UNIVERSIDADE FEDERAL DO MARANHÃO

Programa de Pós-Graduação em Ciência da Computação

Alexandre de Carvalho Araújo

Metodologia baseada em aprendizado profundo para agrupamento de séries temporais univariadas

São Luís
2021

UNIVERSIDADE FEDERAL DO MARANHÃO

Programa de Pós-Graduação em Ciência da Computação

Alexandre de Carvalho Araújo

Metodologia baseada em aprendizado profundo para agrupamento de séries temporais univariadas

São Luís - MA

2021

Alexandre de Carvalho Araújo

Metodologia baseada em aprendizado profundo para agrupamento de séries temporais univariadas

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

Programa de Pós-Graduação em Ciência da Computação

Universidade Federal do Maranhão

Orientador: Prof. Dr. Anselmo Cardoso de Paiva

Coorientador: Prof. Dr. João Dallyson Almeida de Sousa

São Luís - MA

2021

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

De Carvalho Araújo, Alexandre.
Metodologia baseada em aprendizado profundo para agrupamento de séries temporais univariadas / Alexandre De Carvalho Araújo. – 2021.
60 p.

Coorientador(a): João Dallyson Almeida de Sousa.
Orientador(a): Anselmo Cardoso de Paiva.
Dissertação (Mestrado) – Programa de Pós-graduação em Ciência da Computação/ccet, Universidade Federal do Maranhão, São Luís, 2021.

1. Agrupamento de séries temporais. 2. Aprendizado profundo. 3. Estimação de número de grupos. I. Almeida de Sousa, João Dallyson. II. Cardoso de Paiva, Anselmo. III. Título.

Alexandre de Carvalho Araújo

Metodologia baseada em aprendizado profundo para agrupamento de séries temporais univariadas

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

Trabalho aprovado em São Luís - MA, 13 de Maio de 2021:

Prof. Dr. Anselmo Cardoso de Paiva
Orientador

Prof. Dr. João Dallyson Almeida de Sousa
Co-Orientador

Prof. Dr. Daniel Lima Gomes Júnior
Instituto Federal do Maranhão

Prof. Dr. Ginalber Luiz de Oliveira Serra
Instituto Federal do Maranhão

São Luís - MA
2021

À minha fámilia, amigos e a todos
que me deram suporte em minha jornada.

Agradecimentos

Agradeço aos meus pais, Maria do Socorro e José Bernardo, por todo o suporte e ajuda que me deram durante minha vida e durante meu tempo no programa de mestrado.

A todos da minha família, que contribuíram para que eu me tornasse quem sou hoje.

A Anselmo Cardoso de Paiva e João Dallyson Sousa de Almeida, pela orientação sem a qual esse trabalho não seria possível.

Aos meus amigos, cuja companhia me alegra e dá paz de espírito.

Aos meus colegas de projeto, pela paciência e colaboração durante nossas atividades.

Aos colegas de laboratório, pelo ambiente de pesquisa, mas também de diversão, que me faz falta durante a pandemia.

A todos aqueles que me ajudaram com caronas, elas ajudaram a diminuir o estresse causado pelos problemas encontrados nesses 2 anos.

Por fim, agradeço a todos aqueles que de alguma maneira moldaram quem eu sou hoje que não foram ainda contemplados.

"Se esperares que a sorte apareça, a vida torna-se maçadora."

(Mikhail Tal)

Resumo

Concessionárias de energia elétrica têm a necessidade de planejar a distribuição de energia de tal maneira a atender a necessidade de seus clientes. Além disso, analisar o consumo dos clientes para detectar irregularidades é uma outra tarefa importante para essas empresas. Para tais tarefas, a descoberta de padrões de consumo é essencial. A descoberta destes padrões se trata de uma tarefa de agrupamento. Existe uma grande quantidade de abordagens para agrupar dados, mas uma área de pesquisa relativamente nova neste campo é o uso de aprendizado profundo. Grande parte do esforço realizado nessa área se direciona a dados estáticos. No entanto aplicações que utilizam dados temporais, como consumo de energia elétrica, se tornam cada dia mais populares e relativamente pouco esforço de pesquisa é realizado nesta área. Este trabalho apresenta uma metodologia baseada na arquitetura Deep Embedded Clustering (DEC), uma abordagem para agrupamento utilizando aprendizado profundo, que foca em dados temporais além de abordar uma das deficiências originais do DEC, a necessidade de definir a priori a quantidade de grupos a serem criados, o que geralmente não é conhecido em problemas reais. Para avaliar a metodologia proposta, foram conduzidos experimentos sobre a estimação do número de grupos em 66 conjunto de dados cujo número de grupos é conhecido e para avaliar a capacidade de descoberta de padrões de consumo, foram realizados experimentos em 2 conjunto de dados de séries de consumo elétricos reais fornecidos por uma concessionária de energia elétrica. Os resultados encontrados apontam que tanto a estimação do número de grupos quanto o agrupamento em si da metodologia proposta são superiores à abordagens encontradas na literatura, indicando que a metodologia pode ser uma ferramenta efetiva para as concessionárias de energia elétrica.

Palavras-chave: Agrupamento de séries temporais, aprendizado profundo, estimação de número de grupos.

Abstract

Electrical energy companies have a need to plan energy distribution to fulfill the needs of it's costumers. They also need to analyze energy consumption to detect irregularities. To perform these tasks, consumption patterns discovery is an essential tool. Discovery of these patterns is a clustering task. There is a great number of approachs to cluster data, but a relatively new research subject for this task is the usage of deep learning. Great effort in this area is allocated to static data. However, applications that use time series data have been growing in the last years and not as much research effort is being made in this area. In this work, we propose a new methodology based in the Deep Embedded Clustering architecture, a popular deep clustering algorithm, to cluster electric consumption time series data while also approaching one of the flaws present on DEC, the need to inform number of clusters a priori, which isn't generally known in real problems. To evaluate our methodology, we conducted tests regarding cluster number estimation in 66 time series datasets with known cluster numbers and to evaluate the consumption pattern discovery, we use 2 real electrical consumption datasets provided by a energy consumption company. Our results show superior performance when compared to approaches found in the literature for both cluster number estimation and clustering, indicating that the proposed methodology may be a effective tool for electrical energy companies.

Keywords: Time series clustering, Deep Clustering, Cluster number estimation, Electrical energy consumption.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ADAM	*Adaptative Moment Estimation*
AE	*AutoEncoder*
CIB	Critério de Informação Bayesiano
DBA	*DTW barycenter averaging*
DEC	*Deep Embbeded Clustering*
DeD	*Depth of Data*
DESC	*Deep Embedded Shape Clustering*
DTC	*Deep Temporal Clustering*
DTCR	*Deep Temporal Clustering Representation*
DTW	*Dynamic Time Warp*
GMVAE	*Gaussian Mixture Variational AutoEncoder*
IDFT	Inversa Da Transformada discreta de Fourier
KL	Divergência Kullback-Leibler
L	Função de perda
ReLU	*Rectified Linear Units*
RMSProp	*Root Mean Square Propagation*
SGD	*Stochastic Gradient Descent*
TPE	*Tree-structured Parzen Estimator*
VaDE	*Variational Deep Embedding*

Sumário

1 Introdução

Conjuntos de dados que apresentam uma sequência de valores ordenados pelo tempo são chamados de séries temporais (JAVED; LEE; RIZZO, 2020). Esse tipo de dado está presente em vários campos, como ciclos sazonais de dengue (POLWIANG, 2020), dados climáticos (SOARES et al., 2018) e consumo de energia (JOHANNESEN; KOLHE; GOODWIN, 2019). Aplicações que utilizam este tipo de dado geralmente realizam análises baseadas em técnicas estatísticas (LIAO, 2005), como agrupamento ou detecção de anomalia (BRAEI; WAGNER, 2020). Durante a aquisição desses dados, dependendo de quantas fontes de aquisição, ou sensores, são utilizados em cada ponto de medição, classificamos esses dados como univariados ou multivariados (BRAEI; WAGNER, 2020).

O agrupamento é uma das técnicas mais aplicadas em séries temporais, por não depender de supervisão humana ou anotação dos dados, um processo que consome muito tempo (PAPARRIZOS; GRAVANO, 2017). Essa técnica permite extrair padrões e correlações ocultas dos dados (HALKIDI; BATISTAKIS; VAZIRGIANNIS, 2001). O uso de agrupamento não se configura somente como um processo de descoberta de informação, mas também como uma etapa de pré-processamento ou sub rotina em outros tipos de tarefas, como predição de valores (SURABHI et al., 2012).

Um exemplo do uso de agrupamento em séries temporais é a análise de consumo elétrico. Energia elétrica é a segunda maior fonte energética consumida no Brasil, estando atrás apenas do petróleo e derivados. No entanto, mesmo com os avanços em eficiência energética, no período entre 2000 e 2019 esse consumo cresceu a uma taxa média geométrica anual de 2,8% no Brasil (ABRAHÃO; SOUZA, 2021). Esse crescimento gera preocupação em relação à segurança energética e cria uma demanda por ampliação da capacidade instalada, assim como um planejamento e distribuição mais efetivos (NETO; CORRÊA; PEROBELLI, 2016). Dada está situação, o agrupamento de séries de consumo elétrico se torna uma ferramenta desejável (CARPANETO et al., 2006), visto que ela permite uma análise mais eficiente, utilizando os padrões descobertos para melhor entender o comportamento dos clientes. Por consequência, essa análise permite um melhor planejamento e distribuição que leva a um aumento na satisfação do cliente (COKE; TSAO, 2010).

Durante o processo de agrupamento, extrair características para agrupa-las permite uma abordagem que consegue filtrar ruídos e informações irrelevantes (MA et al., 2019b). As técnicas de aprendizado profundo surgiram como uma estratégia poderosa para aprendizado a partir de grandes volumes de dados de maneira supervisionada. Essas técnicas tem sido aplicadas em diversas áreas como visão computacional (AKHTAR;

MIAN, 2018), reconhecimento de voz (LIU et al., 2017), processamento de linguagem natural (OTTER; MEDINA; KALITA, 2020), entre outras. O uso de aprendizado profundo aplicado em problemas de agrupamento se popularizou nos últimos anos, sendo utilizado para aprendizado de características. A aplicação de aprendizado profundo ao problema de agrupamento geralmente requer aprender padrões e características antes desconhecidas em dados não rotulados e o uso de algum algoritmo clássico de agrupamento, como o K-Means (LLOYD, 1982) para classificar essas características e organizar os dados em grupos a partir de alguma métrica de semelhança ou distância (MIN et al., 2018).

Pela natureza dos problemas de agrupamento, não há qualquer informação prévia sobre os dados, logo, a quantidade de grupos existente em um determinado conjunto é uma informação desconhecida (SINAGA; YANG, 2020). No entanto, muitos algoritmos de agrupamento necessitam que o usuário informe o número de grupos a serem criados (JAIN, 2010). Determinar a quantidade de grupos relevantes de um conjunto é um problema complexo, visto que esse valor varia dependendo do conjunto de dados, e como não existe informação na qual se basear, é necessário depender de suposições sobre a distribuição, escala, formato e correlação estrutural dos dados (FU; PERRY, 2020).

Deep Embedded Clustering (DEC) (XIE; GIRSHICK; FARHADI, 2016) é um método para agrupamento com aprendizado profundo que foi em parte responsável pelo aumento de popularidade no uso de aprendizado profundo para agrupamento (MIN et al., 2018). Esse método apresenta bons resultados e uma arquitetura relativamente simples, o que o fez se tornar um benchmark para comparação na área. Este método no entanto apresenta algumas deficiências, como a necessidade da definição a priori do número de grupos e seu baixo desempenho em dados temporais.

A metodologia apresentada nesta dissertação tem como premissa agrupar séries temporais, sem necessidade de que se informe a quantidade de grupos a serem criados, para assim se adequar a situações reais do uso de técnicas de agrupamento. O método proposto, *Deep Embedded Shape Clustering* (DESC), que se baseia no DEC, visa aprimorá-lo para que ele obtenha uma performance superior com dados temporais baseados em suas formas e automaticamente estime um número de grupos a serem criados em um conjunto de séries temporais.

1.1 Objetivos

O objetivo geral é apresentar um novo método de agrupamento de séries temporais utilizando aprendizado profundo que identifica automaticamente a quantidade de grupos. Para tal, é possível listar os seguintes objetivos específicos:

- Desenvolver uma metodologia que utilize algoritmo de agrupamento em conjunto

com aprendizagem profunda levando em conta as peculiaridades de dados temporais e características de forma.

- Encontrar automaticamente um número de grupos dentro de um conjunto de séries temporais.
- Avaliar e comparar metodologias de agrupamento de dados em uma situação onde não há rótulos previamente definidos com os quais realizar validação.

1.2 Organização da Dissertação

Este trabalho segue a seguinte organização: No Capítulo 2, são demostrados conceitos importantes para o bom entendimento do trabalho desenvolvido, como aprendizado profundo aplicado em agrupamento e estimação de grupos.

Capítulo 3 traz alguns trabalhos que agrupam séries temporais, assim como trabalhos sobre estimação de número de grupos são apresentados.

Em seguida, no Capítulo 4 é explicada a metodologia desenvolvida para agrupamento de séries temporais, enquanto o Capítulo 5 apresenta os experimentos realizados, resultados obtidos e discussões referente aos mesmos.

Por fim, o Capítulo 6 apresenta as principais contribuições desta pesquisa e oportunidades que podem ser investigadas como trabalhos futuros.

2 Fundamentação Teórica

Este capítulo apresenta os conceitos relacionados a agrupamento de séries temporais, uso de aprendizado profundo para agrupamento e estimação de número de grupos necessários para o bom entendimento do trabalho desenvolvido.

2.1 Séries temporais

Séries temporais são uma coleção de observações ordenadas no tempo. Essas observações têm como característica a dependência entre observações vizinhas. O grande interesse quando se analisa séries temporais é o reconhecimento e modelagem dessas dependências. Uma série temporal é um conjunto de observações $X_t, t \in T$ onde X é a variável de interesse e T é o conjunto de índices temporais

A formação de uma série temporal pode ser feita através de funções matemáticas, ou seja, é possível definir uma função que gera uma série temporal. Esta função, nomeada função formadora, pode ter uma ou mais variáveis e pode ser de qualquer grau, como linear ou quadrática. A função formadora que gera uma série temporal pode ter 2 termos, o determinístico e o estocástico. Dada uma série temporal linear Y, pode-se determinar que a função formadora da Equação 2.1 é completamente determinística, pois a medição Y_t depende unicamente da variável x_t.

$$Y_t = a + b \cdot x_t \tag{2.1}$$

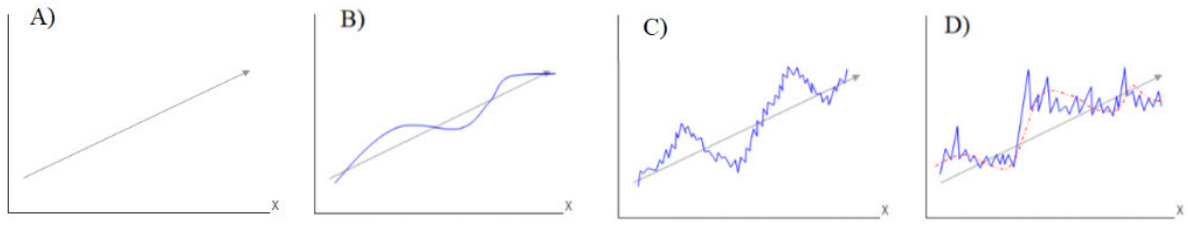
O segundo termo, o termo estocástico, se dá por um termo aleatório que é gerado por um processo estocástico. Um exemplo de função formadora linear pode ser visto na Equação 2.2, onde o termo ϵ_t é gerado a partir de um processo estocástico. Nota-se que uma série temporal estocástica pode ter o termo determinístico, mas a existência de um termo estocástico torna impossível determinar o valor exato para a função em um determinado tempo t.

$$Y_t = a + b \cdot x_t + \epsilon_t \tag{2.2}$$

Uma série temporal pode apresentar até 4 componentes: Tendência, Ciclo, Sazonalidade e Aleatoriedade. Tendência se refere aos elementos de longo prazo que estão relacionados à série temporal. Esse componente pode ser entendido como a direção geral da série temporal, como indicado pela linha reta na Figura 1(A). Ciclos são longas ondas que apresentam certa regularidade em torno de uma linha de tendência. Na Figura 1(B) a

linha ondulada ao redor da linha de tendência representa o ciclo de uma linha temporal. O terceiro componente, Sazonalidade, representa os padrões regulares que ocorrem na série temporal ao longo do tempo. Na Figura 1(C), os padrões que ocorrem na linha de ciclo da Figura 1(B) representam a Sazonalidade nesta série temporal. Por fim, a Aleatoriedade, ou resíduo, engloba todos os elementos que não se encaixam nos outros 3 componentes e pode ser observado na Figura 1(D), onde os componentes já apresentados estão presentes, com a adição do componente aleatório.

