FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Mapeamento automático da topologia de redes inteligentes de baixa tensão

João Afonso da Silva Picão

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Doutora Clara Gouveia

Co-orientador: Doutora Conceição Rocha

6 de Julho de 2020

Resumo

O Sistema Elétrico de Energia tem vindo a sofrer mudanças como consequência das políticas energéticas atuais que procuram viabilizar novos serviços de energia focados no consumidor e também aumentar a integração da produção de energia elétrica através de fontes renováveis. No sentido de promover a gestão ativa de das redes de distribuição, o conceito de Rede Inteligente (*Smart Grid*) tem implementado, o que representa uma mudança no paradigma destas redes. A infraestrutura destas redes inteligentes é sobreposta por uma camada de monitorização e controlo digital que permite que a integração de recursos energéticos distribuídos seja efetuada de um modo seguro.

A variabilidade associada aos recursos renováveis e a participação ativa dos consumidores introduz uma maior incerteza no diagrama de carga e operação da rede. Esta incerteza requer uma capacidade de monitorização e controlo melhorada para que sejam identificadas possíveis violações dos limites técnicos da rede e, com essa informação, definir estratégias de controlo para otimizar a operação do sistema.

A implementação da infraestrutura de contagem inteligente (*Advanced Metering Infrastructure*) é fundamental para a implementação de redes inteligentes. Para além da informação detalhada do consumo, esta infraestrutura permite também recolher informação adicional da rede, nomeadamente informação relativa aos valores de tensão nos nós da rede. A infraestrutura de contagem inteligente apresenta assim uma oportunidade para aumentar a observabilidade da rede. No entanto, existem ainda limitações significativas na caracterização da topologia e inclusivamente da fase de ligação dos contadores à rede de baixa tensão. Esta falta de informação limita a possibilidade de implementação de funcionalidades de gestão ativa da rede de baixa tensão. Como tal é necessário definir estratégias para efetuar o levantamento do mapeamento de fases e da topologia das redes de baixa tensão. Atualmente, a abordagem para a obtenção desta informação implica a deslocação de equipas ao terreno ou a instalação de equipamento adicional no Posto de Transformação que realize a identificação das fases de ligação dos clientes. Esta abordagem acarreta enormes esforços a nível financeiro e de recursos humanos pelo que não é viável a sua utilização generalizada nas redes de baixa tensão.

Esta dissertação tem como principal objetivo propor uma metodologia que permita realizar o mapeamento de fases da rede de baixa tensão de um modo mais viável. Tirando partido do histórico de medidas recolhidas pela infraestrutura de contagem inteligente, a metodologia tem como objetivo identificar a fase e a saída do transformador a que o consumidor está ligado. Com a realização desta identificação e mapeamento é possível reduzir os trabalhos de terreno e possivelmente evitar a instalação de equipamento adicional.

Abstract

Power Systems have been experimenting significant changes because of current energy policies, fostering the large-scale integration of renewable energy sources and enable new energy services focused on the consumer. To promote an active management of electricity grids, smart grid concept has been implemented, representing a change of paradigm particularly in distribution networks operation. The grid infrastructure is overlaid with a digital monitoring and control layer enabling the safe integration of distributed energy resources.

The variability associated with renewable resources and the active participation of consumers introduces greater uncertainty in the load and operation diagram of the network. This uncertainty requires an improved monitoring and controlling capacity of the network to identify possible violations of the technical limits and, with this information, define control strategies to optimize the operation of the system.

The implementation of an Advanced Metering Infrastructure is key in the implementation of smart grids. In addition to detailed consumption information, smart metering can also provide additional information related with the grid, namely voltage magnitude. The smart metering infrastructure thus presents an opportunity to increase the observability of the network. However, there are still significant limitations in the characterization of the low voltage network topology and even the connection phase of the smart meters. This lack of information limits the possibility of implementing features for active management of the low voltage network. In this sense, it is necessary to define strategies for phase mapping and the topology of low voltage networks. Currently, the approach used to obtain this information implies field work or the installation of additional equipment at the distribution transformers to carry out the identification of the customer phase connection. This approach results in enormous financial and human resource effort.

The main purpose of this dissertation is to propose a methodology that allows the phase and mapping of the low voltage network in a more viable way. Taking advantage of the history of measurements collected by the Advanced Metering Infrastructure, the methodology aims to identify the phase and the transformer feeder to which the consumer is connected. With this identification and mapping, it is possible to reduce the field work and possibly to avoid the installation of additional equipment.

Agradecimentos

Agradeço em primeiro lugar às minhas orientadoras, Doutora Clara Gouveia e Doutora Conceição Rocha, pela disponibilidade, ajuda e dedicação demonstrada na orientação deste trabalho.

Quero também agradecer ao INESC TEC por me ter acolhido na realização desta dissertação e ao Eng. Henrique Teixeira por me ter fornecido e ajudado na compreensão do código utilizado em partes desta dissertação.

Agradeço à minha família por todo o apoio dado ao logo do meu percurso académico.
Agradeço também a todos os amigos e colegas por todas as palavras de apoio e incentivo.

João Picão

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas e Símbolos

ACF	Autocorrelation Function
BT	Baixa Tensão
CH	Calinski - Harabasz
DTW	Dynamic Time Warping
EDP	Energias de Portugal
KL	Krzanowski - Lai
LCM	Local Cost Matrix
PACF	Partial Autocorrelation Function
PCA	Principal Component Analysis
PLC	Power Line Communication
PT	Posto de Transformação
TIC	Tecnologias de Comunicação e Informação

Capítulo 1

Introdução

Neste capítulo é apresentado o enquadramento e a motivação deste trabalho tendo em conta os objetivos ambiciosos para a descarbonização do sistema e o impacto que tem vindo a ter nos Sistemas Elétricos de Energia, nomeadamente com a implementação do novo paradigma das redes inteligentes (*Smart Grids*). Com base na motivação são também apresentados os objetivos específicos deste trabalho. Por fim, é apresentada a estrutura que esta dissertação segue.

1.1 Enquadramento

A União Europeia, em concordância com os objetivos definidos pelo Acordo de Paris de redução das emissões de gases de efeito de estufa e descarbonização da economia, tem seguido um conjunto de políticas que visam aumentar a eficiência energética dos sistemas elétricos de energia, aumentar a implementação de produção de energia através de fontes renováveis, criar apoios a projetos de redes inteligentes, aumentar a presença de veículos elétricos no parque automóvel europeu, e proteger os direitos dos consumidores de energia. Estas políticas têm conduzido a uma mudança de paradigma da operação da rede elétrica, em particular nas redes de distribuição de média e baixa tensão.

Relativamente a medidas de eficiência energética e implementação de produção de energia através de fontes renováveis, a União Europeia com a diretiva (EU) 2018/2001 impõe como objetivos até 2030: reduzir para metade a emissão dos gases de efeito de estufa no mínimo 40% relativamente aos níveis da década de 1990 e aumentar para 32% da energia total consumida nos estados membros a percentagem de energia produzida via fontes renováveis. Também na Diretiva (EU) 2018/844 é imposto um melhoramento de 32,5% de eficiência energética do sistema elétricos de energia e das suas instalações. Com a diretiva (EU) 2019/631 são também criadas medidas ainda mais restritivas relativamente às emissões de dióxido de carbono de veículos com motores de combustão interna e são também sugeridos incentivos à comercialização e compra de veículos de zero emissões. Paralelamente, a União Europeia também tem incentivado a criação e desenvolvimento de projetos na área das redes inteligentes (*Smart Grids*). Uma das principais preocupações das políticas europeias é o desenvolvimento de um sistema centrado no consumidor,

permitindo que este tenha acesso à informação do seu consumo detalhado e a sua participação ativa no sistema. Assim, no sentido de facilitar o acesso à informação sobre o seu consumo de energia, a União Europeia lançou também a Diretiva Europeia (EU) 2019/944 para que os estados membros tomem medidas de substituição dos equipamentos de medida analógicos de modo a criarem infraestruturas de contagem digital e inteligente (*Advanced Metering Infrastructure*). Em Portugal, foi publicado em Diário da República, a 2 de Agosto de 2019, o Regulamento n.º 610/2019 – Regulamento dos Serviços das Redes Inteligentes de Distribuição de Energia Elétrica, que vem estabelecer o enquadramento legal aplicável à prestação dos serviços no âmbito das redes inteligentes de distribuição de energia elétrica, nomeadamente no que respeita à disponibilização de dados aos consumidores e comercializadores.

Em consequência das políticas europeias, os sistemas elétricos europeus têm vindo a adaptar-se por forma a aumentar a capacidade de integração de produção de energia por fontes renováveis e acomodar a carga resultante do carregamento dos veículos elétricos, que na sua maioria serão integrados nas redes de distribuição de baixa tensão.

Face às metas ambiciosas para 2030 e ainda 2050, é expectável um aumento ainda mais significativo na integração destes recursos, o que irá aumentar a complexidade de operação na rede de distribuição. A variabilidade destes recursos (principalmente associados à produção de energia solar e eólica) introduz incertezas relativamente aos diagramas de cargas, que por sua vez também contribui para o aumento da probabilidade do sistema ultrapassar os seus limites técnicos de operação. Poderão assim surgir novos desafios à operação das redes de distribuição como consequência da integração destes recursos energéticos distribuídos. Estes desafios podem-se manifestar como: flutuações de tensão, trânsitos de potência inversos, injeção de harmónicos, entre outros [1]. É assim expectável que a rede seja operada mais próxima dos limites técnicos.

Para garantir que a rede de distribuição opere de forma segura e eficiente com a introdução dos recursos energéticos distribuídos, é necessário aumentar observabilidade e controlabilidade da rede de distribuição, em particular na rede de baixa tensão. A implementação de infraestruturas de contagem inteligente na rede de baixa tensão, através dos contadores inteligentes aumentou a capacidade de monitorização da rede, sendo capaz de recolher perfis de carga e tensão em todos os nós da rede de baixa tensão. Contudo, devido a limitações significativas na caracterização da topologia e das características elétricas dos condutores, a utilização destes dados neste momento é limitada para efeitos de monitorização e controlo.

A gestão ativa da rede de baixa tensão, exige assim o desenvolvimento de soluções inovadoras que tirem partido dos dados disponibilizados pela infraestrutura de contagem inteligente, de modo a caracterizar a topologia da rede e assim contribuir para o aumento da observabilidade e controlabilidade da rede.

1.2 Objetivos e Motivação

No seguimento das diretivas europeias, a implementação de dispositivos de telecontagem ou contagem inteligente, disponibiliza um conjunto de medidas relevantes de todos os nós da rede

de baixa tensão. são. Estas medidas são recolhidas e processadas pelo operador da rede de distribuição, e para além de ser utilizadas para efeitos de faturação de energia apresentam também um enorme potencial no sentido de melhorar a observabilidade e controlabilidade da rede de baixa tensão. Numa primeira fase, esta informação pode ser utilizada para realizar o mapeamento de fases e da topologia da rede, que, numa fase posterior, viabiliza a implementação de funções de gestão ativa da rede. Como tal, o mapeamento de fases e da topologia da rede de baixa tensão é um ponto fulcral no paradigma em que o Sistema Elétrico de Energia está a entrar.

A solução atual para o mapeamento de fases envolve a mobilização de equipas de técnicos para efetuar visitas a todas as instalações dos consumidores de modo a realizar a identificação tanto da fase, como da saída do posto de transformação a que estes se encontram ligados. Esta é a solução usada para realizar identificações de fases e saídas pontuais e que sejam necessárias, por exemplo, para efetuar uma intervenção da rede. No entanto, e apesar de ser uma solução exequível, não é uma solução em que o seu aumento de escala de atuação para toda a rede de baixa tensão seja fácil de implementar já que exige um número elevado de recursos humanos envolvidos. Para além do problema da escalabilidade, esta solução está sempre sujeita à introdução de erros associados ao modo como esta solução realiza o mapeamento de fases e da topologia.

No mercado existem também soluções que implicam a instalação de equipamentos nos postos de transformação da rede de baixa tensão. Dependendo do equipamento usado, o mapeamento de fases e da topologia é feito de diversos modos, por exemplo, comunicando com os equipamentos de contagem inteligente dos consumidores ou realizando a análise da corrente nas diferentes saídas do transformador e as suas variações. A generalização desta solução para toda a rede de baixa tensão do Sistema Elétrico de Energia também exige investimentos avultados devido à aquisição dos equipamentos e respetiva instalação.

Também já se começam a estudar soluções que permitam aproveitar os dados de medidas feitas pela infraestrutura de contagem inteligente que já se encontra instalada nas redes de baixa tensão. Neste trabalho será estudado o potencial desses dados para a realização do mapeamento de fases e da topologia das redes de baixa tensão. Assim sendo, o objetivo principal deste trabalho é o desenvolvimento de uma metodologia que recorre a técnicas de análise de dados e a métodos de *clustering* de modo a realizar a divisão dos dados recolhidos pela infraestrutura de contagem inteligente e obter o mapeamento de fases e da topologia da rede.

Para que o objetivo principal do trabalho seja atingido, existem alguns pontos que o trabalho deve abordar e desenvolver com sucesso:

1. Estudar e identificar metodologias para o mapeamento de fases e topologia da rede de baixa tensão;

2. Estudar e identificar os métodos de *clustering* e diferentes parâmetros aplicáveis ao problema do mapeamento de fase. Identificar ferramentas que permitam avaliar o desempenho de cada um dos métodos de *clustering*;

3. Realizar uma análise comparativa dos resultados dos métodos de *clustering* identificados no mapeamento de fases e topologia de redes de baixa tensão. A metodologia desenvolvida

deve ter em conta restrições específicas ao problema que permitam descartar métodos que eventualmente as violem, bem como utilizar as ferramentas de avaliação de *clustering* de modo a avaliar a robustez dos resultados;

A metodologia desenvolvida foi testada numa rede de baixa tensão típica, tendo em conta o histórico das medidas de tensão simulado. O estudo teve como objetivo comparar os resultados dos diferentes métodos relativamente à sua eficácia na identificação das fases corretas com o mapeamento real da rede.

1.3 Estrutura da Dissertação

O documento encontra-se dividido em cinco capítulos. O capítulo 1 descreve o contexto, motivação e objetivos do trabalho. O capítulo 2 descreve o conceito das redes inteligentes e identifica as funções que beneficiariam da informação da fase e topologia da rede. Ainda neste capítulo são apresentados alguns trabalhos desenvolvidos no sentido de realizar o mapeamento de fases e de topologia tendo como base os dados provenientes de dispositivos de contagem inteligente. No capítulo 3 são descritas as diferentes medidas de distância e algoritmos que podem ser implementados nos métodos de *clustering*. Também são apresentados alguns que tipicamente são utilizados para determinar os números ótimos de *clusters*. No capítulo 4 é apresentada a metodologia usada para determinar qual, ou quais, os métodos de *clustering* que obtém melhores resultados para o problema do mapeamento de fases e de topologia. Também são apresentados os resultados obtidos que resultam da implementação dessa metodologia com um conjunto de dados em que se tem conhecimento relativamente às fases de cada medida. Finalmente, no capítulo 5 são tiradas as devidas conclusões sobre o trabalho realizado e também são sugeridos trabalhos futuros que dão seguimento a esta dissertação.

Capítulo 2

Revisão bibliográfica/Estado de arte

2.1 Redes de Distribuição Inteligente: Conceito

A integração em larga escala de recursos energéticos distribuídos e de produção de energia elétrica a partir de fontes renováveis para fazer face à descarbonização do setor da energia e dos transportes, colocam alguns desafios para a operação das redes de distribuição, como por exemplo [2], flutuações de tensão e trânsitos de potência inversos [1]. Assim, para garantir a fiabilidade e robustez da operação é imperativo que se introduzam mudanças significativas no modo como estas redes são controladas através da implementação do conceito de redes inteligentes (*Smart Grids*).

A implementação do conceito de rede inteligente tem como objetivo transformar as redes elétricas para no futuro se tornarem [3]:

- Acessíveis. As redes elétricas devem ser capazes de acomodar com segurança a elevada integração de fontes de energia renovável, o carregamento de veículos elétricos e permitir a participação ativa na procura. Deve-se usar uma abordagem orientada para o mercado que inclua o desenvolvimento de novos serviços de valor acrescentado que premeiem a flexibilidade;

- Flexíveis. Devem ser capazes de gerir a elevada variabilidade associada às fontes de energia renovável e cargas, respondendo simultaneamente às necessidades dos consumidores;

- Fiáveis. Devem assegurar e melhorar a segurança e fiabilidade no fornecimento de energia elétrica;

- Económicas, através da melhoria da eficiência das redes, desde a produção até às instalações dos consumidores, e do desenvolvimento de serviços de energia inovadores e dedicados à participação ativa dos consumidores.

As redes inteligentes surgem com o objetivo de monitorizar e controlar de forma ativa e automática as redes de distribuição dotadas de recursos energéticos distribuídos como a produção

e armazenamento de energia elétrica. A monitorização e o controlo ativo que é levado a cabo nas redes inteligentes é suportado por Tecnologias de Comunicação e Informação (TIC) onde se podem também incluir a infraestrutura de contagem inteligente (*smart metering*).

O grupo de trabalho da União Europeia criado no âmbito das redes inteligentes desenvolveu um modelo conceptual representado na figura 2.1. A operação das redes segundo este modelo de redes inteligentes pode ser dividida em quatro domínios, que são: Operações, Mercado de Energia, Serviços de Energia e Utilizadores da Rede. O domínio das Operações é responsável garantir que o sistema opera de um modo seguro, estável e robusto enquanto também permite a participação do Utilizadores da Rede no Mercado de Energia. A interação dos Utilizadores da Rede no Mercado de Energia no sentido de efetuarem compras ou vendas de energia ou em serviços de suporte são garantidos pelo domínio dos Serviços de Energia. Os Utilizadores da Rede incluem todos os atores que intervêm na produção ou no consumo de energia elétrica [4].

