

## Capítulo

# 2

## Aprendizado de Máquina Supervisionado para Séries Temporais na Área da Saúde

Diego F. Silva, Guilherme G. Arcencio, José Gilberto B. M. Júnior, Vinícius M. A. de Souza, Yuri G. A. da Silva

### *Abstract*

*Machine Learning for time series is the knowledge domain that aims at creating models from data collected over time. This type of data is especially interesting for the healthcare field, as it represents measurements of physiological signals such as electrocardiography and photoplethysmography, among other clinical exams. Models obtained through Machine Learning algorithms in this context can assist healthcare professionals in disease diagnosis and constant patient monitoring. This chapter presents techniques for handling temporal healthcare data from theoretical and practical perspectives. In addition to discussing preprocessing and feature extraction, it addresses the two main Machine Learning tasks for this context: classification and extrinsic regression.*

### *Resumo*

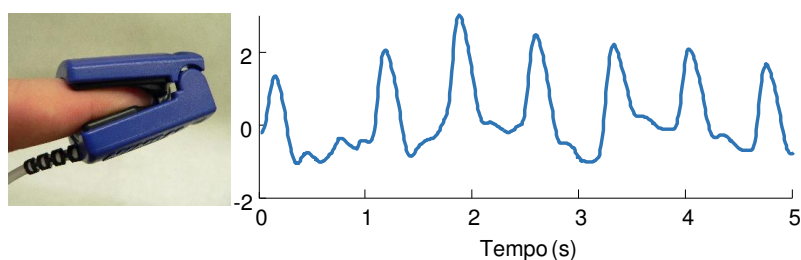
*O Aprendizado de Máquina para séries temporais é o domínio do conhecimento relacionada à criação de modelos a partir de dados coletados ao longo do tempo. Esse tipo de dados é especialmente interessante para a Saúde, uma vez que é a forma de representar medições de sinais fisiológicos, como eletrocardiografia e fotopletismografia, além de outros exames clínicos. Modelos obtidos por algoritmos de Aprendizado de Máquina, nesse contexto, podem auxiliar profissionais de saúde no diagnóstico de doenças e no monitoramento constante de pacientes. Este capítulo apresenta técnicas para lidar com dados temporais para a área da saúde, sob os pontos de vista teórico e prático. Além de discutir a respeito de pré-processamento e extração de características, aborda as duas principais tarefas de Aprendizado de Máquina para este contexto: classificação e regressão extrínseca.*

## 2.1. Introdução

Com o desenvolvimento de novas tecnologias como veículos autônomos, cidades inteligentes e a Internet das Coisas, sensores móveis se tornaram cada vez mais baratos, precisos, pequenos e eficientes em uso de bateria e capacidade de processamento. Além disso, o barateamento de meios de armazenamento e o surgimento de algoritmos capazes de processar grandes volumes de dados permite, cada vez mais, a coleta de dados medidos ao longo do tempo em variados domínios do conhecimento, como saúde, economia e monitoramento ambiental. Como resultado, aplicações que coletam grandes volumes de dados temporais usando sensores surgem em diferentes domínios do conhecimento [Ang and Seng 2016, Yeh et al. 2018, Hu et al. 2019].

Especialmente, a área da Saúde conta com variadas formas de obtenção de dados temporais, bem como diferentes aplicações. Por exemplo, sinais relativos à atividade cardiorrespiratória podem auxiliar na prevenção de ataques cardíacos [Alsuliman et al. 2020] e sinais de unidades de medida inercial, como acelerômetros, pode auxiliar na identificação de doenças como o mal de Parkinson [El Maachi et al. 2020].

Dentre as fontes de dados temporais, é possível citar equipamentos para registro de sinais fisiológicos, como oxímetros e eletrocardiógrafos. A Figura 2.1 ilustra um fotopletismógrafo de dedo e um gráfico da variação de luz coletada por ele ao longo do tempo.



**Figura 2.1.** Um oxímetro de dedo (*esquerda*) e uma série temporal de fotopletismografia de 5 segundos obtida com esse aparelho (*direita*). Adaptado de [Silva et al. 2018].

Enquanto esses dados podem ser utilizados para uma infinidade de aplicações, como o monitoramento de sinais vitais e auxílio ao diagnóstico, grande parte do volume de séries temporais coletadas acaba sendo subutilizada. O Aprendizado de Máquina oferece um conjunto de técnicas altamente eficazes para melhor aproveitar esses dados em aplicações práticas. Nesse contexto, este capítulo aborda conceitos e técnicas de Aprendizado de Máquina aplicados a séries temporais na área da Saúde<sup>1</sup>. Dado o amplo escopo desse domínio, o foco deste trabalho está nas tarefas de classificação e regressão extrínseca de séries temporais. Além disso, são discutidas diversas aplicações desse tipo de dados na área da Saúde, com especial ênfase em dados de eletrocardiografia e fotopletismografia.

<sup>1</sup>Há uma parte prática neste capítulo, em que se utiliza Python e algumas bibliotecas escritas para essa linguagem de programação. Portanto, espera-se que os leitores possuam um conhecimento prévio básico sobre a linguagem. Além disso, este capítulo apresenta apenas a parte principal dos códigos utilizados. Para melhor detalhamento de cada passo, disponibilizamos *notebooks* no repositório GitHub <https://github.com/diegofurts/SBCAS2023>.

A introdução aos conceitos básicos de séries temporais e Aprendizado de Máquina, bem como a discussão de algumas aplicações na Saúde, é realizada na Seção 2.2. Nela, também são brevemente discutidas a decomposição de uma série temporal e o domínio de frequências. A seguir, na Seção 2.3, são apresentados conceitos teóricos e práticos do pré-processamento de séries temporais, com ênfase na remoção de tendência, interpolação, filtragem e extração de características. Como a classificação e a regressão extrínseca possuem muitos conceitos em comum, bem como diversos algoritmos baseados em premissas similares, esses conceitos e alguns algoritmos são apresentados juntos, na Seção 2.4. Por fim, a Seção 2.2.2 apresenta alguns estudos de caso práticos, utilizando ferramentas de código aberto para realizar as tarefas estudadas utilizando dados de eletrocardiograma e fotopleletismografia.

## 2.2. Conceitos Básicos

Com o constante aumento na geração, coleta e armazenamento de séries temporais, também é crescente o interesse em algoritmos capazes de realizar previsões e análises avançadas com base nesse tipo de dados. Este capítulo aborda o uso do Aprendizado de Máquina em séries temporais, com maior foco em aplicações na área da Saúde. Nesse contexto, o objetivo principal desta seção é apresentar os conceitos fundamentais relacionados às séries temporais e ao Aprendizado de Máquina, além de discutir algumas aplicações específicas na área da Saúde.

### 2.2.1. Séries Temporais

O principal objeto deste curso são séries temporais, com foco naquelas originadas de aplicações relacionadas à Saúde. Esse tipo de dado tem se destacado em uma infinidade de aplicações, em áreas como economia, finanças, meteorologia, engenharia e ciências sociais. De forma geral, esse tipo de dados é utilizado em aplicações em que se há o interesse de realizar previsões relacionadas a um comportamento no tempo. Por exemplo, a observação de um instante da resposta de um eletrodo fixado na pele de um paciente não é capaz de indicar informações relevantes sobre sua saúde. Por exemplo, uma única observação de um eletrodo fixado na pele de um paciente pode não fornecer informações relevantes sobre sua saúde. No entanto, quando essa resposta é observada ao longo de um intervalo de tempo, ela pode ser utilizada para diagnósticos, como a detecção de irregularidades no ritmo cardíaco por eletrocardiograma [Becker 2006].

Formalmente, uma série temporal  $S$  é uma sequência de  $n$  valores ordenados, conforme a Equação 1.

$$S = (s_1, s_2, \dots, s_n), \text{ tal que } s_t \in \mathbb{R}^d \forall t \in [1, n] \quad (1)$$

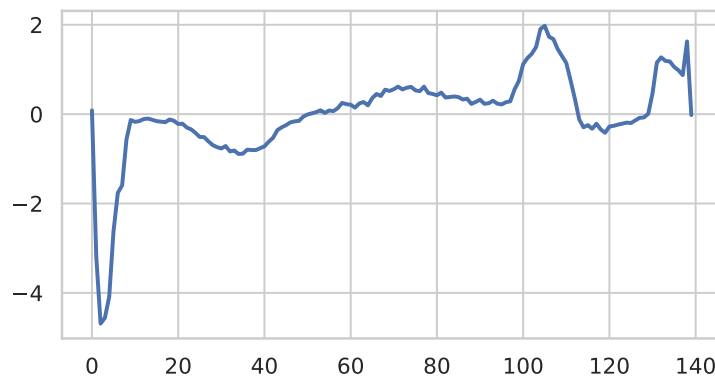
Seguindo essa definição, quando  $d = 1$ , a série temporal é chamada unidimensional ou univariada. Caso contrário, ou seja,  $d > 1$ , tem-se uma série temporal multidimensional ou multivariada. Por exemplo, a fotopleletismografia, observada isoladamente, é representada por uma série temporal univariada. Ao ser observada junto a outros parâmetros, como eletrocardiograma e pressão arterial, observa-se uma série temporal multivariada.

Cada um dos valores  $s_t$  é referido como uma observação do fenômeno que gera esse dado temporal. Usualmente, assume-se que duas observações consecutivas são igualmente espaçadas no tempo. Essa suposição é comum, pois os dispositivos de coleta normalmente medem o valor aferido em uma frequência constante. Caso essa suposição não seja verdadeira, o procedimento usual é utilizar alguma técnica de interpolação dos dados, uma vez que os algoritmos desenvolvidos para séries temporais comumente assumem essa premissa.

Por fim, uma subsequência da série temporal é um subconjunto contíguo de suas observações. Formalmente, uma subsequência de comprimento  $m$  é definida pela Equação 2.

$$\mathbb{S} = (s_i, s_{i+1}, \dots, s_{i+m-1}), \text{ tal que } s_i \in \mathbb{R}^d \forall t \in [1, n - m] \quad (2)$$

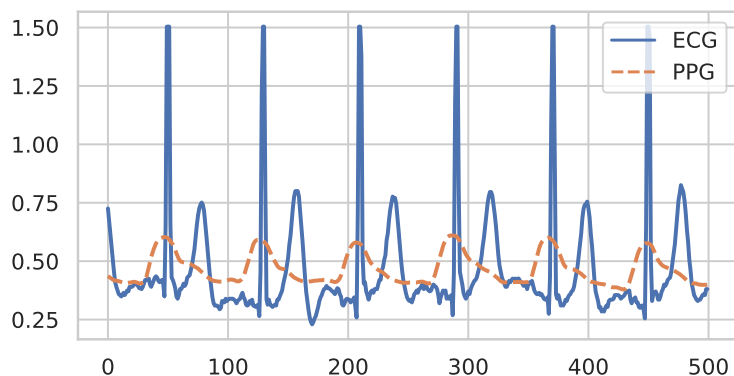
A Figura 2.2 apresenta a forma gráfica de uma série temporal obtida por meio de um eletrocardiograma. Por simplicidade de representação, por vezes será desconsiderada a escala do tempo real no eixo temporal neste capítulo, exibindo-se os índices das observações no eixo. Isso se deve ao fato de que será considerado que todas as séries temporais de um mesmo conjunto de dados foram amostradas na mesma frequência, ou seja, possuam o mesmo número de observações por segundo. Para os algoritmos que serão estudados, o valor real do eixo temporal não é utilizado. Porém, caso seja relevante observar o tempo do evento, a conversão de unidades é bastante simples.



**Figura 2.2. Exemplo de uma série temporal univariada com 140 observações, extraída do conjunto de dados ECG5000 [Dau et al. 2018].**

Enquanto a Figura 2.2 apresenta uma série univariada contendo apenas um evento, a Figura 2.3 exibe uma subsequência de 500 observações de uma série temporal, que contém dados de uma eletrocardiografia (ECG) e uma fotopletismografia (PPG), retirada do repositório de dados para regressão extrínseca de séries temporais [Tan et al. 2020].

Esses dois conjuntos de dados representam exemplos de séries temporais na área da Saúde. Há diversas aplicações nesse contexto, sendo algumas delas discutidas a seguir.



**Figura 2.3. Exemplo de uma série temporal multivariada com 500 observações, extraída do conjunto de dados BIDMC32HR [Tan et al. 2020].**

### 2.2.2. Utilização de Séries Temporais na Saúde

As séries temporais são utilizadas há décadas em aplicações no domínio da Saúde. Porém, observa-se um grande crescimento no interesse por esse tipo de dados em anos recentes. A utilização de Aprendizado de Máquina para dados temporais têm sido aplicado da previsão de ocupação de leitos e tempo de permanência dos pacientes até como auxílio a diagnósticos em diferentes especialidades, como doenças cardiovasculares, renais e respiratórias [Morid et al. 2023].

Uma gama de aplicações vêm se tornando cada vez mais comuns em dados obtidos por meio de eletrocardiografia [Hong et al. 2020]. Dentre essas aplicações, pode-se exemplificar a detecção de apneia do sono [Salari et al. 2022] e a identificação de batimentos cardíacos anômalos, como arritmias [Ebrahimi et al. 2020].

Outro dado que tem sido comumente utilizado em tarefas de Aprendizado de Máquina para séries temporais em Saúde é a fotopletismografia, que consiste na variação da resposta óptica baseada em uma fonte de luz (geralmente um diodo emissor de luz - LED) voltada para a pele. Aparelhos para se realizar a fotopletismografia se tornaram comuns durante o período da pandemia de COVID-19, pois é uma forma precisa de se estimar a oxigenação no sangue de forma não invasiva. Além desse parâmetro, estudos mostram que o Aprendizado de Máquina também pode ser utilizado para se estimar pressão sanguínea [El-Hajj and Kyriacou 2020], detectar ansiedade [Perpetuini et al. 2021], hipertensão [Martinez-Ríos et al. 2022], fibrilação atrial [Pereira et al. 2020], entre outras, a partir de dados de fotopletismografia.

É importante ressaltar que se percebe em novos produtos e na literatura recente haver um crescente interesse no desenvolvimento de novas tecnologias e aprimoramento da comunicação na medicina. A situação social, como o grande número de pessoas que moram em áreas vulneráveis ou remotas, tendo dificuldade de acesso a diversos serviços de saúde, e eventos recentes, como a pandemia de COVID-19, tem estimulado o crescimento do uso da telemedicina e o desenvolvimento de ferramentas para monitoramento de pacientes.

Nesse contexto, encontram-se as ferramentas de *mobile health* (mHealth). A Organização Mundial da Saúde (OMS) definiu mHealth como “prática médica e de saúde pública suportada por dispositivos móveis, como telefones celulares, dispositivos de monitoramento de pacientes, assistentes digitais pessoais e outros dispositivos sem fio” (tradução nossa) [WHO 2011]. Alguns dispositivos relevantes para a saúde móvel são os *smartwatches* e os *smartphones*. Por meio deles, é possível obter dados (comumente séries temporais) que podem ser utilizados para uma infinidade de aplicações.

Por exemplo, acelerômetros podem ser utilizados para o monitoramento de pacientes com doença de Parkinson [Lee et al. 2022]. Mas, além disso, vale ressaltar que os sensores contidos nesses dispositivos móveis podem simular diferentes equipamentos médicos. Por exemplo, a câmera de um *smartphone* pode ser utilizada para simular uma fotoplestimografia [Mazzu-Nascimento et al. 2020] e seu microfone pode ser aplicado como fonocardiografia [Thoms et al. 2017]. Por esse e outros motivos, vivemos um cenário muito favorável ao crescimento da demanda por técnicas de Aprendizado de Máquina para séries temporais na área da Saúde.

Porém, para melhor aproveitar esse cenário, é preciso compreender melhor as características dos dados que estão sendo coletados e como utilizar diferentes técnicas para garantir a qualidade dos dados e das tarefas executadas a partir deles.