Figura 1 – Componentes de um série temporal: (A) Tendência; (B) Ciclo; (C) Sazonalidade; (D) Aleatoriedade.



Fonte: Adaptado pelo autor[1]

2.2 Agrupamento de séries temporais

Agrupamento de séries temporais é substancialmente diferente de agrupar dados atemporais, logo o esforço de pesquisa neste tópico é centralizado na extração de características e em modelos específicos para séries temporais, como a extensão do uso de métricas comumente aplicadas em dados atemporais para dados temporais (MAHARAJ; D'URSO; CAIADO, 2019). Esse processo é relevante em diversos campos do conhecimento, como geologia, medicina, finanças e economia. Para compreender as diferenças entre dados temporais e atemporais e como isso afeta o processo de agrupamento, primeiro temos que entender agrupamento.

2.2.1 Algoritmos de agrupamento

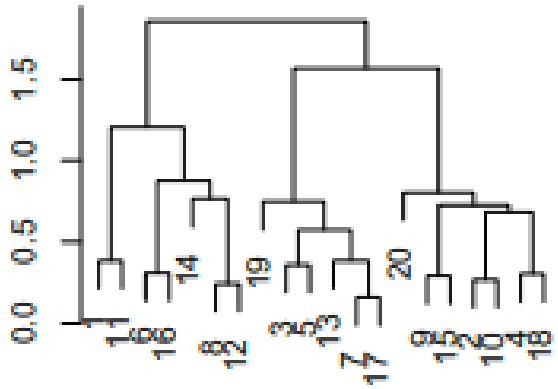
Em termos genéricos, técnicas de agrupamento se dividem em dois tipos, hierárquico ou particional (EVERITT et al., 2011). Ambos visam atribuir os objetos analisados a grupos de tal maneira que os objetos dentro de um grupo sejam similares. Essa similaridade é medida a partir de alguma métrica, como a distância euclidiana. Um exemplo de método hierárquico é o agrupamento aglomerativo (MÜLLNER, 2011). Métodos hierárquicos produzem uma série de partições nos dados, a primeira partição consiste de grupos com

[1] <https://itfeature.com/time-series-analysis-and-forecasting/component-of-time-series-data>, acessado em 17/05/2021

apenar 1 objeto, a última é um grupo que contêm todos os objetos. Essa abordagem é conhecida como *Bottom-Up*

O agrupamento aglomerativo começa com uma quantidade I de grupos, cada um com apenas 1 objeto. Uma série de operações é então realizada para juntar essas partições que eventualmente forçam todos os objetos em um mesmo grupo. A junção de dois grupos é dependente da definição de uma função de similaridade entre os grupos. Os resultados de um agrupamento hierárquico geralmente são apresentados em um gráfico chamado dendrograma, exemplificado na Figura 2.

Figura 2 – Exemplificação de um dendrograma, onde o eixo Y representa o valor de corte de uma métrica para diferenciar dois grupos e o eixo X representa a quantidade de objetos em cada grupo



Fonte: (MURTAGH; LEGENDRE, 2014)

Entre os métodos particionais, o método mais famoso é o K-Means (MACQUEEN et al., 1967). Esse método minimiza a soma dos erros quadrados para obter grupos utilizando um processo iterativo. Esse algoritmo pode ser resumido nos seguintes passos:

1. Inicialização aleatória ou baseada em algum conhecimento prévio de um número K de grupos e calculo dos centroides destes grupos.
2. Atribuição de cada objeto ao grupo mais próximo.
3. Recalcular os centroides baseado nos objetos atuais que pertencem aos grupos.
4. Repetir os passos 2 e 3 até que não haja mudanças nos grupos ou até que um número limite de iterações seja alcançado.

O algoritmo K-Means necessita como entrada a quantidade K de grupos a serem criados. Existem na literatura adaptações do algoritmo K-Means. Nestas adaptações geralmente se modifica a métrica de distância, que geralmente é a distância euclidiana, e

o cálculo dos centroides para adequar o algoritmo a diferentes domínios de dados, como características extraídas com Transformada discreta de Wavelets (VLACHOS et al., 2003) ou séries temporais (HUANG et al., 2016).

2.2.2 Aplicação de algoritmos de agrupamento para séries temporais

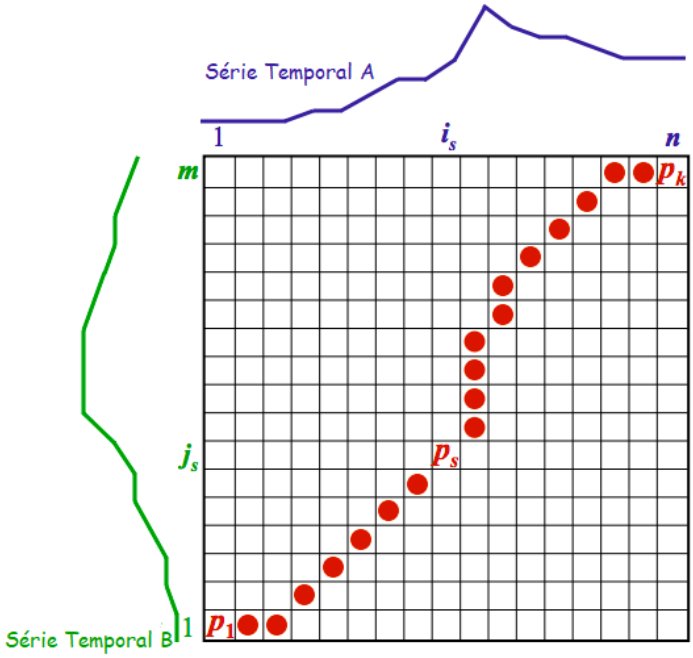
Para agrupar séries temporais, geralmente são feitas adaptações de técnicas de dados atemporais para se adequarem a dados temporais. Exemplo de adaptações são o uso de diferentes tipos de características como características temporais (GLIGOR; AUSLOOS, 2008), no domínio da frequência (MAHARAJ; D'URSO, 2011), extraídas por transformada de Wavelets (ZHANG; SHEIKHOLESLAMI; CHATTERJEE, 2005). Outra possibilidade é a modelagem de séries temporais e agrupamento dos parâmetros (D'URSO; GIOVANNI; MASSARI, 2016). Existem também abordagens que agrupam as séries temporais diretamente (D'URSO, 2000) (D'URSO, 2005).

Outro tipo comum de adaptação é o uso de métricas de distância desenvolvidas para séries temporais. *Dynamic Time Warp* (DTW) (BERNDT; CLIFFORD, 1994) é uma das métricas mais famosas nesta área. DTW é um algoritmo para comparar e alinhar duas séries temporais encontrando o alinhamento não-linear ótimo entre essas séries. Assim, é possível encontrar padrões dentro de séries que tem um ritmo diferente. Esse algoritmo cria uma matriz de distâncias m por m, sendo m o tamanho das séries, onde cada ponto na matriz representa a distância euclidiana entre dois pontos das séries x e y. Um conjunto continuo $W = \{w_1, w_2, ..., w_k\}$ de elementos da matriz, chamado de *warping path*, com $k \geq m$ define o mapeamento entre x e y.

Para otimizar a solução obtida, algumas restrições são criadas, evitando o crescimento exponencial das possibilidades de escolha do algoritmo. Essas restrições são:

- O alinhamento não vai para trás no eixo temporal, logo o caminho, representado como os pontos na matriz da Figura 3, deve sempre se manter ou avançar no tempo.
- O alinhamento não pula no eixo temporal, logo o caminho é completamento conectado.
- Na Figura 3, o alinhamento começa no canto esquerdo inferior, que indica o começo do eixo temporal para ambas as séries e termina no canto superior direito, que indica o fim do eixo temporal para ambas as séries, garantido que o alinhamento não vai considerar somente parte de uma das sequências.
- Um ponto do alinhamento não pode estar a uma distância maior que r da diagonal da matriz, sendo r o tamanho da janela de *warping*. Outra interpretação dessa regra é que o alinhamento não pode permanecer no mesmo momento no tempo por mais que r.

Figura 3 – Visualização das distâncias armazenadas do DTW



Fonte: Adaptado de (RIOFRÍO et al., 2017) pelo autor

- O caminho de alinhamento não deve ser muito íngreme ou plano. Para tal, após no máximo q passos em um eixo, o caminho deve dar um passo no outro eixo, prevenindo que pequenos trechos de uma série sejam alinhados com longos trechos da outra.

2.2.3 K-Shape

K-Shape (PAPARRIZOS; GRAVANO, 2015) é um algoritmo de agrupamento desenvolvido especificamente para agrupar séries temporais baseado no formato das séries. Este algoritmo utiliza a medida de correlação cruzada para comparar séries temporais durante o processo de agrupamento.

A correlação cruzada define a similaridade entre duas sequências $x = (x_1, ..., x_m)$ e $y = (y_1, ..., y_m)$ de tamanho m[2], mesmo que elas não estejam alinhadas. Essa medida estatística alcança invariância ao alinhamento das séries ao realizar um deslocamento s da série x sobre a série y e calcular a relação entre as séries. Esse deslocamento é definido na Equação 2.3.

$$x_{(s)} = \begin{cases} (\overbrace{0, ..., 0}^{|s|}, x_1, x_2, ..., x_{m-s}), & s \geq 0 \\ (x_{1-s}, ..., x_{m-1}, x_m, \underbrace{0, ..., 0}_{|s|}), & s < 0 \end{cases} \tag{2.3}$$

[2] Foram considerados tamanhos iguais para as sequências, mas a correlação cruzada pode ser aplicada para sequências de tamanhos diferentes

Quando todos os deslocamentos $x_{(s)}$ são considerados, com $s \in [-m, m]$, é possível calcular a correlação cruzada $CC_w(x, y) = (c1, ..., c_w)$ com tamanho 2m-1 de acordo com a Equação 2.4:

$$CC_w(x, y) = R_{w-m}(x, y), w \in \{1, 2, ..., 2m - 1\} \tag{2.4}$$

Onde $R_{w-m}(x, y)$ é calculado de acordo com a Equação 2.5:

$$R_k(x, y) = \begin{cases} \sum_{l=1}^{m-k} x_l + k \cdot y_l, & k \geq 0 \\ R_{-k}(x, y), & k < 0 \end{cases} \tag{2.5}$$

O objetivo é encontrar a posição w em que $CC_w(x, y)$ é maximizada. Baseado nesse valor de w, o deslocamento ótimo de x em relação à y é $X_{(s)}$, onde $s = w - m$.

Para possibilitar que a correlação cruzada seja usada para comparar a distância entre as séries, é necessário normalizar a correlação cruzada. Essa normalização é realizada a partir da Equação 2.6

$$NCC(x, y) = \frac{CC_w(x, y)}{\sqrt{R_0(x, x) \cdot R_0(y, y)}} \tag{2.6}$$

onde $R_0(x, x)$ e $R_0(y, y)$ são as autocorrelações de x com x e y com y e $NCC(x, y)$ é a correlação cruzada normalizada. Essa normalização é usada para que o valor do coeficiente esteja sempre entre -1 e 1.

A partir de $NCC(x, y)$ e da detecção da posição w que maximiza $NCC(x, y)$ a função que compara a forma de duas séries ou função de distância baseada na forma (DBF) é definida na Equação 2.7 como:

$$DBF(x, y) = 1 - \max_{w} \left(\frac{CC_w(x, y)}{\sqrt{R_0(x, x) \cdot R_0(y, y)}} \right) \tag{2.7}$$

que resulta em valores entre 0 e 2, com 0 indicando uma similaridade perfeita entre as séries

Este cálculo no entanto tem um tempo computacional de $O(m^2)$ sendo m o tamanho da série temporal. Para reduzir o tempo computacional, utiliza-se do teorema convolucional (KATZNELSON, 2004) que diz que a convolução de duas séries temporais pode ser computada como a Inversa da Transformada Discreta de Fourier (IDFT, do inglês, *Inverse Discrete Fourier Transform*) do produto da Transformada Discreta de Fourier

(DFT, do inglês, Discrete Fourier Transform) individual das duas séries temporais, onde DFT é definido na Equação 2.8:

$$DFT(x_k) = \sum_{r=0}^{|x|-1} x_r e^{\frac{-2frk\pi}{|x|}}, \quad k = 0, ..., |x| - 1 \tag{2.8}$$

e IDFT é definido na Equação 2.9:

$$IDFT(X_r) = \frac{1}{|x|} \sum_{k=0}^{|x|-1} DFT(x_k) e^{\frac{2frk\pi}{|x|}}, \quad r = 0, ..., |x| - 1 \tag{2.9}$$

onde $j = \sqrt{-1}$. A correlação cruzada é computada como a convolução de das duas séries temporais com uma das séries reversa no tempo, ou $x^t = x^{-t}$(KATZNELSON, 2004), que é igual ao conjugado de um numero complexo(representado por $*$) no domínio da frequência. Sendo assim, a Equação 2.4 pode ser computada para cada m como na Equação 2.10:

$$CC(x, y) = IDFT\{DFT(x) * DFT(y)\} \tag{2.10}$$

DFT e IDFT ainda são computados em $O(m^2)$, mas utilizar a implementação otimizada da transformada de Fourier, *Fast Fourier Transform* (COOLEY; TUKEY, 1965), é possível reduzir o tempo de computação para $O(mlog(m))$.

O passo de definição do centroide do grupo é feito baseado na minimização da soma das distâncias quadradas para todas as outras séries. No entanto, como a correlação cruzada é uma medida de similaridade e não de dissimilaridade, defini-se esse passo, a extração dos centroides, como a maximização das similaridades ao quadrado como demostrado na Equação 2.11:

$$\mu_k^* = \underset{\mu_k}{argmax} \sum_{x_i \in P_k} NCC(X_i, \mu_k)^2 \tag{2.11}$$

onde μ_k representa um centroide k. Na Equação 2.11 é necessário encontrar o deslocamento ótimo para cada $X_i \in P_k$. Partindo do principio de que este processo é iterativo e que os novos centroides estão próximos dos antigos centroides, os centroides já computados são usados como referência e todas as séries são alinhadas à essas referências. Para o alinhamento é utilizado a DBF, que identifica o deslocamento ótimo para cada $x_i \in P_k$. Consequentemente, como as séries já estão alinhadas à referência antes da computação dos centroides, o denominador(normalização) na Equação 2.11 pode ser omitido. Assim, combinando a Equação 2.4 e a Equação 2.5 obtêm-se a Equação 2.12:

$$\mu_k^* = \underset{\mu_k}{argmax} \sum_{x_i \in P_k} \left(\sum_{l \in [1,m]} x_{il} \cdot \mu_{kl} \right)^2 \tag{2.12}$$

Assumindo que as sequências x_i estão normalizadas com o escore padrão, e utilizando vetores para expressar a equação, obtêm-se a :

$$\begin{aligned} \vec{\mu_k} &= \underset{\vec{\mu_k}}{argmax} \sum_{\vec{x}_i \in P_k} (\vec{x}_i^T \cdot \vec{\mu_k})^2 \\ &= \underset{\vec{\mu_k}}{argmax} \vec{\mu_k}^T \cdot \sum_{\vec{x}_i \in P_k} (\vec{x}_i \cdot \vec{x}_i)^T \cdot \vec{\mu_k} \end{aligned} \tag{2.13}$$

Na Equação 2.13, somente $\vec{\mu_k}$ não está normalizado. Para lidar com a centralização de $\vec{\mu_k}$, realiza-se a operação $\vec{\mu_k} = \vec{\mu_k} \cdot Q$, onde $Q = I - \frac{1}{m}O$, onde i é a matriz identidade e O é uma matriz composta de 1's. Além disso, para que $\vec{\mu_k}$ tenha norma unitária, a Equação 2.13 é dividida por $\vec{\mu_k}^T \cdot \vec{\mu_k}$ e substituindo S por $\sum_{\vec{x_i} \in P_k} (\vec{x_i} \cdot \vec{x_i}^T)$, têm-se a Equação 2.14:

$$\begin{aligned} \vec{\mu_k}* &= \underset{\vec{\mu_k}}{argmax} \frac{\vec{\mu_k}^T \cdot Q^T \cdot S \cdot Q \cdot \vec{\mu_k}}{\vec{\mu_k}^T \cdot \vec{\mu_k}} \\ &= \underset{\vec{\mu_k}}{argmax} \frac{\vec{\mu_k}^T \cdot M \cdot \vec{\mu_k}}{\vec{\mu_k}^T \cdot \vec{\mu_k}} \end{aligned} \tag{2.14}$$

onde $M = Q^T \cdot S \cdot Q$. Através dessas transformações, a otimização da Equação 2.12 é reduzida à otimização da Equação 2.14, que é um problema já conhecido como Quociente de Rayleigh (GOLUB; LOAN, 2013). É possível encontrar o maximizador $\vec{\mu_k}*$ como o *eigenvector* que corresponde ao maior *eigenvalue* da matrix simétrica real M.