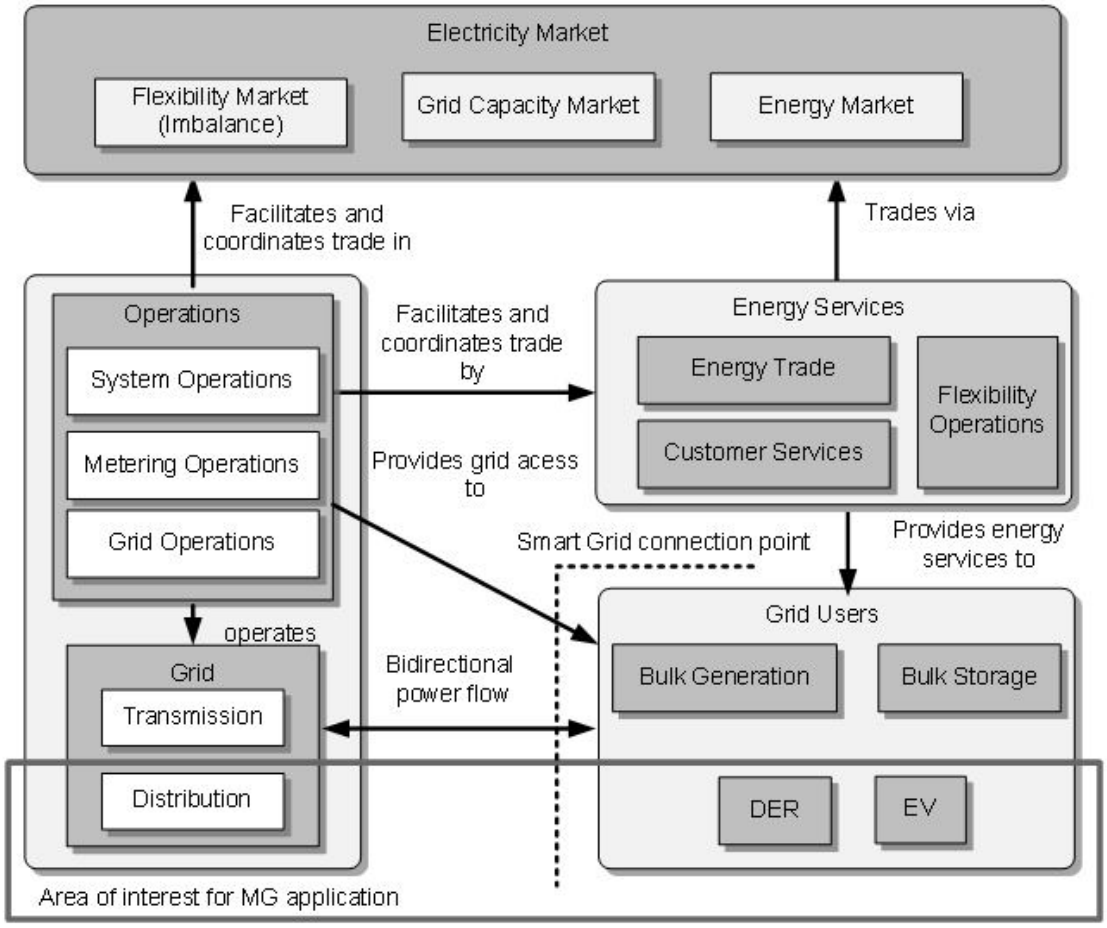

Figura 2.1: Modelo conceptual de redes inteligentes definido pela União Europeia (Adaptado [4]).

A implementação deste novo paradigma de gestão e operação da rede associado às redes inteligentes, assume a capacidade de monitorizar e controlar todo o sistema. Isso significa, no caso da rede de distribuição, estender essa capacidade até ao nó ao consumidor de baixa tensão.

Este aumento da observabilidade e controlabilidade da rede é um fator importante e que influencia o sucesso da implementação de estratégias de operação preventivas e ativas, compatíveis com os novos atores do sistema.

2.2 Arquitetura de gestão e controlo da rede de distribuição

No âmbito do desenvolvimento do conceito das redes inteligentes surgiram na Europa um conjunto de projetos de referência, incluindo em Portugal.O *InovGrid* foi o projeto desenvolvido pela EDP Distribuição no âmbito das redes inteligentes. Este projeto contou com a participação de equipas multidisciplinares que participaram no seu desenvolvimento. O *InovGrid* também foi selecionado pelo organismo da Comissão Europeia, *Joint Research Center*, como caso de estudo no sentido de desenvolver as diretrizes de usadas por este organismo na condução de análises de custo/benifício de projetos de redes inteligentes. Foi também o primeiro projeto nacional a receber a etiqueta de apoio da *European Electricity Grid Initiative*.

O *InovGrid* teve como objetivo responder aos seguinte desafios, que passam por: assegurar que a rede opera com a maior eficiência possível e, desse modo, contribuir para a redução de custos de operação; integrar de modo seguro na rede um grande número de dispositivos de produção distribuída e de veículos elétricos; e apoiar e possibilitar os consumidores a interagirem e participarem nos serviços de energia disponíveis.

Este projeto destacou-se, pela visão integrada para as redes inteligentes: que combina a implementação da infraestrutura de contagem inteligente com a definição de um conjunto de requisitos e aplicações para a implementação de conceitos de gestão ativa da rede de distribuição.

A arquitetura *InovGrid* especificada procura assim explorar as capacidades dos contadores inteligentes e dos controladores de transformadores de distribuição de modo a que, para além de funcionarem apenas como sistemas de contagem inteligente, possam também fornecer informações em tempo real sobre a rede que pode ser utilizada no sentido de ultrapassar os desafios já referidos. Ou seja, a informação recolhida pelos sistemas de contagem inteligentes, serve para alimentar diversas funções avançadas de monitorização, controlo e gestão ativa das redes de distribuição. Claro que toda esta informação tem de ser transmitida até uma central de dados onde são executadas as aplicações de monitorização e gestão avançada da rede. Como tal, o *InovGrid* também considera na sua arquitetura os equipamentos que possibilitam a transmissão de toda a informação que circula entre os diversos pontos das redes de distribuição. A arquitetura descrita encontra-se representada na figura 2.2 e é a arquitetura de referência da EDP Distribuição [5].

As funcionalidades de controlo gestão avançadas referidas na descrição da arquitetura do *InovGrid* não foram idealizadas apenas para beneficiarem o operador de sistema, mas para também poderem estar ao dispor do cliente. Assim sendo, do ponto de vista da gestão avançada e operação das redes foram consideradas funções como:

- Gestão de dados para efeitos de faturação e deteção de fraude;
- Reconfiguração da rede de média tensão;

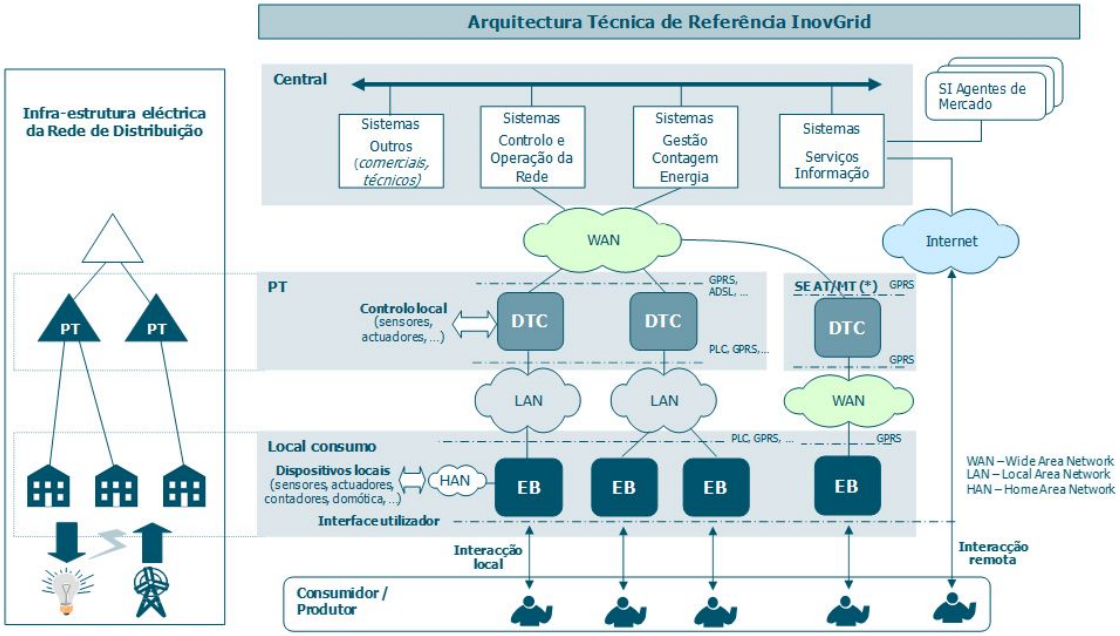

Figura 2.2: Arquitetura do projeto *InovGrid* [5]

- Regulação coordenada de tensão;
- Gestão e controlo da integração de produção dispersa e da rede de mobilidade elétrica;
- Operação da rede em modo isolado;
- Realização de ações de reposição de serviço;
- Controlo e comando dos sistemas de proteção da rede.

Do ponto de vista do consumidor, foram consideradas funções de gestão ativa de consumos e outros serviços de sistema.

A implementação do projeto *Inovgrid* no terreno foi focada principalmente nas funções de gestão de dados recolhidos por contadores inteligentes. Todas as restantes funções têm sido testadas em contexto de projetos piloto relacionados com as redes inteligentes, tais como o projeto *SuSTAINABLE* [6] e *InteGrid* [7].

2.3 Funcionalidades avançadas de gestão da rede de baixa tensão

Devido aos problemas associados à crescente implementação, na rede de baixa tensão, de pequenas fontes de produção de energia elétrica por via renovável e de veículos elétricos são necessárias ferramentas que permitam manter a segurança da operação da rede e tirar partido da flexibilidade de outros recursos energéticos distribuídos como os sistemas de armazenamento de energia. Estas ferramentas têm como objetivo dar suporte à monitorização de rede para a deteção

de condições de operação anómalas ou de emergência e a definição automática das soluções de controlo que permitem evitar ou repor as condições de operação normais.

Na figura 2.3 está representado o conceito definido no âmbito do projeto *InteGrid* para o controlo da rede de baixa tensão tendo em conta as funcionalidades referidas. Nesta secção será feita uma descrição mais aprofundada das mesmas.

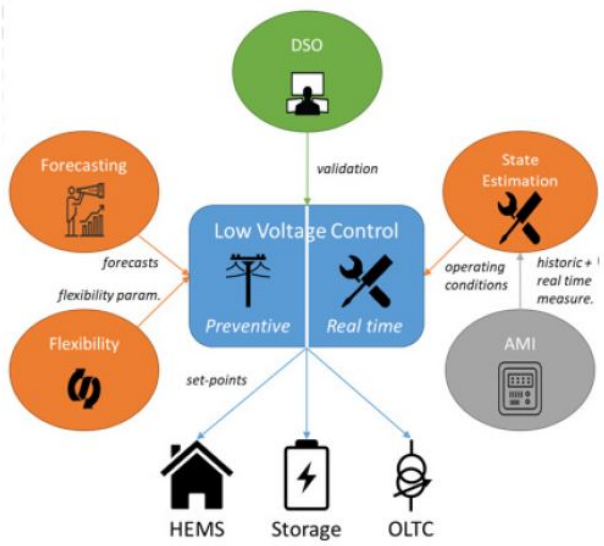

Figura 2.3: Conceito de controlo da rede de baixa tensão [7].

2.3.1 Estimação de Estados na rede de baixa tensão

A função de estimação de estados tem como objetivo criar uma imagem em tempo real da rede de baixa tensão com a informação sobre a tensão nos nós da rede, recolhida dos contadores inteligentes. No caso de existirem violações relacionadas com os valores de tensão nos diferentes nós da rede, o estimador de estados também gera esses alarmes [7]. Para realizar a estimação do estado da rede, o algoritmo desenvolvido utiliza os dados históricos dos contadores inteligentes e medidas recolhidas em tempo-real de sensores, caso estejam disponíveis. O principal objetivo é o de procurar evitar a instalação massiva de sensores para monitorizar a rede de baixa tensão e valorizar a informação disponibilizada pelos contadores inteligentes. No projeto *ADMS4LV*, o estimador de estados utiliza técnicas baseadas em Inteligência Artificial para conseguir determinar o estado de redes em que existe pouca informação relativamente à sua topologia. Neste caso em específico, os contadores inteligentes que alimentam o estimador vão mudando consoante a sua localização e outras variáveis de modo a maximizar a qualidade dos resultados [8]. A informação resultante desta função serve para alimentar diversas outras funções dos sistemas de gestão da rede, já que, para além de dar uma imagem em tempo real do funcionamento da rede, permite também calcular as restantes grandezas físicas da rede.

2.3.2 Controlo de Tensão coordenado na rede de baixa tensão

As funções de controlo de tensão atuam principalmente em equipamentos que permitam o controlo de potência ativa e reativa. Por exemplo, no âmbito do projeto europeu *SuSTAINABLE* foi desenvolvida uma aplicação para o controlo de tensão na rede de baixa tensão realizando a regulação da potência de unidades flexíveis de produção de energia por via renovável, sistemas de armazenamento e de cargas, com o objetivo maximizar a energia proveniente de fontes renováveis e minimizar os custos de operação da rede [6]. Como o número de equipamentos com este tipo de flexibilidade está a aumentar em redes de baixa tensão, consequência da crescente integração de recursos energéticos distribuídos, então as soluções de controlo de tensão têm evoluído no sentido de utilizarem algoritmos inteligentes nesse mesmo controlo.

Dependendo da quantidade de recursos controláveis existentes na rede e da informação relativa à caracterização da rede relativamente à sua topologia, identificação de fases e dos seus equipamentos, podem ser implementados diferentes tipos de funções de controlo de tensão ao nível da sua complexidade. Podem então ser implementadas funções de controlo de tensão que atuam reativamente, ou seja, o controlo de tensão só é feito caso sejam detetadas violações ao nível de limites de tensão através de sucessivas ordens enviadas aos equipamentos até que a violação deixe de se verificar. Também existem funções de controlo de tensão ativas e mais complexas que tiram proveito das funções de estimador de estados e trânsitos de potência de modo a prever os efeitos na rede das ordens que são enviadas aos equipamentos no sentido de resolver os problemas de tensão que possam existir e também promover a operação mais eficiente possível.

As funções baseadas no cálculo do trânsito de potências são usadas fundamentalmente para realizar simulações que avaliem o impacto na rede de estratégias de controlo tomadas pelo sistema de gestão da rede [7]. Os resultados das simulações dos trânsitos de potência auxiliam a determinar quais as melhores soluções de gestão da rede. As funções baseadas nos trânsitos de potência dependem de informação de caracterização da rede, como por exemplo, da sua topologia e da identificação de fases dos consumidores.

2.3.3 Deteção de fraudes

As funcionalidades de deteção de fraudes baseiam-se na análise e processamento dos dados provenientes das infraestruturas de contagem digital e inteligente. A informação detalhada sobre consumos de energia e os respetivos perfis de cada consumidor que é recolhida pelas infraestruturas de contagem inteligente permite que se identifiquem, com maior precisão, situações anómalas de perdas da rede. A existência de perdas pode ter diferentes causas, como por exemplo, existência de fraude, mau funcionamento de equipamentos ou até mesmo erros na recolha ou no processamento dos dados. Como tal, as funções de deteção de fraude correlacionam a informação das infraestruturas de tele contagem com informação de outras funções executadas pelo sistema de gestão da rede, de modo a identificar quais as causas de situações de perdas anómalas [9].

2.4 Mapeamento de fases e de topologia das redes de baixa tensão

Enquanto que as redes de muito alta, alta e média tensão possuem documentação muito detalhada e precisa quanto às ligações das três fases em que a energia é transmitida e distribuída [10], nas redes de baixa tensão existe pouca e, nalguns casos, até mesmo nenhuma informação relativamente à ligação dos consumidores às fases do sistema elétrico de energia. Esta falta de informação deve-se a registos pouco precisos sobre a ligação das instalações e da sua topologia e a ações de manutenção e de reparação que podem impor mudanças. Para além dessa falta de informação, nas redes de baixa tensão os consumidores têm a opção de efetuar a sua ligação com a rede em regime monofásico ou trifásico, sendo que a maioria das instalações elétricas estão ligadas em regime monofásico [11].

Acontece que a maioria das funções implementadas no âmbito das redes inteligentes e dos seus sistemas de gestão e monitorização da rede, e descritas no capítulo 2.3, requerem, para o seu funcionamento, informação sobre o mapeamento de fases e de topologia da rede que controlam. Como tal, é imperativo que se encontrem soluções eficazes na determinação dessa informação. Atualmente, existem duas principais soluções para a falta de informação sobre o mapeamento de fases e da topologia das redes: soluções baseadas na instalação de equipamento de medida na rede e soluções baseadas no tratamento e análise dos dados recolhidos pelos dispositivos de contagem inteligente.

2.4.1 Soluções baseadas em equipamento de medida

Já existem no mercado dispositivos que podem ser integrados em transformadores de distribuição que têm diversas funcionalidades de medição e contagem, comunicação com os sistemas de controlo e gestão da rede, mapeamento de fases e topologia, entre outras.

Um desses dispositivos é o modelo *4SLV* do fabricante *ZIV*. Este dispositivo é aplicado em transformadores de distribuição de média/baixa tensão e é apresentado como uma solução avançada para a supervisão de redes de baixa tensão. O mapeamento de fases e de topologia neste dispositivo pode ser feito de duas maneiras diferentes. A primeira baseia-se na análise nas variações de correntes devido a variações de carga. A segunda está relacionada com a robustez dos sinais PLC [12].

Também a o dispositivo *EWS DTVI* que integra a *DeepGrid* da *Eneida*, que é montado também nos transformadores de distribuição de média/baixa tensão, permite realizar o mapeamento de fases e de topologia, em conjunto com outras funções de monitorização de redes de baixa tensão [13].

2.4.2 Soluções baseadas na análise de dados recolhidos por contadores inteligentes

No sentido de realizar a identificação de fases a partir dos históricos de medidas provenientes de contadores inteligentes já exite trabalho realizado com diversas abordagens e métodos [10, 11, 14, 15, 16, 17, 18, 19].

Em [10] são consideradas medidas de tensão, corrente e energia consumida que são adquiridas pelos contadores inteligentes de 15 em 15 minutos. A partir destes dados é estudada a hipótese de haver uma boa correlação entre o perfil de tensão que pertençam à mesma fase, isto é, medidas de diferentes contadores que estejam ligados na mesma fase de uma saída do transformador irão apresentar uma correlação maior do que com as restantes medidas. Esta abordagem tem como inspiração a utilização deste método em aplicações de deteção e comparação de sinais por via da sua correlação [20, 21, 22, 23]. Neste trabalho foram identificados como potenciais problemas como dessincronizações temporais entre os contadores inteligentes e perdas nas linhas. No entanto, esses problemas foram ultrapassados e esta abordagem identificou corretamente a fase de ligação de 71 dos 75 contadores inteligentes da rede. Para além disso, deslocações ao terreno vieram a descobrir que a informação usada na comparação com os resultados estava na realidade errada e que na realidade, esta abordagem, para estes dados, foi capaz de identificar corretamente a fase de 74 dos 75 contadores inteligentes da rede.

Em [16] é apresentado um método baseado na análise da componente principal (PCA - *Principal Component Analysis*) e na sua interpretação teórico-gráfica para determinar a topologia de uma rede em regime permanente. Os dados usados neste trabalho são séries temporais de medidas de energia. Também em [18] é proposto um método semelhante ao de [16] usando medidas provenientes de contadores inteligentes nos mesmos intervalos de tempo. O método proposto neste trabalho deve ser capaz de considerar que as medidas usadas tenham ruído.