### 2.2.3. Componentes de uma série temporal

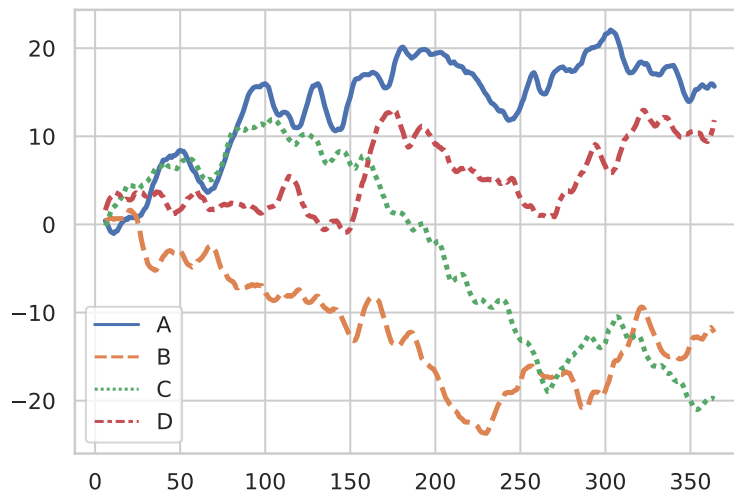
Uma série temporal pode ser decomposta em diferentes propriedades que definem sua variação no tempo. Em muitos casos, a decomposição de uma série pode ser fundamental para uma análise aprofundada dos dados. Ainda, remover a influência de uma ou mais dessas componentes pode ser crítico em diferentes tarefas. Para melhor compreensão desses fatos, esta seção trata dessas componentes.

As componentes mais amplamente descritas na literatura são a tendência, a sazonalidade e o ruído. A tendência é uma componente que representa a direção dos dados ao longo do tempo. Ela indica o comportamento da série a médio ou longo prazo, indicando se os valores observados estão em ascensão, diminuição ou permanecem estáveis. A tendência pode ser linear ou não e pode ser global, que ocorre durante toda a observação do fenômeno, ou local, que representa a direção do movimento das observações em uma subsequência.

A Figura 2.4 mostra um exemplo de quatro caminhadas aleatórias com tendências diferentes.

A sazonalidade se refere a flutuações regulares que ocorrem em intervalos fixos de tempo. Pode ser vista como um comportamento padrão que ocorre em um determinado intervalo de um período. Por exemplo, a série temporal de volume de vendas de uma de brinquedos deve possuir um pico muito evidente próximo ao dia das crianças, seguido de uma queda e outro pico próximo ao Natal. Esse é um comportamento que se repete a cada ano, portanto é um comportamento sazonal.

Também é possível encontrar na literatura a descrição de outra componente, que possui relações com a sazonalidade, conhecida como ciclo. Assim como a sazonalidade, o ciclo também está relacionado a padrões de variação. Porém, esta componente não



**Figura 2.4.** Exemplo de quatro séries temporais com tendências distintas. A série A possui uma clara tendência crescente. A série B também possui uma elevação global, porém com tendências locais de estabilidade (como nas primeiras 150 observações) e decrescente (aproximadamente entre as observações 175 e 275). As séries C e D possuem em tendência global decrescente. Porém, é possível observar uma tendência local de queda nas primeiras 100 observações da série C.

considera os padrões sazonais, mas aqueles que ocorrem em intervalos irregulares e, geralmente, que abrangem períodos mais longos. São, portanto, interessantes para o estudo do comportamento da série a médio/longo prazo.

Além das componentes mencionadas anteriormente, as séries temporais também podem conter variações irregulares ou aleatórias que não podem ser atribuídas a tendência, sazonalidade ou ciclo. Para essas flutuações, comumente atribui-se o termo “ruído”. Variações causadas por fatores imprevisíveis e imprecisões de medida são os principais causadores de irregularidade. Porém, as variações não observadas na tendência e sazonalidade também podem representar informação relevante, pois podem ter sido causadas por perturbações no fenômeno observado.

Uma série temporal pode ser, então, constituída por uma combinação dessas componentes. Especificamente, essa combinação pode ser aditiva ou multiplicativa. Formalmente, a decomposição aditiva representa a série temporal pela soma das componentes, ou seja,  $S = S_{tend} + S_{sazon} + S_{irreg}$ , em que  $S_{tend}$ ,  $S_{sazon}$  e  $S_{irreg}$  são os valores de tendência, sazonalidade e variações irregulares associados a cada observação da série, respectivamente. De forma similar, a decomposição multiplicativa considera que as componentes devem ser multiplicadas para produzir a série temporal associada, ou seja,  $S = S_{tend} * S_{sazon} * S_{irreg}$ .

A decomposição das séries temporais em suas componentes individuais permite analisar separadamente cada aspecto do fenômeno em estudo. Além de facilitar a compreensão do fenômeno, o correto tratamento das componentes, como a remoção de componentes indesejadas, pode melhorar significativamente a qualidade das análises e previsões realizadas a partir de séries temporais. Devido à sua importância, este capítulo apresenta

uma forma prática de realizar essa decomposição como uma etapa do pré-processamento de séries temporais, na Seção 2.3.

#### 2.2.4. Domínio da frequência

Outra forma de se representar uma série temporal é por meio de suas componentes de frequência. A intuição por trás dessa representação vem da Transformada de Fourier, que expressa funções em termos de funções de base sinusoidal [Bloomfield 2004].

Uma vez que trabalhamos com um número finito de observações para representar o fenômeno observado ao longo do tempo, pode-se utilizar a Transformada Discreta de Fourier para se representar a série, originalmente no domínio do tempo, para o domínio de frequências. Usualmente, essa operação é realizada por meio do algoritmo da Transformada Rápida de Fourier. Como resultado, cada componente é caracterizada por sua frequência e amplitude.

A Figura 2.5 ilustra a fotopleitismografia apresentada na Figura 2.3 junto à sua representação no domínio de frequências.

A amplitude de cada componente representa a contribuição relativa dessa frequência específica na série temporal. Valores maiores de amplitude indicam uma maior influência daquela frequência na série. Dessa forma, é possível identificar quais frequências têm um papel mais significativo na série temporal. Essa análise pode levar à observância de determinados padrões ou fenômenos representados na série. Por exemplo, a análise das componentes de frequência de um eletrocardiograma pode evidenciar padrões de ritmos cardíacos, auxiliando na identificação de batimentos cardíacos anormais.

No entanto, é importante ressaltar que a Transformada de Fourier pressupõe que a série temporal seja estacionária e linear. Caso essas suposições não sejam atendidas, pode ser necessário realizar a análise no domínio de tempo-frequência, por meio da Transformada de Fourier de Tempo Curto e suas variantes [Durak and Arikan 2003].

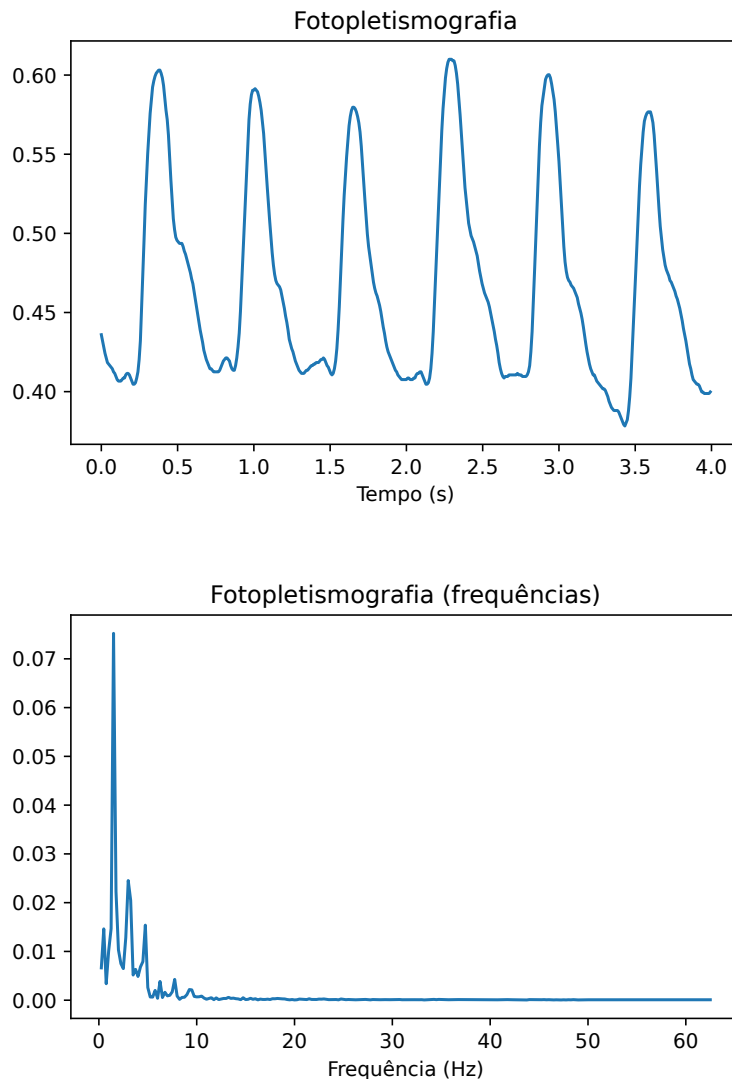
Em alguns casos, as séries temporais podem não atender a essas suposições, e técnicas mais avançadas, como a análise espectral de tempo-frequência, podem ser necessárias.

#### 2.2.5. Aprendizado de Máquina Supervisionado

Aprendizado de Máquina (AM) é um subcampo da Inteligência Artificial (IA) no qual algoritmos apresentam a capacidade de aprender modelos para lidar com uma determinada tarefa a partir da experiência representada por observações anteriores do fenômeno estudado, ou seja, com um conjunto de dados relacionado ao problema a ser resolvido [Mitchell 1997]. Nesse contexto, os principais tipos de aprendizado são o supervisionado, não-supervisionado e por reforço [Faceli et al. 2020].

Este capítulo é voltado exclusivamente para o AM supervisionado. Nesse tipo de aprendizado, considera-se que cada dado possua um atributo alvo, ou variável dependente, a ser predito. A partir de um conjunto de dados rotulados, um algoritmo de AM supervisionado deve induzir um modelo capaz de generalizar o conhecimento descrito pelos dados a fim de prever valores para os atributos alvo de dados ainda não conhecidos. Em





**Figura 2.5.** Fotopletismografia coletada a 125 observações por segundo e sua representação no domínio de frequências.

outras palavras, quando uma nova observação do mesmo fenômeno é feita, esse modelo deve estimar um valor para a variável dependente de tal observação.

Considere um conjunto de dados  $X = x_1, x_2, x_3, \dots, x_n$ ,  $n \in \mathbb{N}^*$ , em que cada elemento  $x_i$  é chamado exemplo. Especificamente,  $x_i = (S_i, a_i)$ ,  $i \in (1, 2, 3, \dots, n)$  é um par ordenado contendo as variáveis que descrevem o evento observado  $S_i$  e um atributo alvo  $a_i$  associado a ele. Neste caso, o conjunto de dados  $X$  contém  $n$  exemplos.

Quando o atributo alvo  $a_i$  contém um valor discreto, podemos chamá-lo de rótulo ou classe. A partir dessa informação, podemos definir a tarefa de classificação como aquela cujo objetivo é obter um modelo tal que, quando uma nova observação é realizada, seja capaz de atribuir corretamente um rótulo a esse novo dado.

Um exemplo de classificação no contexto de séries temporais é a tarefa de reconhecimento de atividades. Nessa tarefa, uma série temporal multidimensional pode ser coletada, por exemplo, por meio do acelerômetro de um celular [Kwapisz et al. 2011] ou outras medidas de inércia de um *smartwatch*, ou outro dispositivo eletrônico vestível. Nesse caso, o conjunto de rótulos a se atribuir para a série temporal observada poderia ser  $R = \{ 'correndo', 'caminhando', 'nadando', 'repousando', \dots \}$ . Dessa maneira, um algoritmo de classificação deve aprender um modelo a partir séries temporais coletadas do acelerômetro de um celular tal que, para uma nova série temporal coletada, seja capaz de atribuir um valor de  $R$  ao atributo alvo  $a_i$ .

Caso o conjunto de valores que podem ser conferidos ao atributo alvo seja contínuo, a tarefa de se estimar um valor para  $a_i$  é chamada regressão. No caso de séries temporais, utiliza-se o termo “regressão extrínseca”, para não se confundir a tarefa com a atividade de previsão de valores futuros (Seção 2.5.3). Seguindo o exemplo do reconhecimento de atividades, o valor estimado pelo modelo pode ser uma estimativa das calorias gastas durante a atividade observada.

Uma vez que um modelo é treinado, ele é utilizado para prever o valor do atributo alvo para novas observações. Uma forma de avaliar a qualidade do modelo é utilizando um conjunto de dados  $Y = \{y_1, y_2, y_3, \dots, y_m\}$ , disjunto de  $X$ . Além disso, cada instância  $y_i$ ,  $i \in (1, 2, 3, \dots, m)$  é definida pelo seu conjunto de séries temporais, mas o valor de seu atributo alvo não é apresentado ao modelo. Ao final, os valores dos atributos alvo observados (omitidos ao modelo) são comparados com aqueles estimados pelos algoritmos de AM. Quanto mais próximos esses valores, melhor a qualidade do modelo.

A etapa de indução de um modelo a partir do conjunto de dados  $X$  é denominada fase de treinamento. Por sua vez, a avaliação a partir do conjunto  $Y$  é conhecida como fase de teste ou avaliação.

### 2.3. Pré-processamento e extração de características

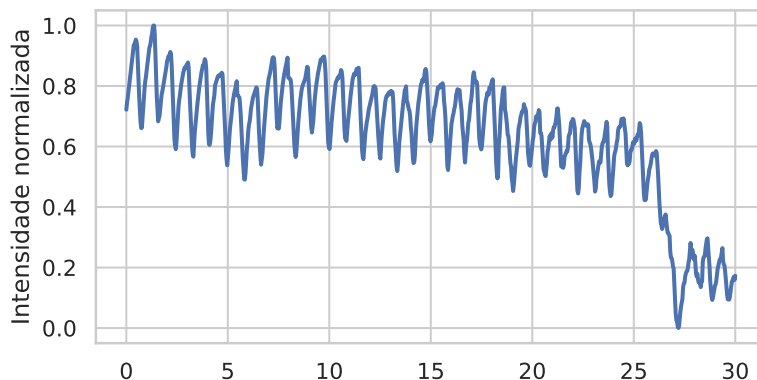
Usualmente, a primeira etapa para se utilizar o Aprendizado de Máquina em séries temporais é a análise exploratória e o pré-processamento dos dados. A análise exploratória permitirá, entre outras coisas, encontrar potenciais problemas nos dados que devem ser tratados para um melhor desempenho na tarefa a ser realizada. Além disso, há muitas abordagens que utilizam a extração de características das séries temporais tanto para exploração quanto para a representação das séries para serem utilizadas como entrada em algoritmos de Aprendizado de Máquina desenvolvidos para dados estruturados.

Nesse contexto, esta seção aborda alguns conceitos importantes de pré-processamento e extração de características de séries temporais.

#### 2.3.1. Remoção de tendência

Quando lidamos com séries temporais na área da Saúde, diversas características indesejadas podem aparecer nos dados. Algumas dessas características podem ser observadas em termos das componentes discutidas na Seção 2.2.3. Por exemplo, a Figura 2.6 mostra uma série temporal de fotopletismografia baseada em *smartphone* [Silva et al. 2023].

Esse dado pode ser sensível a pequenas variações da luz externa, por exemplo, causando uma tendência global decrescente e um artefato (queda brusca) próximo ao final do sinal.