2.3 Agrupamento com aprendizado profundo

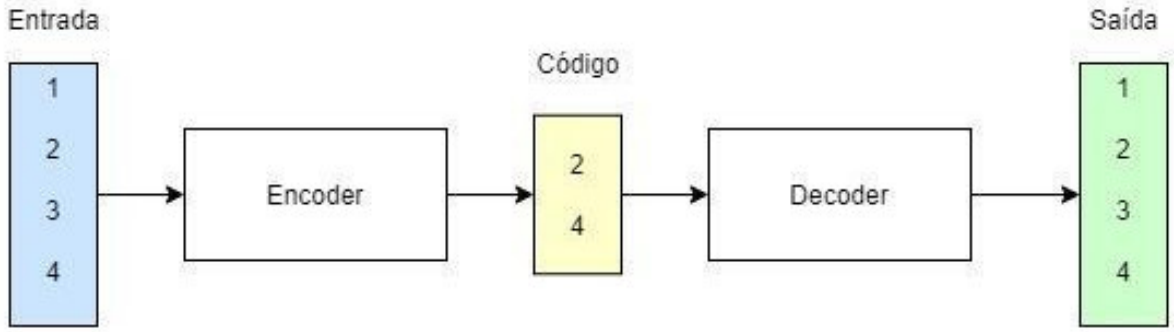
Nos últimos anos, muitos trabalhos focaram em utilizar redes neurais profundas para aprender representações dos dados que melhoram a performance de técnicas de agrupamento (MIN et al., 2018). Diferentes tipos de redes são utilizados para tal propósito como redes totalmente conectadas (FCN, do inglês, *Fully-Connected Networks*), redes convolucionais (CNN, do inglês, *Convolutinal Neural Network*), *AutoEncoders*, *Deep Belief Networks* e Varitional *AutoEncoders*. Cada abordagem tem suas vantagens, mas o objetivo geral é criar representações que ajudem no processo de agrupamento. Nesse trabalho será abordado especificamente o uso de *AutoEncoders*.

2.3.1 AutoEncoder (AE)

Um *AutoEncoder* é um tipo de rede neural treinada para aprender a recriar sua entrada dado uma codificação. A aplicação deste tipo de rede visa aprender uma representação, ou codificação, dos dados que seja suficientemente representativa para diferenciar os dados em uma dimensionalidade reduzida enquanto ainda permite uma reconstrução fiel dos valores originais dos dados utilizando exclusivamente a sua forma codificada. Na Figura 4 é possível ver um modelo geral de um AE. A primeira parte, o

Encoder, é responsável por criar uma codificação representativa da entrada de maneira que a segunda parte, o *Decoder*, consiga utilizar essa codificação para recriar a entrada.

Figura 4 – Modelo de um *AutoEncoder*



Fonte: Acervo do autor

Um AE pode ser definido como duas funções separas, a do encoder, representada na Equação 2.15 e a do decoder, representada na Equação 2.16, onde x são dados de entrada da rede e Z a codificação.

$$enc(x) : x \rightarrow Z \tag{2.15}$$

$$deco(z) : Z \rightarrow x \tag{2.16}$$

Em termos de função de redes, o encoder e o decoder são duas redes diferentes com saidas Z e x', como visto na Equação 2.17 e na Equação 2.18 respectivamente.

$$Z = \sigma(W_x + b) \tag{2.17}$$

$$x' = \sigma'(W'_Z + b') \tag{2.18}$$

onde Z é a codificação, ou saída do encoder, σ é a função de ativação, W_x são os pesos do encoder de entrada x, W'_Z são os pesos do decoder de entrada z e b é o bias e x' é a saída do decoder ou a reconstrução de x. Dadas as equações 2.15 e 2.16 a função de um AE é definida na Equação 2.19 como:

$$enc, deco = \underset{enc,deco}{argmin} ||x - (deco \circ enc)x||^2 \tag{2.19}$$

A função de *loss* L de um AE, também conhecida como erro de reconstrução, pode ser descrita em termos de funções de rede aplicando as equações 2.17 e 2.18 na

Equação 2.19, resultado na função de *loss* L da Equação 2.20 usada para treinar a rede através do processo de *backpropagation*.

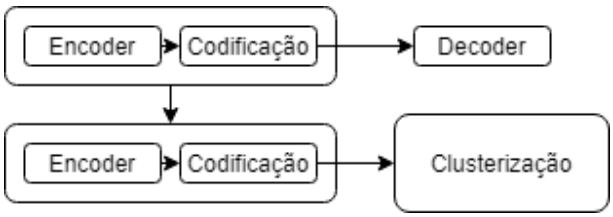
$$L(x, x') = ||x - x'||^2 = ||x - \sigma'(W'(\sigma(W_x + b)) + b'))||^2 \tag{2.20}$$

A aplicação desse tipo de rede neural no contexto de agrupamento se dá em utilizar a codificação para diminuir a dimensionalidade dos dados enquanto mantém uma representação que facilita o agrupamento. Geralmente AEs são utilizados como uma etapa inicial (XIE; GIRSHICK; FARHADI, 2016)(MADIRAJU et al., 2018). Nesse tipo de aplicação, após a codificação, geralmente é utilizado alguma técnica clássica de agrupamento, comumente o K-Means, para realizar o agrupamento dos dados codificados.

2.3.2 Deep Embedded Clustering (DEC)

Deep Embedded Clustering (XIE; GIRSHICK; FARHADI, 2016), uma das mais conhecidas arquiteturas de agrupamento com uso de aprendizagem profunda (MIN et al., 2018), consiste de um AutoEncoder (AE) que é pré-treinado para aprender uma função f_θ que mapeia dados do espaço X original em um espaço de características Z e uma etapa de otimização de parâmetros que realiza o agrupamento, onde são feitas alternadamente a computação da função auxiliar de distribuição alvo e a minimização da divergência Kullback-Leibler (KL) de um conjunto inicial de rótulo.

Figura 5 – Visão geral da metodologia DEC



Fonte: Acervo do Autor

O DEC melhora um agrupamento inicial realizado pelo algoritmo K-means em dois passos que se repetem até convergência. No primeiro passo, uma atribuição probabilística é computada entre os pontos no espaço de características Z e os centroides. No segundo passo, a função de mapeamento e a atribuição do agrupamento são refinados utilizando o aprendizado obtido das atribuições de alta confiança usando uma distribuição auxiliar alvo. Esse processo se repete ate que um critério de convergência seja atingido.

Tendo-se k grupos definidos no agrupamento inicial e n elementos, baseado em (MAATEN; HINTON, 2008), é utilizada a distribuição t-student para medir a similaridade

entre um ponto Z_i e o centroide μ_j como demostrado na Equação 2.21.

$$Q_{ij} = \frac{(1 + ||z_i - \mu_j||^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'=1}^{k}(1 + ||z_i - \mu_{j'}||^2/\alpha)^{+\frac{\alpha+1}{2}}} \tag{2.21}$$

Onde z_i representa um elemento i do espaço original X no espaço de características Z, Q_{ij} é a probabilidade de que o elemento z_i seja atribuído ao centroide μ_j e α é o grau de liberdade para a distribuição de t-student. Usa-se 1 para o valor de α (MAATEN, 2009). A função de divergência Kullback-Leibler (KL) usada como função de perda (L) entre as atribuições probabilísticas e a distribuição auxiliar é definida na Equação 2.22:

$$L = KL(P||Q) = \sum_{i=1}^{n}\sum_{j=1}^{k} P_{ij} \log \frac{P_{ij}}{Q_{ij}} \tag{2.22}$$

Onde o calculo de p_i é feito conforme a Equação 2.23:

$$P_{ij} = \frac{q_{ij}^2/f_j}{\sum_{j'=1}^{n} q_{ij'}^2/f_{j'}} \tag{2.23}$$

E $f_j = \sum_i q_{ij}$ são as frequências dos grupos probabilísticos.

A ideia base para este treinamento é utilizar uma função de agrupamento inicial e um conjunto de dados não classificados. Os dados são inicialmente classificados e a partir das classificações com maior fidelidade é o modelo é treinado para se obter uma melhor função de agrupamento.

A otimização dos μ_j centroides e parâmetros θ da rede são otimizados conjuntamente usando *Stochastic Gradient Descent* (SGD). A computação dos gradientes de L em relação a cada ponto Z_i e cada centroide μ_j são calculados usando as Equações 2.24 e 2.25:

$$\frac{\partial L}{\partial z_i} = \frac{\alpha+1}{\alpha}\sum_{j=1}^{k}(1 + \frac{||z_i - \mu_i||^2}{\alpha})^{-1} \times (p_{ij} - q_{ij})(z_i - \mu_j) \tag{2.24}$$

$$\frac{\partial L}{\partial \mu_i} = -\frac{\alpha+1}{\alpha}\sum_{i=1}^{n}(1 + \frac{||z_i - \mu_i||^2}{\alpha})^{-1} \times (p_{ij} - q_{ij})(z_i - \mu_j) \tag{2.25}$$

Os gradientes $\frac{\partial L}{\partial z_i}$ são passados para a rede com o uso de *backpropagation* para computar o gradiente $\frac{\partial L}{\partial \theta}$ dos parâmetros da rede.

A inicialização do DEC utiliza AEs empilhados. Para o *encoder*, cada camada é inicializada subcamada por subcamada, onde cada uma dessas é um *denoising* AE. *Denoising* AE são AEs, onde a adiciona-se ruído na entrada passada para o *encoder* para garantir que a rede aprenda as caracteriscas que melhor permitem reconstruir essa entrada. Para todas as camadas exceto a primeira e a última, *Rectified Linear Units* (ReLUs) são

usadas. Depois de um treinamento guloso em cada camada, todos as camadas de *encoder* são concatenadas com o *decoder*, que apresenta uma estrutura de treinamento invertida para formar um AE profundo. Após essa inicialização, o processo de treino da camada de AE é realizada.

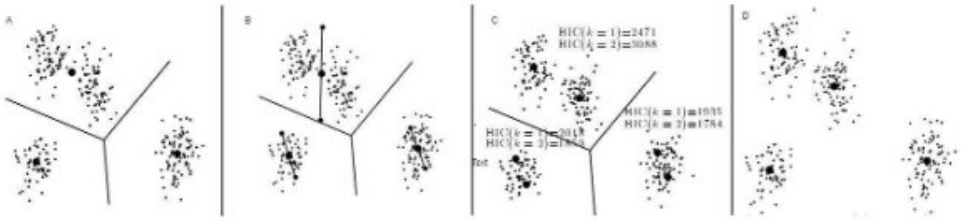
2.4 Estimação do número de grupos

Estimar o número de grupos em um conjunto de dados é essencial para vários algoritmos de agrupamento, visto que este valor tem grande influência nos resultados a serem obtidos. Para estimar esse valor são comumente utilizadas métricas de validação interna, que não necessitam de nenhuma informação sobre os dados além dos valores de suas características, em conjunto com algoritmos de agrupamento. Baseado na métrica de validação, realiza-se a maximização ou minimização da métrica escolhida para estimar o valor a ser usado para agrupar os dados. Um exemplo de algoritmo que realiza essa estimação é o X-means (PELLEG; MOORE et al., 2000). Este algoritmo maximiza o Critério de Informação Bayesiano(CIB), definido na Equação 2.26, para determinar o número de grupos. O processo desse algoritmo utiliza um algoritmo de agrupamento, por padrão o K-Means, de maneira iterativa para decidir até que ponto é benéfico criar grupos. Essa decisão é feita com a escolha da alternativa que tem o maior CIB, definido na Equação 2.26:

$$CIB(M_j) = l_j(D) - \frac{P_j}{2} \cdot logR \tag{2.26}$$

Onde $l_j(D)$ é o *log-likelihood* dos dados D de acordo com o j-ésimo modelo obtido no ponto de maior semelhança, P_j é o número de parâmetros no modelo M_j e R é a quantidade de elementos no conjunto de dados. Esse algoritmo necessita de um valor máximo para o número de grupos como parâmetro de entrada.

Figura 6 – Visualização do X-Means: (A) Agrupamento inicial com $k = 3$; (B) Agrupamento de cada subgrupo; (C) Escolha das soluções baseadas no CIB; (D) Solução final com 4 grupos



Fonte: Adaptado de (PELLEG; MOORE et al., 2000) pelo autor

Esse processo, que pode ser visualizado na Figura 6 tem 3 passos principais: (1) Refinamento de parâmetros. (2) Aprimoramento da estrutura e (3) verificação da

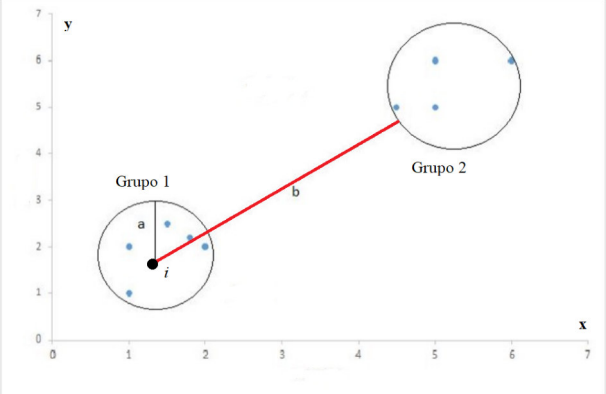
condição de parada. Na etapa (1), exemplificado na Figura 6(A), é aplicado o algoritmo de agrupamento, por padrão o K-Means, com $k = 2$ ou $k = 3$, até que ele convirja. Na etapa (2), representada na Figura 6(B) e (C), cada grupo gerado na etapa (1) é avaliado utilizando o CIB e depois o algoritmo de agrupamento é aplicado em cada grupo cada grupo gerado na etapa (1). O resultado desta segunda execução é também avaliado com o CIB e então compara-se ambas as soluções. Aquela que obteve o maior CIB é mantido. A etapa (3) verifica se houve a criação do número máximo de grupos ou se nenhum grupo foi subdivido pela maximização do CIB. O resultado final desse processo é representado na Figura 6(D), onde os grupos que maximizaram o CIB foram escolhidos.

2.5 Coeficiente de silhueta

O Coeficiente de Silhueta (ROUSSEEUW, 1987) é uma métrica de validação interna utilizada para quantificar a qualidade de um agrupamento. Métricas de validação interna são assim chamadas pois utilizam somente os valores das características dos dados para quantificar a qualidade do processo de agrupamento, não necessitando assim, de nenhuma informação sobre os dados. Esse coeficiente mostra o quão próximos entre si estão os pontos intra-grupo e o quão separados os grupos estão entre si. A silhueta de um ponto varia entre -1 e 1, onde uma silhueta próxima de 1 indica grupos que o ponto está distante dos grupos vizinhos, uma silhueta próxima de 0 indica que o ponto está próxima ou sobre a borda de decisão e um coeficiente negativo indica que o elemento pode ter sido atribuído à um grupo erroneamente.