Tanto [15] e [17] são usados métodos baseados no *clustering k-means* do histórico das medidas para fazer o mapeamento de fases. Em [17] faz-se o *clustering* tendo em conta a correlação das medidas. À matriz com a correlação entre medidas aplica-se também a transformada de *Fisher Z* de modo a normalizar as distribuições das correlações. Com o resultado dessa normalização é que se aplica o algoritmo de *clustering k-means*. Em [15] é usado também algoritmo *k-means* mas sem o uso da transformada de *Fisher Z* nem da matriz com as correlações entre medidas. Contudo, os resultados deste trabalho sugerem que o método usado para o *clustering* teve uma precisão de 90% na identificação das fases dos contadores que adquiriram as medidas utilizadas.

Em [14] é apresentado um método que também tira partido da correlação entre medidas e conjuga-as com o método da teoria dos grafos para obter a identificação de fases. Os resultados obtidos por este método foram melhores quando comparados com os resultados obtidos com a aplicação do algoritmo de *clustering k-means*. Os dados usados para identificação de fases são dados reais adquiridos por contadores inteligentes de uma rede de baixa tensão belga.

Já em [11] é testado outro método com uma lógica diferente dos expostos anteriormente. Neste artigo, é proposto que a identificação de fases seja feita tendo em conta a Lei da Conservação de Energia. Isto é, a energia que é transmitida por cada fase do transformador tem de ser igual à soma da energia consumida no cliente e às perdas que ocorram no trajeto do transformador até ao cliente. A estimação das perdas nas linhas é um problema, no entanto os autores assumem que as perdas são conhecidas. Para realizar o mapeamento de fases, são analisadas as medidas dos transformadores e dos contadores inteligentes dos clientes. Essas medidas são dispostas em sistemas de equações lineares que pretendem descrever a Lei da Conservação de Energia e a partir

daí determinar quais os clientes que estão ligados a cada uma das fases à saída do transformador. Os resultados deste trabalho indicam que, caso os erros não cresçam significativamente com as medidas e o número destas medidas seja suficiente, é possível realizar a identificação da fase de cada um dos contadores. No entanto, para este método ser aplicado é imperativo que se tenha as medidas de energia do transformador e que as medidas realizadas em todos os contadores estejam sincronizadas.

A figura 2.4 esquematiza as diferentes abordagens usadas para realizar o mapeamento de fases baseado na análise dos dados recolhidos pelos contadores inteligentes.

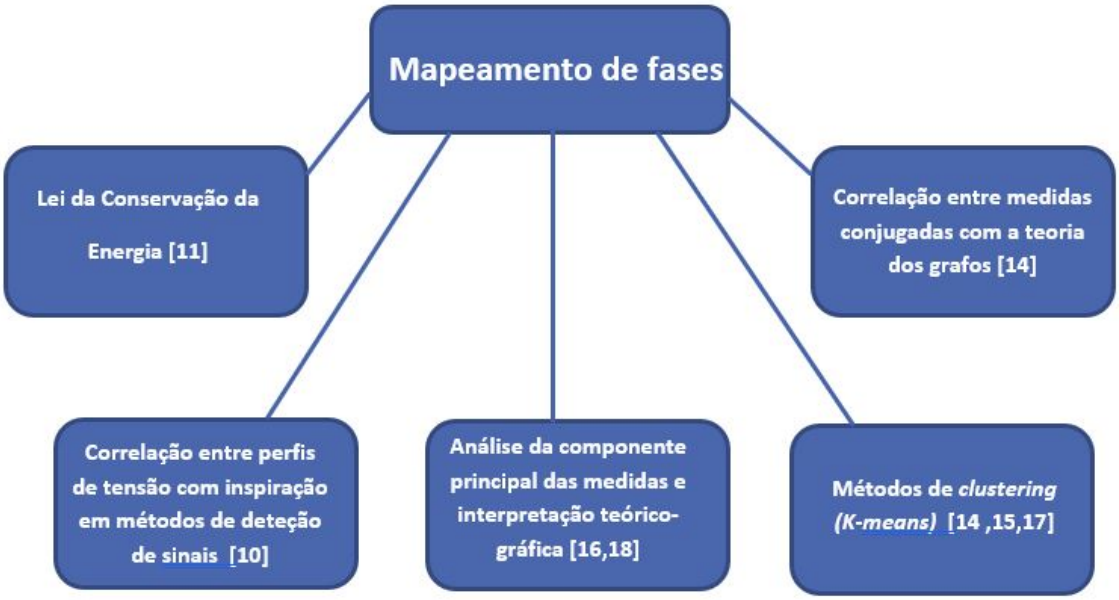

Figura 2.4: Esquema com as diferentes abordagens ao problema do mapeamento de fases.

A decisão de utilizar técnicas *clustering* apresentam resultados bastante interessantes, tendo por base a informação das tensões e não dependendo de outros dados como o consumo e estimação de perdas que poderão ter erros significativos associados. Considerando a utilização recente destas metodologias, carece uma análise mais aprofundada do impacto da escolha dos algoritmos de *clustering* e medidas de distância.

Capítulo 3

Métodos

3.1 Introdução

Atualmente vivemos num mundo em que somos constantemente assolados com imensuráveis quantidades de informação e dados. No entanto, todos estes dados de nada servem se não forem corretamente analisados e geridos. Um modo de analisar e tratar corretamente dados consiste em tentar encontrar similaridades ou diferenças nos mesmos de modo a agrupá-los ou separá-los consoante diferentes critérios para que seja possível classificá-los. Este tipo de abordagem remonta às próprias origens do ser humano em que, quando deparados com a necessidade de compreender novos objetos ou fenómenos, procedia à análise das similaridades ou dissimilaridades dos mesmos tendo como termo de comparação outros objetos ou fenómenos. [24].

Assim sendo, *clustering* consiste em juntar objectos ou indivíduos com características ou atributos semelhantes entre si num mesmo grupo e, simultaneamente, separar os que apresentam características diferentes para diferentes grupos. De um modo formal, para um dado conjunto de dados S podem ser criados k grupos (ou *clusters*) $C = C_1, ..., C_k$ [25]. De acordo com a teoria subjacente ao *clustering*, os métodos podem ser divididos em dois grupos, os métodos *hard clustering* e os métodos *soft clustering*. Os métodos *hard clustering* partem do pressuposto que cada elemento pertence a um e um só grupo, isto é, as suas características ditam automaticamente o grupo a que pertencem. Por outro lado, os métodos *soft clustering* partem do pressuposto que há elementos que apresentam características de mais de um grupo, isto é, há sobreposição entre os grupos e há elementos que pertencem a mais do que um grupo. Como tal, nos métodos *soft clustering* é associada a cada elemento uma probabilidade de pertencer a cada um dos grupos. Como exemplo de *hard clustering* temos os algoritmos de *k-means* e o hierárquico, já os algoritmos *fuzzy*, como por exemplo o *c-means*, são algoritmos de *soft clusterin* [24, 25].

O *clustering* também pode ser classificado como supervisionado ou não supervisionado. É possível que para um certo conjunto de dados exista alguma informação sobre a divisão dos dados. No caso de essa informação existir *a priori*, é possível usar esse conhecimento para alimentar algoritmos de aprendizagem ou usar essa informação de modo a poder classificar os restantes dados sobre os quais não se possui informação. Em situações deste género, o *clustering*

é classificado como sendo supervisionado [24]. Também pode ocorrer o caso de não se saber a classe a que cada elemento pertence ou mesmo quantas classes existem nos dados.. Nesse caso, espera-se que o método de *clustering* utilizado seja capaz de encontrar características e atributos intrínsecos aos dados e que encontre uma solução adequada para o *clustering*, que idealmente deverá representar a realidade desses mesmos dados. Este tipo de *clustering* é classificado como não supervisionado [24, 25].

Quando se quer realizar o *clustering* de uma série de dados é necessário ter em atenção 3 grandes passos que influenciam o resultado final do mesmo. O primeiro grande passo é extrair e selecionar atributos que permitam evidenciar semelhanças e diferenças de modo a que se possam criar os grupos em que os dados devem ser divididos. O segundo passo é responsável por determinar as similaridades ou dissimilaridades entre os diferentes elementos dos dados, e, com essa informação, executar o algoritmo de *clustering* de modo a efetuar a divisão e classificação dos dados. Neste passo são definidas as distâncias usadas para tratar os dados bem como os critérios de alocação dos dados no respetivo *cluster*, tendo em conta os atributos evidenciados no primeiro passo. Por fim, é necessário validar o *clustering* efetuado. Por essa razão, o último passo tenta avaliar se a seleção de atributos, as distâncias e o algoritmos de *clustering* apresentam resultados adequados e que espelham a realidade do *clustering* efetuando o cálculo de diversos índices de validação [24]. Neste trabalho apenas foram considerados o segundo passo e o terceiro, já que os dados usados correspondem a séries temporais do valor da tensão de cada consumidor e o objectivo é usar toda a informação contida nas mesmas.. Nos capítulos 3.2, 3.3 e 3.4 são aprofundados os temas das distâncias, algoritmos de *clustering* e os índices de validação, respetivamente, que foram usados neste trabalho.

3.2 Distâncias

Existem diversas abordagens e métodos para medir distâncias e similaridades entre os elementos que integram os dados. Na grande maioria das vezes os atributos destes elementos são descritos em vetores multi-dimensionais. No entanto, esses atributos podem ser quantitativos ou qualitativos, contínuos ou binários, nominais ou ordinários, e, consoante a sua classificação, são determinadas quais as distâncias ou similiridades mais adequadas.

Para que uma medida possa ser considerada uma distância ou dissimilaridade tem de possuir algumas propriedades [24]:

1. Simetria. A distância entre x_i e x_j deve ser igual à distância entre x_j e x_i, isto é:

$$D(x_i, x_j) = D(x_j, x_i) \tag{3.1}$$

2. Positividade. Para quaisquer x_i e x_j, a distância entre as duas variáveis deve ser sempre positiva, isto é:

$$D(x_i, x_j) \geq 0 \tag{3.2}$$

No entanto, se uma distância ou dissimilaridade tiver as propriedades abaixo, nesse caso é denominada de métrica. Como algumas das distâncias usadas neste trabalhado não se podem classificar como métricas, a partir deste ponto, todas as medidas serão denominadas como distâncias, mesmo que possuam propriedades de métricas.

3. Igualdade do triangular. A distância entre x_i e x_j tem de ser sempre menor ou igual que a soma das distâncias entre x_i, x_k e x_k, x_j:

$$D(x_i,x_j) \leq D(x_i,x_k) + D(x_k,x_j) \tag{3.3}$$

4. Reflexividade. A distância entre x_i e x_j tem de ser 0 no caso de $x_i = x_j$:

$$D(x_i,x_j) = 0 \qquad \text{, se } x_i = x_j \tag{3.4}$$

Analogamente, para que uma medida seja considerado como medida de similaridade tem de possuir uma série de propriedades e caso possua outro conjunto de propriedades também pode ser denominado como métrica de similaridade. Contudo, neste trabalho como não foram consideradas medidas de similaridade, não se entrará em mais detalhes relativamente a esse tema.

De um modo geral, as distâncias ou dissimilaridades são usadas quando os atributos das variáveis são contínuos, enquanto que as funções de similaridade são tipicamente usadas em variáveis com atributos qualitativos [24].

Quando se calculam as distâncias de conjunto de dados com N indivíduos o resultado é apresentado numa matriz NxN denominada matriz de proximidades, que é preenchida com as distâncias entre os indivíduos i e j.

As diferentes distâncias usadas neste trabalho são descritas nas sub-secções seguintes.

3.2.1 Distância de *Minkowski*

Considerando duas variáveis com p atributos, $x_i = (x_{i1},x_{i2},...,x_{ip})$ e $x_j = (x_{j1},x_{j2},...,x_{jp})$ a distância de *Minkowski* é calculada por [26]:

$$D(x_i,x_j) = \left(\sum_{l=1}^{p} |x_{il} - x_{jl}|^g \right)^{1/g} \tag{3.5}$$

Esta distância é invariável a qualquer translação ou rotação apenas quando $g = 2$. Quando uma variável tem atributos com valores elevados, esses atributos irão sobrepor-se aos restantes [24].

3.2.2 Distância Euclidiana

A distância Euclidiana é usada frequentemente em inúmeras aplicações e é o caso específico da distância de *Minkowski* em que g toma o valor dois, $g = 2$:

$$D(x_i, x_j) = \left(\sum_{l=1}^{p} |x_{il} - x_{jl}|^2 \right)^{1/2} \tag{3.6}$$

Esta distância, quando usada em métodos de *clustering*, tende a formar *clusters* hiperesféricos.

3.2.3 Distância DTW - *Dynamic Time Warping*

Cada vez mais surgem cenários em que os dados se encontram organizados em séries temporais, como por exemplo, em dados relacionados com o mercado da bolsa, dados médicos, monitorização de sistemas e maquinarias e com eles surge a necessidade de realizar o *clustering* destes mesmos dados [27].

No *clustering* de séries temporais a distância ou dissimilaridade DTW (*Dynamic Time Warping*) é a distância mais popular. Esta distância destaca-se das outras por ser extremamente eficiente no cálculo de dissimilaridades para este tipo de dados, já que minimiza os efeitos de distorções temporais. Para além disso, a distância DTW, permite procurar semelhanças entre séries temporais de comprimentos diferentes [28]. Por exemplo, se considerarmos a distância Euclidiana, esta é eficiente em muitas aplicações, mas na manipulação de dados representados por séries temporais tem as desvantagens de não conseguir lidar com séries de tamanhos diferentes, bem como ser sensível a ruído, escala e desvios temporais [27].

Apesar de eficiente, a distância DTW requer um grande esforço computacional quando comparado com outras distâncias. Considerando duas séries temporais com comprimentos n e m, o esforço computacional da distância DTW é $O(mn)$, o que é bastante [27]. Importa também referir que, apesar da distância DTW ser capaz de lidar com séries temporais de diferentes comprimentos, quando os comprimentos das séries são próximos, obtém-se melhores resultados ao realizarem-se interpolações lineares de modo a que as séries passem a ter comprimentos exatamente iguais [29].

O cálculo da distância DTW pode ser dividida em dois grandes passos.

No **primeiro passo** é construida uma matriz de dissimilaridades entre duas séries temporais, $X = (x_1, x_2, ..., x_n)$ e $Y = (y_1, y_2, ..., y_m)$ com comprimentos n e m, respetivamente, usando outras distâncias. Esta matriz é denominada por matriz dos custos locais ou LCM (*Local Cost Matrix*). Tipicamente, a distância usada no cálculo da matriz dos custos locais é a distância Euclidiana.

$$LCM(i, j) = \left(\sum |x_i - x_j|^2 \right)^{1/2} \quad , \; i \in [1, n] \; e \; j \in [1, m] \tag{3.7}$$

No **segundo passo**, com a matriz dos custos locais já construída, determina-se qual o caminho ótimo que percorre esta matriz com o menor custo acumulado possível.

Formalmente, esse caminho ótimo, com comprimento k, será uma sequência de pontos, $p = (p_1, p_2, ..., p_k)$, em que: $p_l = (i, j)$, $l \in [1, k]$, $i \in [1, n]$ e $j \in [1, m]$. Caminho esse que tem de cumprir uma série de restrições:

1. Condições de fronteira: O início e fim do caminho têm de corresponder aos pontos iniciais e finais das séries temporais. $p_1 = (1,1)$ e $p_k = (n,m)$;

2. Condição de monotonicidade: Esta condição tem como objetivo preservar a ordem cronológica dos pontos das séries temporais;

3. Condição do tamanho de salto: O tamanho do salto dado em cada iteração na procura do caminho ótimo deve ser limitada. A condição mais básica que deve cumprir é: $p_{l+1} - p_l \in \{(1,1),(1,0),(0,1)\}$.

A função de custo para qualquer que seja o caminho que percorre a matriz dos custos locais, c_p, é dada pela equação 3.8.

$$c_p(X,Y) = \sum_{l=1}^{k} c(x_i, y_j) \tag{3.8}$$

O caminho que se pretende encontrar é precisamente o que minimiza esta função. Uma maneira de encontrar o caminho ótimo é testar todos os caminho possíveis e a partir daí escolher o que tem o menor custo agregado. Contudo, esta solução significa que o número de caminhos possíveis cresce exponencialmente caso alguma das séries temporais aumente o seu comprimento. Para tentar ultrapassar esta situação, no cálculo da distância DTW é empregue métodos de programação dinâmica com complexidade $O(mn)$ já referida anteriormente. Assim sendo, a distância DTW é dada pelo custo agregado mínimo representado na equação 3.9.

$$DTW(X,Y) = min\{c_p(X,Y)\} \tag{3.9}$$

O algoritmo de programação dinâmica, para determinar a distância DTW, primeiro constrói uma matiz de custos D para depois determinar o custo mínimo. O processo de construção da matriz D é feito do seguinte modo:

1. Primeira linha: $D(1,j) = \sum_{v=1}^{j} c(x_1, y_v)\ ,\ j \in [1,m]$;

2. Primeira coluna: $D(i,1) = \sum_{v=1}^{i} c(x_v, y_i)\ ,\ i \in [1,n]$;

3. Restantes elementos: $D(i,j) = min\{D(i-1,j-1),\ D(i-1,j),\ D(1,j-1)\} + c(x_i, y_j)\ ,\ i \in [1,n]\ e\ j \in [1,m]$.

Com a matriz D construída, o caminho ótimo pode ser encontrado partindo do ponto $p = (m,n)$ até ao ponto $p = (1,1)$ e percorrendo o percurso com o menor custo e assim, determinar a distância DTW [28].

3.2.4 Distância ACF - *Autocorrelation Function*

Como o próprio nome indica, a distância ACF (*Autocorrelation Function*) tira proveito das funções de autocorrelação estimadas a partir dos dados para calcular as suas dissimilaridades.

Se considerarmos dois conjuntos de dados X e Y e que as suas autocorrelações foram estimadas e são $\rho_X = (\rho_{1,X}, ..., \rho_{l,X})^T$ e $\rho_Y = (\rho_{1,Y}, ..., \rho_{l,Y})^T$, então a distância ACF entre estas duas séries de dados pode ser definida pela equação 3.10 [30].

$$d_{ACF}(X,Y) = \sqrt{(\rho_X - \rho_Y)^T \Omega (\rho_X - \rho_Y)} \tag{3.10}$$

Onde Ω é a matriz de pesos associada a cada medida.

Existem duas escolhas comuns relativamente ao que fazer com esta matriz de pesos Ω:

1. Considerar a matriz dos pesos como uma matriz de identidade ($\Omega = I$). Com esta consideração, a distância ACF passa a ser, essencialmente, a distância Euclidiana entre as funções de autocorrelação dos dados. A distância ACF passa ser calculada a partir da equação 3.11.

$$d_{ACF}(X,Y) = \sqrt{\sum_{i=1}^{l} (\rho_{i,X} - \rho_{i,Y})^2} \tag{3.11}$$

2. Considerar um declínio geométrico dos pesos com o atraso **(lag)** das autocorrelações, pelo que a distância ACF, neste caso, passa a ser dada pela equação 3.12:

$$d_{ACF}(X,Y) = \sqrt{\sum_{i=1}^{l} p(1-p)^i (\rho_{i,X} - \rho_{i,Y})^2} \quad \text{, com } 0 < p < 1. \tag{3.12}$$

Distâncias análogas podem ser construídas considerando, por exemplo, as funções de autocorrelação parciais (PACF - *Partial Autocorrelation Function*). Também as considerações feitas para a matriz dos pesos Ω podem ser aplicadas nestes casos [31].