**Figura 2.6.** Série temporal representando uma fotopletismografia de 30 segundos obtida por *smartphone*.

Para demonstrar a decomposição de séries na prática, podemos aplicar a decomposição sazonal da biblioteca `statsmodel` na série temporal apresentada na Figura 2.6, conforme exibido pelo Código-Fonte 2.1<sup>2</sup>. Nesse caso, consideramos a decomposição aditiva e o período, de acordo com observação do comportamento, um período de 24 unidades. Em outras palavras, consideramos que o ciclo de subida e descida da intensidade da luz dura 24 observações. Uma vez que este dado possui uma amostragem de 30 observações por segundo, esse ciclo corresponde a aproximadamente 0,8 segundos.

#### Código-Fonte 2.1. Decompor uma série temporal.

```

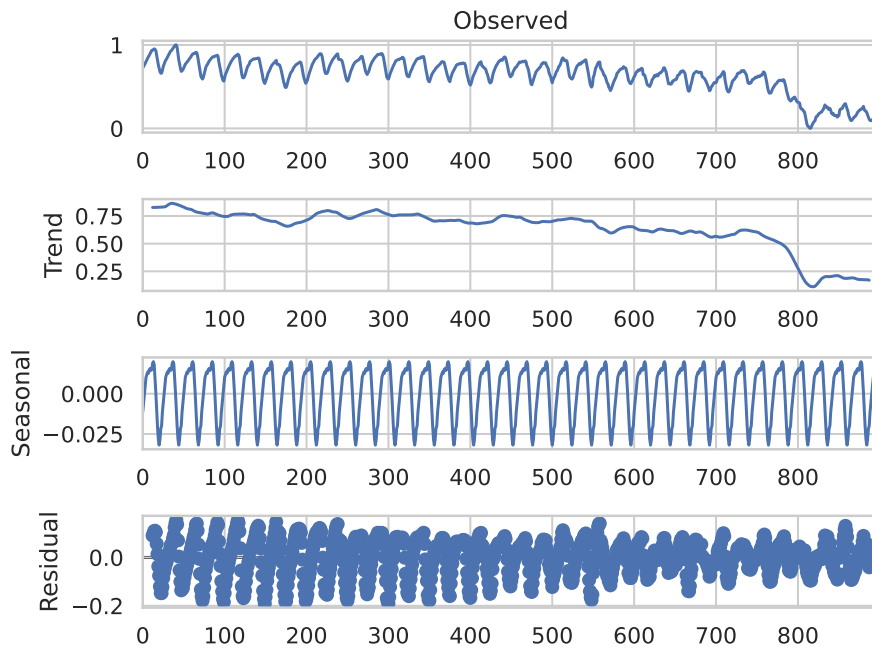
1 from statsmodels.tsa.seasonal import seasonal_decompose
2
3 import matplotlib.pyplot as plt
4 import numpy as np
5
6 series = np.genfromtxt('data.tsv', delimiter='\t')
7 result = seasonal_decompose(series,
8                             model='additive',
9                             period=24)
10
11 plt.figure(figsize=[6,9])
12 result.plot()
13 plt.show()

```

O resultado dessa operação é exibido na Figura 2.7.

Seguindo-se esse exemplo, é possível verificar a série temporal resultante da remoção da componente de tendência. A Figura 2.8 mostra o resultado da soma das componentes de sazonalidade e ruído, resultante do Código-Fonte 2.2.

<sup>2</sup>Para melhor entendimento da leitura dos dados, recomendamos o leitor se direcionar à Seção 2.5.1



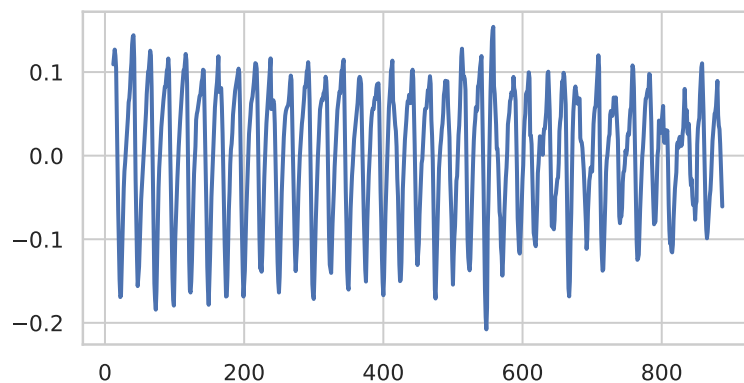
**Figura 2.7.** Resultado da decomposição aditiva da série exibida na Figura 2.6 (*topo*) em tendência, sazonalidade e ruído (do segundo quadro em diante, de cima para baixo, respectivamente).

**Código-Fonte 2.2.** Exibir a série resultante da remoção da componente de tendência.

```

1 import seaborn as sns
2 sns.set_theme(style='whitegrid')
3
4 plt.figure(figsize=[6,3])
5 sns.lineplot((result.seasonal + result.resid), linewidth=2)
6 plt.plot()

```



**Figura 2.8.** Resultado da remoção da componente de tendência.

### 2.3.2. Interpolação

Além de problemas relacionadas às suas componentes, há outras características indesejadas que podem ser observadas em séries temporais de quaisquer domínios de aplicação. Dois dos mais comuns, que também podem ocorrer no domínio da Saúde, são a amostragem irregular e os valores ausentes. No primeiro caso, a suposição de amostragem uniforme é invalidada. No segundo, alguns algoritmos podem falhar ao fazer previsões. Nos dois casos, é possível utilizar técnicas de interpolação, que é o processo de estimar valores intermediários entre observações com valores conhecidos a fim de preencher lacunas causadas por valores ausentes. Existem várias abordagens de interpolação, variando-se a suposição da complexidade que os valores ausentes podem assumir.

A interpolação linear pressupõe uma relação linear entre os pontos conhecidos adjacentes. Dessa forma, calcula os valores intermediários usando uma combinação linear ponderada entre esses vizinhos de valores conhecidos. A interpolação polinomial generaliza a interpolação linear ao aplicar uma função polinomial para estimar os valores intermediários. Para isso, ajusta-se o polinômio aos pontos conhecidos da série temporal e, então a função é utilizada para estimar os valores ausentes. Por fim, outra técnica de interpolação muito conhecida é a aplicação de *splines*. Assim como a interpolação polinomial, ajusta uma curva aos valores conhecidos para estimar valores para as observações com valores ausentes. Porém, nesse caso, é utilizada uma combinação de polinômios por partes, conhecida como *spline*.

Para realizar a interpolação, é comum a utilização da biblioteca `pandas`, bastante conhecida para a leitura e manipulação de dados. No Código-Fonte 2.3, são removidas 300 observações (um terço) da série temporal aleatoriamente. A seguir, são exibidas as séries com tais valores ausentes e o resultado da interpolação linear. O gráfico resultante é exibido na Figura 2.9.

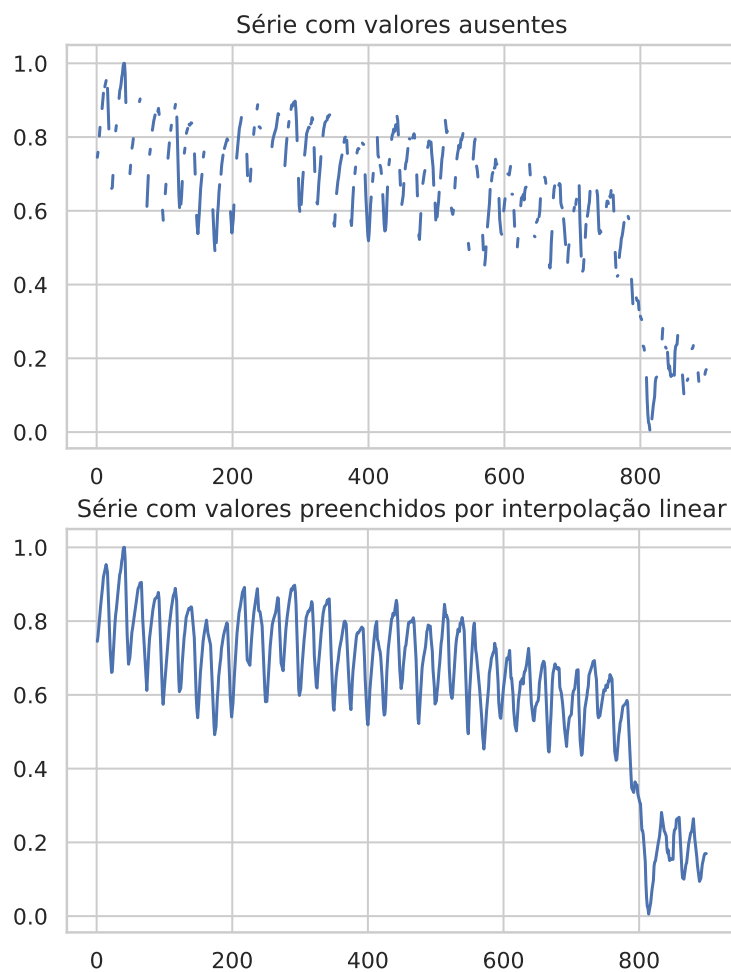
#### Código-Fonte 2.3. Remover observações e interpolar valores para substituí-los.

```

1 import random
2 import pandas as pd
3
4 random.seed(42)
5 to_remove = random.sample(range(series.size), 300)
6 series[to_remove] = np.nan
7
8 fig, (ax1, ax2) = plt.subplots(2, 1, figsize=[6,8])
9
10 ax1.plot(series)
11 ax1.set_title('Série com valores ausentes')
12
13 ax2.plot(pd.DataFrame(series).interpolate(method='linear'))
14 ax2.set_title('Série com interpolação linear')
15
16 plt.show()

```

No caso de haver uma frequência de amostragem irregular, é possível utilizar interpolação para realocar os pontos observados, a partir deles, interpolar a série temporal



**Figura 2.9. Série temporal com valores ausentes e o resultado da interpolação linear.**

nos pontos em que o valor está ausente. Essa operação permite uma análise mais consistente e facilita a aplicação de técnicas específicas para séries temporais, em especial, de Aprendizado de Máquina.

### 2.3.3. Filtragem

Além das flutuações aleatórias que podem ser observadas na decomposição de séries temporais, os sinais obtidos a partir da observação de um fenômeno também podem sofrer ruídos que não são inerentes ao que está sendo estudado. Um exemplo disso são os ruídos relacionados a problemas de medição. Geralmente, esse tipo de ruído ocorre devido a limitações do *hardware* utilizado para a coleta dos dados.

Seguindo o exemplo do uso de acelerômetros para monitorar pacientes portadores da doença de Parkinson, apresentado na Seção 2.2.2, deve-se considerar que o acelerôme-

tro pode possuir uma imprecisão que o faz registrar um ruído de alta frequência. Assim, a série temporal coletada não representa apenas a informação motora do paciente de forma clara, mas também adiciona uma variação que não adiciona qualquer informação para a aplicação.

Esse tipo de problema é muito comum em diversas aplicações. Para resolvê-lo, utilizamos técnicas de filtragem. A filtragem é o processo utilizado para remover ou atenuar componentes indesejáveis das séries temporais, preservando as características relevantes do fenômeno analisado. Esse conjunto de técnicas é amplamente utilizado para reduzir ruído, destacar padrões e facilitar a interpretação dos dados.

Formalmente, a filtragem em séries temporais consiste em aplicar um filtro a uma série temporal  $S = (s_1, s_2, \dots, s_n)$ , visando obter uma nova série temporal filtrada  $S_f = (s_{f_1}, s_{f_2}, \dots, s_{f_n})$ . O filtro utilizado pode ser representado por uma função matemática ou algoritmo que opera sobre os valores da série temporal.

Uma das técnicas mais comuns de filtragem é o filtro de passa-banda (ou passa-faixa). Esse filtro define uma faixa de frequências desejadas, o que é usualmente definido por conhecimento de domínio e pode variar significativamente entre uma aplicação e outra. A partir disso, o filtro elimina ou atenua as componentes fora da faixa definida. O passa-banda pode ser visto como a combinação dos filtros de passa-baixa, que elimina componentes de alta frequência, e de passa-alta, que elimina componentes de baixa frequência.

Outro tipo de filtro muito comum é o de médias móveis. Esse filtro substitui o valor de um ponto pelo valor médio de um subconjunto de pontos temporalmente vizinhos a ele. O filtro de médias móveis pode suavizar flutuações de curto prazo e remover ruídos de alta frequência. Um exemplo claro de médias móveis é a sua utilização na análise do número de casos de uma determinada doença, como foi comum para a COVID-19. O padrão do número de casos sofria uma variação muito significativa conforme o dia da semana. Por isso, optou-se pela utilização da média móvel de sete dias, o que remove a flutuação diária e apresentava uma informação mais clara sobre a evolução da pandemia.

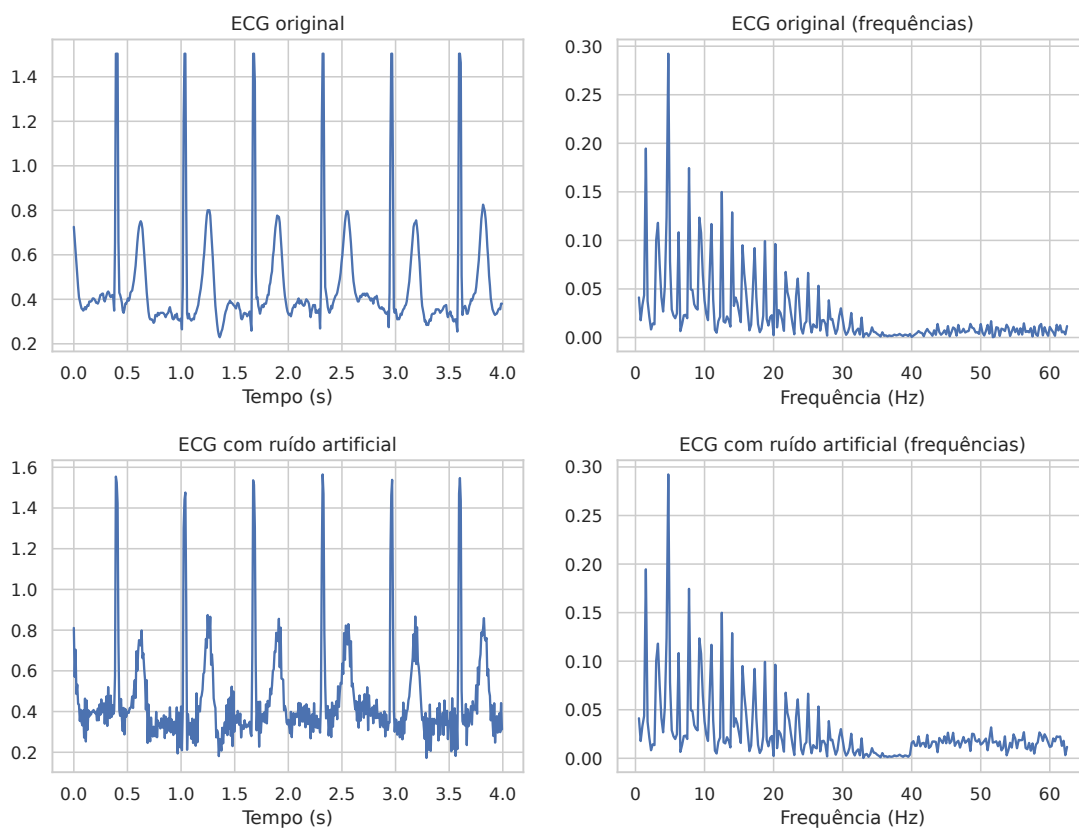
Na linguagem Python, a biblioteca mais comumente utilizada para se calcular a média móvel de séries temporais é a biblioteca `pandas`. Para visualizar o efeito desse filtro, vamos primeiramente adicionar ruído nas frequências entre 40 Hz e 62 Hz do eletrocardiograma exibido na Figura 2.3. A Figura 2.10 mostra a influência desse ruído nos domínios de tempo e de frequência do sinal.