Na Figura 7, onde os eixos x e y representam coordenadas euclidianas, quanto menor a distância intra-grupo, representada pela linha a, mais denso será o grupo, e quanto maior for a distância inter-grupo, representada pela linha b, maior será a silhueta, pois um bom agrupamento é aquele em que os elementos de um grupo são muito próximos entre si e os grupos são distantes uns dos outros.

Figura 7 – Visualização dos elementos da silhueta



Fonte: Acervo do autor

Na Equação 2.27 temos a definição da silhueta $s(i)$ para um ponto i. O termo $a(i)$, representado na Figura 7 pelo circulo de raio a, é a média da distância entre o ponto i e todos os outros pontos que pertencem ao mesmo grupo que i. Já $b(i)$, representado como a linha b na Figura 7, representa a menor distância média entre o ponto i e os pontos do grupo vizinho à ele, ou seja, o grupo mais próximo à ele ao qual ele não pertence. O coeficiente de silhueta é calculado como a média aritmética da silhueta de todos os pontos.

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{2.27}$$

2.6 Escore Padronizado

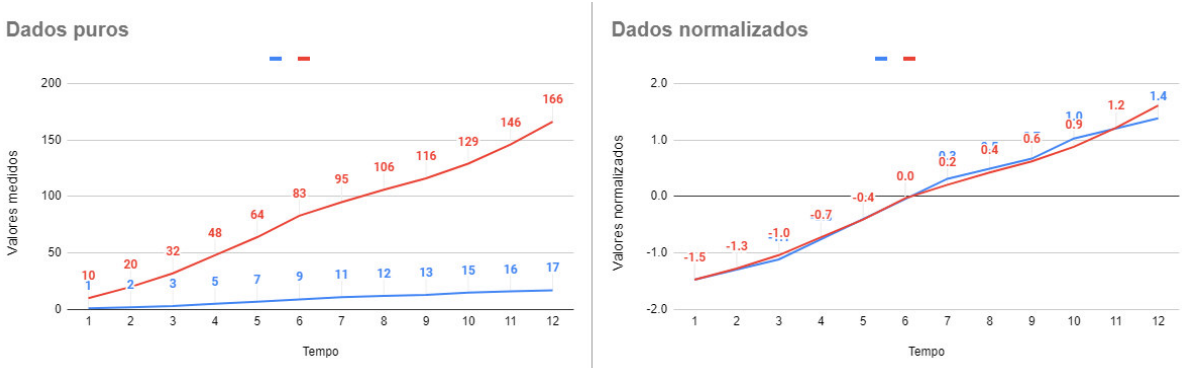
Na estatística, o Escore Padronizado, (Z-Score), é uma técnica de normalização onde os dados são representados pela distância em desvios padrões da média do conjunto. A Equação 2.28 apresenta a aplicação do Z-score em séries temporais.

$$X_i = \frac{(X_i - \mu) * std_{alvo}}{\sigma + \mu_{alvo}} \tag{2.28}$$

Onde X_i é um ponto da série temporal, μ é a media da série, std_{alvo} é o desvio padrão desejado, σ é o desvio padrão da série e μ_{alvo} é a média alvo. Por padrão, o valor usado para std_{alvo} é 1 e μ_{alvo} é 0, já que esses valores fazem com que não se perca a distribuição original no espaço, ou no caso de séries temporais, o formato da série.

O objetivo da aplicação desta técnica em um contexto de agrupamento de séries temporais é retirar as diferenças de escala dos dados, que podem influenciar o resultado, enquanto mantêm-se os padrões temporais (TURECZEK et al., 2019). Uma visualização da aplicação desta técnica pode ser vista na Figura 8 em duas séries temporais. Como ilustrado, essa técnica facilita a comparação da forma de duas séries temporais sem que a escala de seus valores afete o resultado.

Figura 8 – Visualização da aplicação do *Escore Padronizado* em séries temporais



Fonte: Acervo do autor

2.7 Otimização de hiperparâmetros

Hiperparâmetros são parâmetros de modelos que devem ser definidos antes do treinamento desse modelo. Todo modelo de aprendizado de máquina contém hiperparâmetros e otimizá-los é otimizar a performance do modelo (FEURER; HUTTER, 2019). Este processo visa reduzir o esforço humano necessário para ajuste dos parâmetros e melhorar o desempenho dos algoritmos de aprendizado de máquina para o problema em questão.

Uma das primeiras conclusões que se obteve nesta área foi que diferentes conjuntos de dados tem melhor performance com diferentes combinações de hiperparâmetros. Outra conclusão que se obteve foi que adaptar os hiperparâmetros gera melhores resultados comparados às configurações padrões em bibliotecas de aprendizado de máquina (THORNTON et al., 2013; SANDERS; GIRAUD-CARRIER, 2017). Os hiperparâmetros podem ser divididos em dois tipos, hiperparâmetros de treinamento e hiperparâmetros de estrutura. Como os nomes sugerem, hiperparâmetros de treinamento controlam o treinamento da rede (épocas de treinamento, taxa de aprendizado, etc) e hiperparâmetros de estrutura controlam a estrutura da rede neural (Número de camadas, função de ativação, etc). Um algoritmo de otimização de hiperparâmetros utiliza um conjunto G de possibilidades de valores para os hiperparâmetros e um histórico experimental H dos valores de *loss* para minimizar a função de *loss* (BERGSTRA; YAMINS; COX, 2013).

2.7.1 *Tree-structured Parzen Estimator Approach* (TPE)

O TPE (KOMER; BERGSTRA; ELIASMITH, 2014) é um modelo sequencial de otimização (Sequential Model-based Optimization - SMBO) que usa modelos construídos sequencialmente para aproximar o desempenho de um conjunto de hiperparâmetros m baseado no histórico dos modelos anteriores.

Uma definição de otimização Bayesiana de hiperparâmetros é construir um modelo probabilístico da função objetivo e usá-lo para selecionar os hiperparâmetros que tem maior probabilidade de gerar melhores resultados e avaliá-los na verdadeira função objetivo. Essa definição pode ser interpretada como uma análise do impacto dos hiperparâmetros, onde aqueles que tem maior chance de ter impacto no desempenho da função objetivo são testados para confirmar essa suposição.

O TPE constrói o modelo probabilístico utilizando o teorema de Bayes e a probabilidade de cada combinação de hiperparâmetros (estimadores) é representado como uma inequação que divide o valor esperado da função objetivo de cada estimador entre acima e abaixo de um determinado limiar. Essa inequação cria então duas distribuições probabilísticas, uma de estimadores que provavelmente irão gerar resultados abaixo do limiar, $G(m)$, e uma de estimadores que provavelmente irão gerar resultados acima do limiar, $L(m)$. Intuitivamente, utilizar os estimadores de $L(m)$ é melhor, logo, estes estimadores

são avaliados na verdadeira função objetivo.

Cada vez que a avaliação de um conjunto m de hiperparâmetros é feita, os resultados da avaliação são salvas no histórico na forma de erro e. Este histórico é usado para construir um novo modelo probabilístico, que por sua vez ira cria novas distribuições $G(m)$ e $L(m)$. O objetivo final é a minimização do erro e utilizando o histórico para guiar a seleção dos hiperparâmetros m. Esse processo reduz o tempo de busca por hiperparâmetros, visto que não é necessário testar todas as combinações, somente aquelas que estima-se levar a uma melhora no resultado.

3 Trabalhos Relacionados

Este capítulo apresenta estudos que envolvem agrupamento de séries temporais e estimação do número de grupos. A grande maioria dos trabalhos retratados neste capítulo utilizam aprendizado profundo para representação dos dados.

3.1 Aprendizado Profundo aplicado à agrupamento

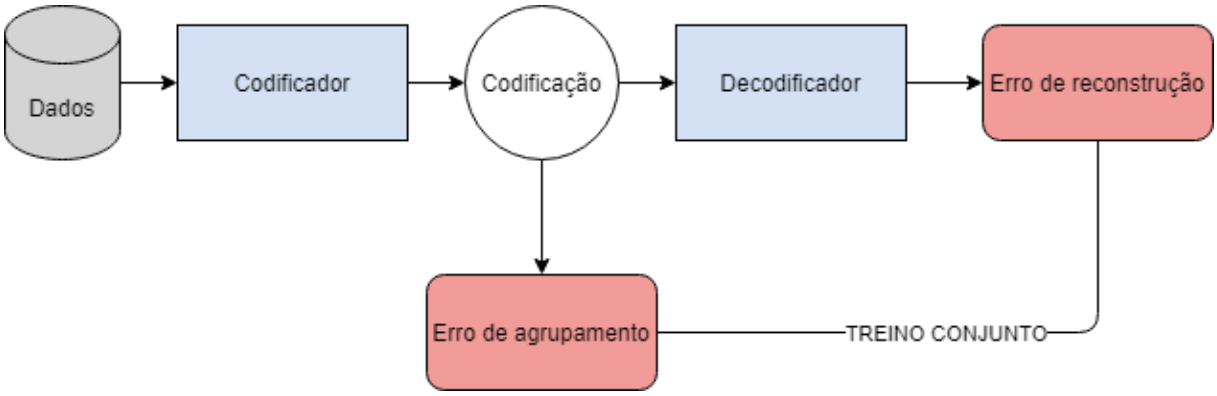
Um dos focos na pesquisa em Aprendizado Profundo aplicada à agrupamento representação de características. Normalmente se utiliza uma arquitetura que cria uma representação dos dados, e esse dados são então clusterizados com alguma técnica clássica (MIN et al., 2018), como o K-means (LLOYD, 1982).

Uma classificação possível para as diferentes aplicações de Aprendizado Profundo para agrupamento é baseada na arquitetura de rede neural utilizada (MIN et al., 2018). Exemplos de arquitetura são AutoEncoder, AutoEncoders Variacionais (VAE, do inglês, Varitional AutoEncoder), redes geradoras adversariais (GAN, do inglês, Generative Adversarial Network) e redes de agrupamento profunda (CDNN, do inglês, Clustering Deep Neural Network).

3.1.1 Algoritmos baseados em AE

Em algoritmos baseadas em AE, a perda da rede é geralmente o erro de reconstrução, que geralmente é a diferença entre a entrada e saída da rede, combinada com o erro de agrupamento, e a arquitetura da rede pode ser adaptada para o tipo de dado alvo. nesses algoritmos, uma representação generalista pode ser vista na Figura 9.

Figura 9 – Visão geral da arquitetura de algoritmos baseados em AE.



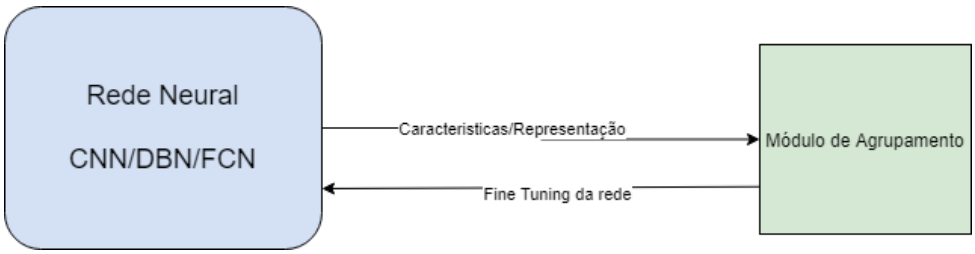
Fonte: Acervo do autor

Trabalhos como Yang et al. (2017) e Huang et al. (2014) se baseiam nesse tipo de arquitetura, utilizando o AE para extração de características e um treino de otimização conjunta entre o erro de reconstrução e o erro de agrupamento de um algoritmo de agrupamento.

3.1.2 Algoritmos baseados em CDNN

Nos algoritmos baseados em CDNN, utiliza-se somente o erro de agrupamento para o treino da rede e a rede pode ser composta de FCN, CNN, entre outros. Por não utilizar o erro de reconstrução na função de perda, esse tipo de rede têm o risco de criar espaços de características corrompidos, onde todos os pontos são mapeados em grupos concisos mas que não têm significância. Uma visão geral de como esse tipo de algoritmo funciona pode ser visto na Figura 10

Figura 10 – Visualização de algoritmos baseados em CDNN.



Fonte: Acervo do autor

Pode-se caracterizar esse tipo de algoritmo em 3 classes, dependendo de como é realizada a inicialização, ou pré-treino, da rede empregada nestes algoritmos (1) Pré-treino não supervisionado, (2) Pré-treino supervisionado e (3) Sem pré-treino.

Em algoritmos do tipo (1), primeiramente é treinado uma rede de forma não supervisionada, como um AE ou uma máquina de Boltzmann restrita (RBM, do inglês, Restricted Boltzmann Machine). As redes utilizadas nesta forma de pré-treino conseguem aprender somente com as informações disponíveis nos dados, sem necessidade de rotulação. Essa capacidade é uma propriedade desejada quando se deseja realizar agrupamento, visto que o objetivo deste é extrair informações previamente desconhecidas sobre os dados. Depois desse pré-treino é feita uma refinação dos parâmetros da rede utilizando a perda de clusterização. Exemplos desse tipo de rede são DEC (XIE; GIRSHICK; FARHADI, 2016), DNC (CHEN, 2015), DBC (LI; QIAO; ZHANG, 2018) e DTC (MADIRAJU et al., 2018), onde a rede é primeiramente pré-treinada para aprender a extrair as características mais discriminativas dos dados para criar uma representação que é então usada para realizar o agrupamento.

Nas redes do tipo (2), a rede é inicialmente treinada de maneira supervisionada para que posteriormente se utilize as características aprendidas a partir de grandes datasets

rotulados para representação dos dados a serem agrupados. Guérin et al. (2017) realizou experimentação extensiva com uma variedade de arquiteturas CNN pré-treinadas no dataset ImageNet (DENG et al., 2009) e os resultados mostram que características extraídas de CNN profundas treinadas em um dataset grande e rotulado, combinadas com algoritmos clássicos de agrupamento, tem uma performance superior ao estado da arte de métodos de agrupamento. Hsu e Lin (2017) é um representativo desse tipo de rede, onde um *framework* baseado em CNN utiliza as características extraídas a partir de um modelo inicial treinado no ImageNet para inicializar os centroides, que são refinados iterativamente.

Por fim, em redes do tipo (3) realiza-se a extração de características diretamente através de funções de perda arquitetadas para este proposito. Yang, Parikh e Batra (2016) e Moriya et al. (2018) utilizam uma CNN para aprendizado de representação e agrupamento aglomerativo. A otimização da função objetivo é feita iterativamente, onde o agrupamento hierárquico é feito no *forward pass* enquanto o aprendizado de representação é feito no *backward pass*. Dessa maneira ambos os aprendizados são feitos conjuntamente.

3.1.3 Algoritmos baseados em VAE

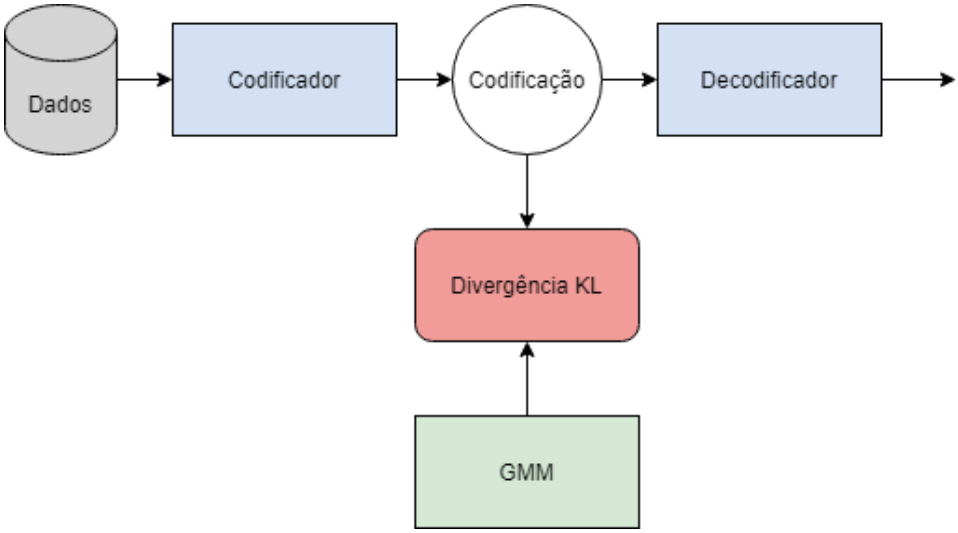
Redes que se baseiam em VAE, além de realizar agrupamento, também podem realizar geração de amostras. O conceito por trás deste processo é aprender como o dados variam entre si dentro do conjunto, permitindo a criação de amostras que assemelham-se aos dados reais. *Variational Deep Embedding* (VaDE) (JIANG et al., 2016) e *Gaussian Mixture Variational AutoEncoder* (GMVAE) (DILOKTHANAKUL et al., 2016) utilizam VAE, exemplificado na Figura 11, combinado com modelos de mistura gaussiana (GMM, do inglês, *Gaussian Mixture Models*) para modelar o processo gerador dos dados em conjunto com o codificador do AE para extração de características através de codificação.