3.2.5 Distância de *Mahalanobis - Wasserstein*

No sentido de se calcularem distâncias entre dados com múltiplas variáveis e representados por histogramas, Verde e Irpino [32] propuseram a distância de *Mahalanobis - Wasserstein*. Esta distância também se adapta facilmente de modo a ser usada quando os dados têm múltiplas variáveis e são representados por distribuições empíricas [33].

A distância de *Wasserstein* entre duas variáveis aleatórias f e g, com distribuições F e G, foi definida em [32] e encontra-se descrita na equação 3.13

$$d_W(F,G) := \left(\int_0^1 (F^{-1}(t) - G^{-1}(t))^2 dt \right)^{1/2} \tag{3.13}$$

Onde F^{-1} e G^{-1} são as funções dos quantis das duas distribuições F e G. A distância de *Wasserstein* pode ser comparada como uma extensão da distância Euclidiana entre as funções dos quantis, quando computada de modo a comparar duas distribuições de probabilidades [33].

Tendo em contas as características dos dados usados neste trabalho, é usada a versão discreta da distância de *Wasserstein*, ou *Mallows*, adaptada aos dados com múltiplas variáveis representadas por distribuições empíricas. Sendo que a distância de *Mallows* entre duas distribuições empíricas é dada pela equação 3.14 [33].

$$d_W(f,g) = \left(\frac{1}{n} \sum_{i=1}^{n} \left| F_{(i)}^{-1} - G_{(i)}^{-1} \right|^2 \right)^{1/2} \tag{3.14}$$

Por razões de facilidade de escrita, a partir deste ponto, esta distância será referida apenas por distância de *Wasserstein*.

3.3 Algoritmos de *clustering*

Dado que já foi abordado o tema da determinação de distâncias e da matriz de dissimilaridades no capítulo 3.2, neste capítulo, ao fazer referência aos métodos de *clustering* apenas se têm em conta procedimentos e metodologias usadas no *clustering* após terem sido determinadas as distâncias e dissimilaridades.

O facto de não existir uma definição precisa sobre quais os métodos ótimos para realizar o *clustering*, surgem diferentes métodos de tratar e dividir os dados iniciais tendo em conta as suas dissimilaridades no sentido de proceder ao seu *clustering* [34]. Estes métodos e algoritmos têm, logicamente, diferenças que vão de encontro às características dos dados e as suas dissimilaridades de modo a que o *clustering* seja o melhor possível tendo em conta o problema em causa. Em [35] é sugerido que os diferentes métodos de *clustering* sejam divididos em dois grandes grupos: os métodos de ***clustering* hierárquico** e os métodos de **partição de dados** [25].

3.3.1 *Clustering* Hierárquico

Os métodos de *clustering* hierárquico baseiam-se na divisão dos dados partindo de apenas um *cluster* que inclui todos os objetos dos dados e ir sucessivamente criando mais divisões até que cada objeto tenha o seu próprio *cluster* ou vice-versa. Quando um método de *clustering* hierárquico parte de apenas um *cluster* que inclui a totalidade dos dados e vai realizando sucessivas divisões, então é considerado um método **divisivo**. Se, por outro lado, o método partir de uma série de *clusters* que apenas contém um dos objetos dos dados e for realizando sucessivas uniões até acabar com apenas um *cluster* que inclui a totalidade dos dados, então é considerado um método de **aglomeração** [25].

Uma vez que neste trabalho apenas se usaram métodos de *clustering* hierárquico de aglomeração, apresenta-se agora o procedimento típico na execução e aplicação deste tipo de métodos [24]:

1. Criação de tantos *clusters* quanto o número de objetos que fazem parte do conjunto de dados que se quer analisar. Cada *cluster* terá apenas um objeto;

2. Procurar a distância mínima entre dois *clusters*:

$$D(C_i, C_j) = min\{D(C_m, C_l)\} \quad , \quad 1 \leq m, l \leq N \quad e \quad m \neq l \tag{3.15}$$

   Em que N é o número de total de objetos dos dados. Os *clusters* com a menor distância entre si, C_i e C_j, serão agregados passando a representar apenas um novo *cluster*;

3. Recalcular a matriz das distâncias entre *clusters* de modo a que se tenha em conta o novo *cluster* formado;

4. Repetir os passos 2 e 3 até que a totalidade dos objetos se encontre em apenas um *cluster*.

A divisão ou agregação referida acima pode ser feita tendo em conta diferentes critérios. Como tal, os métodos de *clustering* hierárquico também podem ser classificados consoante o critério que usam:

- *Clustering* de ligação única: Neste método, a distância entre dois *clusters* é dada pela menor distância entre dois objetos em que cada um desses objetos pertence ao respetivo *cluster* [36]. A desvantagem associada a este método é que pode acontecer que dois *clusters* que apresentem atributos distintos entre eles, mas que dois dos seus objetos próximos das fronteiras se encontrem próximos. Essa proximidade pode levar à agregação errada desses dois *clusters* [37];

- *Clustering* de ligação completa: Neste método, a distância entre dois *clusters* é considerada como sendo a máxima distância entre dois objetos que pertençam a *clusters* distintos [38].

- *Clustering* de ligação média: Neste método, a distância entre distância entre dois *clusters* é dada pela ditância entre as médias de todos os objetos que pertençam ao mesmo *cluster* [39, 40].

Tendo em conta o critério usado na definição das distâncias entre *clusters*, pode-se dizer que os métodos de ligação única e ligação completa são métodos gráficos já que apenas avaliam distâncias entre os objetos de cada um dos *clusters* sem realizarem outro tipo de cálculos. Já o método de ligação média pode ser considerado como um método geométrico já que são feitos cálculos geométricos para determinar os centros dos *cluster* e só depois é avaliada a distância entre os *clusters* [24].

De um modo geral, os métodos de *clustering* hierárquico são vantajosos em termos de versatilidade e de observação de resultados. Em termos de versatilidade, por exemplo, os métodos de ligação única mantém boas performances quando os *clusters* não são isotrópicos, quando se encontram bem separados ou também quando os estes são concêntricos. Relativamente à

visualização de resultados, os métodos de *clustering* hierárquico apresentam os seus resultados em dendogramas que permitem ao utilizador definir onde quer realizar o corte do mesmo de acordo com os seus critérios e verificar qual o número de *clusters* obtido no ponto de corte. Se por algum motivo o utilizador opta por outro critério que mude a localização do corte no dendograma e que altere o número de *clusters* obtido não é necessário realizar outro *clustering*, basta apenas modificar o valor de corte do dendograma [25].

Por outro lado, o *clustering* hierárquico também apresenta algumas limitações e desvantagens. Uma desvantagem apontada muitas vezes aos métodos de *clustering* hierárquico é o facto de quando um objeto é atribuído a um dado *cluster*, este nunca mais é considerado durante o algoritmo. Isto é, a atribuição é final e não é alvo de reconsiderações [25, 24]. Estes métodos também apresentam limitações ao nível do esforço computacional que requerem. Considerando que o conjunto de dados em análise tem n objetos ou variáveis, o esforço computacional do *clustering* hierárquico é de $O(n^2)$, o que significa que, para grandes conjuntos de dados, estes métodos são muito demorados e que requerem muita memória em termos de *Inputs/Outputs* [25]. Como também já foi referido, estes métodos são sensíveis a ruído e a objetos das zonas exteriores dos *clusters* o que, nalguns casos, pode comprometer a robustez do método [24].

3.3.2 Métodos de Partição de dados

Ao contrário dos métodos de *clustering* hierárquico, os métodos de partição de dados partem de um *clustering* inicial, definido aleatoriamente ou tendo em conta algum critério imposto pelo utilizador, e tentam realocar os elementos dos dados em diferentes *clusters* até obter um resultado que otimize ou minimize os critérios de convergência definidos. O número k de *clusters* em que se pretende dividir os dados também tem de ser especificado pelo utilizador [24, 25].

De um ponto de vista ideal, o resultado ótimo do *clustering* obtido por este tipo de métodos, seja qual for o critério de convergência que é utilizado, deve ser encontrado tendo em conta todas as hipóteses de *clustering* possíveis [24, 25]. Contudo, existem inúmeras hipóteses de *clustering* que podem ser obtidas e a exploração de todas representa um esforço computacional que simplesmente não é viável. Tendo em conta a inviabilidade da exploração de todas as possibilidades de *clustering*, desenvolveram-se métodos heurísticos que têm como objetivo obter soluções muito aproximadas àquelas que deveriam ser obtidas pelo método ideal [24, 25].

Existem vários critérios de convergência usados pelos métodos de partição de dados, critérios esses que são importantes para os sucesso dos mesmos [41]. O critério que é usado com mais frequência baseia-se na soma dos quadrados dos erros [24, 25] e todos os métodos de partição de dados usados neste trabalho usam todos este critério.

O somatório do quadrado dos erros de um dado *clustering* com k *clusters* e n objetos, é dado por:

$$J = \sum_{i=1}^{k} \sum_{j=1}^{n} \gamma_{ij} ||x_j - m_i||^2 \tag{3.16}$$

Em que: m_i é o centro do *cluster i* e γ_{ij} é uma variável que toma valor 1 quando o objeto x_j pertence ao *cluster i* e 0 quando tal não é verdadeiro.

Um dos métodos de partição de dados que utiliza o critério da soma do quadrado dos erros é o ***k-means*** [42].

O *k-means* é um método simples e de fácil implementação, em que se consideram que o centro de um *cluster k* com *n* elementos, é dado pela média, μ_k, de todos os objetos que integram esse mesmo *cluster* [25]:

$$\mu_k = \frac{1}{n_k} \sum_{q=1}^{n_k} x_q \tag{3.17}$$

O procedimento que é usado para realizar o *clustering* utilizando o *k-means* é:

1. Inicializar aleatoriamente ou com algum conhecimento *a priori* uma série de *k clusters* e calcular a matriz $M = [m_1, m_2, ..., m_k]$ com os respetivos centros;

2. Realocar os objetos dos dados nos *clusters* que lhes forem mais próximos. Isto é, $x_j \in C_w$ se $||x_j - m_w|| \leq ||x_j - m_i||$, para $j \in [1,n]$, $i \neq w$ e $i \in [1,k]$;

3. Recalcular a matriz com os centros dos *clusters* tendo em conta as modificações efetuadas em 2;

4. Repetir os passos 2 e 3 enquanto o critério não for cumprido ou enquanto não for ultrapassado o número de máximo iterações que se possa ter definido.

O *k-means* é um método vantajoso já que tem um esforço computacional $O(nkd)$ em que, tipicamente, *k* e *d* são valores muito inferiores aos valores de *n*, pelo que é um método interessante quando se quer realziar o *clustering* a dados com muitos elementos *n* [24]. Outras vantagens passam também pela facilidade de implementação, interpretação de resultados, velocidade de convergência e adaptabilidade a dados esparsos [43].

Contudo, o método *k-means* também apresenta algumas desvantagens que não devem ser descuradas.

Uma dessas desvantagens passa pelos *clusters* iniciais do algoritmo. Como foi referido anteriormente, estes *clusters* são definidos aleatoriamente ou então consoante algum conhecimento que se possa ter *a priori*, sendo que não existem modos universais e eficientes de garantir que estes primeiros *clusters* são os ideais para a obtenção do resultado ótimo no fim do algoritmo. Uma maneira de tentar minimizar esta desvantagem é pela realização de várias corridas, isto é, correr o algoritmo diversas vezes fazendo com que o *clustering* de partida é diferente de corrida para corrida [24].

Quando se implementa um algoritmo *k-means*, também é preciso ter em conta que o resultado final apresentado pode não ser o ótimo global. Ou seja, este algoritmo não tem capacidade de reconhecer se o resultado ótimo a que está a chegar é um ótimo global ou local. Pode acontecer

que este ótimo seja local e o algoritmo não tenha capacidade para sair deste ótimo e assim nunca chega ao ótimo global [24].

Por fim, o modo como os centros dos *clusters* são calculados no *k-means* também apresenta limitações. A mais óbvia é que a utilização deste algoritmo está limitado a aplicações em que seja possível calcular as médias dos objetos, o que nem sempre é o caso. Outra limitação do *k-means* é que ao considerar todos os pontos do *cluster* para o cálculo do seu centro, pode-se estar a considerar ruído e pontos exteriores do *cluster* que podem distorcer o resultado do seu centro. A solução, para esta limitação em particular, passa por considerar outras maneiras de definir o centro dos *clusters* [24, 25]. O algoritmo ***k-medoids*** foi desenvolvido tendo em conta este facto e em vez de calcular as médias dos objetos para definir os centros dos *clusters*, usa a mediana dos objetos de cada *cluster* para definir o seu centro [34]. Deste modo, este algoritmo torna-se mais robusto relativamente ao ruído e a pontos exteriores dos *clusters* [25].

3.4 Índices para o cálculo do número ótimo de *clusters*

Como já foi referido anteriormente, tanto as distâncias e dissimilaridades como os algoritmos de *clustering* usados influenciam o resultado final do *clustering*. Como tal, é essencial que também sejam desenvolvidos índices que avaliem a confiança dos resultados obtidos pelos métodos de *clustering*. Para além da avaliação dos resultados obtidos, os índices também são uma ferramenta importante na determinação do número ótimo de *clusters*. Para o mesmo conjunto de dados usando a mesma distância e o mesmo algoritmo de *clustering* calculam-se os índices de validação para resultados com diferentes números de *clusters*, o resultado que tiver o melhor índice é aquele que determina o número ótimo de *clusters*.

Os índices de validação combinam a informação relativa à compactação de cada *cluster*, bem como a separação entre os *clusters* e ainda outros fatores como propriedade geométricas e estatísticas dos dados.

Nesta secção são apresentados os índices usados neste trabalho no sentido de determinar o número ótimo de *clusters* com base na revisão feita em [44].

Antes de apresentar os índices, considera-se que:

$n =$ número de observações das variáveis dos dados;

$p =$ número de variáveis dos dados;

$q =$ número de *clusters*;

$X = \{x_{ij}\}$, $i = 1,2,...,n$ e $j = 1,2,...,p$ Onde X é a matriz nxp com p variáveis, medidas ao longo de n observações;

$\overline{X} =$ Matriz qxp com as médias dos *clusters*;

$\overline{x} =$ Centróide dos dados da matriz X;

$n_k =$ Número de objetos que pertencem ao *cluster* C_k;

$c_k =$ Centróide do *cluster* C_k;

$x_i =$ Vetor com dimensão p com as observações do objeto i do *cluster* C_k;

$||x|| = (x^T x)^{1/2}$;

$W_q =$ Matriz de dispersão interna dos dados divididos em q *clusters*:

$$W_q = \sum_{k=1}^{q} \sum_{i \in C_k} (x_i - c_k)(x_i - c_k)^T \tag{3.18}$$

$B_q =$, Onde B_q Matriz de dispersão entre *clusters* dos dados divididos em q *clusters*:

$$B_q = \sum n_k (c_k - \overline{x})(c_k - \overline{x})^T \tag{3.19}$$

$N_t =$ Número total de pares de observações de um conjunto de dados:

$$N_t = \frac{n(n-1)}{c} \tag{3.20}$$

$N_w =$ Número total de pares de observações que pertencem ao mesmo *cluster*:

$$N_w = \sum_{k=1^q} \frac{n_k(n_k-1)}{c}; \tag{3.21}$$

$N_b =$ Número total de pares de observações que pertencem a diferentes *clusters*:

$$N_b = N_t - N_w \tag{3.22}$$

$S_w =$ Soma das distâncias internas dos *clusters*:

$$S_w = \sum_{k=1}^{q} \sum_{i,j \in C_k i \leq j} d(x_i, x_j) \tag{3.23}$$

$S_b =$ Soma das distâncias entre *clusters*:

$$S_b = \sum_{k=1}^{q-1} \sum_{l=k+1}^{q} \sum_{i,j} d(x_i, x_j) \quad , i \in C_k \ , \ j \in C_l \tag{3.24}$$

3.4.1 Índice *Krzanowski - Lai* - KL

Índice proposto por Krzanowski e Lai em que o número de *clusters* q que maximiza o índice é considerado como o número ótimo de *clusters*. Este índice é definido por:

$$KL(q) = \left| \frac{(q-1)^{2/p} trace(W_{q-1}) - q^{2/p} trace(W_q)}{(q)^{2/p} trace(W_q) - (q+1)^{2/p} trace(W_{q+1})} \right| \tag{3.25}$$

3.4.2 Índice *Calinski - Harabasz* - CH

Índice proposto por Calinski e Harabasz em que o número de *clusters* q que maximiza o índice é considerado como o número ótimo de *clusters*. Este índice é definido por:

$$CH(q) = \frac{trace(B_q)/(q-1)}{trace(W_q)/(n-q)} \tag{3.26}$$

3.4.3 Índice *Hartigan*

Índice proposto por Hartigan é definido por:

$$\text{Hartigan} = \left(\frac{trace(W_q)}{trace(W_{q+1})} - 1 \right) (n-q-1) \tag{3.27}$$

Em que $q \in \{1,...,n-2\}$. Para o índice de Hartigan, o número de *clusters* q que maximiza a diferença entre níveis hierárquicos é considerado como o número ótimo de *clusters*.

3.4.4 Índice de *McClain*

Índice proposto por McClain e Rao em que o número de *clusters* q que minimiza o índice é considerado como o número ótimo de *clusters*. Este índice consiste no cálculo do rácio entre a média das distâncias internas de um *cluster*, divida pelo número de distâncias internas de um *cluster* e a média das distâncias entre *clusters*, divida pelo número de distâncias entre *clusters*:

$$\text{McClain} = \frac{S_w/N_w}{S_b/N_b} \tag{3.28}$$

3.4.5 Índice *Gamma*

Índice proposto por Baker e Hubert e que representa uma adaptação da estatística *Gamma* de Goodman e Kriskal. O número de *clusters* q que maximiza o índice é considerado como o número ótimo de *clusters*. Este índice faz comparações entre as dissimilaridades internas de um *cluster* e as dissimilaridades entre *clusters*. Se as dissimilaridades internas de um *cluster* forem menores que dissimilaridades entre *clusters* então considera-se que a comparação é concordante ($s(+)$). Se as dissimilaridades internas de um *cluster* forem maiores que dissimilaridades entre *clusters* então considera-se que a comparação é discordante ($s(-)$). A possibilidade de ambas as dissimilaridades serem iguais é ignorada por este índice. Assim sendo, o índice *Gamma* é definido por:

$$\text{Gamma} = \frac{s(+) - s(-)}{s(+) - s(+)} \tag{3.29}$$

3.4.6 Índice *Gplus*

Índice examinado por Milligan em que o número de *clusters* q que minimiza o índice é considerado como o número ótimo de *clusters*. Este índice é definido por:

$$\text{Gplus} = \frac{2s(-)}{N_t(N_t-1)} \tag{3.30}$$

Em que: $s(-)$ é o número de vezes que dois pontos que pertencem ao mesmo *cluster* têm uma distância maior que dois que pertencem a *clusters* distintos.