O Código-Fonte 2.4 ilustra como é simples realizar a filtragem por média móvel no sinal por meio da biblioteca.

#### **Código-Fonte 2.4. Utilizar o filtro de médias móveis de tamanho 3.**

```
1 new_ecg = pd.DataFrame(ecg_noisy).rolling(3).mean()
```

O resultado dessa operação é apresentado pela Figura 2.11. É possível notar que o sinal se parece muito mais com o original do que o sinal ruidoso se parecia. No domínio das frequências, nota-se que as magnitudes das frequências acima dos 40Hz foi bastante reduzido. No entanto, também se nota uma atenuação do vale de curta duração correspondente a onda Q do complexo QRS, o que pode prejudicar a interpretação do



**Figura 2.10. Série temporal (original acima) com ruído artificialmente inserido (abaixo).**

dado [Kashani and Barold 2005]. Por isso, ao mesmo tempo que a filtragem de séries temporais pode ser extremamente relevante para se obter sucesso na tarefa realizada, ela deve ser executada com cautela.

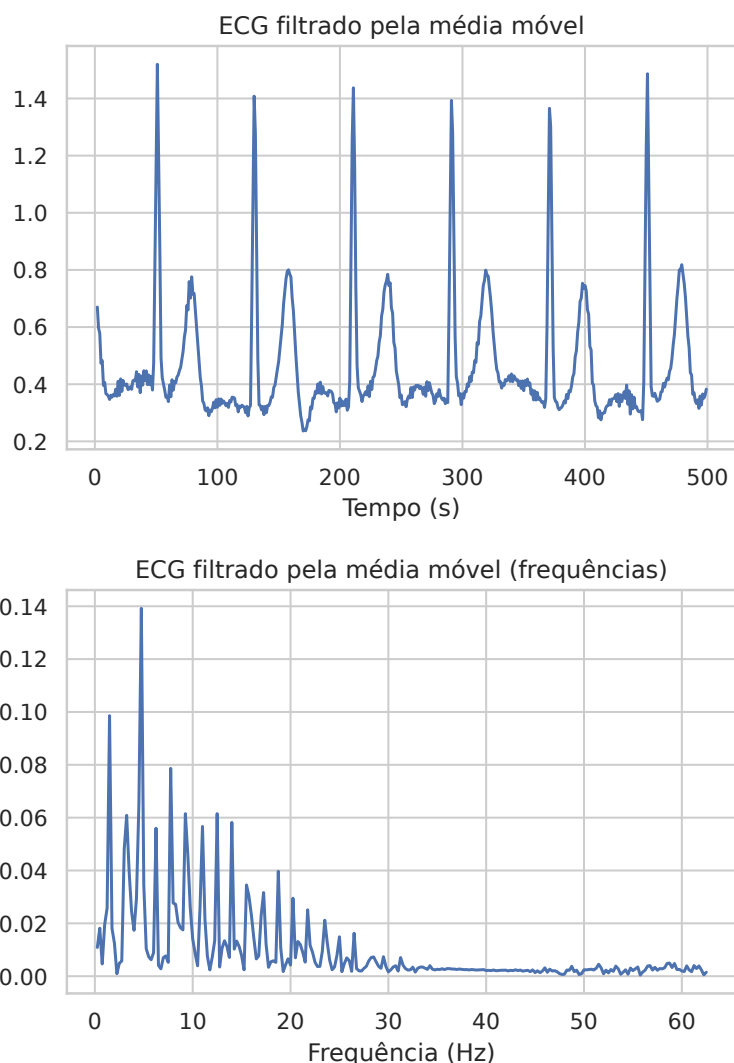
### 2.3.4. Extração de características

Uma das principais tarefas a ser desenvolvidas em análise de séries temporais é a extração de características. Formalmente, a extração de características em séries temporais consiste em mapear uma série temporal  $S = (s_1, s_2, \dots, s_n)$  em um vetor de características  $\mathbf{X} = (x_1, x_2, \dots, x_m)$ , onde  $m$  é o número de características extraídas. Cada característica  $x_i$  representa uma medida específica ou propriedade derivada dos dados.

Descrever a série temporal por meio de atributos significativos pode fornecer informação relevante para a interpretação do fenômeno observado. Por exemplo, atributos interpretáveis podem ser utilizados em etapas de análise exploratória, escolha de algoritmos para diferentes tarefas e até mesmo entrada para algoritmos de Aprendizado de Máquina projetados para lidar com dados estruturados [Ruiz et al. 2020, Gay et al. 2021].

Devido à sua importância, foram construídas diversas ferramentas para a extração de séries temporais. A maioria das bibliotecas de extração de características para séries temporais lançadas nos últimos anos, como a `tsfel` [Barandas et al. 2020] e a





**Figura 2.11. ECG filtrado por média móvel de tamanho 3 e seu efeito no domínio de frequências.**

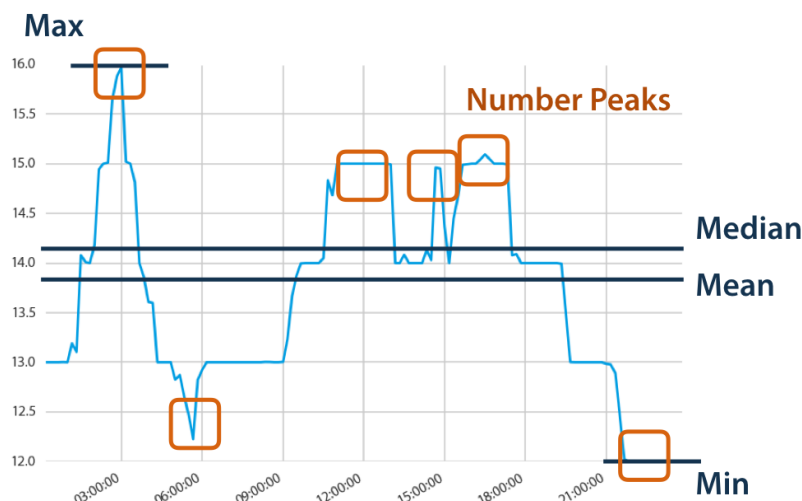
`tsfresh` [Christ et al. 2018], foram desenvolvidas para a linguagem Python. Há ferramentas em outras linguagens, como a `hctsa` [Fulcher and Jones 2017], para Matlab.

Além disso, diferentes ferramentas de Aprendizado de Máquina para séries temporais possuem módulos específicos para a extração de características, visto que diversos algoritmos usam essa tarefa como uma das etapas de seu *pipeline*. Porém, como ocorre com as bibliotecas `aeon` e `tsai`, a extração de características é feita, parcial ou completamente, pela aplicação da biblioteca `tsfresh`.

Geralmente, as características extraídas por essas ferramentas estão em categorizadas conforme as transformações realizadas ou domínio em que são extraídas. Por exemplo, a `tsfresh` possui 63 métodos de caracterização distintos, sendo que alguns extraem um conjunto de características. No total, a biblioteca extrai, por padrão, 794 ca-

racterísticas de cada série temporal. Essas características podem ser extraídas no domínio do tempo, da frequência ou de tempo-frequência.

A Figura 2.12 ilustra algumas das características que a `tsfresh` calcula no domínio do tempo. Nela, são ilustradas as características de média, mediana, máximo, mínimo e número de picos. Além dessas, diversas outras estatísticas descritivas, como curtose das suas observações e contagem de pontos acima da média são extraídas nesse domínio.



**Figura 2.12.** Esquema ilustrativo de algumas características que podem ser extraídas da série temporal. Fonte: Documentação da biblioteca `tsfresh` (<https://tsfresh.readthedocs.io/>)

Algumas características possuem uma interpretação relativamente simples de seu significado em relação ao comportamento da série temporal. Por exemplo, para dados oriundos da fotopleletismografia de dois pacientes com a maior e menor frequência cardíaca do conjunto de dados BIDMCHR (ver Figura 2.18), é possível notar diferenças significativas em alguns dos valores obtidos. A complexidade estimada do sinal, por exemplo, é diretamente proporcional à variação da série temporal. Intuitivamente, quanto mais picos e vales uma série possuir, maior será sua complexidade. No exemplo da estimativa de frequência cardíaca, a complexidade estimada a partir da fotopleletismografia do paciente com frequência cardíaca de 60,09 bpm é de 4,05. Para o paciente com frequência cardíaca de 125,28 bpm, a complexidade estimada é 8,75.

Há diferentes conjuntos padrão de características que a `tsfresh` utiliza para representar as séries por meio de atributos. Por exemplo, a ferramenta pode extrair todas as características disponíveis, apenas aquelas marcadas como “eficientes”, apenas as características extraídas no domínio do tempo, entre outros. Além disso, é possível definir manualmente um conjunto de características a serem calculadas.

Essas ferramentas, incluindo o `tsfresh`, são capazes de extrair características de séries uni ou multivariadas. No caso de séries multivariadas, é possível extrair subconjuntos diferentes de características de cada variável. Por exemplo, um especialista pode entender que um determinado subconjunto de atributos é mais relevante para representar um eletrocardiograma e outro subconjunto, potencialmente com intersecção com o primeiro, seja melhor para fotopleletismografia.

É importante notar que todas essas características são não-supervisionadas. Em outras palavras, não é necessária a informação de rótulo para estimá-las. Esse fato possui duas grandes vantagens. Primeiro, não depender de um atributo-alvo para realizar a extração de características costuma levar a técnicas mais computacionalmente eficientes. Além disso, a não dependência de um atributo alvo leva a características que generalizam melhor a caracterização das séries, melhorando a interpretação de seus significados mesmo em domínios e conjuntos de dados completamente diferentes.

A ferramenta `hctsa` (acrônimo para *Highly Comparative Time Series Analysis*) [Fulcher and Jones 2017] possui um conjunto consideravelmente maior de características que a `tsfresh`. Ela pode realizar a extração de mais de 7500 características. Enquanto, por um lado, esse número seja vantajoso no sentido de descrever a série temporal em abrangência e detalhe, isso torna a extração muito custosa. Além disso, mesmo os algoritmos aplicados sobre esses atributos podem sofrer tanto com o custo computacional quanto com a maldição da dimensionalidade [Köppen 2000].

Porém, muitas dessas características são muito correlacionadas e outras possuem pouco poder discriminativo. Quando elas são aplicadas em problemas, digamos, de classificação, muitas características acabam possuindo pouca relevância para a tarefa.

Foi pensando nessa problemática que foi criado o conjunto de características canônicas para séries temporais (`catch22`, do inglês *CAnonical Time-series CHaracteristics*) [Lubba et al. 2019]. Por meio de um processo supervisionado, foram selecionadas 22 atributos que são bem discriminativos transversalmente em muitos domínios de aplicação e sejam pouco redundantes entre si.

Nesse contexto, alguns dos algoritmos que alcançam os melhores resultados nas tarefas de classificação e regressão extrínseca de séries temporais se baseiam na extração de características, em especial das bibliotecas mais exploradas nesta seção: da `tsfresh` ou `catch22` [Middlehurst and Bagnall 2022, Guijo-Rubio et al. 2023].

## 2.4. Algoritmos e Técnicas de Classificação e Regressão Extrínseca

As duas tarefas abordadas neste capítulo, classificação e regressão extrínseca, possuem uma relação muito próxima. Além de ambas serem tarefas de Aprendizado de Máquina supervisionado, compartilham as características dos dados, como a dependência temporal entre as observações. Conseqüentemente, muitas das premissas adotadas para a proposta de algoritmos de classificação valem para algoritmos de regressão extrínseca (e vice-versa).

Nesta seção, serão abordadas as diferentes categorias de algoritmos existentes na literatura para classificação e regressão extrínseca de séries temporais. Serão discutidas quatro principais categorias de algoritmos: baseados em distância; baseados em intervalos; baseados em dicionários; aprendizado profundo; e comitês.

### 2.4.1. Algoritmos Baseados em Distância

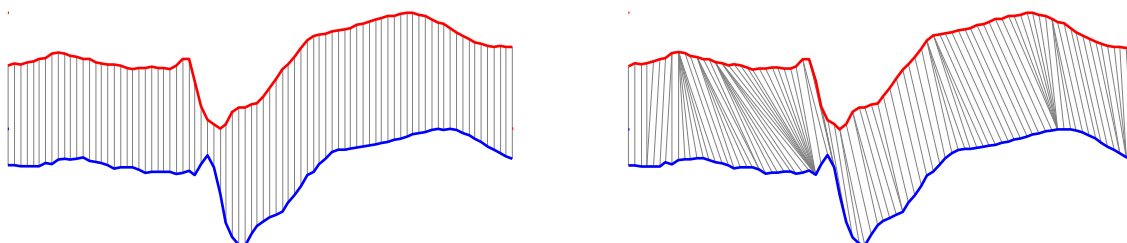
Os algoritmos de Aprendizado de Máquina para classificação e regressão baseados em distâncias operam comparando a similaridade entre as instâncias, neste caso as séries temporais, por meio de medidas de distância, tais como a distância Euclidiana, a distância

Manhattan ou *Dynamic Time Warping* (DTW). Neste caso, um rótulo é atribuído a uma instância baseando-se na similaridade com outras instâncias já rotuladas.

Nesta categoria, identificar e entender a medida de similaridade mais adequada para o problema é um fator crucial para obter um bom desempenho na tarefa. Desta forma, otimizar esse parâmetro é uma tarefa essencial, caso contrário a medida utilizada não irá capturar adequadamente a similaridade existente entre os dados, tornando o modelo ineficaz para a tarefa.

Uma primeira ideia, mais direta e amplamente utilizada, para comparar a similaridade entre duas séries é calcular a distância euclidiana entre os seus valores em cada instante. Contudo, esse tipo de medida não considera pequenas variações no eixo do tempo da série. Por exemplo, basta que uma série receba um *shift* de alguns pontos para que se torne muito dissimilar a si mesma, mesmo capturando, em essência, o mesmo fenômeno.

Por isso, foram criadas diversas medidas de distância entre séries que consideram essas pequenas distorções no eixo do tempo. Exemplos populares são a *Dynamic Time Warping* (DTW), *Derivative Dynamic Time Warping* (DDTW) e *Longest Common Subsequence* (LCSS). A Figura 2.13 mostra como a DTW considera o aspecto temporal das séries: ela “distorce” o pareamento de pontos para reduzir o impacto de deslocamentos temporais.



**Figura 2.13. Alinhamento de pontos de séries temporais na distância euclidiana (esquerda) e na DTW (direita) [Silva 2017].**

Para construir um modelo que lance mão de medidas de distância/similaridade, pode-se utilizar o algoritmo *k*-Vizinhos Mais Próximos, no qual a cada instância é atribuída a média dos atributos-alvo das *k* instâncias rotuladas mais similares, no caso de regressão, ou a classe mais frequente dentre as *k* instâncias rotuladas mais similares, no caso de classificação.

Outros modelos mais elaborados que utilizam medidas de distância são o *Elastic Ensemble* (EE) [Lines and Bagnall 2015], uma combinação de onze modelos *k*-Vizinhos com diferentes medidas e conjuntos de valores de parâmetros; a *Proximity Forest* (PF) [Lucas et al. 2019], que constrói árvores de decisão cujo critério de decisão é a distância (dentre as onze medidas usadas pelo EE) a exemplares escolhidos aleatoriamente; e a *Proximity Forest 2* (PF2) [Herrmann et al. 2023], que simplifica e reduz as possíveis medidas de distância da PF.