3.2 Agrupamento de séries temporais com Aprendizado Profundo

O objetivo de agrupar séries temporais é atribuir séries temporais à grupos tal que todos as séries temporais dentro de um grupo sejam similares entre si e que os grupos como um todo sejam diferenciáveis. De maneira generalista, pode-se dividir os métodos de agrupamento de séries temporais em agrupamento das séries em si e agrupamento das características extraídas.

Algoritmos que realizam agrupamento das séries temporais diretamente geralmente mudam a função de distância de técnicas clássicas para se adaptar à características presentes em séries temporais, como por exemplo escala e distorção. Alguns algoritmos utilizam o *Dynamic Time Warp* (DTW) e o *DTW barycenter averaging* (DBA) (ZHANG; HEPNER, 2017; LIU; ZHANG; ZENG, 2019) em conjunto com técnicas clássicas como o K-Means e implementam mudanças nos mesmos para melhorar performance. Outros

Figura 11 – Visualização de um VAE



Fonte: Acervo do autor

algoritmos implementam novas distâncias e métodos para definição de centroides (YANG; LESKOVEC, 2011), ou utilizam versões normalizadas de métricas já existentes para comparar as séries temporais (PAPARRIZOS; GRAVANO, 2015).

Métodos que agrupam as características extraídas utilizar algoritmos de agrupamento sobre as características extraídas em vez de agrupar os dados diretamente, o que diminui o impacto de ruído e *outliers*, assim como diminui a dimensionalidade dos dados sobre os quais se trabalha (MA et al., 2019b). Essa classificação pode ainda ser subdividida em dois tipos: Abordagens em duas partes, que agrupam após extração das características (i) e abordagens que conjuntamente otimizam o aprendizado de características e agrupamento(ii).

Exemplos de abordagens do tipo (i) podem ser vistos em Guo, Jia e Zhang (2008) e Zakaria, Mueen e Keogh (2012) onde respectivamente se usa análise de componentes independentes e u-shapelets para extração das características a serem agrupadas. As características extraídas por esse tipo de abordagem podem não ser adequadas para agrupamento visto que a extração de características é tratada como uma etapa de pré-processamento (MA et al., 2019b).

Já nas abordagens do tipo (ii) um ponto a se levar em consideração é que as características a serem extraídas devem ser não-lineares (MA et al., 2019a). Exemplos de abordagens para agrupar séries temporais são o *Deep Temporal Clustering* (DTC) (MADIRAJU et al., 2018) e *Deep Temporal Clustering Representation* (DTCR) (MA et al., 2019b), que combinam um codificador com uma camada de agrupamento para aprender uma representação não linear dos grupos. O aprendizado é guiado a uma representação significativa através da camada de agrupamento e da minimização da divergência de Kullback-Leibler. Lee e Schaar (2020) utiliza a mesma ideia de ter um codificador que cria

uma representação codificada com o objetivo de agrupar pacientes a partir do prognostico em vez dos sintomas em si. O diferencial apresentado é o uso de uma rede de seleção e uma rede de predição que combinadas com o codificador levam ao aprendizado de uma representação que melhor descreve a distribuição dos prognósticos.

3.3 Estimação do número de grupos

Uma deficiência de vários algoritmos de agrupamento é a necessidade que o número de grupos a serem criados seja definido à priori. No entanto, a quantidade de grupos de um conjunto normalmente é desconhecido em casos reais (SINAGA; YANG, 2020). Naturalmente, para resolver esse problema, diferentes metodologias foram criadas para estimar a quantidade de grupos existentes em um conjunto de dados. Uma maneira de abordar este problema é o uso de índices de validação. Estes índices podem ser divididos em índices internos e externos (RENDÓN et al., 2011). Os índices externos comparam o resultado do agrupamento com com algum conhecimento prévio, como rótulos de classes. Já índices internos são aqueles que avaliam a qualidade da estrutura dos grupos utilizando somente as informações presentes nos dados. Alguns exemplo famosos de índices internos usados para estimação do número de grupos são: Índice de Davies-Bouldin (DAVIES; BOULDIN, 1979), índice de Calinski-Harabasz (CALIŃSKI; HARABASZ, 1974), e o coeficiente de Silhueta.

Através do conceito de agrupamento de consenso, Ünlü e Xanthopoulos (2019) propõe a utilização de 5 técnicas clássicas de agrupamento (K-Means, Gaussiana, Espectral, *Fuzzy* e Hierárquica) em conjunto com 3 índices de validação interna (coeficiente de silhueta , índice de Calinski-Harabasz e índice de Davies-Bouldin). As diferentes combinações entre técnica de agrupamento e métrica de validação é um esquema de agrupamento com consenso ponderado baseada em grafos para determinar qual o número de grupos que obtém os melhores resultados. Os resultados mostram que a melhor métrica foi o coeficiente de silhueta, no entanto a taxa de acerto entre as métricas não foi discrepante o suficiente para que as outras métricas fossem descartadas.

Outra abordagem foi a versão não supervisionada da validação cruzada de Gabriel desenvolvida por Fu e Perry (2020). Essa técnica simula validação cruzada através de um esquema de partições nos dados para poder calcular um erro de validação cruzada para cada valor possível para K no K-Means.

Como a aplicação repetida de técnicas de agrupamento nestas técnicas podem levar a um custo computacional elevado, (PATIL; BAIDARI, 2019) propõe uma técnica chamada de *Depth of Data* (DeD), para estimar o número de grupos, sem que se aplique técnicas de agrupamento. A técnica é baseada na profundidade de Mahalanobis (MAHALANOBIS, 1936). Calculando a mediana da profundidade do conjunto, o conjunto é divido em k

subconjuntos, com k variando dentro de um intervalo definido. Após essa divisão, o algoritmo calcula a mediana da profundidade de cada subconjunto e a diferença entre a profundidade mediana do subconjunto e a profundidade mediana do conjunto completo. O valor de k é encontrado com a maximização do da diferença de profundidade dos subconjuntos com o conjunto completo.

Rodriguez e Laio (2014) utilizam a ideia de densidade e que um centroide é mais denso que os elementos vizinhos e tem uma distância grande com outros elementos de alta densidade. A ideia do *clustering by fast search* (C-FS) é encontrar então os picos de densidade do conjunto,

3.4 Considerações finais

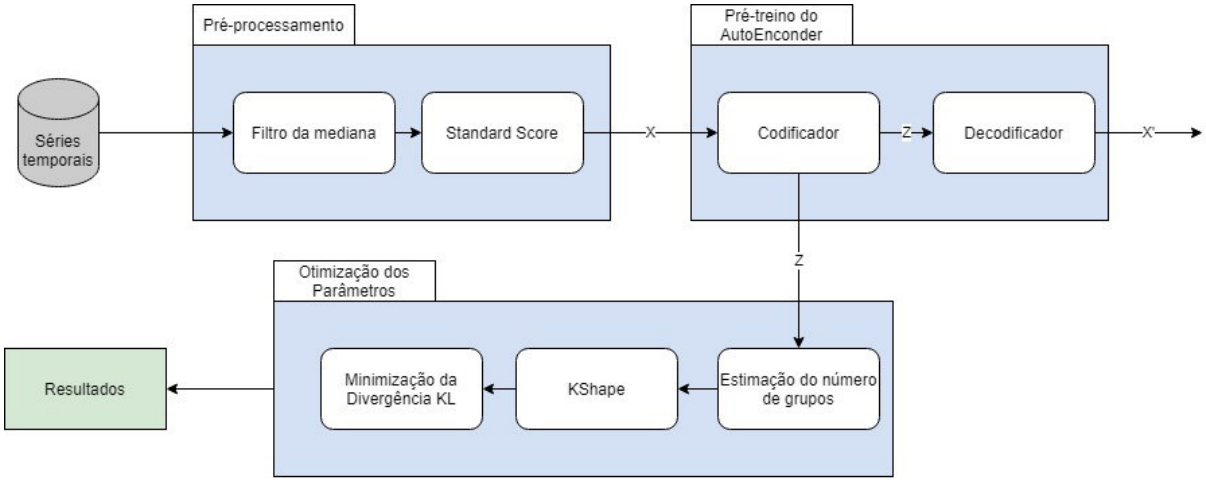
Como foi apresentado, pesquisa ao redor de agrupamento é tema recorrente. Ao longo de mais de duas décadas foram propostas diversas técnicas para obter-se agrupamentos mais significativos.

Uma deficiência presente em muitos algoritmos e métodos de agrupamento é uma abordagem para se adequar a casos reais, onde o número de grupos a serem criados é dificilmente conhecido. Apesar de separadamente existirem trabalhos que realizam agrupamento e trabalhos que estimam o número de grupos, poucos são os que realizam as duas tarefas conjuntamente. Outro ponto que não é explorado nos trabalhos citados é o uso de aprendizado profundo para agrupamento de séries temporais e estimação do número de grupo em dados temporais.

4 Metodologia

Este capítulo apresenta a metodologia proposta denominada *Deep Embedded Shape Clustering* (DESC). Esta metodologia utiliza os conceitos e técnicas introduzidos no Capítulo 2 para agrupar séries temporais baseado na sua forma. Na Figura 12 é ilustrada uma visão geral da metodologia.

Figura 12 – Visão geral do DESC



Fonte: Acervo do autor

Inicialmente é apresentada a etapa de pré-processamento dos dados. Em seguida são demostradas as etapas de pré-treino do AutoEncoder e estimação do número de grupos. Posteriormente é apresentado o processo de agrupamento em si. Por fim, são descritos os experimentos realizados para validar a metodologia proposta nesta dissertação bem como a etapa de avaliação dos resultados.

4.1 Pré-processamento

Como primeiro passo da metodologia, é feito um pré-processamento nos dados que consiste na inclusão de dados ausentes, filtro da mediana e aplicação da técnica Escore padronizado sobre os dados.

O primeiro passo é realizar um tratamento nos dados para abordar medições faltantes. Este fenômeno é um problema comum em séries temporais (MORITZ; BARTZ-BEIELSTEIN, 2017). Para abordar tal problema é feita a imputação ou preenchimento dos valores faltantes. O método de imputação utilizado é repetição da ultima medição válida ou, caso não existam medições prévias, utiliza-se a próxima medição válida.

Após isso, é aplicado o filtro da mediana, com o intuito de remover anomalias geradas por medições equivocadas ou por erro humano. A janela deslizante utilizada foi de 3 meses, logo a mediana do conjunto mês atual, mês anterior e mês seguinte, substitui o valor do mês atual. O uso desse filtro têm como objetivo remover valores discrepantes do resto da séries causados por problemas de medição, visto que esses valores afetam o resultado da técnica de normalização aplicada a seguir.

Por último, é aplicado o *standardization* ou escore padronizado. O uso do desta técnica é essencial, visto que esta técnica trata as diferenças de escala dos dados reescalando os valores sem que se perca a forma da séries temporais. Esse passo é representado pela Caixa 1 na Figura 12.

4.2 Pré-treino do AutoEncoder (AE)

Após o pré-processamento, é realizado o pré-treino do AE. Essa etapa é análoga ao pré-treino realizado no DEC, onde utiliza-se a minimização da função de *loss* da Equação 2.20 para treinamento da rede . A rede neural irá reduzir a dimensionalidade dos dados enquanto preserva as características mais representativas. Os hiperparâmetros do treinamento, como otimizador da rede durante esse treinamento e número de épocas são decididos através de otimização da hiperparâmetros que é discutida mais a frente.

Ao final deste passo, o *decoder* é removido e o *encoder* é utilizado para codificar os dados de entrada pré-processados em uma representação que realça os pontos mais representativos das séries temporais. Esse passo é representado pela etapa 2 na Figura 12.

4.3 Otimização dos parâmetros

Nesta etapa é feito a otimização dos parâmetros que definem os grupos finais da metodologia. Esse processo começa com a estimação do número de grupos presente no conjunto de dados e segue para o refinamento dos parâmetros.

4.3.1 Estimação do número de grupos

Um fator importante para a metodologia é a estimação do número de grupos a serem criados. No DEC este valor deve ser determinado à priori pelo usuário, uma limitação que nem sempre pode ser atendida devido à natureza dos problemas de agrupamento. Para abordar esse problema, utiliza-se a maximização do CIB, conforme a Equação 2.26. Durante o processo de maximização do CIB é aplicado o *K-Shape* como algoritmo de agrupamento. O *K-Shape* foi escolhido pois ele é especializado em agrupar séries temporais com base em suas formas, através das permitindo assim um melhor agrupamento das séries em comparação com o K-Means utilizado usualmente neste processo.

O resultado final deste processo é um agrupamento inicial com um número de grupos estimados através da maximização do CIB. Esse agrupamento inicial é então refinado na etapa seguinte, a otimização dos parâmetros.

4.3.2 Refinamento dos parâmetros

Após a estimação da quantidade de grupos e obtenção de um agrupamento inicial, temos o treino da arquitetura como um todo. É anexado na saída do encoder uma camada de agrupamento que irá refinar o agrupamento inicial.

Esse processo é realizado de maneira similar ao DEC e pode ser entendido como a utilização das atribuições que apresentam maior confiabilidade (Maior similaridade com um único centroide) como base para as atribuições com menos confiabilidade (similaridade parecida com vários centroides). Isto permite o treinamento da arquitetura em uma base onde não se tem nenhuma informação sobre os dados além dos seus valores intrínsecos.

Iterativamente são calculadas atribuições probabilísticas, demostradas na Equação 2.23, a partir das divisões dos grupos e similaridades entre elementos e centroides. A partir dessa atribuição, é calculada uma função de distribuição auxiliar, demostrada na Equação 2.21 que permite utilizar a divergência de Kullback-Leibler(KL) para calcular o erro que será propagado para a rede como um todo, criando assim o aprendizado. Os parâmetros do encoder são então otimizados utilizando esse erro, usando os exemplos de alta confiabilidade para direcionar o aprendizado para uma representação que permite um melhor agrupamento.

4.4 Otimização dos hiperparâmetros da rede

Para definir automaticamente os hiperparâmetros, é usado o *Tree-structured Parzen Estimator* (TPE) proposto por (KOMER; BERGSTRA; ELIASMITH, 2014). Definir automaticamente os hiper-parâmetros permite que o método se torne mais independente do conjunto de dados, visto que cada conjunto de dados usará os hiperparâmetros que tenham melhor desempenho. O espaço de busca definido para os hiperparâmetros é:

- Otimizador durante Pré-treino do AE: *Stochatic Gradiente Descent* (SGD) (BOTTOU, 2012), *Root Mean Square Propagation* (RMSProp) (ILIEVSKI et al., 2016) , *Adaptative Moment Estimation* (ADAM) (KINGMA; BA, 2014)
- Otimizador durante treino do DESC: SGD, RMSProp, ADAM
- Intervalo de atualização da distribuição auxiliar: 1, 2 e 5 iterações
- Épocas de pré-treino: 100, 150, 200, 500 épocas
- Tamanho do *Batch*: 16, 32, 64, 128, 256

- Número máximo de iterações: 200, 1000, 2000, 5000 iterações
- Limiar de parada: 0.1, 0.01, 0.001, 0.0001

Durante a busca dos hiperparâmetros, limita-se a busca para 300 combinações únicas de parâmetros escolhidos aleatoriamente e essas combinações são avaliadas a partir do coeficiente de silhueta obtido a partir dessas combinações.