3.4.7 Índice *Tau*

Índice testado por Milligan em que o número de *clusters* q que maximiza o índice é considerado como o número ótimo de *clusters*. Este índice é definido por:

$$\text{Tau} = \frac{s(+) - s(-)}{[(N_t(N_t-1)/2 - t)(N_t(N_t-1)/2)]^{1/2}} \tag{3.31}$$

Em que: $s(-)$ é o número de vezes que dois pontos que pertencem ao mesmo *cluster* têm uma distância maior que dois que pertencem a *clusters* distintos e $s(+)$ é o número de vezes que dois pontos que pertencem ao mesmo *cluster* têm uma distância menor que dois que pertencem a *clusters* distintos.

3.4.8 Índice de *Dunn*

Índice definido por Dunn em que o número de *clusters* q que maximiza o índice é considerado como o número ótimo de *clusters*. Este índice calcula o rácio entre a mínima distância interna de um *cluster* e a máxima distância entre *clusters*:

$$\text{Dunn} = \frac{min_{1\leq i\leq j\leq q} d(C_i, C_j}{max_{1\leq k\leq q} diam(C_k)} \tag{3.32}$$

Em que: $d(C_i, C_j$ é a função de dissimilaridade entre dois *clusters* C_i e C_j definida por $d(C_i, C_j) = min_{x\in C_i, y\in C_j} d(x,y)$ e $diam(C)$ é o diâmetro de um *cluster* C definido por $diam(C) = max_{x,y\in C} d(x,y)$.

3.4.9 Índice SD

Índice que se baseia na média de dispersão para *clusters* e na sua separação. O número de *clusters* q que minimiza o índice é considerado como o número ótimo de *clusters*. O índice é definido por:

$$\text{SDindex}(q) = \alpha \text{Scat}(q) + \text{Dis}(q) \tag{3.33}$$

Em que: o primeiro termo, $\text{Scat}(q)$, é calculado segundo a equação 3.34 e que indica a média das distâncias internas de um *cluster*; o segundo termo, $\text{Dis}(q)$ é calculado segundo a equação 3.35 e que indica a separação entre *clusters* e α é o fator de peso e que se considera como sendo $\text{Dis}q_{max}$ onde q_{max} é número máximo de *clusters* que vêm como entrada.

$$\text{Scat}(q) = \frac{\frac{1}{q}\sum_{k=1}^{q} ||\sigma^{(k)}||}{||\sigma||} \tag{3.34}$$

Em que: σ é o vetor com as variâncias de cada objeto dos dados e $\sigma^{(k)}$ é o vetor com as variâncias de cada *cluster* C_k.

$$\text{Dis}(q) = \frac{D_{max}}{D_{min}} \sum_{k=1}^{q} \left(\sum_{z=1}^{q} ||c_k - c_z \right)^{-1} \tag{3.35}$$

Em que: $D_{max} = max(||c_k - c_z||)\ \forall k,z \in \{1,2,...,q\}$ é a distância máxima entre centros de *clusters* e $D_{min} = min(||c_k - c_z||)\ \forall k,z \in \{1,2,...,q\}$ é a distância mínima entre centros de *clusters*.

3.4.10 Índice SDbw

Índice que se baseia no critérios de compactação de um *cluster* e na separação entre *clusters*. O número de *clusters q* que minimiza o índice é considerado como o número ótimo de *clusters*. O índice SDbw é definido por:

$$\text{SDbw}(q) = \text{Scat}(q) + \text{Densidade.bw}(q) \tag{3.36}$$

Em que: $\text{Scat}(q)$ é igual ao definido na equação 3.34 na descrição do índice SD e $\text{Densidade.bw}(q)$ é definido na equação 3.37.

$$\text{Densidade.bw}(q) = \frac{1}{q(q-1)} \sum_{i=1}^{q} \left(\sum_{j=1, i \neq j}^{q} \frac{\text{densidade}(u_{ij})}{max(\text{densidade}(c_i), \text{densidade}(c_j))} \right) \tag{3.37}$$

Em que: u_{ij} é o ponto médio do segmento de reta entre os centróides c_i e c_j e densidade u_{ij} é dada pela equação 3.38.

$$\text{densidade}(u_{ij}) = \sum_{l=1}^{n_{ij}} f(x_l, u_{ij}) \tag{3.38}$$

Onde:

n_{ij} é o número de objetos de que pertencem aos *clusters* C_i e C_j;

$f(x_l, u_{ij})$ toma o valor 0 se $d(x, u_{ij}) > \frac{1}{q} \left(\sum_{k=1}^{q} ||\sigma^{(k)}|| \right)^{1/2}$ ou 1 no caso contrário.

3.4.11 Índice C

Índice definido por Hubert e Levin em que o número de *clusters q* que minimiza o índice é considerado como o número ótimo de *clusters*. Este índice é definido por:

$$\text{Cindex} = \frac{S_w - S_{min}}{S_{max} - S_{min}} \ , \ S_{min} \neq S_{max} \ , \ \text{Cindex} \in [0,1] \tag{3.39}$$

Em que: S_{min} é a soma das menores distâncias N_w entre todos os pares de pontos dos dados e S_{max} é a soma das maiores distâncias N_w entre todos os pares de pontos dos dados.

3.4.12 Índice *Silhouette*

Índice introduzido por Rousseeuw em que o número de *clusters q* que maximiza o índice é considerado como o número ótimo de *clusters*. Este índice é definido por:

$$\text{Silhouette} = \frac{\sum_{i=1}^{n} S(i)}{n}, \text{ Silhouette} \in [-1,1] \tag{3.40}$$

Em que:

$S(i) = \frac{b(i)-a(i)}{max\{a(i);b(i)\}}$,

$a(i) = \frac{\sum_{j\in\{C_r\ i\}} d_{ij}}{n_r-1}$, é a média das dissimilaridades entre o objeto i e os restantes objetos de C_r,

$b(i) = min_{s\neq r}\{d_{iC_s}\}$,

$d_{iC_s} = \frac{\sum_{j\in C_s} d_{ij}}{n_s}$, é a média das dissimilaridades entre o objeto i e os restantes objetos de C_s.

3.4.13 Índice de *Ball*

Índice proposto por Ball e Hall em que o número de *clusters q* que maximizar a diferença entre níveis é considerado como o número ótimo de *clusters*. Este índice baseia-se na distância média dos objetos de um dado *cluster* até ao centróide desse mesmo *cluster* e é definido por:

$$\text{Ball} = \frac{W_q}{q} \tag{3.41}$$

3.4.14 Índice *Ptbiserial*

Índice examinado por Milligan e Kraemer em que o número de *clusters q* que maximiza o índice é considerado como o número ótimo de *clusters*. Este índice é definido por:

$$\text{Ptbiserial} = \frac{[(S_b/N_b) - (S_w/N_w)][N_w N_b 7 N_t^2]^{1/2}}{s_d} \tag{3.42}$$

Em que s_d é o desvio-padrão de todas as distâncias.

3.4.15 Índice Gap

A estatística de Gap foi proposta por Tibshirani e é calculada segundo a equação 3.43:

$$\text{Gap}(q) = \frac{1}{B}\sum_{b=1}^{B} logW_{qb} - log(W_q) \tag{3.43}$$

Em que B é o número de dados de referência criados a partir do uso da prescrição uniforme descrita por Tibshirani e W_{qb} é a matriz de de dispersões internas definida como no índice de Hartigan.

O número ótimo de *clusters* é escolhido para o menor q que verifique a seguinte condição:

$$\text{Gap}(q) \geq \text{Gap}(q+1) - s_{q+1} \quad , \quad (q = 1, ..., n-2) \tag{3.44}$$

Onde:

$s_q = sd_q\sqrt{1+1/B}$,

sd_q é o desvio-padrão de $\{logW_{bq}\}$, $b = 1, ..., B$: $sd_q = \sqrt{\frac{1}{B}\sum_{b=1}^{B}(logW_{qb} - \bar{l})^2}$,

$\bar{l} = \frac{1}{B}\sum_{b=1}^{B} logW_{bq}$.

Capítulo 4

Metodologia e Resultados

Neste capítulo apresenta-se a metodologia aplicada neste trabalho para determinar o par distância, algoritmo, ou seja, o modo de fazer o *clustering*, que apresenta melhores resultados para o problema específico do mapeamento de fases.

Uma parte deste trabalho foi desenvolvido e implementado em *C* e outra parte em *R*, e sempre que possível recorreu-se ao uso de pacotes desenvolvidos e implementados no *software R*. As rotinas implementadas no *software R* encontram-se no anexo A. As rotinas usadas para retirar alguns resultados em *C* foram cedidas pela equipa do INESC TEC integrada no projeto *EUniversal*.

4.1 Metodologia

Com base no número de distâncias e de algoritmos de *clustering* em consideração neste trabalho, a abordagem proposta consiste em proceder a sucessivas etapas de comparações e análises, sendo que em cada uma delas eliminam-se distâncias ou pares de distância, algoritmos.

O primeiro passo nesta metodologia é a seleção das distâncias. Considerando o objectivo deste trabalho torna-se claro que as distâncias têm de ser capazes de distinguir as três fases. Como tal, o critério para a eliminação das distâncias neste passo é a fraca distinção entre objetos que se sabem que pertencem a fases diferentes, isto é, trios de objetos que fazem parte de um contador inteligente trifásico.

Selecionadas as distâncias, no segundo passo determinam-se os diversos índices que permitem seleccionar o número óptimo de *clusters* associado a cada par de *clustering*. Como, regra geral, e uma vez que os índices conduzem a resultados distintos, foi adoptado o critério da maioria para decidir qual o número óptimo de *clusters* a considerar. Para a implementação deste passo foi usada a função *NbClust*, que faz parte de um pacote com o mesmo nome, e que permite determinar todos os índices usados neste trabalho. Nesta fase eliminam-se quer os pares que não verificam um número ótimo de *clusters* maior ou igual que três, já que se querem identificar pelo menos as três fases da rede de baixa tensão, quer os pares que apresentam um número de *clusters* ótimo superior a 3 x n, em que n é o número de saídas do transformador. O número ótimo de *clusters*

definido neste passo deve ser considerado em todos os algoritmos, mas com particular destaque na utilização do algoritmo *k-means*, já que, como foi referido no capítulo 3, este é um algoritmo que necessita que o utilizador defina o número de *clusters* em que quer dividir os dados.

Tendo identificado o número ótimo de *clusters* para cada método e após a eliminação dos que não verificam os limites inferiores e superiores que estes devem verificar, passa-se então ao passo seguinte que é a realização do *clustering* com os ótimos definidos. Neste passo, os diferentes *clusterings* são realizados a partir das funções do *software RStudio*: *hclust* para o *clustering* hierárquico, *kml* para o algoritmo *k-means*. Neste passo, também foi usado um código em *C* fornecido pela equipa do INESC TEC integrada no projeto *EUniversal*, que executa o algoritmo *k-medoids* para algumas distâncias. Os resultados obtidos para os diferentes algoritmos considerando as diferentes distâncias podem, numa primeira fase, ser alvo de uma comparação com a informação das fases de cada objeto que foi fornecida em conjunto com os dados. E numa segunda fase, podem-se obter informações importantes sobre os *clusterings* obtidos com recurso à função *cluster.stats* do *RStudio* que permite avaliar o desempenho de cada par algoritmo, distância usado no *clustering*.

4.2 Dados utilizados

Os dados utilizados na obtenção de resultados tendo em conta a metodologia apresentada em 4.1 foram também fornecidos pela equipa do INESC TEC integrada no projeto *EUniversal*.

Sobre estes dados sabe-se que:

- São sintetizados a partir de uma rede de distribuição de baixa tensão típica;
- São constituídos por 43 medidas de tensão, em que 3 pertencem a um contador inteligente trifásico e as restantes 40 são medidas de contadores inteligentes monofásicos. As observações destas medidas foram feitas de 15 em 15 minutos durante um período de 42 dias;
- Tem-se informação relativamente às ligações das fases da rede de baixa tensão a que cada contador está ligado;
- Sabe-se também que o transformador desta rede tem um total de 6 saídas sendo que cada fase tem 2 saídas.

4.3 Resultados

Esta secção apresenta os resultados obtidos de acordo com a metodologia proposta em 4.1 e os dados apresentados em 4.2.

4.3.1 Seleção com base nas distâncias

Numa abordagem inicial ao problema da identificação de fases, foram selecionadas as distâncias usadas na criação da matriz de dissimilaridades. As distâncias usadas foram: a Euclidiana, ACF, PACF e DTW, computadas pelas funções: *diss.EUCL, diss.ACF, diss.PACF* e *diss.DTWARP*, respetivamente. As funções usadas nesta fase pertencem ao pacote *TSclust* do *RStudio*. Também foi considerada a distância de *Wasserstein*, no entanto, a matriz das dissimilaridades que resultam desta distância foi obtida por recurso a uma rotina já implementada em *R* e que me foi cedida para este trabalho. Para além de se usarem os dados originais, estes também foram padronizados de duas maneiras diferentes dando origem a mais matrizes de dissimilaridades. Os dados originais encontram-se organizados numa matriz em que as colunas representam os objetos dos dados e as linhas representam as observações feitas ao longo do tempo desses mesmos objetos. As transformações feitas aos dados antes da implementação de algumas das distâncias acima referidas fora a sua normalização, standarização e diferença de gradientes. O processo de normalização dos dados é feito a partir da função *scale* do *RStudio* que subtrai a média das observações de um objeto aos valores desse mesmo objeto e divide esse resultado pelo desvio padrão das observações do objeto. A standarização da matriz dos dados corresponde à mudança de escala dos seus valores de modo a que o valor mínimo de cada coluna da matriz é 0 e o valor máximo de cada coluna é 1. Quando se aplica a diferença de gradientes aos dados, então essa matriz passa a representar a diferença entre sucessivas observações das séries temporais dos dados. No caso dos dados de entrada para o cálculo da distância de *Wasserstein*, eles correspondem a um rearranjo dos dados originais. Isto é, originalmente temos um vector (série temporal) para cada objeto e passamos a ter uma matriz p x m em que p é o número de dias e m o número de observações recolhidas por dia. Deste modo os dados passaram de uma matriz n x (pm) para uma matriz (np) x m.

Tendo em conta as transformações que se podem aplicar aos dados bem como as distâncias, foram criadas dez matrizes de dissimilalidades resultantes de: dados originais e distância Euclidiana; dados standarizados e distância Euclidiana; dados normalizados e distância Euclidiana; gradientes dos dados e distância Eclidiana; dados originais e distância ACF; dados originais e distância PACF; dados originais e distância DTW; dados normalizados e distância DTW; dados originais e distância de *Wasserstein*; dados normalizados e distância de *Wasserstein*.

As matrizes de dissimilaridades criadas devem conseguir evidenciar diferenças entre objetos que pertençam a fases distintas da rede de baixa tensão. Sabe-se à partida que no caso dos contadores trifásicos a medida de cada fase pertencerá a *clusters* distintos. Como tal, o método utilizado para perceber se a distinção de fases é percetível é verificar se nos objetos que se sabe que pertencem a contadores inteligentes trifásicos têm dissimilaridades entre si. Esta verificação foi feita visualmente a partir do gráfico da primeira linha da matriz de dissimilaridades que representa a distância entre os objetos dos dados com a referência, que neste caso é a primeira variável dos dados. No entanto, com a representação gráfica de toda a primeira linha da matriz de dissimilaridades, a identificação dos objetos que pertencem ao contador inteligente trifásico

torna-se complicada. Assim sendo, nas figuras de 4.1 a 4.10 encontram-se representados os gráficos referentes às distâncias dos objectos que pertencem ao contador inteligente trifásico.

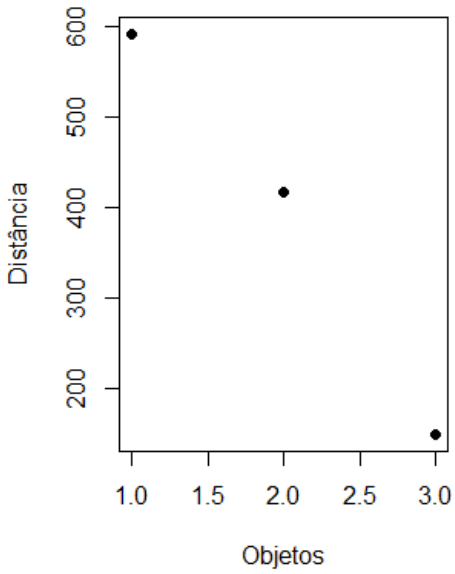

Figura 4.1: Distância Euclidiana, com os dados originais, entre os objetos do contador inteligente trifásico e o primeiro objeto.

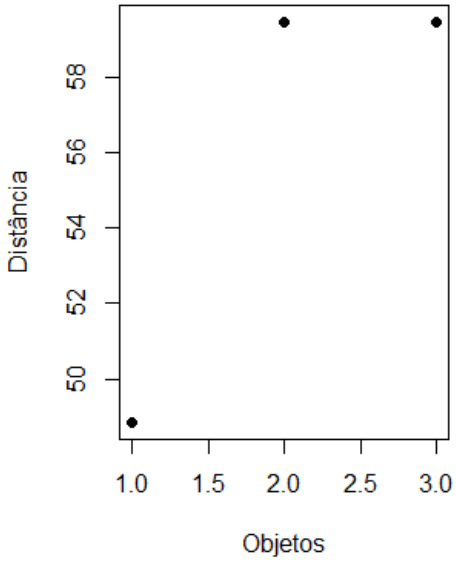

Figura 4.2: Distância Euclidiana, com os dados normalizados, entre os objetos do contador inteligente trifásico e o primeiro objeto.

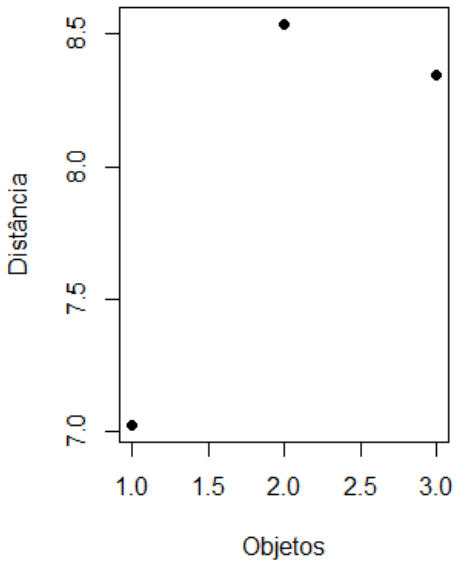

Figura 4.3: Distância Euclidiana, com os dados standarizados, entre os objetos do contador inteligente trifásico e o primeiro objeto.