### 2.4.2. Algoritmos Baseados em Intervalos

Na Seção 2.3.4, foram descritas maneiras de extrair características de séries temporais. Contudo, alguns algoritmos trabalham com a extração de características de intervalos delimitados das séries, em vez de sua totalidade. Em geral, esses algoritmos constroem combinações de modelos nas quais cada modelo se ajusta a um ou mais intervalos de um conjunto de séries.

Um dos mais simples modelos baseados em intervalos é a *Time Series Forest* (TSF) [Deng et al. 2013]. A TSF constrói uma árvore de decisão para cada intervalo escolhido aleatoriamente, sendo que a média, o desvio padrão e o *slope* de cada intervalo são usados como as características de entrada de cada árvore.

O *Random Interval Spectral Ensemble* (RISE) [Lines et al. 2018] funciona de forma muito similar à TSF. Porém, as características extraídas de cada intervalo vêm do domínio da frequência, e não do tempo, como o *Power Spectrum* (PS) e a *Autocorrelation Function* (ACF).

Por fim, os mais recentes avanços em algoritmos baseados em intervalos são a *Canonical Interval Forest* (CIF) [Middlehurst et al. 2020] e a *Diverse Representation Canonical Interval Forest* (DrCIF) [Middlehurst et al. 2021]. A CIF funciona da mesma forma que a TSF, ou seja, construindo árvores de decisão baseadas em características extraídas de intervalos. Entretanto, são adicionadas às características extraídas na TSF os 22 atributos presentes no *catch22* (descrito na Seção 2.3.4), o que contribui para que o CIF tenha grande poder discriminatório em vários conjuntos de dados. A DrCIF, por sua vez, expande a gama de atributos extraídos ao considerar, para cada instância, a sua primeira derivada e o seu periodograma, além dela própria, como entradas para a extração.

### 2.4.3. Algoritmos Baseados em Dicionários

Os algoritmos baseados em dicionário tentam discretizar as séries em sequências de padrões (ou *palavras*), criando, assim, dicionários que representam a frequência observada de cada palavra nessas séries. Os modelos, por fim, discriminam as classes ou valores-alvo de um conjunto de séries ao comparar os dicionários gerados a partir de cada uma.

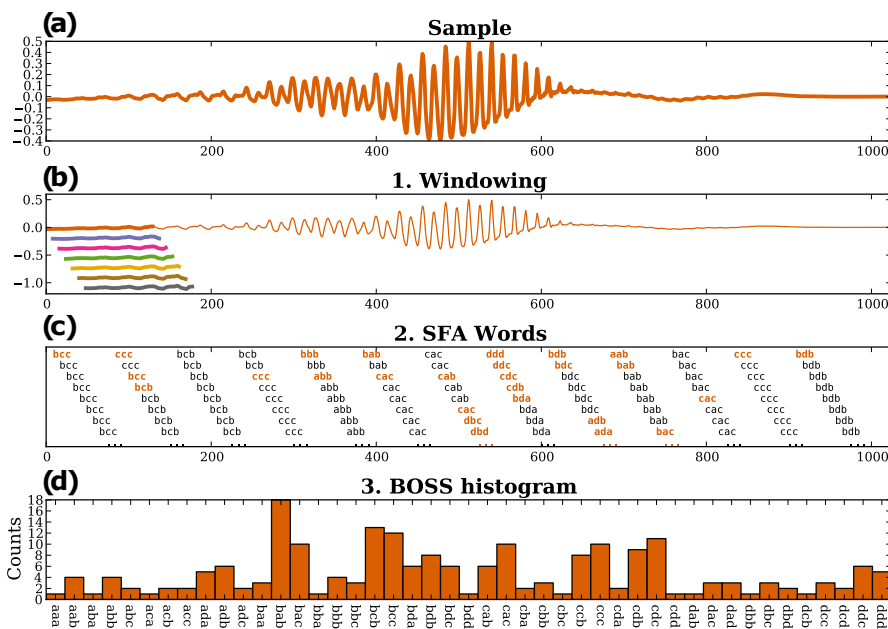
Um exemplo desse tipo de algoritmo é o *Bag of SFA Symbols* (BOSS) [Schäfer 2015], que consiste em uma sequência de diversas etapas. Primeiro, são extraídas janelas deslizantes das séries, isto é, são considerados trechos de tamanho fixo  $w$  que se movem uma observação à frente a cada iteração. Cada janela, então, passa por um processo chamado *Symbolic Fourier Approximation* (SFA) [Schäfer and Höggqvist 2012], no qual seus  $l$  primeiros coeficientes espectrais (obtidos da sua Transformada Discreta de Fourier) são discretizados em  $c$  compartimentos (ou *letras*) de forma que cada compartimento contenha quase o mesmo número de instâncias.

Após a SFA de cada janela, cada série se torna uma sequência de palavras de  $l$  letras, formadas a partir de um alfabeto de tamanho  $c$ . Uma redução de redundância é então aplicada a essas sequências, ou seja, em caso de palavras iguais consecutivas, apenas o primeiro caso é considerado. O número de ocorrências de cada palavra é, então, contado em cada sequência, obtendo-se assim um conjunto de histogramas, dicionários ou sacola de palavras (do inglês, *bag of words*).

Por fim, a classificação de instâncias é feita por meio de um algoritmo de 1-Vizinho Mais Próximo, que considera os dicionários obtidos. A medida de distância considerada por ele é definida na Equação 3, chamada de *Distância BOSS* pelos autores originais. Note que, nesta medida, apenas as palavras não ausentes na primeira sacola são consideradas, tornando a distância não comutativa. Uma instância recebe, por fim, a classe da instância rotulada mais próxima.

$$\text{dist}(B_1, B_2) = \sum_{w \in B_1 | B_1(w) > 0} [B_1(w) - B_2(w)]^2 \quad (3)$$

O processo de transformação em histogramas realizado pelo BOSS é ilustrado na Figura 2.14.



**Figura 2.14. Ilustração das etapas do algoritmo BOSS utilizando-se palavras de tamanho 3 e alfabeto de tamanho 4. As palavras em preto são desconsideradas devido à redução de numerosidade [Schäfer 2015].**

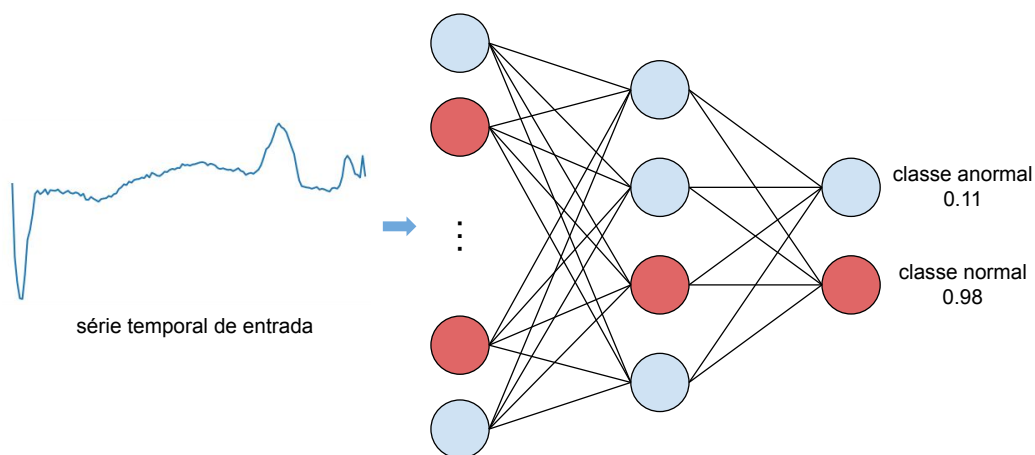
#### 2.4.4. Aprendizado Profundo

O aprendizado profundo, ou em inglês *deep learning*, têm recentemente recebido grande atenção na literatura. Essa categoria de algoritmos de Aprendizado de Máquina tem ultrapassado o desempenho de outros algoritmos mais clássicos em diversas tarefas. Nesta seção, serão explicados os conceitos básicos para o entendimento em alto nível do aprendizado profundo e de redes neurais.

Esta categoria de algoritmos inspirou-se na estrutura de funcionamento do cérebro humano, englobando técnicas de treinamento de redes neurais para realizar tarefas complexas envolvendo processamento massivo de dados. Essas redes neurais consistem em camadas de funções conectadas que, assim como neurônios naturais, trocam informação entre si, mas processam seu sinal de entrada individualmente. Ou seja, este conceito pode ser entendido superficialmente como um modelo simplificado do cérebro humano.

Assim como o cérebro trabalha com as conexões entre os neurônios para desempenhar tarefas complexas, as redes neurais (artificiais) simulam essas conexões para capturar padrões nos dados por meio de pesos atribuídos às suas entradas, assim como as sinapses do cérebro humano.

Na Figura 2.15, é possível observar uma rede neural de exemplo, na qual uma série temporal de eletrocardiografia é tomada como entrada. Nessa representação, os neurônios ativados estão na cor vermelha. Na saída, é possível observar que o neurônio ativado com maior valor é o que indica a classe normal, ou seja, o ECG de entrada é pertencente a um paciente saudável, sem nenhuma anomalia no ritmo cardíaco.



**Figura 2.15. Exemplo de uma rede neural artificial na qual a entrada é uma série temporal de um eletrocardiograma e a saída são as classes normal e anormal, que indica uma anomalia.**

Para o treinamento destas redes, faz-se necessária a existência de um conjunto de dados rotulados. Desta forma, a rede irá aprender as relações importantes nos dados para que o rótulo seja predito. Sejam essas relações importantes para definir um dado como pertencente a uma classe, como uma anomalia em batimento cardíaco pode indicar uma doença, ou uma relação em uma fotopletismografia indicando um valor de saturação do oxigênio, em uma tarefa de regressão extrínseca. Entretanto, é necessário um grande volume de dados para que a rede possa aprender essas relações complexas nos dados, o que pode ser um fator limitante para o uso de redes neurais.

Tratando-se especificamente de redes neurais no domínio de séries temporais, estudos comprovam a competitividade dessas técnicas quando comparadas com métodos do estado da arte, tanto para classificação quanto para regressão extrínseca [Fawaz et al. 2019, Foumani et al. 2023]. Dentre as arquiteturas de redes neurais utilizadas para obter bons desempenhos reportados na literatura, destaca-se o InceptionTime [Fawaz et al. 2020], que demonstrou uma boa capacidade de obter um bom desempenho para séries temporais de variados domínios.

### 2.4.5. Algoritmos Baseados em Comitês

Com esses e muitos outros algoritmos, a área de Aprendizado de Máquina para séries temporais também se enquadra em um clássico problema do Aprendizado de Máquina: o teorema “não há almoço grátis” [Wolpert 2002]. Como cada algoritmo compreende um viés indutivo diferente, não há um algoritmo que melhor desempenhe em quaisquer conjuntos de dados. Comparações experimentais em larga escala [Bagnall et al. 2017] podem mostrar que um algoritmo  $A$  é melhor que  $B$  na média dos resultados para os conjuntos de dados utilizados no experimento, mas sempre há uma fração dos conjuntos de dados nos quais  $B$  desempenha melhor que  $A$ .

Os vieses intrínsecos a modelos de Aprendizado de Máquina podem ser reduzidos combinando-os em modelos coletivos. Ao ajustar diferentes modelos suficientemente variados a um conjunto de dados e rotular uma instância levando-se em conta o voto de cada um, é possível obter um desempenho melhor do que aquele obtido com cada modelo individual. Essa estratégia é comumente referenciada como comitê.

Um dos primeiros modelos de comitê para classificação de séries temporais foi o *Collective of Transformation-Based Ensembles* (COTE, ou Flat-COTE) [Bagnall et al. 2015], que combinou modelos construídos a partir de diversos domínios das séries temporais: tempo, frequência, mudança e *shapelets*. Pouco adiante, foi proposto o *Hierarchical Vote Collective of Transformation-based Ensembles* (HIVE-COTE) [Lines et al. 2018], que melhorou o Flat-COTE ao adicionar uma estrutura hierárquica com votos probabilísticos, dois novos classificadores nos domínios pré-existentes e modelos em dois novos domínios. Os domínios considerados pelo HIVE-COTE são ilustrados por séries simuladas na Figura 2.16.

Por fim, o mais recente HIVE-COTE 2.0 [Middlehurst et al. 2021] substituiu dois classificadores já existentes e adicionou um novo, melhorando ainda mais o desempenho do algoritmo nos conjuntos de dados disponíveis.

## 2.5. Exemplos de Tarefas de Classificação e Regressão Extrínseca

Esta seção apresenta alguns exemplos práticos de classificação e regressão extrínseca de séries temporais utilizando as ferramentas `tsai` e `aeon`.

### 2.5.1. Conjunto de Dados

Neste capítulo, foram utilizados diferentes conjuntos de dados nos exemplos exibidos nas seções anteriores. Esses conjuntos de dados foram propostos para as tarefas de classificação e regressão extrínseca. Dois deles serão utilizados para realizar demonstrações práticas. Vale ressaltar que optamos por utilizar conjuntos de séries temporais já segmentadas e pré-processadas. No entanto, recomendamos ao leitor a exploração do repositório de dados PhysioBank<sup>3</sup>, que possui uma grande quantidade e diversidade de dados, majoritariamente séries temporais, provenientes da área da Saúde.

Para a tarefa de classificação, será utilizado o conjunto de dados ECG5000, disponibilizado no repositório *UCR Time Series Classification Archive* [Dau et al. 2018]. O conjunto é composto por 5000 trechos de eletrocardiogramas, das quais são encontradas

<sup>3</sup><https://archive.physionet.org/physiobank/database/>



500 séries no conjunto de treino e 4500 séries no conjunto de teste. Cada série é univariada, possui 140 observações e pode estar rotulada em uma das 5 classes do conjunto. Neste contexto, esse conjunto simula um cenário onde a existência de instâncias rotuladas é limitada se comparado ao número de instâncias não rotuladas.

A Figura 2.17 mostra um exemplo de série para cada uma das classes contidas no conjunto ECG5000.

Os conjuntos utilizados na tarefa de regressão são: BIDMCRR, BIDMCHR e BIDMCSpO2, disponibilizados pelo repositório de dados para regressão extrínseca de séries temporais mantido pela universidade de Monash, a UEA e a UCR [Tan et al. 2021]. Os conjuntos dados BIDMCHR e BIDMCSpO2 dividem-se em 5550 instâncias para o conjunto de treino e 2399 instâncias para o conjunto de teste. Já o BIDMCRR, apesar de possui o mesmo número de instâncias no conjunto de teste, compreende 5471 instâncias para treino.

Todos os conjuntos são compostos por séries temporais multivariadas de 4000 observações. As duas variáveis observadas para esses conjuntos de dados são eletrocardiograma e fotopletismografia. A maior diferença entre esses conjuntos de dados se refere ao atributo alvo, sendo a frequência respiratória (BIDMCRR), a frequência cardíaca (BIDMCHR) e a oxigenação no sangue (BIDMCSpO2).

As Figuras 2.18, 2.19 e 2.20 mostram exemplos de instâncias dos conjuntos BIDMCHR, BIDMCRR e BIDMCSpO2, respectivamente. Cada uma das instâncias está acompanhada de seu valor alvo, sendo que foram selecionados exemplos cujos valores para os atributos-alvo fossem bastante distintos.

### 2.5.2. Classificação

A classificação de séries temporais é a tarefa de maior foco neste capítulo. A tarefa de classificação pode ser definida, em alto nível, como a indução de um modelo capaz de inferir rótulos para novas observações de um fenômeno. Por exemplo, considere um eletrocardiograma com anotações relativas à apneia. A tarefa de classificação consiste em induzir um modelo a partir desse conjunto de dados anotados e, para um novo paciente, utilizá-lo para inferir, a cada batimento, a presença ou ausência de apneia.