5 Resultados

Neste capítulo são apresentados e discutidos os resultados obtidos nos experimentos realizados para a validação da metodologia proposta. Como não há avaliação prévia, usa-se uma métrica de avaliação interna de agrupamento. Além dos resultados numéricos, são apresentados resultados visuais, que permitem observar as semelhanças no comportamento entre os integrantes de cada grupo. Para tal, são apresentados gráficos com os centroides e séries do mesmo grupo. Os resultados apresentados foram obtidos de acordo com três experimentos, um para avaliar a qualidade da estimação do número de grupos e outros dois para avaliar a qualidade do agrupamento em si.

5.1 Conjuntos de dados

Para realizar os experimentos projetados foram utilizados os conjuntos de dados descritos nas seções a seguir.

5.1.1 Estimação do número de grupos

A quantidade de grupos de um conjunto normalmente é desconhecido em casos reais (SINAGA; YANG, 2020). Logo, um fator importante para uma metodologia de agrupamento é a descoberta desta quantidade. Afim de validar a estimação do número de grupos realizada, foram coletados conjuntos de dados de séries temporais onde a quantidade de grupos fosse conhecida. Foram selecionados 66 conjuntos de dados do UCR *Archive* (DAU et al., 2018), contendo conjuntos que tem entre 2 e 60 grupos pré-classificados.

5.1.2 Agrupamento

Para avaliar a capacidade de agrupamento do método proposto, foram utilizados dados de consumo elétrico, medidos entre maio de 2018 e abril de 2019. Temos dois conjuntos de dados, titulados Comercial e Residencial disponibilizados por uma empresa de fornecimento de energia elétrica. A base Comercial contém séries de 4.540 clientes e a base residencial contém séries de 86.360 clientes e todos os clientes de ambas as bases pertencem ao mesmo município. As séries contêm até 12 meses de medições mensais e foram medidas manualmente por funcionários da empresa. Existem casos onde as séries não são completas, faltando um ou múltiplos meses de medições. Além disso, existem medições com valores fora dos limites de consumo estabelecidos pela empresa. É importante notar esses pontos, pois todos os clientes devem ser classificados, mesmo que apresentem essas particularidades.

5.2 Experimentos

Para validar a metodologia proposta, foram propostos três experimentos. O primeiro deles buscou validar a estimação do número de grupos da metodologia proposta e os dois outros experimentos buscaram validar a qualidade do agrupamento da metodologia proposta.

5.2.1 Estimação do número de grupos

O primeiro dos experimentos avaliou a etapa de estimação do número de grupos proposta e a comparou com o algoritmo *Density Peaks* (RODRIGUEZ; LAIO, 2014). Cada um dos algoritmos foi executado 5 vezes. Todos os dados passaram pela mesma etapa de pré-processamento. Ambos os algoritmos necessitam de parâmetros a serem pré-definidos. Para o X-means, foi utilizado um valor máximo de 100 grupos a serem criados. Para o *Density Peaks* é necessário determinar 2 parâmetros, Delta e Limiar de Densidade. Como a estimação destes parâmetros é altamente dependente dos dados (MEHMOOD et al., 2016), foram usadas várias combinações e manteve-se a combinação que obteve os melhores resultados.

Na Tabela 1, são apresentados os resultados obtidos para esse experimento. Foram utilizados a média do erro de estimação e o desvio padrão do erro de estimação para avaliar tanto o X-means como o *Density Peaks*. Como o *Density Peaks* sempre apresenta o mesmo resultado para os mesmos parâmetros, o desvio padrão é 0, logo essa coluna foi omitida.

Os valores realçados na Tabela 1 demostram o algoritmo que obteve o menor erro médio de estimação para aquele conjunto. No geral, o X-Means foi superior em 72% dos conjuntos de dado. Alguns dos resultados obtidos pelo *Density Peaks*, como por exemplo, no conjunto **FordB**, onde o algoritmo erra por 3634 grupos, são devidos aos parâmetros escolhidos para o algoritmo. Neste mesmo conjunto, o X-means obtêm um ótimo resultado, onde nas 5 estimações ele erra por somente 1 grupo. Existem no entanto casos como o do conjunto **Yoga**, onde o *Density Peaks* acerta exatamente o número de grupos existentes, enquanto o X-means obtém um erro médio de 47. Esses resultados demostram que a estimação do número de grupos do método proposto consegue em geral ter um resultado superior, além de ter uma parametrização mais simples, levando a uma facilidade maior de uso e adaptabilidade à diversos conjuntos de dados.

Tabela 1 – Resultados obtidos para a estimação do número de grupos em 66 conjuntos do UCR

Nome	# Elementos	# Grupos	Density Peaks	X-means	
			Média	Média	Desvio padrão
ACSF1	100	10	9	8	1.26
Adiac	390	37	36	24	7.96
ChlorineConcentration	467	3	1	5	2.45
Computers	250	2	248	11	21.00
CricketX	390	12	378	18	11.21
CricketY	390	12	378	33	16.57
CricketZ	390	12	378	20	13.62
DistalPhalanxOutlineCorrect	600	2	1	16	20.08
DistalPhalanxTW	400	6	5	5	0.40
DodgerLoopDay	78	7	38	24	23.30
Earthquakes	322	2	320	1	0.00
ECG200	100	2	1	25	29.32
ECG5000	500	5	2	40	18.47
ElectricDevices	8926	7	6500	5	0.80
EOGHorizontalSignal	362	12	121	10	0.75
EOGVerticalSignal	362	12	166	17	16.80
EthanolLevel	504	4	2	53	4.84
FaceAll	560	14	500	17	10.46
FacesUCR	200	14	126	20	14.00
FiftyWords	450	50	382	49	0.00
Fish	175	7	6	44	0.00
FordA	3601	2	3599	1	0.00
FordB	3636	2	3634	1	0.00
FreezerRegularTrain	150	2	1	56	4.92
GestureMidAirD1	208	26	79	25	0.00
GestureMidAirD2	208	26	113	25	0.00
GestureMidAirD3	208	26	39	25	0.00
GesturePebbleZ1	132	6	27	5	0.00
GesturePebbleZ2	146	6	21	5	0.40
Ham	109	2	42	1	0.00
HandOutlines	1000	2	165	53	0.00
Haptics	155	5	50	43	19.72
InsectWingbeatSound	220	11	9	21	13.29
LargeKitchenAppliances	375	3	372	2	0.80
MedicalImages	381	10	6	31	22.50
MelbournePedestrian	1194	10	354	8	1.55
MixedShapesRegularTrain	500	5	216	3	22.57
NonInvasiveFetalECGThorax1	1800	42	11	17	0.00
NonInvasiveFetalECGThorax2	1800	42	1	17	2.50
OSULeaf	200	6	58	28	19.26
PhalangesOutlinesCorrect	1800	2	1	14	23.15
Phoneme	214	39	175	38	0.00
PigArtPressure	104	52	20	5	0.47
PigCVP	104	52	30	49	2.32
PLAID	537	11	8	40	15.06
PowerCons	180	2	63	11	20.50
RefrigerationDevices	375	3	372	2	0.00
ScreenType	375	3	372	2	1.50
SemgHandGenderCh2	300	2	82	51	0.00
SemgHandMovementCh2	450	6	123	5	0.40
SemgHandSubjectCh2	450	5	122	32	23.36
ShapesAll	600	60	47	20	19.77
SmallKitchenAppliances	375	3	369	2	0.00
StarLightCurves	1000	3	356	42	20.68
SwedishLeaf	500	15	14	34	11.50
SyntheticControl	300	6	189	24	22.78
TwoPatterns	1000	4	996	3	0.00
UWaveGestureLibraryAll	896	8	888	36	0.00
UWaveGestureLibraryX	896	8	603	48	4.49
UWaveGestureLibraryY	896	8	367	46	6.02
UWaveGestureLibraryZ	896	8	567	46	6.36
Wafer	1000	2	41	32	15.41
WordSynonyms	267	25	136	24	0.00
Worms	181	5	152	13	21.73
WormsTwoClass	181	2	155	1	0.49
Yoga	300	2	0	47	9.58

Fonte: Elaborada pelo Autor

5.2.2 Agrupamento

Para avaliar a qualidade de agrupamento foram realizados dois experimentos, um utilizando a base Comercial e outro usando a base Residencial. Foram utilizados para comparação alguns algoritmos clássicos, como o K-means (ARTHUR; VASSILVITSKII, 2006), DBSCAN (SCHUBERT et al., 2017), agrupamento espectral (SHI; MALIK, 2000), hierárquico (JR, 1963) e gaussiano (MCLACHLAN; BASFORD, 1988). Além disso, compara-se também o K-Shape e métodos que utilizam aprendizado profundo na forma de *AutoEncoders* para representação de características como o DEC original e *Deep Temporal Clustering* (DTC) (MADIRAJU et al., 2018), uma especialização do DEC para séries temporais.

Cada método foi executado 5 vezes para cada um dos conjuntos a serem apresentados. Para uma avaliação justa, o pré-processamento utilizado pelo DESC é usado em todos os métodos de comparação. O mesmo número de grupos criados pelo DESC foi utilizado para todos os algoritmos de comparação testados que necessitam desse parâmetro.

Para avaliação dos resultados, utilizou-se o coeficiente de silhueta (ZAKI; JR; MEIRA, 2014). Para computar a distância entre as séries temporais, foi utilizado o *Dynamic Time Warp* (DTW) (MÜLLER, 2007) em vez da distância euclidiana, visto que o DTW é mais adequado para comparação entre séries temporais (THIOLLIERE et al., 2015; YU et al., 2019; WICHT; FISCHER; HENNEBERT, 2016). Um problema encontrado durante o uso desta métrica é a escalabilidade do uso de memória, principalmente durante o experimento com a base Residencial. Para resolver este problema, foi calculado a silhueta de 10 subconjuntos aleatórios de 500 séries temporais e então calculado a média do coeficiente de silhueta obtido para esses 10 subconjuntos. Esta abordagem foi utilizada em todos os testes. Na avaliação dos experimentos para o agrupamento, além do coeficiente de silhueta, utilizou-se uma avaliação visual, visto que se deseja que a forma das séries temporais dentro de um grupo sejam similares.

5.2.2.1 Resultados no Conjunto Comercial

No conjunto Comercial, 4 grupos foram encontrados. A Figura 13 demonstra os centroides destes grupos encontrados pela metodologia proposta. A partir destes centroides é possível afirmar que foram encontrados 4 padrões de consumo entre os clientes:

- Centroide 1: Aumento gradual de consumo seguido por uma queda de consumo.
- Centroide 2: Nível de consumo estável.
- Centroide 3: Constante crescimento de consumo.
- Centroide 4: Constante declínio de consumo.

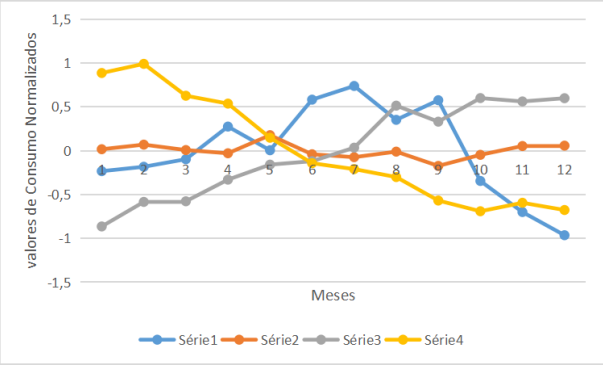


Figura 13 – Centroides dos grupos criados para o conjunto Comercial

Os centroides 3 e 4 apresentam comportamentos contrastantes, o que torna fácil distinguir as séries que pertencem a cada um desses grupos. O centroide 2 tem um comportamento estável, onde os clientes sempre mantém o mesmo nível de consumo ao longo do tempo. Por fim, o centroide 1 contem o comportamento mais complexo entre os 4 comportamentos encontrados. Ele apresenta aumento de consumo gradual nos meses iniciais seguido por queda nos níveis de consumo nos meses finais. Por esse tipo de comportamento ser um agregado de dois outros comportamentos encontrados (Comportamento crescente e decrescente) as séries desse grupo apresentam proximidade com o comportamento dos demais grupos. Isso pode ser observado pelo coeficiente de silhueta, visto que um valor de 0.42, como visto na Tabela 2 pode ser interpretado como alta similaridade dentro de cada grupo e pontos de similaridade no comportamento entre elementos de grupos diferentes, causados principalmente por elementos do grupo representado pelo centroide 1.

Tabela 2 – Resultados de 5 execuções para o conjunto Comercial

Método	Coeficiente de Silhueta	Desvio Padrão
DBSCAN	-0.34	0.012
GMM	-0.08	0.008
K-MEANS	0.18	0.008
ESPECTRAL	0.19	0.010
HIERÁRQUICO	0.24	0.004
K-SHAPE	0.29	0.012
DEC	0.27	0.021
DTC	0.28	0.021
DESC(Proposto)	**0.42**	**0.012**

A Tabela 2 demonstra a comparação dos resultados obtidos com métodos e algoritmos conhecidos para cinco execuções. Como pode ser observado, o método proposto obtém uma melhora significativa no coeficiente de silhueta, indicando uma melhor separação dos grupos em comparação aos outros métodos. Pode-se observar também que o DEC realiza o que é proposto no trabalho original, um refinamento do resultado obtido pelo K-means, e o DTC melhora, mesmo que ligeiramente, os resultados do DEC quando se

trata de séries temporais.

Para realizar uma análise visual dos resultados, clientes e centroides são apresentados nas figuras 14, 15, 16, 17. Como o centroide representa um grupo, é esperado que o comportamento do centroide seja similar ao comportamento dos clientes do mesmo grupo.

A Figura 14 apresenta um exemplo da série de consumo de um cliente (Linha contínua) em comparação com o centroide do grupo ao qual o cliente foi atribuído (Linha tracejada). O comportamento esperado de um cliente representado pelo centroide 1 é de crescimento gradual seguido por uma queda no nível de consumo. No caso demonstrado observa-se tanto o crescimento quanto o declínio no nível de consumo, demostrando que o centroide representou bem o comportamento geral do cliente.

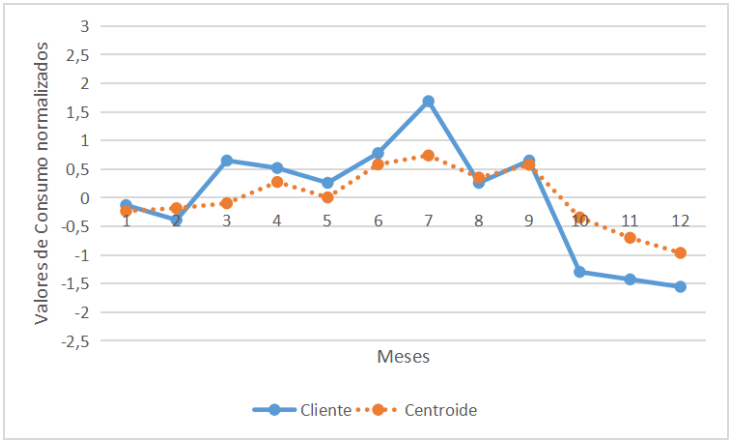


Figura 14 – Cliente do conjunto Comercial em comparação com seu centroide (Centroide 1)

Já na Figura 15, tem-se um exemplo no qual a série de consumo do cliente apresenta um grau de diferença maior em relação ao centroide. Nesta série temos um crescimento rápido de consumo nos meses finais, mas como na maior parte do tempo o cliente manteve um nível estável de consumo, consequentemente foi atribuído ao grupo representado pelo centroide 2.