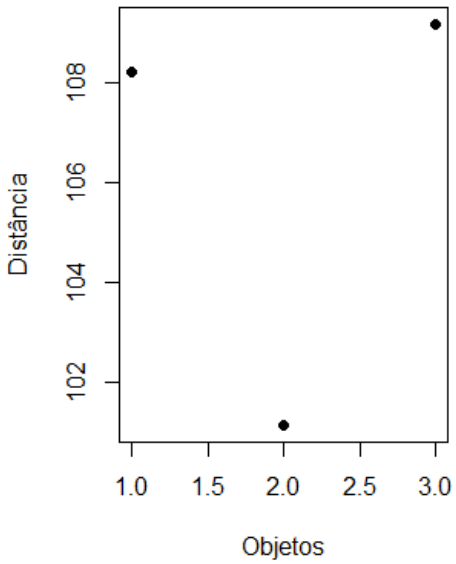

Figura 4.4: Distância Euclidiana, com a diferença de gradientes, entre os objetos do contador inteligente trifásico e o primeiro objeto.

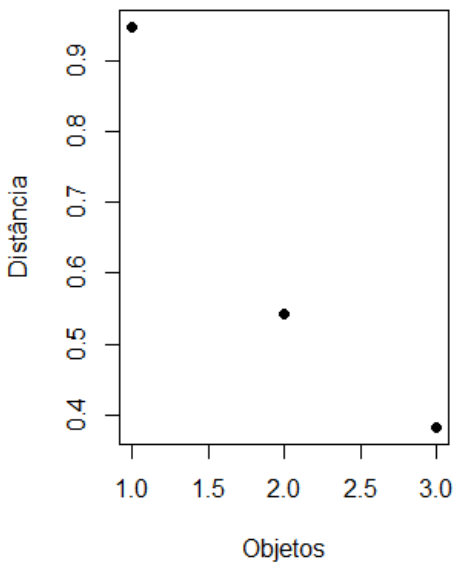

Figura 4.5: Distância ACF, com os dados originais, entre os objetos do contador inteligente trifásico e o primeiro objeto.

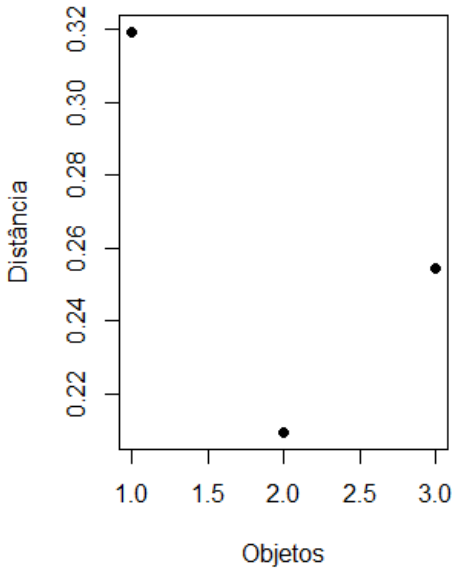

Figura 4.6: Distância PACF, com os dados originais, entre os objetos do contador inteligente trifásico e o primeiro objeto.

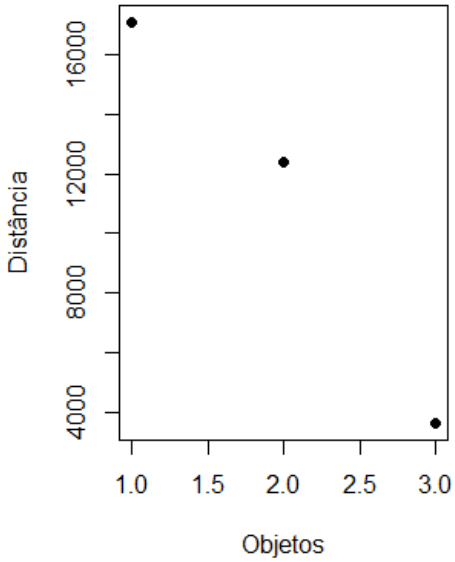

Figura 4.7: Distância DTW, com os dados originais, entre os objetos do contador inteligente trifásico e o primeiro objeto.

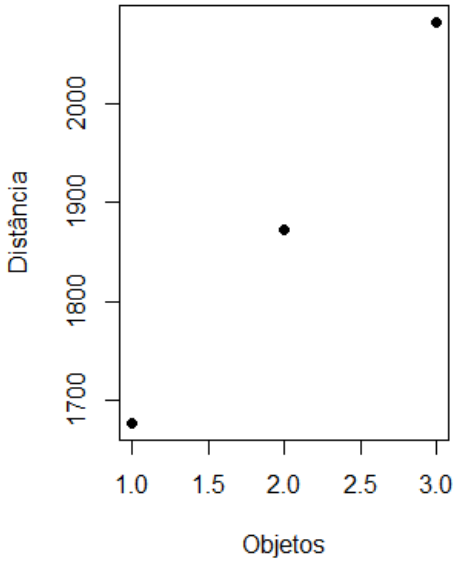

Figura 4.8: Distância DTW, com os dados normalizados, entre os objetos do contador inteligente trifásico e o primeiro objeto.

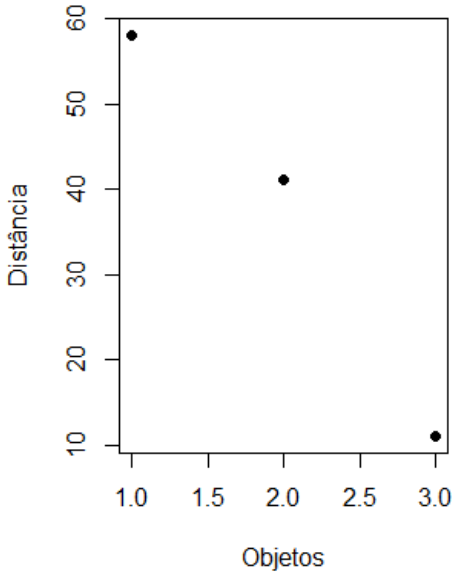

Figura 4.9: Distância de *Wasserstein*, com os dados originais, entre os objetos do contador inteligente trifásico e o primeiro objeto.

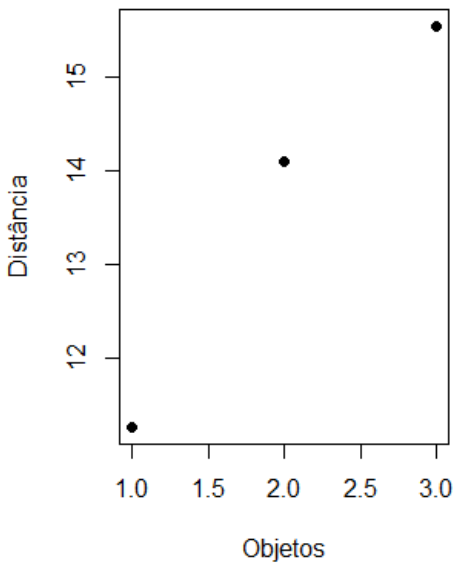

Figura 4.10: Distância de *Wasserstein*, com os dados normalizados, entre os objetos do contador inteligente trifásico e o primeiro objeto.

Como se pode observar pelos traçados do gráficos anteriores, a grande maioria das distâncias aplicadas aos dados, transformados ou não, permitem que se identifiquem diferenças entre objetos de fases diferentes. A única exceção ocorre para o traçado da figura 4.2, que corresponde à aplicação da distância Euclidiana aos dados normalizados. Neste caso, a distância usada, para os dados usados, não foi capaz de distinguir de forma clara o segundo e terceiro objeto do contador inteligente trifásico. As distâncias em que os dados foram transformados revelam uma capacidade inferior para distinguir as três fases quando comparadas com a que usa os dados originais. Consequentemente, nos próximos passos não se procederá à transformação dos dados.

4.3.2 Determinação do número ótimo de *clusters*

Depois de selecionar as matrizes de dissimilaridades que distinguem objetos que se sabem ser de fases diferentes, é necessário determinar o número ótimo de *clusters* que se deve impor ao aplicar os diferentes algoritmos de *clustering* com as diversas matrizes de dissimilaridades.

O uso da função *NbClust* permite não só calcular os diferentes índices descritos em 3.4 como, também, introduzir para além dos dados originais a matriz das distâncias. Isto é necessário porque algumas das distâncias usadas neste trabalho não fazem parte do leque de opções já implementadas no *software*. Os algoritmos de *clustering* usados foram: o *clustering* hierárquico com o método de agregação de *Ward.D2* que é de ligação média e o algoritmo *K-means* que um método de partição de dados. Para cada um dos dois algoritmos foram calculados os números ótimos de *clusters*, entre os valores 2 e 15, para as matrizes de dissimilaridades selecionadas em **??**. Os resultados obtidos encontram-se expostos nas tabelas 4.1 e 4.2, para o *clustering* hierárquico e para o *K-means*, respetivamente.

Tabela 4.1: Número ótimo de *clusters* para o *clustering* hierárquico.

	Número ótimo de *clusters* por índice															
Distâncias	kl	ch	hartigan	mcclain	gamma	gplus	tau	dunn	sdindex	sdbw	cindex	silhouette	ball	ptbiserial	gap	Melhor Nº de *clusters*
Euclidiana	3	10	3	2	5	5	3	5	5	15	6	6	3	3	3	3
ACF	8	8	8	2	15	15	2	2	5	15	15	2	3	2	2	2
PACF	9	4	4	15	15	15	5	3	4	15	15	5	3	5	2	15
DTW	9	9	5	2	5	5	4	5	5	15	3	5	3	2	3	5
Wasserstein	3	10	3	2	5	5	3	15	5	15	4	5	3	3	3	3

Dada a natureza deste problema em específico, que pretende identificar pelo menos três *clusters* distintos que correspondem às três fases da rede de baixa tensão, podem-se excluir os métodos de *clustering* em que o número ótimo de *clusters* é inferior a três. Dado que os postos de transformação têm várias saídas por fase, é importante que também se consideram os pares de *clustering* que tenham números ótimos de *clusters* superiores a três, já que, tendo mais do que três *clusters* pode estar a ser feita, não só a divisão dos objetos pelas três fases, mas também pelas diversas saídas do transformador. Sendo esse o caso, não só se está a obter informação sobre o

Tabela 4.2: Número ótimo de *clusters* para o *K-means*.

	Número ótimo de *clusters* por índice															
Distâncias	kl	ch	hartigan	mcclain	gamma	gplus	tau	dunn	sdindex	sdbw	cindex	silhouette	ball	ptbiserial	gap	Melhor Nº de *clusters*
Euclidiana	8	5	5	2	5	5	3	5	5	13	8	5	3	3	3	5
ACF	8	5	5	2	13	15	3	5	5	13	15	8	3	3	3	5 e 3
PACF	8	5	5	2	15	15	2	2	5	13	15	8	3	8	3	8, 5, 2 e 15
DTW	8	5	5	2	5	5	3	5	5	13	9	5	3	2	3	5
Wasserstein	8	5	5	2	5	5	3	5	5	13	8	5	3	3	3	5

mapeamento de fases, mas também se fica a saber informação relativamente à topologia da rede. Contudo, sabe-se que no caso dos dados que foram utilizados, apenas existem duas saídas por fase no transformador, pelo que, também se podem excluir os métodos com número ótimo de *clusters* superior a seis.

Resumindo, a determinação do número ótimo de *clusters* permite que sejam excluídas algumas combinações de distâncias e algoritmos de *clustering*. No *clustering* hierárquico é excluída a distância ACF por não ser capaz de identificar o mínimo de 3 *clusters*. Já que, para os dados usados temos um posto de transformação com duas saídas por fase, então é excluído, também, o método: *clustering* hierárquico com a distância PACF. Finalmente, com a algoritmo *K-means* e a distância PACF os resultados do número ótimo de *clusters* foram inconclusivos já que existe empate entre os números ótimos 8, 5, 2 e 15, pelo que este par também foi excluído.

4.3.3 *Clustering*

Com os métodos de *clustering* que verificaram as condições impostas em **??** e **??** e os respetivos números ótimos de *clusters* é possível efetuar o *clustering* e analisar os resultados de cada combinação de distância/algoritmo. Nesta fase do trabalho, também se incluem os resultados de *clustering* do algoritmo *K-medoids*, realizado em *C*, com as distâncias DTW, Euclidiana e de *Minkowski* com $g = 3$. Considerou-se a distância de *Minkowski* com $g = 3$ no sentido de perceber se o aumento do pârametro g influenciava de um modo percetível os resultados finais. Importa também referir que os métodos de *clustering* que foram corridos no *RStudio* não usam quaiquer restrições associadas à natureza do problema, enquanto que o *K-medoids* implementado em *C* apresenta restrições que obrigam a que os objetos trifásicos fiquem em *clusters* distintos.

O ***clustering* hierárquico** foi realizado através da função *hclust* do pacote *stats* do *RStudio*. O algoritmo selecionado na função foi o *ward.D2* de modo a corresponder ao algoritmo que se considerou em **??**. Os resultados para as diferentes matrizes de dissimilaridades resultantes do uso das diferentes distâncias são apresentados em dendogramas. Conforme as características do *clustering* hierárquico, já referidas no capítulo 3, o algoritmo *ward.D2* é um algoritmo de agregação pelo que parte de um número de *clusters* igual ao número de objetos e vai fazendo a sua agregação até apenas existir um *cluster* com todos os elementos. Nos dendogramas é possível

verificar essa agregação à medida que o dendograma cresce em altura. Quando dois *clusters* apresentam poucas diferenças, a altura do dendograma até que sejam agregados é pequena. Se, pelo contrário, dois *clusters* apresentarem grandes diferenças, a altura do dendograma até que esses *clusters* sejam agregados é maior. Os dendogramas obtidos para as diferentes distâncias utilizadas encontram-se traçados nas figuras 4.11 a 4.13.

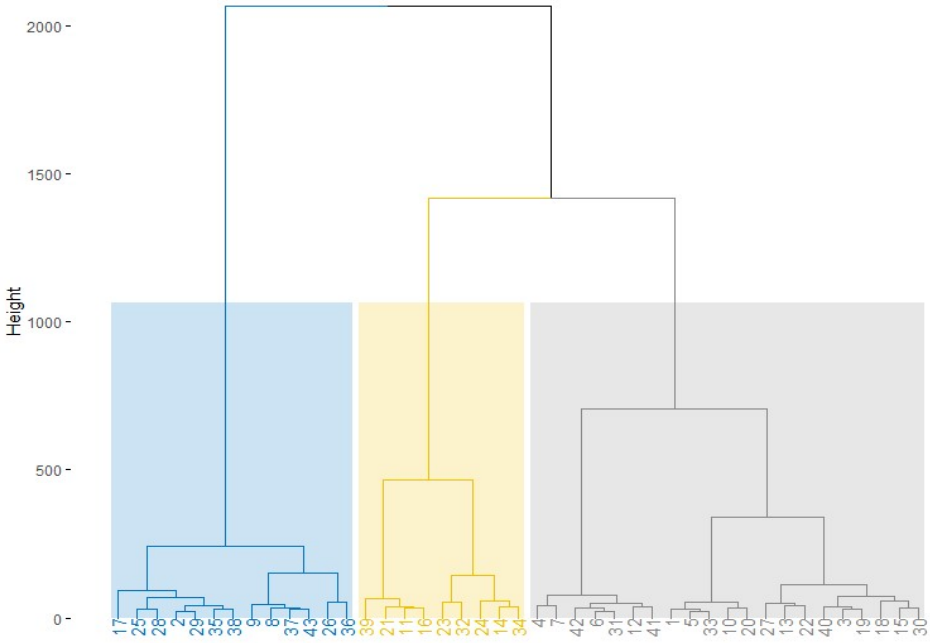

Figura 4.11: *Clustering* hierárquico com distância Euclidiana, dados originais e número ótimo de clusters = 3.

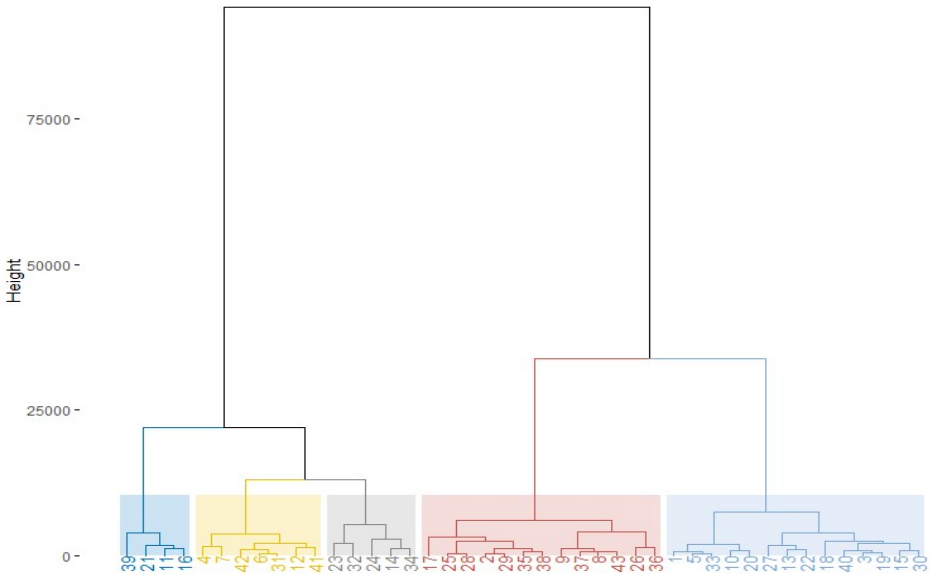

Figura 4.12: *Clustering* hierárquico com distância DTW, dados originais e número ótimo de clusters = 5.