Historicamente, a classificação de séries temporais por meio de algoritmos muito simples se mostrou muito eficiente e eficaz. Como exemplo, o simples algoritmo do vizinho mais próximo foi utilizado com sucesso em uma ampla gama de aplicações. Na última década, inúmeros algoritmos surgiram para essa tarefa. A literatura de classificação de séries temporais tem buscado categorizar esses métodos e entende-se que há ao menos quatro categorias principais de algoritmos: baseados em distância, intervalos, dicionários e aprendizado profundo, já descritos na Seção 2.4. Além disso, há algoritmos que usam ideias contidas em mais de uma dessas categorias, além de algoritmos baseados em comitês de classificadores [Middlehurst et al. 2021].

É importante retomar o fato que há um compromisso entre custo computacional e eficácia de cada algoritmo [Ruiz et al. 2020]. Somado a isso, considera-se o teorema do “não há almoço grátis” [Wolpert 2002]. Sendo assim, é necessário conhecer técnicas baseadas em estimar o desempenho de diferentes algoritmos utilizando o conjunto de

treinamento, para ser decidido qual algoritmo utilizar nas próximas etapas. No exemplo dado a seguir, será respeitada a divisão de treino e teste do conjunto de dados. Assim, o classificador será induzido apenas com o conjunto de treino e será avaliado apenas com o conjunto de teste.

A seguir serão mostrados exemplos práticos de como utilizar cada um destes algoritmos com o conjunto de dados ECG5000.

O Código-Fonte 2.5 mostra um exemplo de aplicação do algoritmo baseado em distância do vizinho mais próximo. Neste caso, se aplica o algoritmo disponível na biblioteca `aeon` no conjunto de treino, instanciando-se os parâmetros para a classificação. Assim, quando se realiza uma predição no conjunto de testes, cada instância não rotulada tem sua similaridade calculada com base nas instâncias de treino. Baseando-se nas distâncias para um número de vizinhos estipulado pelo parâmetro  $k$ , o algoritmo vai determinar a qual rótulo aquela instância de teste pertence.

#### Código-Fonte 2.5. Classificação com o Algoritmo KNN.

```

1 from aeon.classification.distance_based import
    KNeighborsTimeSeriesClassifier
2 import numpy as np
3
4 ecg5000_train = np.genfromtxt('ECG5000_TRAIN.tsv',
    delimiter='\t')
5 ecg5000_test = np.genfromtxt('ECG5000_TEST.tsv',
    delimiter='\t')
6
7 x_train, y_train = ecg5000_train[1:], ecg5000_train[0]
8 x_test, y_test = ecg5000_test[1:], ecg5000_test[0]
9
10 knn = KNeighborsTimeSeriesClassifier()
11 knn.fit(x_train, y_train)
12
13 y_pred = knn.predict(x_test)
14
15 print(accuracy_score(y_test, y_pred))

```

Ao final do código é possível notar o cálculo da acurácia, que determina quantas instâncias de cada classe o algoritmo previu corretamente. A Figura 2.21 mostra a matriz de confusão para o algoritmo KNN no conjunto de dados. Nela é possível observar cada rótulo que o algoritmo previu corretamente, e nos casos de erro qual foi o rótulo atribuído pelo algoritmo para aquela instância no lugar do seu rótulo real. Como ocorre um desbalanceamento de classes neste conjunto, onde a maioria das classes são pertencentes aos rótulos 1 e 2, as outras classes que possuem poucos exemplos concentram grande parte dos erros do algoritmo. Desta forma, há pouco volume de instâncias nestas classes para o algoritmo aprender alguma espécie de padrão que permita uma predição correta.

No Código-Fonte 2.6 é possível observar a aplicação do modelo de aprendizado profundo InceptionTime para classificação no conjunto de dados ECG5000. A implementação baseia-se na utilização da biblioteca `tsai` [Oguiza 2022]. Note que com poucas

linhas de código é possível aplicar um algoritmo do estado-da-arte para classificação de séries temporais. O InceptionTime baseia-se em *InceptionBlocks* para capturar padrões e relações lineares e não lineares na estrutura das séries temporais. Contudo para utilizá-lo é preciso transformar as séries que normalmente são carregadas em um *array* de duas dimensões para três dimensões, sendo elas: o número de instâncias, o número de canais e a última dimensão o tamanho da série ou a quantidade de observações.

### Código-Fonte 2.6. Classificação com o InceptionTime.

```

1 from tsai.all import *
2 import numpy as np
3
4 ecg5000_train = np.genfromtxt('ECG5000_TRAIN.tsv',
5     delimiter='\t')
6 ecg5000_test = np.genfromtxt('ECG5000_TEST.tsv',
7     delimiter='\t')
8
9 x_train, y_train = ecg5000_train[1:], ecg5000_train[0]
10 x_test, y_test = ecg5000_test[1:], ecg5000_test[0]
11
12 x_train = np.reshape(x_train, (x_train.shape[0], 1,
13     x_train.shape[1]))
14 x_test = np.reshape(x_test, (x_test.shape[0], 1,
15     x_test.shape[1]))
16
17 train_ds = TSDataset(x_train, y_train - 1, types=(TSTensor,
18     TSLabelTensor))
19 train_dl = DataLoader(train_ds, bs=128, num_workers=0)
20
21 test_ds = TSDataset(x_test, y_test - 1, types=(TSTensor,
22     TSLabelTensor))
23 test_dl = DataLoader(test_ds, bs=128, num_workers=0)
24 dls = DataLoaders(train_dl, test_dl,
25     device=default_device())
26
27 num_classes = len(np.unique(y_train))
28 model = InceptionTime(1, num_classes)
29
30 learn = Learner(dls, model, loss_func=nn.CrossEntropyLoss(),
31     metrics=accuracy)
32 learn.fit_one_cycle(25, lr_max=1e-2)
33
34 preds, labels = learn.get_preds(dl=test_dl)
35 print(accuracy_score(preds.argmax(dim=1).numpy(),
36     labels.numpy()))

```

A Figura 2.22 mostra a matriz de confusão resultante da predição do InceptionTime realizada nos dados de teste do conjunto ECG5000. Assim como na matriz de confusão do KNN, é possível notar que o comportamento é semelhante. Com métricas muito

semelhantes, necessita-se uma análise de outros fatores para a escolha dos modelos, tais como: custo computacional, volume de dados necessários, arquitetura da solução, entre outros fatores externos.

No caso da comparação entre o algoritmo de vizinho mais próximo e o InceptionTime, sempre que uma nova instância é inserida para a predição, o KNN precisa calcular todas as distâncias desta nova instância e das existentes para determinar o seu rótulo. Essa tarefa por muitas vezes pode ser custosa e tornar-se computacionalmente inviável em grandes volumes de dados. Porém, no caso do InceptionTime, o maior custo computacional está no treinamento da rede neural, uma vez treinada, para cada instância serão somente utilizados os pesos internos da rede para determinar seu rótulo, eliminando a necessidade de recalculá-las. Desta forma, mesmo com métricas e resultados semelhantes, é possível determinar um modelo que melhor se adéqua ao caso proposto, contudo vale lembrar que não existe um modelo melhor para todos os casos.

### 2.5.3. Regressão extrínseca

Com o crescente interesse em séries temporais nas últimas décadas, que vem se intensificando nos últimos anos, pesquisadores propuseram dezenas de algoritmos específicos para a classificação [Ruiz et al. 2020] e outras tarefas, como *forecasting*, de séries temporais. Por outro lado, há ainda desafios não superados pela literatura para este tipo de dados. Um exemplo se dá pela lacuna por diferentes tarefas de aprendizado, como a regressão extrínseca de séries temporais. Por “extrínseca”, entende-se que o alvo da predição é um valor externo à série, ao contrário do que acontece com o *forecasting*. Por exemplo, ao considerar séries temporais de fotopletismografia, a regressão extrínseca teria por objetivo estimar parâmetros clínicos, como a saturação de oxigênio no sangue ou concentração de hemoglobina.

Recentemente, um grupo de pesquisadores notou esse fato e constatou que a tarefa de regressão é uma necessidade em diversos domínios de aplicação [Tan et al. 2021]. Dessa forma, criaram um repositório de conjuntos de dados de regressão em séries temporais e realizaram experimentos iniciais com diversos algoritmos<sup>4</sup>. A partir dessa iniciativa, foram propostas diversas novas técnicas para essa tarefa, incluindo algoritmos tradicionais aplicados a características extraídas das séries [Gay et al. 2021], adaptações de algoritmos bem estabelecidos em classificação [Guijo-Rubio et al. 2023] e de arquiteturas de redes neurais [Foumani et al. 2023].

Alguns algoritmos de classificação de séries temporais podem ser convertidos para lidar com regressão extrínseca de forma relativamente simples. Isso se deve ao fato de que muitos desses algoritmos consistem de uma ou mais etapas de transformação seguidas de um classificador final nos dados transformados.

Esse é o caso de um dos algoritmos mais bem avaliados em regressão extrínseca: o Random Convolutional Kernel Transform (ROCKET). O ROCKET utiliza um grande número de filtros convolucionais (em geral, 10000), construídos aleatoriamente a partir de diversos parâmetros, aplicando-os às séries temporais e extraíndo características da série resultante, por padrão, valor máxima (max) e proporção de valores positivos (ppv). Essas

<sup>4</sup><http://tseregression.org/>

características formam uma grande tabela atributo-valor, que é então usada como entrada para um classificador linear.

Para adaptar o ROCKET à regressão extrínseca, basta substituir o classificador linear na ponta final por um regressor linear. No Código-Fonte 2.7, por exemplo, mostra-se a utilização de um regressor ROCKET utilizando a biblioteca `aeon` para a predição de frequência cardíaca por meio do conjunto de dados BIDMCHR.

#### Código-Fonte 2.7. Regressão com o ROCKET.

```

1 from aeon.datasets import load_from_tsfile
2 from aeon.transformations.panel.rocket import Rocket
3
4 from sklearn.pipeline import Pipeline
5 from sklearn.linear_model import RidgeCV
6 from sklearn.metrics import mean_squared_error
7
8 X_train, y_train = load_from_tsfile('BIDMCHR_TRAIN.ts')
9 X_test, y_test = load_from_tsfile('BIDMCHR_TEST.ts')
10
11 model = Pipeline([
12     ('transformer', Rocket()),
13     ('regressor', RidgeCV())
14 ])
15 model.fit(X_train, y_train)
16 y_pred = model.predict(X_test)
17
18 print(mean_squared_error(y_test, y_pred, squared=False))

```

Ao final, o Código-Fonte imprime o valor do erro quadrático médio entre os valores observados e os valores preditos. A título de reforçar o exemplo da utilização da biblioteca `aeon` para regressão extrínseca, o Código-Fonte 2.8 mostra como realizar a predição de valores de frequência respiratória por meio do conjunto de dados BIDMCRR.

#### Código-Fonte 2.8. Exemplo de utilização da biblioteca Aeon.

```

1 from aeon.datasets import load_from_tsfile
2 from aeon.regression.distance_based import \
3     KNeighborsTimeSeriesRegressor
4
5 # Carregar o dataset para a memoria
6 X_train, y_train = load_from_tsfile('BIDMCRR_TRAIN.ts')
7 X_test, y_test = load_from_tsfile('BIDMCRR_TEST.ts')
8
9 # Definir o modelo
10 model = KNeighborsTimeSeriesRegressor(
11     distance='dtw',
12     n_neighbors=3
13 )
14
15 # Ajustar o modelo aos dados

```

```

16 model.fit(X_train, y_train)
17
18 # Obter as previsões
19 y_pred = model.predict(X_test)
20
21 print(mean_squared_error(y_test, y_pred, squared=False))

```

## 2.6. Considerações Finais

A área de Aprendizado de Máquina para séries temporais está em plena expansão. Graças ao surgimento de novas demandas e tecnologias na área da Saúde, como saúde móvel (*mHealth*), esse domínio de conhecimento tem sido um dos principais motivadores para esse crescimento. Para que essas aplicações em Saúde possam estar cada vez mais presentes no nosso dia-a-dia, ainda é necessário trabalhar em propostas e avaliação tanto de novas técnicas quanto novas aplicações.

No entanto, o estado-da-arte em classificação e regressão extrínseca de séries temporais enfrenta um problema relacionado ao compromisso entre eficiência e eficácia. Os estudos experimentais mais recentes para essas tarefas apontam para o fato que os algoritmos mais precisos são também os mais custosos.

Ao se utilizar séries temporais como medições de sinais para auxiliar médicos e pacientes com os cuidados da saúde, é preciso que as previsões realizadas pelos modelos de Aprendizado de Máquina sejam corretas. Por isso, é necessário se investigar por algoritmos que sejam menos custosos e possam fornecer respostas iguais ou melhores do que os de maior custo.

Uma tendência na literatura recente da área é o aparecimento de novas propostas no contexto de aprendizado profundo. Porém, muitas das arquiteturas investigadas nesse domínio são adaptações diretas de arquiteturas projetadas para outros tipos de dados, como imagens e vídeos. Para avançar significativamente na criação de modelos neurais para séries temporais, é necessário investigar por técnicas específicas para esse tipo de dados.

Ainda, uma vez que há um custo considerável para a obtenção de dados rotulados no domínio da Saúde, há um grande espaço para se pesquisar por técnicas que trabalhem com diferentes suposições de rótulos. Por exemplo, a criação de modelos a partir de pouco volume de dados, de aprendizado semissupervisionado ou autossupervisionado, aprendizado de uma classe, entre outros.

Por fim, os autores gostariam de agradecer ao apoio a este trabalho. O desenvolvimento deste documento só foi possível graças aos auxílios relacionados aos processos nº 2022/03176-1, nº 2022/00305-5, nº 2023/02680-0 e nº 2023/05171-0, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

## Referências

[Alsuliman et al. 2020] Alsuliman, T., Humaidan, D., and Sliman, L. (2020). Machine learning and artificial intelligence in the service of medicine: Necessity or potentiality?

*Current research in translational medicine*, 68(4):245–251.

- [Ang and Seng 2016] Ang, L.-M. and Seng, K. P. (2016). Big sensor data applications in urban environments. *Big Data Research*, 4:1–12.
- [Bagnall et al. 2017] Bagnall, A., Lines, J., Bostrom, A., Large, J., and Keogh, E. (2017). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery*, 31:606–660.
- [Bagnall et al. 2015] Bagnall, A., Lines, J., Hills, J., and Bostrom, A. (2015). Time-series classification with cote: The collective of transformation-based ensembles. *IEEE Transactions on Knowledge and Data Engineering*, 27(9):2522–2535.
- [Barandas et al. 2020] Barandas, M., Folgado, D., Fernandes, L., Santos, S., Abreu, M., Bota, P., Liu, H., Schultz, T., and Gamboa, H. (2020). Tsfel: Time series feature extraction library. *SoftwareX*, 11:100456.
- [Becker 2006] Becker, D. E. (2006). Fundamentals of electrocardiography interpretation. *Anesthesia progress*, 53(2):53–64.
- [Bloomfield 2004] Bloomfield, P. (2004). *Fourier analysis of time series: an introduction*. John Wiley & Sons.
- [Christ et al. 2018] Christ, M., Braun, N., Neuffer, J., and Kempa-Liehr, A. W. (2018). Time series feature extraction on basis of scalable hypothesis tests (tsfresh—a python package). *Neurocomputing*, 307:72–77.
- [Dau et al. 2018] Dau, H. A., Keogh, E., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., Yanping, Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G., and Hexagon-ML (2018). The ucr time series classification archive. [https://www.cs.ucr.edu/~eamonn/time\\_series\\_data\\_2018/](https://www.cs.ucr.edu/~eamonn/time_series_data_2018/).
- [Deng et al. 2013] Deng, H., Runger, G., Tuv, E., and Vladimir, M. (2013). A time series forest for classification and feature extraction. *Information Sciences*, 239:142–153.
- [Durak and Arikan 2003] Durak, L. and Arikan, O. (2003). Short-time fourier transform: two fundamental properties and an optimal implementation. *IEEE Transactions on Signal Processing*, 51(5):1231–1242.
- [Ebrahimi et al. 2020] Ebrahimi, Z., Loni, M., Daneshtalab, M., and Gharehbaghi, A. (2020). A review on deep learning methods for ecg arrhythmia classification. *Expert Systems with Applications: X*, 7:100033.
- [El-Hajj and Kyriacou 2020] El-Hajj, C. and Kyriacou, P. A. (2020). A review of machine learning techniques in photoplethysmography for the non-invasive cuff-less measurement of blood pressure. *Biomedical Signal Processing and Control*, 58:101870.
- [El Maachi et al. 2020] El Maachi, I., Bilodeau, G.-A., and Bouachir, W. (2020). Deep 1d-convnet for accurate parkinson disease detection and severity prediction from gait. *Expert Systems with Applications*, 143:113075.