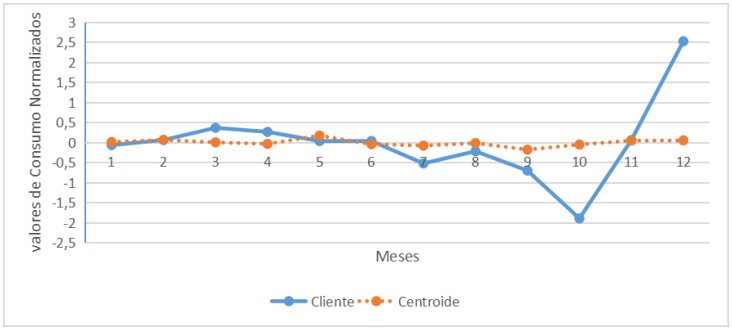


Figura 15 – Cliente do conjunto Comercial em comparação com seu centroide (Centroide 2)

Vale ressaltar que como o centroide é um elemento que representa o grupo como um todo, espera-se que ele seja parecido com os membros do grupo, mas cada cliente vai ter um grau diferente de similaridade. Essa situação é esperada e está presente em todos os grupos, como pode ser visto nas Figuras 16 e 17

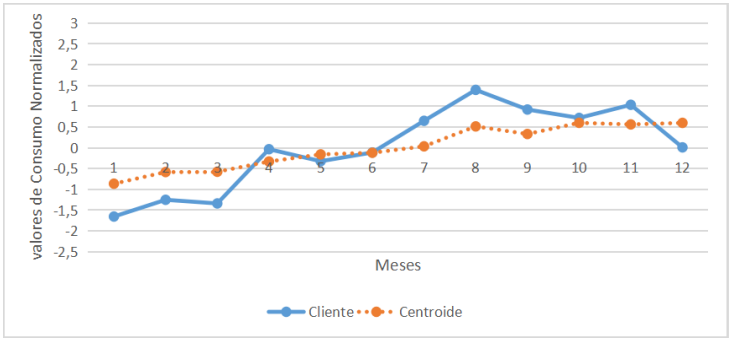


Figura 16 – Cliente Comercial levemente diferente de seu centroide (Centroide 3)

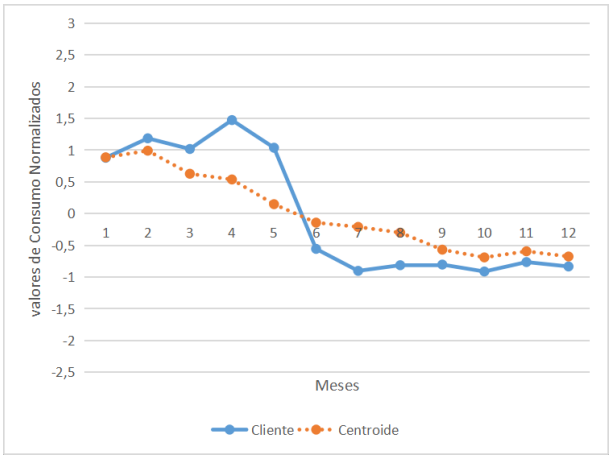


Figura 17 – Cliente Comercial levemente diferente de seu centroide (Centroide 4)

5.2.2.2 Resultados no Conjunto Residencial

Para esse conjunto de dados, as técnicas de agrupamento espectral e hierárquica requerem grandes quantidades de memória (SHAHAM et al., 2018), visto que ambos usam matrizes de similaridades, no caso criando uma matriz 86.360 x 86.360, e por este motivo, não estão presentes nos resultados desta seção. Além disso, utilizou-se os mesmos hiperparâmetros encontrados na otimização do conjunto Comercial.

Os resultados encontrados são similares aos resultados encontrados no conjunto Comercial, onde o DESC obtém um coeficiente de silhueta maior que os métodos e técnicas comparados, como pode ser observado na Tabela 3. Para este conjunto foram encontrados 3 grupos, apresentados na Figura 18. Todos os métodos e algoritmos testados apresentam desempenho próximo aos resultados encontrados no conjunto Comercial. A mesma abordagem utilizada no conjunto Comercial foi aplicada, quando necessária, onde o

mesmo número de grupos que foi encontrado pelo DESC foi informado para os algoritmos que necessitam de um número de grupos na inicialização.

Tabela 3 – Resultados de 5 execuções para o conjunto Residencial

Método	Coeficiente de Silhueta	Desvio Padrão
DBSCAN	-0.21	0.036
GMM	0.01	0.001
K-SHAPE	0.34	0.007
K-MEANS	0.23	0.012
DEC	0.10	0.043
DTC	0.18	0.071
DESC	**0.41**	**0.026**

Os padrões de comportamento encontrado, ilustrados na Figura 18, foram mais generalistas que no conjunto Comercial. Dois dos grupos encontrados são similares aos encontrados no conjunto Comercial. Os comportamentos encontrados podem ser interpretados como:

- Centroide 1: Aumento constante de nível de consumo.
- Centroide 2: Declínio constante de nível de consumo.
- Centroide 3: Decréscimo nos meses iniciais, seguido por um crescimento no nível de consumo.

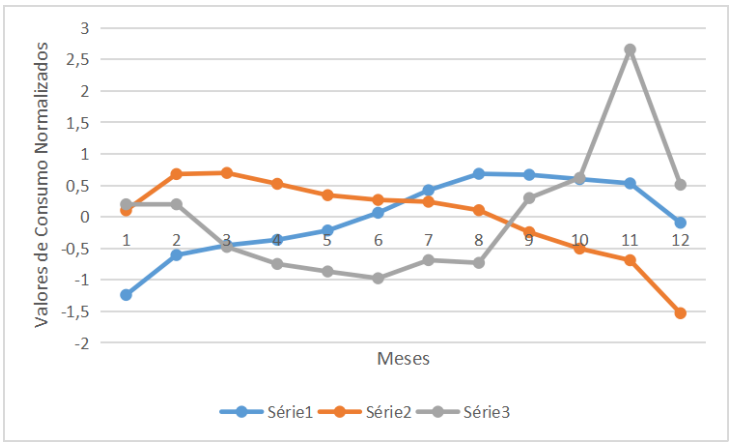


Figura 18 – Centroides dos grupos criados para o conjunto Residencial

O mesmo fenômeno encontrado no conjunto Comercial pode ser observado no conjunto Residencial, onde o comportamento geral representado pelo centroide pode ser observado nos clientes. As Figuras 19 e 20 apresentam ocorrências deste fenômeno, com padrões dos centroides 1 e 2, que também foram encontrados no conjunto Comercial.

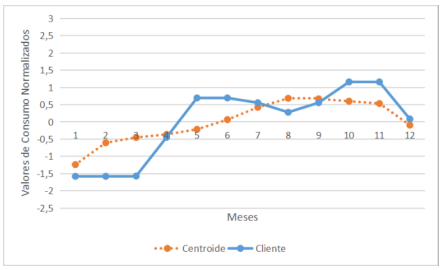


Figura 19 – Cliente Residencial com o centroide do grupo ao qual ele foi atribuído (Centroide 0)

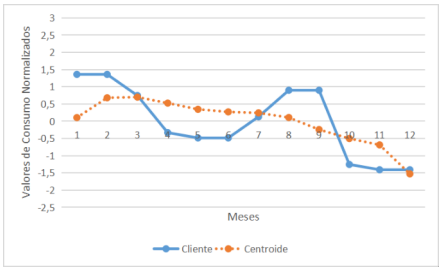


Figura 20 – Cliente residencial com o centroide do grupo ao qual ele foi atribuído (Centroide 1)

No terceiro grupo, muitos dos clientes tem um alto grau de similaridade com o centroide do grupo, como ilustrado na Figura 21, o que pode indicar que talvez esse grupo tenha sido criado como o resultado de um *overfitting* para este centroide.

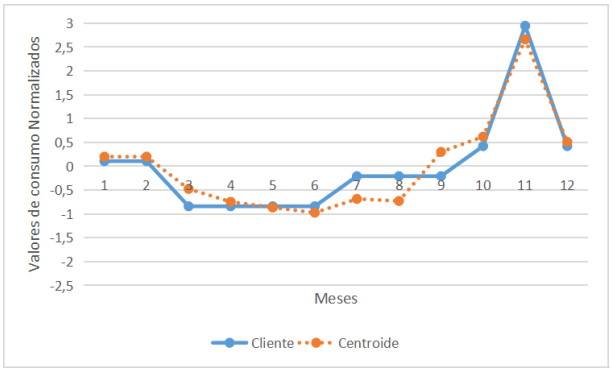


Figura 21 – Cliente residencial com o centroide do grupo ao qual ele foi atribuído (Centroide2)

5.3 Discussão

No primeiro dos experimentos, estimação do número de grupos, pode-se observar na Tabela 1 que o X-means adaptado a séries temporais obtém resultados mais consistente em relação ao *Density Peaks*. Essa consistência se dá em relação ao tipo de parâmetro que cada um dos algoritmos necessita. Enquanto o X-means utiliza um limiar superior para o número de grupos, o *Density Peaks* utiliza dois parâmetros mais abstratos, chamados de Delta e Limiar de densidade. Esses parâmetros são muito mais voláteis em relação ao conjunto de dados em que se deseja estimar o número de grupos. A vantagem apresentada pelo *Density Peaks* é a reprodutibilidade da estimação. Enquanto o X-Means raramente obtém os mesmo resultados, o *Density Peaks* sempre apresenta o mesmo resultado para um conjunto com os mesmo parâmetros. Assim se de alguma maneira for automatizada

a escolha dos parâmetros para um conjunto de dados, é possível obter resultados mais consistentes.

Nos outros dois experimentos, que avaliam o agrupamento, observa-se que no conjunto Comercial alcançou-se um bom resultado, onde a grande maioria dos clientes é bem representado pelo centroide do grupo ao qual foi atribuído. Isso é reforçado pelo coeficiente de silhueta, como demostra a Tabela 2.

Na avaliação do conjunto Residencial, observa-se que o número de grupos encontrados reduziu, contrário a expectativas. Uma possibilidade para essa ocorrência é o uso dos pesos do conjunto Comercial no AutoEncoder da rede. O coeficiente de silhueta encontrado para esse experimento pelo método proposto foi superior aos outros métodos testados, mostrando que mesmo essa ocorrência não leva à uma perca significativa nos resultados.

5.4 Considerações finais

Os resultados expostos neste capítulo demonstram a aplicação da metodologia proposta para um conjunto de dados real de séries temporais de consumo elétrico. Os experimentos realizados indicam que a metodologia é promissora considerando os resultados obtidos. Além de auxiliar na análise de dados temporais, a metodologia pode ainda implicar na melhoria da classificação de séries temporais. Entretanto, são necessários mais experimentos, principalmente com conjuntos de dados mais variados. Apesar disso, os resultados iniciais mostram-se promissores.

O capítulo seguinte traz as conclusões e discussões gerais sobre este trabalho de mestrado, além de propor melhorias e trabalhos futuros.

6 Conclusão

Agrupamento de séries temporais é uma tarefa complexa com vários usos possíveis na indústria. Um desses uso se apresenta na análise de consumo de energia elétrica. Técnicas de agrupamento que utilizam aprendizado profundo são uma tendência dos últimos anos, mostrando resultados promissores.

Este trabalho propõe uma metodologia para agrupar séries temporais de consumo elétrico. Para tal objetivo, tomou-se como base a arquitetura DEC. A metodologia adequa a arquitetura DEC para o domínio das séries temporais, mais especificamente, séries de consumo elétrico. O algoritmo K-Shape é utilizado em conjunto com a arquitetura para melhor agrupar as características de forma das séries temporais. Uma dificuldade presente na aplicação de muitos algoritmos usados para agrupamento, incluindo o DEC, é a estimação do número de grupos existentes em um conjunto de dados. Em problemas reais essa informação é geralmente desconhecida. Para tratar este problema adequou-se a maximização do coeficiente de informação Bayesiano para uso em séries temporais, permitindo uma melhor estimação do número de grupos dentro de um conjunto de séries temporais. Esta abordagem proposta neste trabalho apresenta uma solução para um problema básico da arquitetura DEC. Por fim, para permitir uma melhor generalização da metodologia proposta, utiliza-se otimização dos hiperparâmetros da rede com o *Tree-structured Parzen Estimator Approach*. A metodologia proposta combina todos estes elementos para atingir uma melhor qualidade no agrupamento de séries temporais de consumo elétrico.

Os resultados indicam que a metodologia proposta consegue criar grupos significativos de padrões de consumo de energia elétrica em dados reais. No experimento realizado para avaliar a estimação de grupos, os resultados mostraram-se favoráveis quando comparados com o da abordagem *Density Peaks*. Já nos experimentos para avaliar a qualidade do agrupamento, observasse que o método proposto conseguiu criar grupos com uma qualidade de agrupamento maior, como indicado pelo coeficiente de silhueta.

Considerando os resultados dos experimentos pode-se ainda afirmar que a metodologia proposta consegue realizar uma boa descoberta de padrões. Compreende-se ainda que os resultados levantam indícios que a metodologia proposta pode auxiliar empresas de fornecimento de energia, ao proporcionar mais informações sobre os padrões de consumo de seus clientes. Portante, se entende que os objetivos deste trabalho foram atingidos

Como trabalhos futuros sugere-se a aplicação da metodologia apresentada noutros *datasets* a fim de obter-se mais robustez em relação às conclusões aqui apresentadas. Outra sugestão é a adaptação da metodologia para comportar séries temporais multivariadas,

permitindo assim uma análise ainda mais significativa das séries de consumo.

Referências

ABRAHÃO, K. C. d. F. J.; SOUZA, R. G. V. d. Estimativa da evolução do uso final de energia elétrica no setor residencial do brasil por região geográfica. *Ambiente Construído*, SciELO Brasil, v. 21, n. 2, p. 383–408, 2021. Citado na página 14.

AKHTAR, N.; MIAN, A. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, IEEE, v. 6, p. 14410–14430, 2018. Citado na página 15.

ARTHUR, D.; VASSILVITSKII, S. *k-means++: The advantages of careful seeding*. [S.l.], 2006. Citado na página 46.

BERGSTRA, J.; YAMINS, D.; COX, D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In: PMLR. *International conference on machine learning*. [S.l.], 2013. p. 115–123. Citado na página 31.

BERNDT, D. J.; CLIFFORD, J. Using dynamic time warping to find patterns in time series. In: SEATTLE, WA, USA:. *KDD workshop*. [S.l.], 1994. v. 10, n. 16, p. 359–370. Citado na página 20.

BOTTOU, L. Stochastic gradient descent tricks. In: *Neural networks: Tricks of the trade*. [S.l.]: Springer, 2012. p. 421–436. Citado na página 41.

BRAEI, M.; WAGNER, S. Anomaly detection in univariate time-series: A survey on the state-of-the-art. *arXiv preprint arXiv:2004.00433*, 2020. Citado na página 14.

CALIŃSKI, T.; HARABASZ, J. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, Taylor & Francis, v. 3, n. 1, p. 1–27, 1974. Citado na página 37.

CARPANETO, E.; CHICCO, G.; NAPOLI, R.; SCUTARIU, M. Electricity customer classification using frequency–domain load pattern data. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 28, n. 1, p. 13–20, 2006. Citado na página 14.

CHEN, G. Deep learning with nonparametric clustering. *arXiv preprint arXiv:1501.03084*, 2015. Citado na página 34.

COKE, G.; TSAO, M. Random effects mixture models for clustering electrical load series. *Journal of time series analysis*, Wiley Online Library, v. 31, n. 6, p. 451–464, 2010. Citado na página 14.

COOLEY, J. W.; TUKEY, J. W. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, JSTOR, v. 19, n. 90, p. 297–301, 1965. Citado na página 23.

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.; HEXAGON-ML. *The UCR Time Series Classification Archive*. 2018. <https://www.cs.ucr.edu/~eamonn/time_series_data_2018/>. Citado na página 43.

DAVIES, D. L.; BOULDIN, D. W. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, n. 2, p. 224–227, 1979. Citado na página 37.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: IEEE. *2009 IEEE conference on computer vision and pattern recognition*. [S.l.], 2009. p. 248–255. Citado na página 35.

DILOKTHANAKUL, N.; MEDIANO, P. A.; GARNELO, M.; LEE, M. C.; SALIMBENI, H.; ARULKUMARAN, K.; SHANAHAN, M. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016. Citado na página 35.

D'URSO, P. Fuzzy clustering for data time arrays with inlier and outlier time trajectories. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 13, n. 5, p. 583–604, 2005. Citado na página 20.

D'URSO, P.; GIOVANNI, L. D.; MASSARI, R. Garch-based robust clustering of time series. *Fuzzy Sets and Systems*, Elsevier, v. 305, p. 1–28, 2016. Citado na página 20.

D'URSO, P. Dissimilarity measures for time trajectories. *Journal of the Italian Statistical Society*, Springer, v. 9, n. 1-3, p. 53, 2000. Citado na página 20.

EVERITT, B. S.; LANDAU, S.; LEESE, M.; STAHL, D. *Cluster analysis 5th ed.* [S.l.]: John Wiley, 2011. Citado na página 18.