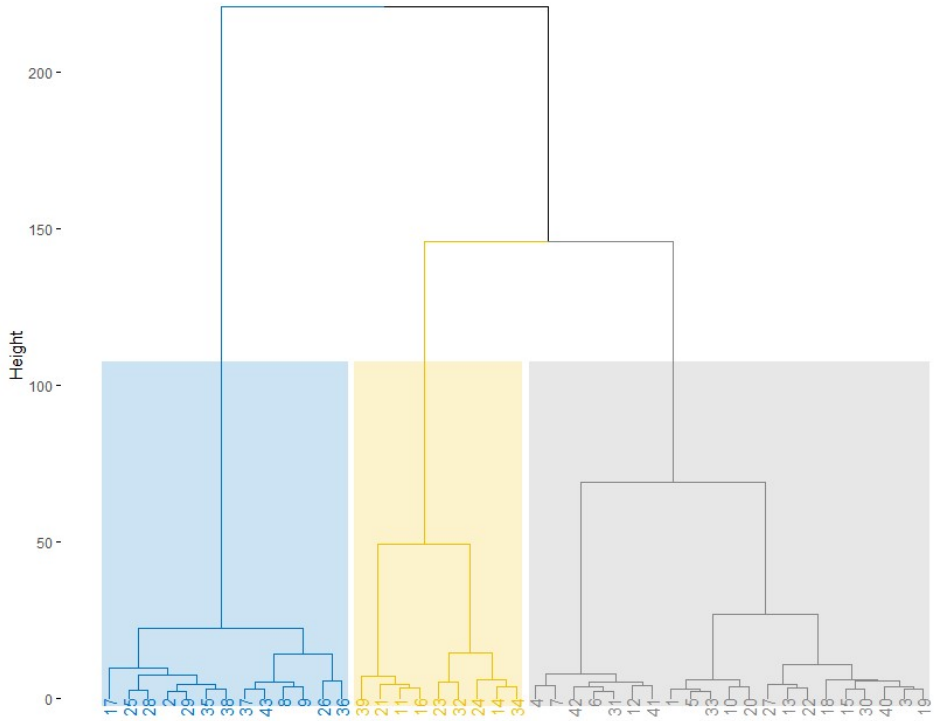

Figura 4.13: *Clustering* hierárquico com distância de *Wasserstein*, dados originais e número ótimo de clusters = 3.

Tendo realizado os *clusterings* hierárquicos para as diversas distâncias, a primeira verificação que é necessário fazer é se de facto, em cada um dos métodos, foi possível separar cada um dos objetos que fazem parte do contador inteligente trifásico em cada um dos três principais *clusters*.

No caso do *clustering* hierárquico com a distância Euclidiana e de *Wasserstein*, ambas com os dados originais, essa verificação é fácil de realizar visualmente já que o número ótimo de *clusters* nestes caso é três. Assim sendo, no caso em que se utiliza a distância Euclidiana com os dados originais - figura 4.11 - os objetos $16, 17$ e 18 (que correspondem aos objetos trifásicos) encontram-se em *clusters* distintos pelo que é verificada a condição de os objetos trifásicos estarem em *clusters* distintos. Também no caso em que se utiliza a distância de *Wasserstein* com os dados originais - figura 4.13 - os objetos trifásicos, representados por $16, 17$ e 18, também se encontram em três *clusters* distintos pelo que este método também verifica essa condição.

Com base na figura 4.12, onde se encontra representado clustering hierárquico com base na distância DTW, é possível verificar que este separa e distingue muito bem dois *clusters* (evidenciado pela altura do dendograma com apenas 2 *clusters*. Contudo, à medida que o número de *clusters* é maior que dois, a altura entre os nós dos diferentes ramos do dendograma é pequena, o que sugere que as diferenças entre eles não é evidente e daí a exclusão deste método.

Dada a natureza dos algoritmos de *clustering K-means* e *K-medoids*, não é possível fazer o mesmo tipo de análise feita para os resultados do *clustering* hierárquico, pelo que, nesta secção não é feita nenhuma análise aos métodos que usam estes algoritmos, sendo que os seus resultados apenas são apresentados nas secções seguintes.

4.3.4 Comparação com a informação original

Nesta secção apresentam-se os resultados relativos à identificação da fase correta, quando comparado com os dados da rede. Para o problema da identificação da fase os números de *clusters* foram limitados a 3.

Relativamente aos métodos que usam o algoritmo de *clustering* hierárquico, o *clustering* foi executado do mesmo modo que em **??**. Apenas os dendogramas foram todos cortados na região em que só exite divisão em três *clusters*.

No caso dos métodos que utilizam os algoritmos *K-means*, estes foram executados também no *RStudio* para as diversas distâncias selecionadas em **??** e **??**. A função usada para executar estes métodos foi a *kml* que pertence ao pacote com o mesmo nome. Esta função não permite que se utilizem diretamente as matrizes de dissimilaridades já calculadas, mas é possível especificar qual a medida de distância que se pretende utilizar. No entanto, a distância de *Wasserstein* usada não está incluída no leque de distâncias que a função *kml* permite utilizar, pelo que só se utilizou o algoritmo *K-means* para as distâncias Euclidiana com os dados originais e as distâncias ACF e DTW, ambas também com os dados originais.

Os métodos de *clustering* que utilizam o algoritmo *K-medoids* foram executados com recurso a um programa desenvolvido na linguagem *C* que foi fornecido.

Dado que apenas se pretende comparar o mapeamento de fases resultante dos métodos de *clustering* com o que é fornecido com os dados, nesta fase, os métodos que utlizam o algoritmo *K-means* e *K-medoids* apenas foram executados para o caso em que os dados são divididos em três *clusters*.

Tabela 4.3: Comparação entre os resultados do *clustering* com o mapeamento de fases fornecido.

Algoritmo	Distância	Certos	Errados
Clustering Hierárquico	Euclidiana	25	18
	Wasserstein	25	18
K-medoids C	DTW - 3 clusters	19	24
	Euclidiana - 3 clusters	26	17
	Minkowski - 3 clusters	27	16
K-means "kml"	Euclidiana - 3 clusters	25	18
	ACF - 3 clusters	25	18
	DTW - 3 clusters	25	18

Tabela 4.4: Identificação dos objetos bem identificados (assinalados a 1) e mal identificados (assinalados com -) para cada um dos métodos testados.

Algoritmo	Distância	Objetos																						
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Clustering Hierárquico	Euclidiana	-	1	-	-	1	-	-	1	-	-	1	-	1	-	-	1	1	1	1	-	1	1	1
	Wasserstein	-	1	-	-	1	-	-	1	-	-	1	-	1	-	-	1	1	1	1	-	1	1	1
	Totais Parciais	**0**	**2**	**0**	**0**	**2**	**0**	**0**	**2**	**0**	**0**	**2**	**0**	**2**	**0**	**0**	2	2	2	**2**	**0**	**2**	**2**	**2**
K-medoids	DTW	-	-	-	-	-	-	-	-	-	-	1	-	1	-	-	1	1	1	1	1	1	1	1
	Euclidiana	-	1	-	-	-	-	-	1	-	-	1	-	1	-	-	1	1	1	1	1	1	1	1
	Minkowski	1	1	-	1	-	1	1	-	-	1	1	1	1	-	-	1	1	1	1	-	1	1	1
	Totais Parciais	**1**	**2**	**0**	**1**	**0**	**1**	**1**	**1**	**0**	**1**	**3**	**1**	**3**	**0**	**0**	3	3	3	**3**	**2**	**3**	**3**	**3**
K-means	Euclidiana	-	1	-	-	1	-	-	1	-	-	1	-	1	-	-	1	1	1	1	-	1	1	1
	ACF	1	1	-	-	-	-	-	1	-	1	1	-	1	-	-	1	1	1	1	-	1	1	1
	DTW	-	1	-	-	1	-	-	1	-	-	1	-	1	-	-	1	1	1	1	-	1	1	1
	Totais Parciais	**1**	**3**	**0**	**0**	**2**	**0**	**0**	**3**	**0**	**1**	**3**	**0**	**3**	**0**	**0**	3	3	3	**3**	**0**	**3**	**3**	**3**
	Totais	**2**	**7**	**0**	**1**	**4**	**1**	**1**	**6**	**0**	**2**	**8**	**1**	**8**	**0**	**0**	8	8	8	**8**	**2**	**8**	**8**	**8**

Tabela 4.5: Identificação dos objetos bem identificados (assinalados a 1) e mal identificados (assinalados com -) para cada um dos métodos testados (continuação).

Algoritmo	Distância	Objetos																				
		24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	**Totais**
Clustering Hierárquico	Euclidiana	1	1	1	-	1	1	-	-	1	-	1	1	-	-	-	1	1	1	1	1	**25**
	Wasserstein	1	1	1	-	1	1	-	-	1	-	1	1	-	-	-	1	1	1	1	1	**25**
	Totais Parciais	**2**	**2**	**2**	**0**	**2**	**2**	**0**	**0**	**2**	**0**	**2**	**2**	**0**	**0**	**0**	**2**	**2**	**2**	**2**	**2**	-
K-medoids	DTW	1	-	-	-	-	-	-	1	1	1	1	-	1	1	-	1	1	-	-	-	**19**
	Euclidiana	1	1	1	-	1	1	-	-	1	1	1	1	-	-	-	1	1	1	1	1	**26**
	Minkowski	1	1	1	-	1	1	-	-	1	-	1	1	-	1	-	1	1	-	-	-	**27**
	Totais Parciais	**3**	**2**	**2**	**0**	**2**	**2**	**0**	**1**	**3**	**2**	**3**	**2**	**1**	**2**	**0**	**3**	**3**	**1**	**1**	**1**	-
K-means	Euclidiana	1	1	1	-	1	1	-	-	1	-	1	1	-	-	-	1	1	1	1	1	**25**
	ACF	1	1	1	-	1	1	-	1	1	-	1	1	-	-	-	1	1	-	-	1	**25**
	DTW	1	1	1	-	1	1	-	-	1	-	1	1	-	-	-	1	1	1	1	1	**25**
	Totais Parciais	**3**	**3**	**3**	**0**	**3**	**3**	**0**	**1**	**3**	**0**	**3**	**3**	**0**	**0**	**0**	**3**	**3**	**2**	**2**	**3**	-
	Totais	**8**	**7**	**7**	**0**	**7**	**7**	**0**	**2**	**8**	**2**	**8**	**7**	**1**	**2**	**0**	**8**	**8**	**5**	**5**	**6**	-

Os número de totais de objetos que cada um dos métodos de *clustering* consegue identificar corretamente ou erradamente comparativamente ao mapeamento de fases fornecido encontram-se expostos na tabela 4.3. O método que utiliza o algoritmo *K-medoids* com a distância de Minkowski ($g = 3$) é o que mais objetos identifica corretamente, com um total de 27 objetos corretos em 43 possíveis. Por outro lado, o método que menos objetos consegue identificar corretamente é o que utiliza o algoritmo de *clustering K-medoids* com a distância DTW. Dos 43 objetos dos dados, este método apenas consegue identificar corretamente 19. Comparativamente com os restantes métodos, que têm um número de objetos corretamente identificados que varia entre 25 e 27, este método tem um desempenho nitidamente pior.

Nas tabelas 4.4 e 4.5 é feita a análise de quais os objetos que foram bem identificados em cada método para tentar verificar se existem padrões que demonstrem se há objetos específicos

que sejam mais fáceis, ou mais díficeis, de identificar pelos diferentes métodos. Da observação das tabelas 4.4 e 4.5, verifica-se que, de facto, os objetos 3, 9, 14, 15, 27, 30 e 38 nunca são corretamente identificados por nenhum dos métodos. Por outro lado, também existem objetos que são corretamente identificados por todos os métodos de *clustering*. Estes resultados podem ter dois significados. Primeiro: os elementos que nunca foram identificados apresentam atributos semelhantes a *clusters* aos quais na realidade não pertencem e assim não são alocados no *cluster* correto. Segundo: os métodos até conseguem fazer a correta identificação deste objetos, mas no entanto a informação das fases dos objetos que foram fornecidas juntamente com os dados não está correta. Dado que os dados utilizados neste caso são dados reais, quando estes foram fornecidos, foi feito o aviso de que existe alguma possibilidade, ainda que baixa, de que a informação das fases dos objetos conter imprecisões.

4.3.5 Análise estatística dos *clusterings* obtidos

Tendo em conta os métodos de *clustering* avaliados ao longo deste capítulo 4 pretende-se, não só obter o mapeamento correto de fases, mas também o mapeamento da topologia da rede. Para avaliar o mapeamento da topologia, os métodos de *clustering* deveriam, não só formar três *clusters* como foi feito no capítulo **??**, mas sim formar o seu número ótimo de *clusters*, que foram calculados no capítulo **??**, desde que este número seja inferior ao número total de saídas no conjunto das três fases, 6 (o transformador tem 2 saídas por fase), e superior ao número de fases do sistema, 3. No entanto, relativamente aos dados usados para correr os métodos de *clustering*, tem-se apenas informação no contexto do mapeamento de fases. Assim sendo, optou-se por se fazer uma análise das estatísticas dos resultados obtidos para cada método quando considerado o seu número ótimo de *clusters* e, quando esse número ótimo for diferente de três, realizar uma comparação com os resultados obtidos para os mesmos métodos mas quando apenas se considera a divisão dos dados em três *clusters*. A análise das estatísticas dos *clusterings* tem como objetivo encontrar ou detetar pârametros que permitam afirmar se o resultado é ou não válido no contexto do problema em análise. Já a comparação com as estatísticas dos resultados em que se considera a divisão dos dados em três *clusters* tem como objetivo verificar quais as fases em que se conseguiram identificar as diferentes saídas.

Para calcular as estatísticas associadas aos *clusterings* realizados pelos diferentes métodos foi utilizada a função *cluster.stats* do pacote *fpc* do *RStudio*. Para a função executar os cálculos pretendidos foram introduzidos na função, a matrizes das dissimilaridades usadas em cada um dos métodos e os vetores com os resultados do *clustering* que se quer avaliar. Os parâmetros ou estatísticas calculados pela função *cluster.stats* que se consideraram relevantes para a avaliação e comparação que se pretende fazer foram: o número de objetos que foi alocado a cada *cluster*, o diâmetro de cada *cluster* e a separação mínima entre os mesmos. Os resultados obtidos encontram-se nas tabelas 4.6, 4.7 e 4.8.

Analisando os resultados expostos na tabela 4.6 é possível perceber que, em todos os métodos, existe uma clara separação entre os *clusters* formados. Dado que o que distingue os diferentes métodos nesta tabela são as distâncias (o algoritmo é sempre o mesmo), é evidente que a separação

Tabela 4.6: Estatísticas para os métodos que utilizam o algoritmo de *clustering* hierárquico.

		Clusters		
Distância	Parâmetros dos clusters	1	2	3
Euclidiana 3 clusters	Nº de objetos	13	9	21
	Diâmetro	207,3	315,7	318,5
	Separação mínima	190,3	176,5	176,5
Wasserstein 3 clusters	Nº de objetos	13	9	21
	Diâmetro	22,0	33,2	31,0
	Separação mínima	18,8	14,6	14,6

entre os *clusters* deve ser feita tendo em conta o diâmetro dos *clusters* formados para esse mesmo método, ao contrário de apenas comparar diretamente o valor da separação entre todos os métodos. Tendo em conta esse aspeto, o método que apresenta a menor separação entre *clusters* é o que emprega a distância de *Wasserstein*. Essa separação mínima relativamente ao diâmetro do *cluster* ocorre no caso do *cluster* número dois. O diâmetro desse *cluster* é de $33,2$ e a separação mínima é de $14,6$, o que significa que esta representa cerca de 45% do diâmetro, ou seja, mesmo no pior caso, a separação entre *clusters* nunca é inferior a 45% do diâmetro do *cluster* e daí a conclusão de que, para todos os métodos que utilizam o algoritmo de *clustering* hierárquico, existe uma clara separação entre os *clusters* criados.

Realizando uma análise semelhante, mas agora para a tabela 4.7, é possível verificar que também neste caso existe uma separação ainda satisfatória entre *clusters*. A separação mínima é de cerca de 30% do diâmetro do *cluster* número 1, para o método que usa a distância ACF.

Para os métodos utilizados com o algoritmo *K-means* (ainda tabela 4.7), verifica-se que foram mais os casos que apresentam uma divisão mais fina dos dados, o que pode significar que foram capazes de realizar o mapeamento da topologia, para além do mapeamento de fases. Contudo, se considerarmos que representam a topologia podemos verificar que nem todas as topologias obtidas foram iguais. Os métodos que utilizam as distâncias Euclidiana e DTW, ambas com os dados originais, obtiveram uma topologia que divide o *cluster* que contém 21 objetos em três novos *clusters* com 9, 7 e 5 objetos respetivamente.

Relativamente aos resultados apresentados na tabela 4.8, os métodos que utilizam as distâncias Euclidiana e de *Minkowski*, ambas com os dados originais, têm uma separação mínima a rondar os 20% em relação ao diâmetro do *cluster*. Já a distância de DTW com os dados originais, tem uma separação mínima de 10% do diâmetro do *cluster*. Estes resultados levam a crer que não existe uma separação clara entre os diferentes *clusters* o que indica uma falta de robustez dos métodos quando comparados com os métodos associados a outros algoritmos de *clustering*. Se tivermos em conta os resultados do capítulo **??**, essa falta de robustez torna-se evidente para o método que utiliza a distância DTW em conjunto com o algoritmo *K-medoids*, já que foi o método que teve o pior resultado no mapeamento de fases quando comparado com as informação que veio com os dados relativamente à fase de cada objeto.

Tabela 4.7: Estatísticas para os métodos que utilizam o algoritmo de *K-means*.

		Clusters				
Distância	Parâmetros dos clusters	1	2	3	4	5
Euclidiana 3 clusters	Nº de objetos	21	13	9	-	-
	Diâmetro	318,5	207,3	315,7	-	-
	Separação mínima	176,5	190,3	176,5	-	-
Euclidiana 5 clusters	Nº de objetos	13	9	9	7	5
	Diâmetro	80,93	68,32	68,91	49,12	43,35
	Separação mínima	66,40	66,40	77,34	83,98	72,91
ACF 3 clusters	Nº de objetos	25	9	9	-	-
	Diâmetro	0,705	0,361	0,435	-	-
	Separação mínima	0,212	0,212	0,609	-	-
DTW 3 clusters	Nº de objetos	21	13	9	-	-
	Diâmetro	7303	5679	8660	-	-
	Separação mínima	4633	4633	4676	-	-
DTW 5 clusters	Nº de objetos	13	9	9	7	5
	Diâmetro	5679	3518	5614	6887	1781
	Separação mínima	4633	2885	2570	2570	2885

Tabela 4.8: Estatísticas para os métodos que utilizam o algoritmo de *K-medoids*.

		Clusters		
Distância	Parâmetros dos clusters	1	2	3
DTW 3 clusters	Nº de objetos	21	14	8
	Diâmetro	13000	18868	22404
	Separação mínima	2247	2885	2247
Euclidiana 3 clusters	Nº de objetos	14	16	13
	Diâmetro	631,4	276,5	207,3
	Separação mínima	120,4	120,4	190,3
Minkowski 3 clusters	Nº de objetos	13	14	16
	Diâmetro	96,3	110,8	129,6
	Separação mínima	22,1	22,1	52,6

Capítulo 5

Conclusões e Trabalho Futuro

Neste capítulo apresentam-se alguns comentários e conclusões sobre o trabalho desenvolvido e finaliza-se com algumas sugestões para trabalho futuro.

5.1 Conclusões

Analisando os resultados obtidos no subcapítulo 4.2 é possível tirar uma série de conclusões sobre a metodologia usada para a determinação do mapeamento de fases e da topologia da rede em estudo.