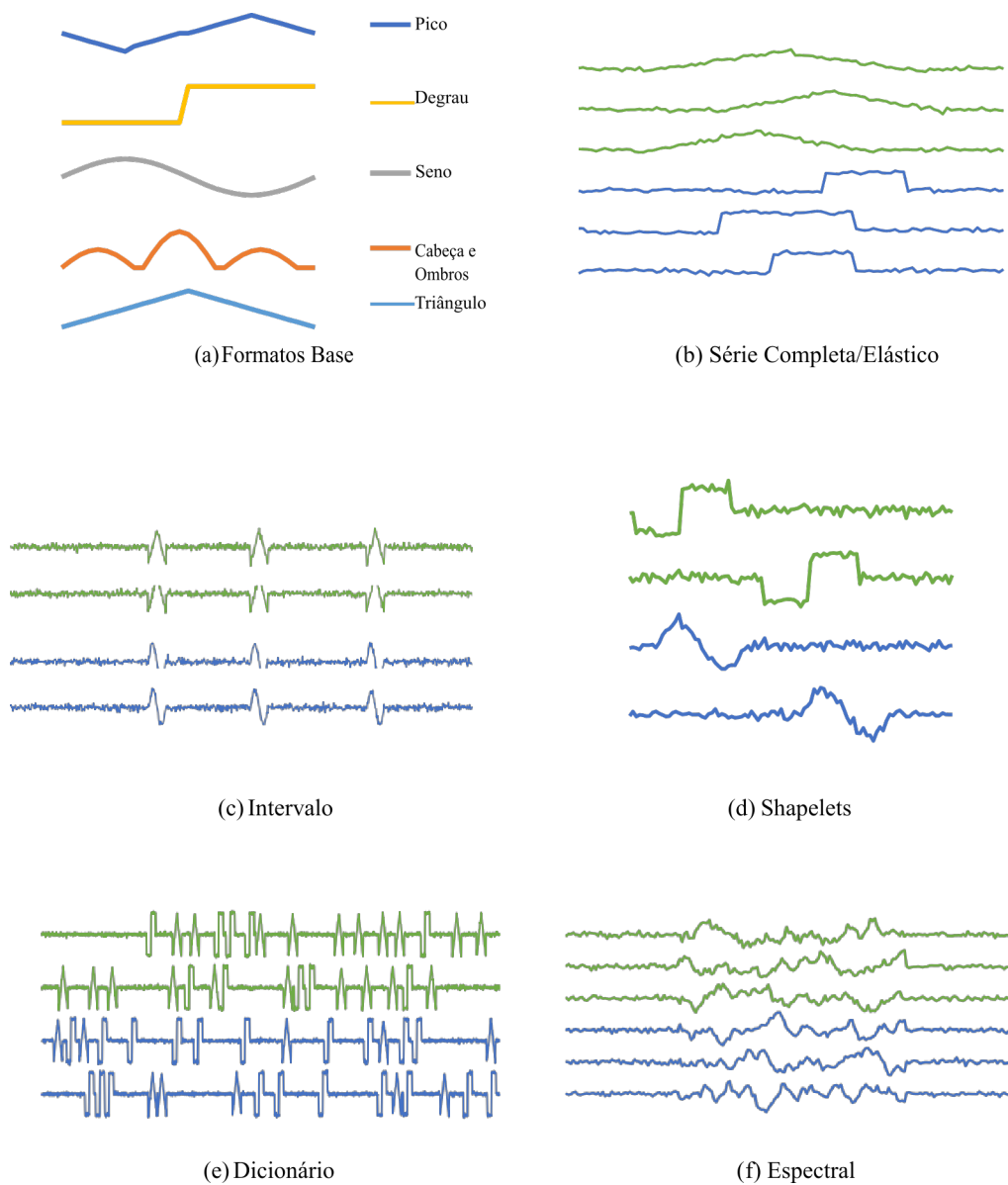
- [Faceli et al. 2020] Faceli, K., Lorena, A. C., Gama, J., Almeida, T. A., and Carvalho, A. C. P. L. F. (2020). *Inteligência Artificial—uma abordagem de aprendizado de máquina*. Rio de Janeiro: LTC, 2 edition.
- [Fawaz et al. 2019] Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963.
- [Fawaz et al. 2020] Fawaz, H. I., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., Webb, G. I., Idoumghar, L., Muller, P.-A., and Petitjean, F. (2020). InceptionTime: Finding AlexNet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962.
- [Foumani et al. 2023] Foumani, N. M., Miller, L., Tan, C. W., Webb, G. I., Forestier, G., and Salehi, M. (2023). Deep learning for time series classification and extrinsic regression: A current survey. *arXiv preprint arXiv:2302.02515*.
- [Fulcher and Jones 2017] Fulcher, B. D. and Jones, N. S. (2017). hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. *Cell systems*, 5(5):527–531.
- [Gay et al. 2021] Gay, D., Bondu, A., Lemaire, V., and Boullé, M. (2021). Interpretable feature construction for time series extrinsic regression. In *Advances in Knowledge Discovery and Data Mining: 25th Pacific-Asia Conference, PAKDD 2021, Virtual Event, May 11–14, 2021, Proceedings, Part I*, pages 804–816. Springer.
- [Guijo-Rubio et al. 2023] Guijo-Rubio, D., Middlehurst, M., Arcencio, G., Silva, D. F., and Bagnall, A. (2023). Unsupervised feature based algorithms for time series extrinsic regression. *arXiv preprint arXiv:2305.01429*.
- [Herrmann et al. 2023] Herrmann, M., Tan, C. W., Salehi, M., and Webb, G. I. (2023). Proximity forest 2.0: A new effective and scalable similarity-based classifier for time series.
- [Hong et al. 2020] Hong, S., Zhou, Y., Shang, J., Xiao, C., and Sun, J. (2020). Opportunities and challenges of deep learning methods for electrocardiogram data: A systematic review. *Computers in biology and medicine*, 122:103801.
- [Hu et al. 2019] Hu, Y., Ji, C., Zhang, Q., Chen, L., Zhan, P., and Li, X. (2019). A novel multi-resolution representation for time series sensor data analysis. *Soft Computing*, pages 1–26.
- [Kashani and Barold 2005] Kashani, A. and Barold, S. S. (2005). Significance of qrs complex duration in patients with heart failure. *Journal of the American College of Cardiology*, 46(12):2183–2192.
- [Köppen 2000] Köppen, M. (2000). The curse of dimensionality. In *World Conference on Soft Computing in Industrial Applications*, volume 1, pages 4–8.



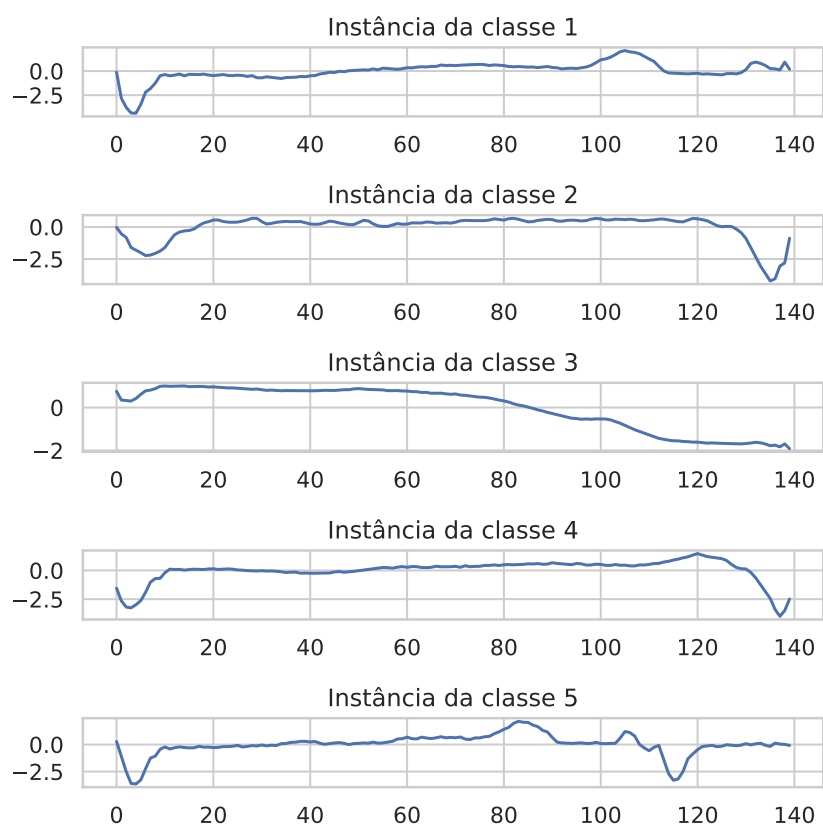
- [Kwapisz et al. 2011] Kwapisz, J. R., Weiss, G. M., and Moore, S. A. (2011). Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82.
- [Lee et al. 2022] Lee, J., Yeom, I., Chung, M. L., Kim, Y., Yoo, S., and Kim, E. (2022). Use of mobile apps for self-care in people with parkinson disease: systematic review. *JMIR mHealth and uHealth*, 10(1):e33944.
- [Lines and Bagnall 2015] Lines, J. and Bagnall, A. (2015). Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery*, 29(3):565–592.
- [Lines et al. 2018] Lines, J., Taylor, S., and Bagnall, A. (2018). Time series classification with hive-cote: The hierarchical vote collective of transformation-based ensembles. *ACM Transactions on Knowledge Discovery from Data*, 12(5):1–35.
- [Lubba et al. 2019] Lubba, C. H., Sethi, S. S., Knaute, P., Schultz, S. R., Fulcher, B. D., and Jones, N. S. (2019). catch22: Canonical time-series characteristics: Selected through highly comparative time-series analysis. *Data Mining and Knowledge Discovery*, 33(6):1821–1852.
- [Lucas et al. 2019] Lucas, B., Shifaz, A., Pelletier, C., O’Neill, L., Zaidi, N., Goethals, B., Petitjean, F., and Webb, G. I. (2019). Proximity forest: an effective and scalable distance-based classifier for time series. *Data Mining and Knowledge Discovery*, 33(3):607–635.
- [Martinez-Ríos et al. 2022] Martinez-Ríos, E., Montesinos, L., and Alfaro-Ponce, M. (2022). A machine learning approach for hypertension detection based on photoplethysmography and clinical data. *Computers in Biology and Medicine*, 145:105479.
- [Mazzu-Nascimento et al. 2020] Mazzu-Nascimento, T., de Oliveira Leal, Â. M., Nogueira-de Almeida, C. A., de Avó, L. R. d. S., Carrilho, E., and Silva, D. F. (2020). Noninvasive self-monitoring of blood glucose at your fingertips, literally!: Smartphone-based photoplethysmography. *International Journal of Nutrology*, 13(02):048–052.
- [Middlehurst and Bagnall 2022] Middlehurst, M. and Bagnall, A. (2022). The fresh-prince: A simple transformation based pipeline time series classifier. In *Pattern Recognition and Artificial Intelligence: Third International Conference, ICPRAI 2022, Paris, France, June 1–3, 2022, Proceedings, Part II*, pages 150–161. Springer.
- [Middlehurst et al. 2020] Middlehurst, M., Large, J., and Bagnall, A. (2020). The canonical interval forest (cif) classifier for time series classification. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 188–195.
- [Middlehurst et al. 2021] Middlehurst, M., Large, J., Flynn, M., Lines, J., Bostrom, A., and Bagnall, A. (2021). Hive-cote 2.0: a new meta ensemble for time series classification. *Machine Learning*, 110(11-12):3211–3243.
- [Mitchell 1997] Mitchell, T. M. (1997). Machine learning.

- [Morid et al. 2023] Morid, M. A., Sheng, O. R. L., and Dunbar, J. (2023). Time series prediction using deep learning methods in healthcare. *ACM Transactions on Management Information Systems*, 14(1):1–29.
- [Oguiza 2022] Oguiza, I. (2022). tsai - a state-of-the-art deep learning library for time series and sequential data. Github.
- [Pereira et al. 2020] Pereira, T., Tran, N., Gadhoumi, K., Pelter, M. M., Do, D. H., Lee, R. J., Colorado, R., Meisel, K., and Hu, X. (2020). Photoplethysmography based atrial fibrillation detection: a review. *NPJ digital medicine*, 3(1):3.
- [Perpetuini et al. 2021] Perpetuini, D., Chiarelli, A. M., Cardone, D., Filippini, C., Rinnella, S., Massimino, S., Bianco, F., Bucciarelli, V., Vinciguerra, V., Fallica, P., et al. (2021). Prediction of state anxiety by machine learning applied to photoplethysmography data. *PeerJ*, 9:e10448.
- [Ruiz et al. 2020] Ruiz, A. P., Flynn, M., Large, J., Middlehurst, M., and Bagnall, A. (2020). The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, pages 1–49.
- [Salari et al. 2022] Salari, N., Hosseinian-Far, A., Mohammadi, M., Ghasemi, H., Khaizaie, H., Daneshkhah, A., and Ahmadi, A. (2022). Detection of sleep apnea using machine learning algorithms based on ecg signals: A comprehensive systematic review. *Expert Systems with Applications*, 187:115950.
- [Schäfer 2015] Schäfer, P. (2015). The boss is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery*, 29(6):1505–1530.
- [Schäfer and Höggqvist 2012] Schäfer, P. and Höggqvist, M. (2012). Sfa: a symbolic fourier approximation and index for similarity search in high dimensional datasets. In *EDBT '12: Proceedings of the 15th International Conference on Extending Database Technology*, pages 516–527.
- [Silva 2017] Silva, D. F. (2017). *Large-Scale Similarity-Based Time Series Mining*. PhD thesis, Universidade de São Paulo, São Carlos.
- [Silva et al. 2023] Silva, D. F., de M. Júnior, J. G. B., Domingues, L. V., and Mazzu-Nascimento, T. (2023). Hemoglobin estimation from smartphone-based photoplethysmography with small data. In *Computer-Based Medical Systems*, page No prelo. IEEE.
- [Silva et al. 2018] Silva, D. F., Giusti, R., Keogh, E., and Batista, G. E. (2018). Speeding up similarity search under dynamic time warping by pruning unpromising alignments. *Data Mining and Knowledge Discovery*, 32:988–1016.
- [Tan et al. 2020] Tan, C. W., Bergmei, C., Petitjean, F., Schmidt, D., Webb, G. I., Bagnall, A., and Keogh, E. (2020). The monash, uea & ucr time series extrinsic regression archivee. <http://tseregression.org/>.

