/s10618-020-00701-z>
- [7] 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 *IX Escola Regional de Computação Aplicada à Saúde (ERCAS 2024)*, 2024.
- [8] P. H. S. Coelho, *Análise comparativa da classificação de séries temporais representadas como grafos*, Dissertação de Mestrado, Universidade Federal de Minas Gerais, Belo Horizonte, Brasil, 2024.
- [9] N. Lucas, C. Hills, J. Lines, and E. Keogh. “Proximity Forest: An effective and scalable distance-based classifier.” *Data Mining and Knowledge Discovery*, 33, pp. 607–635, 2019. Available at: <https://arxiv.org/abs/1808.10594>.
- [10] 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*, pp. 178–189, Porto Alegre, RS, Brasil, 2022. SBC.
- [11] 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(3), pp. 606–660, 2017.
- [12] M. Köppen. “The curse of dimensionality.” In *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*, volume 1, pages 4–8, 2000.
- [13] 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), pp. 10466–10485, 2024.
- [14] 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.
- [15] L. Lacasa, B. Luque, F. Ballesteros, J. Luque, and J. C. Nuño. “From time series to complex networks: The visibility graph.” *Proceedings of the National Academy of Sciences*, 105(13), pp. 4972–4975, 2008.
- [16] 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
- [17] Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping Chen, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, Gustavo Batista, and Hexagon-ML. “The UCR Time Series Classification Archive,” October 2018. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.
- [18] Mengya Jin, Huan Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey Webb, Irwin King, and Shirui Pan. “A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection.” arXiv preprint arXiv:2307.03759, 2023.
- [19] Ulrik Brandes. “On Variants of Shortest-Path Betweenness Centrality and their Generic Computation.” *Social Networks*, 30(2):136–145, 2008. <https://doi.org/10.1016/j.socnet.2007.11.001>.
- [20] Xiang Lan, Hong Mo, Shuyue Chen, Qian Liu, and Yong Deng. “Fast transformation from time series to visibility graphs.” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(8):083105, 2015.
- [21] S. Wernicke and F. Rasche. “FANMOD: a tool for fast network motif detection.” *Bioinformatics*, 22(9):1152–1153, 2006. <https://doi.org/10.1093/bioinformatics/btl038>.
- [22] 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.