FEURER, M.; HUTTER, F. Hyperparameter optimization. In: *Automated Machine Learning*. [S.l.]: Springer, Cham, 2019. p. 3–33. Citado na página 31.

FU, W.; PERRY, P. O. Estimating the number of clusters using cross-validation. *Journal of Computational and Graphical Statistics*, Taylor & Francis, v. 29, n. 1, p. 162–173, 2020. Citado 2 vezes nas páginas 15 e 37.

GLIGOR, M.; AUSLOOS, M. Clusters in weighted macroeconomic networks: the eu case. introducing the overlapping index of gdp/capita fluctuation correlations. *The European Physical Journal B*, Springer, v. 63, n. 4, p. 533–539, 2008. Citado na página 20.

GOLUB, G. H.; LOAN, C. F. V. *Matrix computations*. [S.l.]: JHU press, 2013. v. 3. Citado na página 24.

GUÉRIN, J.; GIBARU, O.; THIERY, S.; NYIRI, E. Cnn features are also great at unsupervised classification. *arXiv preprint arXiv:1707.01700*, 2017. Citado na página 35.

GUO, C.; JIA, H.; ZHANG, N. Time series clustering based on ica for stock data analysis. In: IEEE. *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*. [S.l.], 2008. p. 1–4. Citado na página 36.

HALKIDI, M.; BATISTAKIS, Y.; VAZIRGIANNIS, M. On clustering validation techniques. *Journal of intelligent information systems*, Springer, v. 17, n. 2, p. 107–145, 2001. Citado na página 14.

HSU, C.-C.; LIN, C.-W. Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data. *IEEE Transactions on Multimedia*, IEEE, v. 20, n. 2, p. 421–429, 2017. Citado na página 35.

HUANG, P.; HUANG, Y.; WANG, W.; WANG, L. Deep embedding network for clustering. In: IEEE. *2014 22nd International conference on pattern recognition*. [S.l.], 2014. p. 1532–1537. Citado na página 34.

HUANG, X.; YE, Y.; XIONG, L.; LAU, R. Y.; JIANG, N.; WANG, S. Time series k-means: A new k-means type smooth subspace clustering for time series data. *Information Sciences*, Elsevier, v. 367, p. 1–13, 2016. Citado na página 20.

ILIEVSKI, I.; AKHTAR, T.; FENG, J.; SHOEMAKER, C. A. Efficient hyperparameter optimization of deep learning algorithms using deterministic rbf surrogates. *arXiv preprint arXiv:1607.08316*, 2016. Citado na página 41.

JAIN, A. K. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, Elsevier, v. 31, n. 8, p. 651–666, 2010. Citado na página 15.

JAVED, A.; LEE, B. S.; RIZZO, D. M. A benchmark study on time series clustering. *arXiv preprint arXiv:2004.09546*, 2020. Citado na página 14.

JIANG, Z.; ZHENG, Y.; TAN, H.; TANG, B.; ZHOU, H. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148*, 2016. Citado na página 35.

JOHANNESEN, N. J.; KOLHE, M.; GOODWIN, M. Relative evaluation of regression tools for urban area electrical energy demand forecasting. *Journal of cleaner production*, Elsevier, v. 218, p. 555–564, 2019. Citado na página 14.

JR, J. H. W. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, Taylor & Francis Group, v. 58, n. 301, p. 236–244, 1963. Citado na página 46.

KATZNELSON, Y. *An introduction to harmonic analysis*. [S.l.]: Cambridge University Press, 2004. Citado 2 vezes nas páginas 22 e 23.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. Citado na página 41.

KOMER, B.; BERGSTRA, J.; ELIASMITH, C. Hyperopt-sklearn: automatic hyperparameter configuration for scikit-learn. In: CITESEER. *ICML workshop on AutoML*. [S.l.], 2014. v. 9, p. 50. Citado 2 vezes nas páginas 31 e 41.

LEE, C.; SCHAAR, M. V. D. Temporal phenotyping using deep predictive clustering of disease progression. In: PMLR. *International Conference on Machine Learning*. [S.l.], 2020. p. 5767–5777. Citado na página 36.

LI, F.; QIAO, H.; ZHANG, B. Discriminatively boosted image clustering with fully convolutional auto-encoders. *Pattern Recognition*, Elsevier, v. 83, p. 161–173, 2018. Citado na página 34.

LIAO, T. W. Clustering of time series data—a survey. *Pattern recognition*, Elsevier, v. 38, n. 11, p. 1857–1874, 2005. Citado na página 14.

LIU, W.; WANG, Z.; LIU, X.; ZENG, N.; LIU, Y.; ALSAADI, F. E. A survey of deep neural network architectures and their applications. *Neurocomputing*, Elsevier, v. 234, p. 11–26, 2017. Citado na página 15.

LIU, Y.-T.; ZHANG, Y.-A.; ZENG, M. Adaptive global time sequence averaging method using dynamic time warping. *IEEE Transactions on Signal Processing*, IEEE, v. 67, n. 8, p. 2129–2142, 2019. Citado na página 35.

LLOYD, S. Least squares quantization in pcm. *IEEE transactions on information theory*, IEEE, v. 28, n. 2, p. 129–137, 1982. Citado 2 vezes nas páginas 15 e 33.

MA, Q.; LI, S.; SHEN, L.; WANG, J.; WEI, J.; YU, Z.; COTTRELL, G. W. End-to-end incomplete time-series modeling from linear memory of latent variables. *IEEE transactions on cybernetics*, IEEE, v. 50, n. 12, p. 4908–4920, 2019. Citado na página 36.

MA, Q.; ZHENG, J.; LI, S.; COTTRELL, G. W. Learning representations for time series clustering. *Advances in neural information processing systems*, v. 32, p. 3781–3791, 2019. Citado 2 vezes nas páginas 14 e 36.

MAATEN, L. V. D. Learning a parametric embedding by preserving local structure. In: *Artificial Intelligence and Statistics*. [S.l.: s.n.], 2009. p. 384–391. Citado na página 27.

MAATEN, L. v. d.; HINTON, G. Visualizing data using t-sne. *Journal of machine learning research*, v. 9, n. Nov, p. 2579–2605, 2008. Citado na página 26.

MACQUEEN, J. et al. Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. [S.l.], 1967. v. 1, n. 14, p. 281–297. Citado na página 19.

MADIRAJU, N. S.; SADAT, S. M.; FISHER, D.; KARIMABADI, H. Deep temporal clustering: Fully unsupervised learning of time-domain features. *arXiv preprint arXiv:1802.01059*, 2018. Citado 4 vezes nas páginas 26, 34, 36 e 46.

MAHALANOBIS, P. C. On the generalized distance in statistics. In: NATIONAL INSTITUTE OF SCIENCE OF INDIA. [S.l.], 1936. Citado na página 37.

MAHARAJ, E. A.; D'URSO, P.; CAIADO, J. *Time series clustering and classification*. [S.l.]: CRC Press, 2019. Citado na página 18.

MAHARAJ, E. A.; D'URSO, P. Fuzzy clustering of time series in the frequency domain. *Information Sciences*, Elsevier, v. 181, n. 7, p. 1187–1211, 2011. Citado na página 20.

MCLACHLAN, G. J.; BASFORD, K. E. *Mixture models: Inference and applications to clustering*. [S.l.]: M. Dekker New York, 1988. v. 38. Citado na página 46.

MEHMOOD, R.; ZHANG, G.; BIE, R.; DAWOOD, H.; AHMAD, H. Clustering by fast search and find of density peaks via heat diffusion. *Neurocomputing*, Elsevier, v. 208, p. 210–217, 2016. Citado na página 44.

MIN, E.; GUO, X.; LIU, Q.; ZHANG, G.; CUI, J.; LONG, J. A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access*, IEEE, v. 6, p. 39501–39514, 2018. Citado 4 vezes nas páginas 15, 24, 26 e 33.

MORITZ, S.; BARTZ-BEIELSTEIN, T. imputets: time series missing value imputation in r. *R J.*, v. 9, n. 1, p. 207, 2017. Citado na página 39.

MORIYA, T.; ROTH, H. R.; NAKAMURA, S.; ODA, H.; NAGARA, K.; ODA, M.; MORI, K. Unsupervised segmentation of 3d medical images based on clustering and deep representation learning. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Medical Imaging 2018: Biomedical Applications in Molecular, Structural, and Functional Imaging*. [S.l.], 2018. v. 10578, p. 1057820. Citado na página 35.

MÜLLER, M. *Information retrieval for music and motion*. [S.l.]: Springer, 2007. v. 2. Citado na página 46.

MÜLLNER, D. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*, 2011. Citado na página 18.

MURTAGH, F.; LEGENDRE, P. Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? *Journal of classification*, Springer, v. 31, n. 3, p. 274–295, 2014. Citado na página 19.

NETO, A. B. F.; CORRÊA, W. L. R.; PEROBELLI, F. S. Consumo de energia e crescimento econômico: uma análise do brasil no período 1970-2009. *Análise Econômica*, v. 34, n. 65, 2016. Citado na página 14.

OTTER, D. W.; MEDINA, J. R.; KALITA, J. K. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, IEEE, 2020. Citado na página 15.

PAPARRIZOS, J.; GRAVANO, L. k-shape: Efficient and accurate clustering of time series. In: ACM. *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. [S.l.], 2015. p. 1855–1870. Citado 2 vezes nas páginas 21 e 36.

PAPARRIZOS, J.; GRAVANO, L. Fast and accurate time-series clustering. *ACM Transactions on Database Systems (TODS)*, ACM New York, NY, USA, v. 42, n. 2, p. 1–49, 2017. Citado na página 14.

PATIL, C.; BAIDARI, I. Estimating the optimal number of clusters k in a dataset using data depth. *Data Science and Engineering*, Springer, v. 4, n. 2, p. 132–140, 2019. Citado na página 37.

PELLEG, D.; MOORE, A. W. et al. X-means: Extending k-means with efficient estimation of the number of clusters. In: *Icml*. [S.l.: s.n.], 2000. v. 1, p. 727–734. Citado na página 28.

POLWIANG, S. The time series seasonal patterns of dengue fever and associated weather variables in bangkok (2003-2017). *BMC Infectious Diseases*, BioMed Central, v. 20, n. 1, p. 1–10, 2020. Citado na página 14.

RENDÓN, E.; ABUNDEZ, I.; ARIZMENDI, A.; QUIROZ, E. M. Internal versus external cluster validation indexes. *International Journal of computers and communications*, v. 5, n. 1, p. 27–34, 2011. Citado na página 37.

RIOFRÍO, S.; POZO, D.; ROSERO, J.; VÁSQUEZ, J. Gesture recognition using dynamic time warping and kinect: A practical approach. In: IEEE. *2017 International Conference on Information Systems and Computer Science (INCISCOS)*. [S.l.], 2017. p. 302–308. Citado na página 21.

RODRIGUEZ, A.; LAIO, A. Clustering by fast search and find of density peaks. *science*, American Association for the Advancement of Science, v. 344, n. 6191, p. 1492–1496, 2014. Citado 2 vezes nas páginas 38 e 44.

ROUSSEEUW, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, Elsevier, v. 20, p. 53–65, 1987. Citado na página 29.

SANDERS, S.; GIRAUD-CARRIER, C. Informing the use of hyperparameter optimization through metalearning. In: IEEE. *2017 IEEE International Conference on Data Mining (ICDM)*. [S.l.], 2017. p. 1051–1056. Citado na página 31.

SCHUBERT, E.; SANDER, J.; ESTER, M.; KRIEGEL, H. P.; XU, X. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, ACM New York, NY, USA, v. 42, n. 3, p. 1–21, 2017. Citado na página 46.

SHAHAM, U.; STANTON, K.; LI, H.; NADLER, B.; BASRI, R.; KLUGER, Y. Spectralnet: Spectral clustering using deep neural networks. *arXiv preprint arXiv:1801.01587*, 2018. Citado na página 49.

SHI, J.; MALIK, J. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, Ieee, v. 22, n. 8, p. 888–905, 2000. Citado na página 46.

SINAGA, K. P.; YANG, M.-S. Unsupervised k-means clustering algorithm. *IEEE Access*, IEEE, v. 8, p. 80716–80727, 2020. Citado 3 vezes nas páginas 15, 37 e 43.

SOARES, E.; JR, P. C.; COSTA, B.; LEITE, D. Ensemble of evolving data clouds and fuzzy models for weather time series prediction. *Applied Soft Computing*, Elsevier, v. 64, p. 445–453, 2018. Citado na página 14.

SURABHI, A.; PAREKH, S. T.; MANIKANTAN, K.; RAMACHANDRAN, S. Background removal using k-means clustering as a preprocessing technique for dwt based face recognition. In: IEEE. *2012 International Conference on Communication, Information & Computing Technology (ICCICT)*. [S.l.], 2012. p. 1–6. Citado na página 14.

THIOLLIERE, R.; DUNBAR, E.; SYNNAEVE, G.; VERSTEEGH, M.; DUPOUX, E. A hybrid dynamic time warping-deep neural network architecture for unsupervised acoustic modeling. In: *Sixteenth Annual Conference of the International Speech Communication Association*. [S.l.: s.n.], 2015. Citado na página 46.

THORNTON, C.; HUTTER, F.; HOOS, H. H.; LEYTON-BROWN, K. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2013. p. 847–855. Citado na página 31.

TURECZEK, A. M.; NIELSEN, P. S.; MADSEN, H.; BRUN, A. Clustering district heat exchange stations using smart meter consumption data. *Energy and buildings*, Elsevier, v. 182, p. 144–158, 2019. Citado na página 30.

ÜNLÜ, R.; XANTHOPOULOS, P. Estimating the number of clusters in a dataset via consensus clustering. *Expert Systems with Applications*, Elsevier, v. 125, p. 33–39, 2019. Citado na página 37.

VLACHOS, M.; LIN, J.; KEOGH, E.; GUNOPULOS, D. A wavelet-based anytime algorithm for k-means clustering of time series. In: CITESEER. *In proc. workshop on clustering high dimensionality data and its applications*. [S.l.], 2003. Citado na página 20.

WICHT, B.; FISCHER, A.; HENNEBERT, J. Keyword spotting with convolutional deep belief networks and dynamic time warping. In: SPRINGER. *International Conference on Artificial Neural Networks*. [S.l.], 2016. p. 113–120. Citado na página 46.

XIE, J.; GIRSHICK, R.; FARHADI, A. Unsupervised deep embedding for clustering analysis. In: *International conference on machine learning*. [S.l.: s.n.], 2016. p. 478–487. Citado 3 vezes nas páginas 15, 26 e 34.

YANG, B.; FU, X.; SIDIROPOULOS, N. D.; HONG, M. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In: JMLR. ORG. *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. [S.l.], 2017. p. 3861–3870. Citado na página 34.

YANG, J.; LESKOVEC, J. Patterns of temporal variation in online media. In: *Proceedings of the fourth ACM international conference on Web search and data mining*. [S.l.: s.n.], 2011. p. 177–186. Citado na página 36.

YANG, J.; PARIKH, D.; BATRA, D. Joint unsupervised learning of deep representations and image clusters. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 5147–5156. Citado na página 35.

YU, Z.; NIU, Z.; TANG, W.; WU, Q. Deep learning for daily peak load forecasting–a novel gated recurrent neural network combining dynamic time warping. *IEEE Access*, IEEE, v. 7, p. 17184–17194, 2019. Citado na página 46.

ZAKARIA, J.; MUEEN, A.; KEOGH, E. Clustering time series using unsupervised-shapelets. In: IEEE. *2012 IEEE 12th International Conference on Data Mining*. [S.l.], 2012. p. 785–794. Citado na página 36.

ZAKI, M. J.; JR, W. M.; MEIRA, W. *Data mining and analysis: fundamental concepts and algorithms*. [S.l.]: Cambridge University Press, 2014. 444-445 p. Citado na página 46.

ZHANG, A.; SHEIKHOLESLAMI, G.; CHATTERJEE, S. *Wavelet-based clustering method for managing spatial data in very large databases*. [S.l.]: Google Patents, 2005. US Patent 6,882,997. Citado na página 20.

ZHANG, Y.; HEPNER, G. F. The dynamic-time-warping-based k-means++ clustering and its application in phenoregion delineation. *International Journal of Remote Sensing*, Taylor & Francis, v. 38, n. 6, p. 1720–1736, 2017. Citado na página 35.