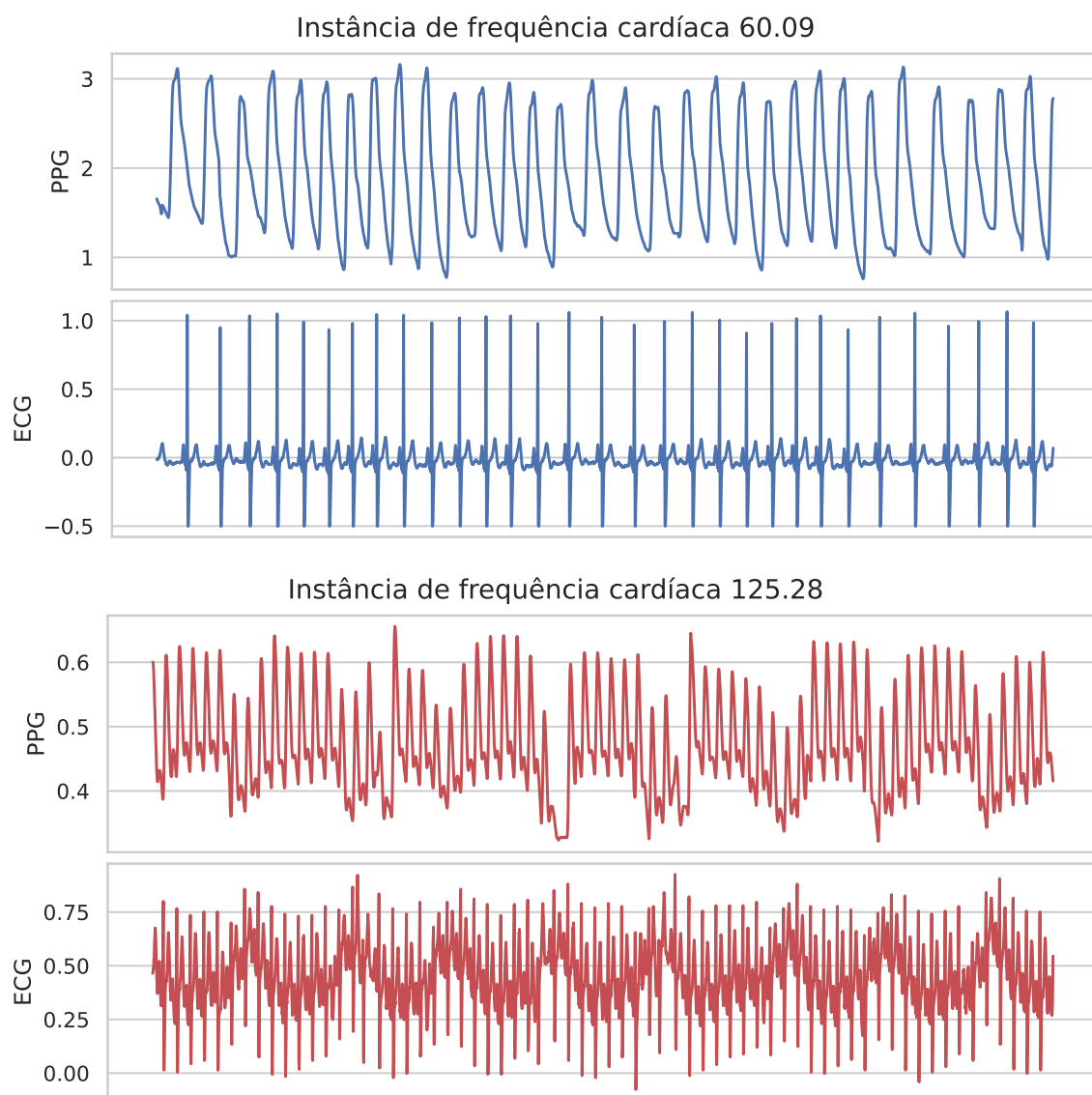
- [Tan et al. 2021] Tan, C. W., Bergmeir, C., Petitjean, F., and Webb, G. I. (2021). Time series extrinsic regression. *Data Mining and Knowledge Discovery*, pages 1–29.
- [Thoms et al. 2017] Thoms, L.-J., Colicchia, G., and Girwidz, R. (2017). Phonocardiography with a smartphone. *Physics Education*, 52(2):023004.
- [WHO 2011] WHO, W. H. O. (2011). mhealth: new horizons for health through mobile technologies. *mHealth: new horizons for health through mobile technologies*.
- [Wolpert 2002] Wolpert, D. H. (2002). The supervised learning no-free-lunch theorems. *Soft computing and industry*, pages 25–42.
- [Yeh et al. 2018] Yeh, C.-C. M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H. A., Zimmerman, Z., Silva, D. F., Mueen, A., and Keogh, E. (2018). Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile. *Data Mining and Knowledge Discovery*, 32(1):83–123.



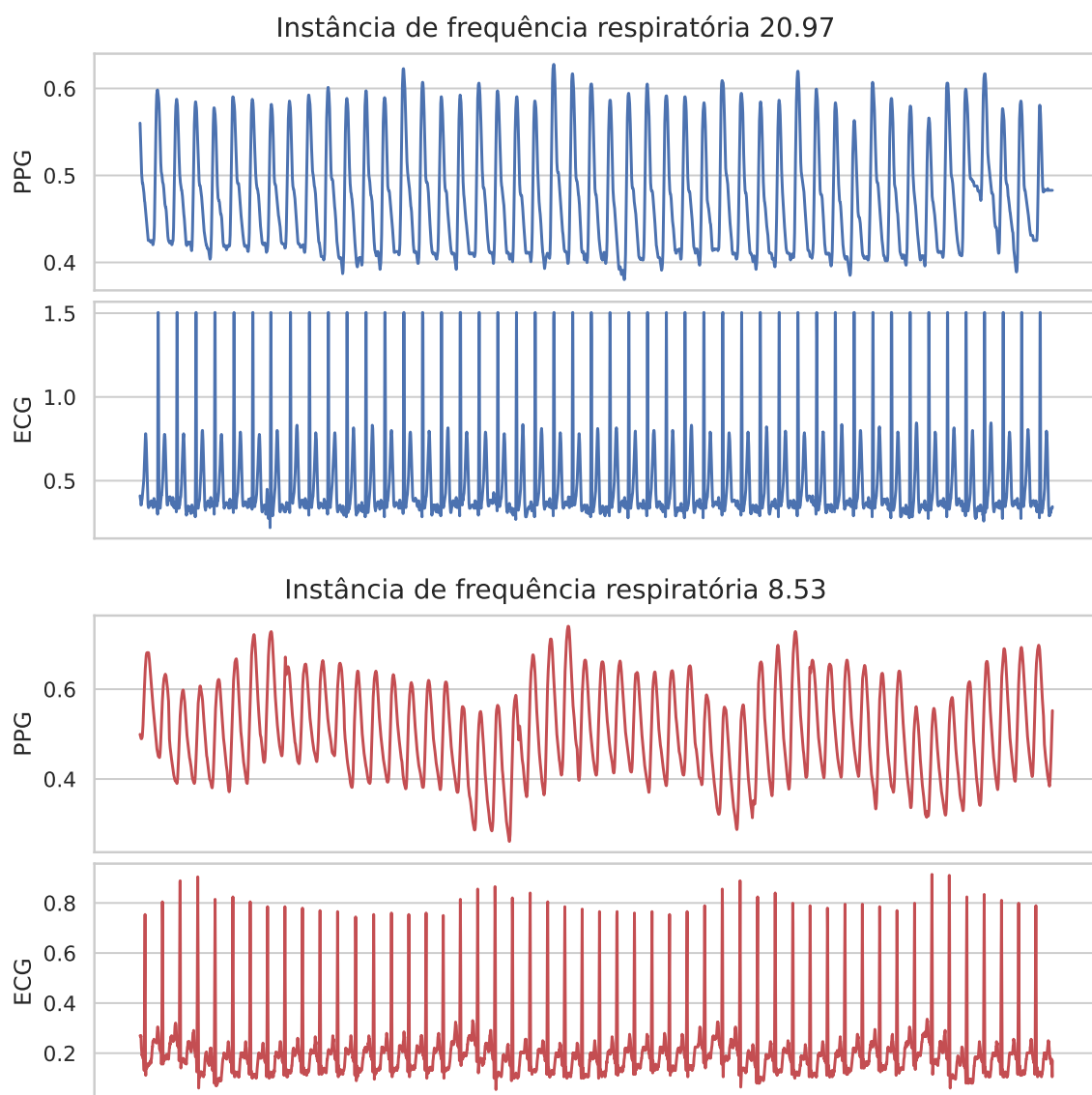
**Figura 2.16. Ilustração dos domínios considerados pelos modelos do HIVE-COTE. Adaptado de [Lines et al. 2018].**



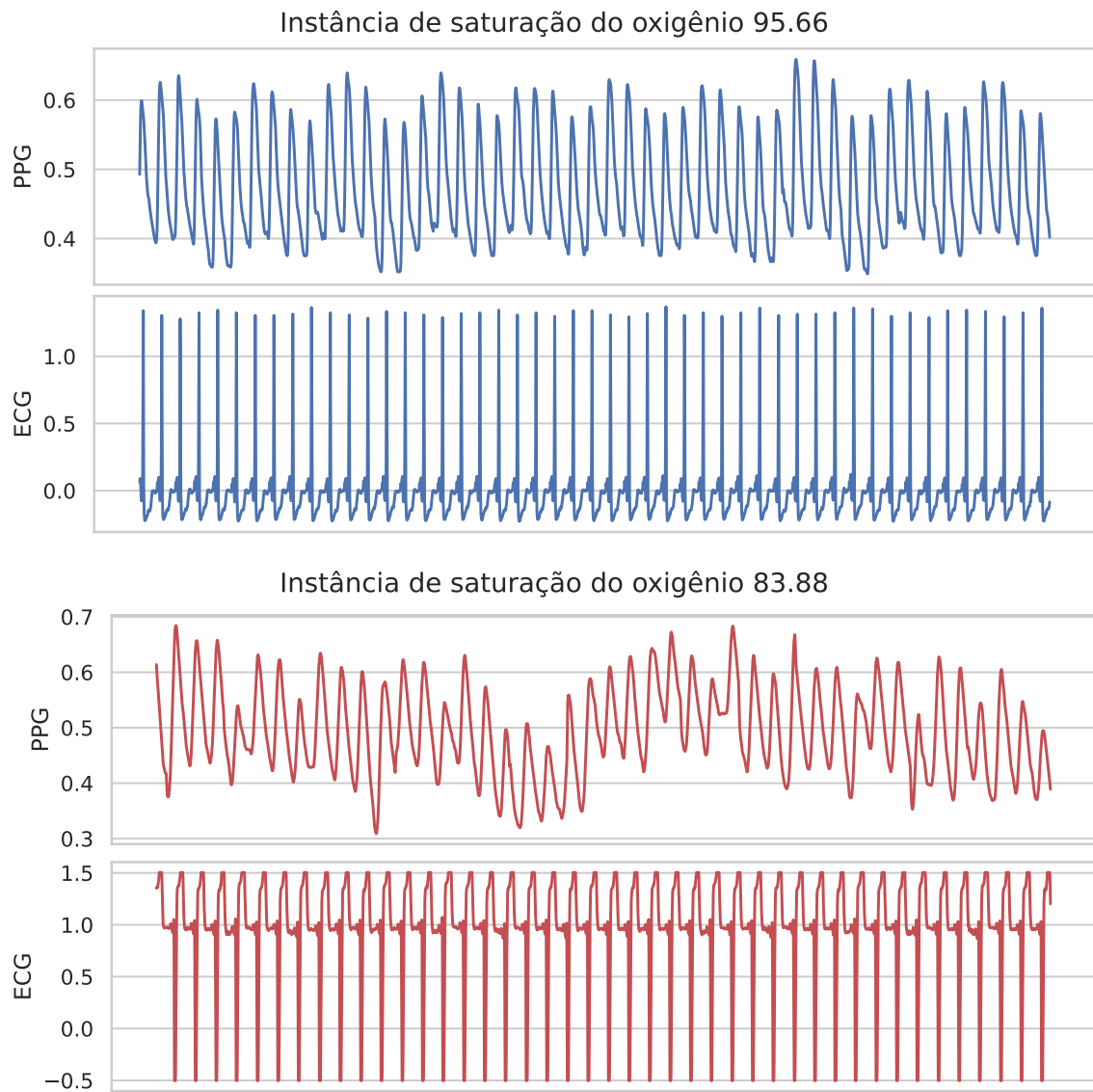
**Figura 2.17. Exemplos de instâncias contidas em cada classe do conjunto de dados ECG5000.**



**Figura 2.18. Exemplos de instâncias do conjunto de dados BIDMCHR com suas respectivas frequências cardíacas.**

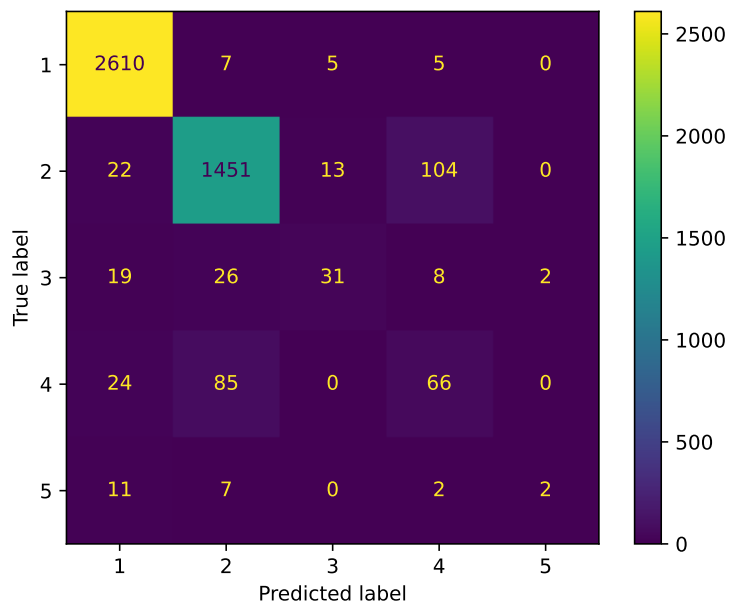


**Figura 2.19. Exemplos de instâncias do conjunto de dados BIDMCRR com suas respectivas frequências respiratórias.**

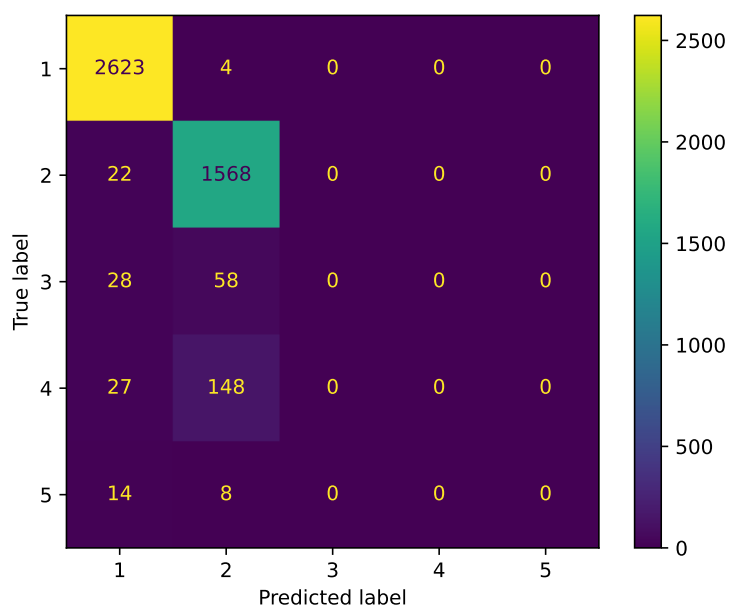


**Figura 2.20.** Exemplos de instâncias do conjunto de dados BIDMCSpO2 com suas respectivas saturações do oxigênio.





**Figura 2.21. Matriz de Confusão gerada pelo Algoritmo KNN.**



**Figura 2.22. Matriz de Confusão gerada pelo InceptionTime após treinamento no conjunto de dados ECG5000.**