1. As abordagens para o mapeamento de fase tem como principal vantagem evitar investimentos na instalação de equipamento e trabalhos no terreno. No âmbito deste trabalho foi identificada um conjunto de metodologias baseadas em dados dos contadores inteligentes que apresentam resultados promissores.

2. Foram identificados um conjunto de metodologias de clustering e parâmetros candidatos à resolução do problema do mapeamento de fases.

3. Foi realizada uma análise comparativa das diferentes abordagens identificadas que conduziram aos seguintes resultados:

   3.1. Da análise da separação entre as três fases do contador inteligente trifásico para as diferentes distâncias consideradas, é possível concluir que nenhuma das transformações aplicadas aos dados foram eficazes. Para os dados utilizados, a distinção entre as medidas trifásicas foi menos clara para os casos em que foram aplicadas transformações do que nos casos em que se usaram apenas os dados originais para se calcularem as distâncias;

   3.2. A determinação do número ótimo de *clusters* permite perceber quais os métodos de *clustering* que, quando realizam o seu *clustering* ótimo e sem restrições, conseguem identificar um número de *clusters* que faça sentido no contexto do problema do mapeamento de fases e da topologia. Ou seja, quando o número ótimo de *clusters* é inferior às 3 fases da rede ou superior ao produto do número de saídas por fase do transformador com o número

de fases, então esse método não está a produzir o seu *clustering* ideal de acordo com o que se pretende pelo que não deve ser considerado.

3.3. Realizando o *clustering* com o número de *clusters* $k = 3$ para todos os métodos selecionados e comparando com a informação sobre as fases de cada contador, verifica-se que todos (com exceção do método *K-medoids* com distância DTW) apresentam resultados muito semelhantes. Não só o número de identificações certas de fases foi próximo, como também a identificação correta foi feita maioritariamente para os mesmos contadores. Apesar de o número total de identificações corretas (entre 25 e 27), em cada método, andar longe do total de contadores (43), a consistência dos resultados nos contadores que foram corretamente identificados e também nos que foram mal identificados levanta dúvidas quanto à veracidade da informação fornecida. Como os dados usados são reais esta questão foi colocada aos responsáveis por esta informação, e estes não colocaram de parte a hipótese de os dados terem incorreções. Comparando estes resultados com os obtidos pela equipa do INESC TEC que integra o projeto *EUniversal*, verifica-se que estes são diferentes o que leva a crer que, consoante os dados utilizados, o par distância, algoritmo com melhor desempenho pode variar;

3.4. Por fim, a análise estatística feita aos métodos selecionados nesta fase permite duas conclusões. A primeira é que a separação entre os *clusters* formados é boa, excepto no método que obteve os piores resultados quando comparado com a informação das fases (método *K-medoids* com distância DTW), o que indica que a separação está, de facto, associada a melhores resultados de *clustering*. A segunda conclusão é que, quando é usado o número ótimo de *clusters*, para tentar realizar o mapeamento da topologia, ambos os métodos chegam à mesma topologia. No entanto, no sentido físico do problema, o mapeamento da topologia obtido é impossível, já que, a rede estudada tem apenas 2 saídas por fase no transformador, e o resultado dos métodos apresenta uma topologia em que uma das fases tem 3 saídas.

Tendo em conta todas estas conclusões, podemos dizer que os objetivos foram atingidos. Os resultados obtidos permitem concluir que o método que conduz ao melhor mapeamento de fases é o que conjuga a distância de *Minkowski* com $g = 3$ e o algoritmo de *clustering K-medoids* com restrições específicas do problema. Para o problema do mapeamento da topologia os resultados são ainda inconclusivos. No entanto, coloca-se a hipótese de o *clustering* estar a identificar proximidade física entre contadores da mesma fase e não saídas. A verificação desta hipótese ficará para trabalho futuro.

5.2 Trabalho Futuro

Dado que a utilização de métodos de *clustering* para a realização do mapeamento de fases e da topologia das redes de baixa tensão é um tema com potencial para mais exploração, ainda há

espaço para diversos trabalhos futuros. Aqui ficam algumas sugestões e que vêm na sequência da realização desta dissertação:

- Sugere-se que a metodologia criada seja utilizada com diferentes conjuntos de redes e dados por forma a permitir confirmar as conclusões relativamente ao mapeamento de fases e tirar mais conclusões relativamente à topologia da rede;
- No futuro também se podem explorar os algoritmos identificados para o problema da identificação das saídas do transformador a que os contadores inteligentes estão ligados e avaliar a informação fornecida relativamente à topologia da rede;
- Seria também interessante, no futuro, desenvolver uma análise final relativamente à confiança dos resultados obtidos para cada contador inteligente;
- Tendo em conta que, consoante os dados que se estão a utilizar, os pares distância, algoritmo com melhor desempenho podem variar, esta metodologia tem potencial para ser incorporada em módulos de *software* de fabricantes, abordando uma estratégia híbrida permitindo a utilização dos pares algoritmo-distância que produzem melhores resultados face aos dados. Deste modo, o método usado para o mapeamento de fases vai-se adaptando tendo em conta a rede e os respetivos dados em estudo;
- Na maior parte dos casos, existe sempre alguma informação relativamente à rede de baixa tensão (por exemplo, números de saídas do transformador, algum consumidor em específico que se saiba a que conhece a que fase da rede está ligado). Como tal, também seria interessante desenvolver métodos de *clustering* que tenham em conta esse conhecimento de modo a que se possam realizar mapeamentos de fases e da topologia da rede mais precisos e eficientes.

Anexo A

A.1 Rotinas implementadas em *RStudio* para a obtenção dos resultados

```
library(TSclust)
library(FactoMineR)
library(factoextra)
library(cluster)
library(pvclust)
library(NbClust)
library(kml)
library(fpc)
library(xlsx)
library(philentropy)

#### Ler e ajustar o dataset para o R ####
data <- read.csv("Historico_tese.csv", sep=" ",header = FALSE)
mallows43 <- read.csv("DMallowa2_43.csv", sep=" ",header = FALSE)
mallows43p <- read.csv("DMallowa2_43pe.csv", sep=" ",header = FALSE)

# Define os cabeçalhos como as três primeiras linhas de data
headers <- data[c(1,2,3),]

#Define novos cabeçalhos como characters e como lista
new_headers <- lapply(headers[3,],as.character)

#Remove os cabeçalhos de data
data <- data[-c(1,2,3),]

#Define as datas das medidas
dates <- data[,1]

#Remove a coluna das datas da variável data
data <- data[,-1]
mallows43 <- mallows43[,-1]
mallows43p <- mallows43p[,-1]

#Elimina as últimas linhas e colunas da distância de Mallows
mallows43 <- mallows43[,-44]
mallows43p <- mallows43p[,-44]

mallows43 <- mallows43[-44,]
mallows43p <- mallows43p[-44,]

#Converte os valores de data (que estão em factor) em characters e
#posteriormente para numeric (o mesmo para Mallows)
data <- data.frame(sapply(data, function(x) as.numeric(as.character(x))))
mallows43 <- data.frame(sapply(mallows43, function(x) as.numeric(as.character(x))))
mallows43p <- data.frame(sapply(mallows43p, function(x) as.numeric(as.character(x))))

#Numera cada linha de data por ordem crescente
rownames(data) <- 1:nrow(data)

#Passa para use_data os valores de data da coluna de VMagValue
use_data <- data[,grepl("VMagValue",new_headers[-1])]

#Define a função colMax que cria uma lista com os máximos de cada coluna
#ignorando as células com NA
colMax <- function(data) sapply(data, max, na.rm = TRUE)

```

```r
#Guarda em max_values os máximos de cada coluna
max_values <- colMax(use_data)

#Elimina as colunas que estão a 0
use_data <- use_data[max_values!=0]

#Elimina a primeira e segunda medida trifásica
use_data <- use_data[,-(2:4)]
use_data <- use_data[,-(6:8)]

#Define a função range01 que calcula a distância de x ao ponto mínimo de 0 a 1
range01 <- function(x){(x-min(x))/(max(x)-min(x))}

#Passa os valores de use_data para uma distância standart de 0 a 1
use_data_stand <- data.frame(lapply(use_data, range01))

#Passa os valores de use_data para uma distância normalizada
use_data_norm <- scale(use_data)

#Cria uma matriz com a correlação dos valores de use_data
corr <- cor(use_data)


#### calculo de distancias entre series temporais ####

#Cálculo da distância Euclidiana entre séries temporais com base em use_data
dist <- matrix(0,nrow = ncol(use_data),ncol = ncol(use_data))
for (i in 1:ncol(use_data)){
  for (j in i:ncol(use_data)){
    dist[i,j] <- diss.EUCL(use_data[,i],use_data[,j])
  }
}


#Cálculo da distância Euclidiana entre séries temporais com base em
#use_data_stand
dist_stand <- matrix(0,nrow = ncol(use_data),ncol = ncol(use_data))
for (i in 1:ncol(use_data)){
  for (j in i:ncol(use_data)){
    dist_stand[i,j] <- diss.EUCL(use_data_stand[,i],use_data_stand[,j])
  }
}


#Cálculo da distância Euclidiana entre séries temporais com base em
#use_data_norm
dist_norm <- matrix(0,nrow = ncol(use_data),ncol = ncol(use_data))
for (i in 1:ncol(use_data)){
  for (j in i:ncol(use_data)){
    dist_norm[i,j] <- diss.EUCL(use_data_norm[,i],use_data_norm[,j])
  }
}

#Cálculo da distância das dissimiliridades simples entre séries temporais
#com base em use_data
dist_acf <- matrix(0,nrow = ncol(use_data),ncol = ncol(use_data))
```

```
for (i in 1:ncol(use_data)){
  for (j in i:ncol(use_data)){
    dist_acf[i,j] <- diss.ACF(use_data[,i],use_data[,j])
  }
}


#Cálculo da distância das dissimiliridades parciais entre séries temporais
#com base em use_data
dist_pacf <- matrix(0,nrow = ncol(use_data),ncol = ncol(use_data))
for (i in 1:ncol(use_data)){
  for (j in i:ncol(use_data)){
    dist_pacf[i,j] <- diss.PACF(use_data[,i],use_data[,j])
  }
}


#Cálculo da distância de tempo warping dinâmica entre séries temporais
#com base em use_data
dist_dtw <- matrix(0,nrow = ncol(use_data),ncol = ncol(use_data))
for (i in 1:ncol(use_data)){
  for (j in i:ncol(use_data)){
    dist_dtw[i,j] <- diss.DTWARP(use_data[,i],use_data[,j])
  }
}

#Cálculo da distância de tempo warping dinâmica entre séries temporais
#com base em use_data_norm
dist_norm_dtw <- matrix(0,nrow = ncol(use_data),ncol = ncol(use_data))
for (i in 1:ncol(use_data)){
  for (j in i:ncol(use_data)){
    dist_norm_dtw[i,j] <- diss.DTWARP(use_data_norm[,i],use_data_norm[,j])
  }
}
plot(dist_dtw[1,c(16,17,18)])


#### calculo de distancias para a variaÃ§Ã£o/ gradiente dos dados ####

#Calcula a diferença entre duas linhas consecutivas de use_data
use_data_dif <- use_data[-1,]-use_data[-nrow(use_data),]

#Calcula a diferença entre duas linhas consecutivas de use_data_stand
use_data_stand_dif <- use_data_stand[-1,]-use_data_stand[-nrow(use_data_stand),]

#Calcula a diferença entre duas linhas consecutivas de use_data_norm
use_data_norm_dif <- use_data_norm[-1,]-use_data_norm[-nrow(use_data_norm),]

#Cálculo da distância Euclidiana entre gradientes com base em use_data_dif
dist_dif <- matrix(0,nrow = ncol(use_data_dif),ncol = ncol(use_data_dif))
for (i in 1:ncol(use_data_dif)){
  for (j in i:ncol(use_data_dif)){
    dist_dif[i,j] <- diss.EUCL(use_data_dif[,i],use_data_dif[,j])
  }
}

```

```
#Traça a distância entre os pontos trifásicos
plot(dist[1,c(16,17,18)], xlab="Objetos", ylab="Distância")
plot(dist_norm33[1,c(16,17,18)], xlab="Objetos", ylab="Distância")
plot(dist_stand33[1,c(16,17,18)], xlab="Objetos", ylab="Distância")
plot(dist_dif33[1,c(16,17,18)], xlab="Objetos", ylab="Distância")
plot(dist_acf33[1,c(16,17,18)], xlab="Objetos", ylab="Distância")
plot(dist_pacf33[1,c(16,17,18)], xlab="Objetos", ylab="Distância")
plot(dist_dtw33[1,c(16,17,18)], xlab="Objetos", ylab="Distância")
plot(dist_norm_dtw33[1,c(16,17,18)], xlab="Objetos", ylab="Distância")
plot(mallows33[1,c(16,17,18)], xlab="Objetos", ylab="Distância")
plot(mallows_norm33[1,c(16,17,18)], xlab="Objetos", ylab="Distância")


#####Rotina para o cálculo do número ótimo de clusters######
lista.methods = c("kl", "ch", "hartigan","mcclain", "gamma", "gplus",
                  "tau", "dunn", "sdindex", "sdbw", "cindex", "silhouette",
                  "ball","ptbiserial", "gap","frey")
lista.distance = c("metodo","euclidean", "maximum", "manhattan", "canberra")

tabla = as.data.frame(matrix(ncol = 3, nrow = length(lista.methods)))


for(i in 1:length(lista.methods)){
  set.seed(123)
  res = NbClust(data = t(use_data), diss = dd, distance = NULL,
                min.nc = 2, max.nc = 15,
                method = "kmeans", index =lista.methods[i])

  tabla[i,1] = lista.methods[i]
  tabla[i,2] = res$Best.nc[1]
  tabla[i,3] = res$Best.nc[2]

}

tabla


#####Clustering Hierárquico#####
dd <- as.matrix(mallows43p)
colnames(dd) <- c(1:43)
dd <- as.dist(t(dd), diag = FALSE, upper = FALSE)

#Função de cálculo do clustering
set.seed(123)
hc <- hclust(dd, method = "ward.D2")

fviz_dend(hc , k=4,
          palette = "jco",               # Color palette see ?ggpubr::ggpar
          rect = TRUE, rect_fill = TRUE, # Add rectangle around groups
          rect_border = "jco",           # Rectangle color
          labels_track_height = 2     # Augment the room for labels
)


#####K-Means#####

```

```
#Cluster com função kml

#Definição dos pârametros para correr kml
data_v1 <- cld(traj=t(use_data), idAll=c(1:ncol(use_data)),
               time=c(1:nrow(use_data)), maxNA=10)
options <- parKml(saveFreq=100, maxIt=50, imputationMethod="copyMean",
                  distanceName = "dtw",power=2,
                  distance = function(x,y)diss.DTWARP(x,y),
                  centerMethod=function(x)mean(x,na.rm=TRUE),
                  startingCond="randomAll",
                  nbCriterion=100, scale=FALSE)

time1 <- Sys.time()
time1
set.seed(123)
kml(object=data_v1, nbClusters=3, nbRedrawing=5, toPlot="none", parAlgo=options)
Sys.time()-time1

#resultado do clustering
partitions_dist <- getClusters(data_v1, 3, 1, asInteger = TRUE)

####rotina para o cálculo das estatísticas####

#Lê os resultados dos clustering guardados em excel
kml_dist_noc5 <- read.xlsx(file = "kmeans.xlsx", sheetIndex = 1 ,
                           sheetName = "dist", rowIndex = NULL, startRow = 2,
                           endRow = 44, colIndex = 2, header = FALSE)
kml_dist_noc3 <- read.xlsx(file = "kmeans.xlsx", sheetIndex = 1 ,
                           sheetName = "dist-noc3", rowIndex = NULL, startRow = 2,
                           endRow = 44, colIndex = 2, header = FALSE)

kml_dist_acf_noc3 <- read.xlsx(file = "kmeans.xlsx", sheetIndex = 4 ,
                               sheetName = "dist_acf-noc3", rowIndex = NULL, startRow = 2,
                               endRow = 44, colIndex = 2, header = FALSE)
kml_dist_dtw_noc5 <- read.xlsx(file = "kmeans.xlsx", sheetIndex = 5 ,
                               sheetName = "dist_dtw-noc5", rowIndex = NULL, startRow = 2,
                               endRow = 44, colIndex = 2, header = FALSE)
kml_dist_dtw_noc3 <- read.xlsx(file = "kmeans.xlsx", sheetIndex = 6 ,
                               sheetName = "dist_dtw-noc3", rowIndex = NULL, startRow = 2,
                               endRow = 44, colIndex = 2, header = FALSE)


ch_dist_noc3 <-  read.xlsx(file = "ch.xlsx", sheetIndex = 1 ,
                           sheetName = "dist", rowIndex = NULL, startRow = 2,
                  endRow = 44, colIndex = 2, header = FALSE)

ch_dist_wass_noc3 <-  read.xlsx(file = "ch.xlsx", sheetIndex = 3 ,
                                sheetName = "dist_wass", rowIndex = NULL, startRow = 2,
                               endRow = 44, colIndex = 2, header = FALSE)


c_dist_dtw_noc3 <-  read.xlsx(file = "Resultados Clustering C.xlsx",
                              sheetIndex = 1 ,sheetName = "C - DTW",
                              rowIndex = 8, colIndex = (2:44), header = FALSE)
c_dist_noc3 <-  read.xlsx(file = "Resultados Clustering C.xlsx",
                          sheetIndex = 2 ,sheetName = "C - Euclidiana",
```

```
                          rowIndex = 8, colIndex = (2:44), header = FALSE)
c_dist_mink_noc3 <-  read.xlsx(file = "Resultados Clustering C.xlsx",
                               sheetIndex = 3 ,sheetName = "C - Minkowski",
                               rowIndex = 8, colIndex = (2:44), header = FALSE)

kml_dist_noc5 <- data.frame(sapply(kml_dist_noc5, function(x) as.numeric(as.character(x))))
kml_dist_noc3 <- data.frame(sapply(kml_dist_noc3, function(x) as.numeric(as.character(x))))
kml_dist_acf_noc3 <- data.frame(sapply(kml_dist_acf_noc3, function(x) as.numeric(as.character(x))))
kml_dist_dtw_noc5 <- data.frame(sapply(kml_dist_dtw_noc5, function(x) as.numeric(as.character(x))))
kml_dist_dtw_noc3 <- data.frame(sapply(kml_dist_dtw_noc3, function(x) as.numeric(as.character(x))))

ch_dist_noc3 <-  data.frame(sapply(ch_dist_noc3, function(x) as.numeric(as.character(x))))
ch_dist_wass_noc3 <-  data.frame(sapply(ch_dist_wass_noc3, function(x) as.numeric(as.character(x))))

c_dist_dtw_noc3 <-  data.frame(sapply(c_dist_dtw_noc3, function(x) as.numeric(as.character(x))))
c_dist_noc3 <-  data.frame(sapply(c_dist_noc3, function(x) as.numeric(as.character(x))))
c_dist_mink_noc3 <-  data.frame(sapply(c_dist_mink_noc3, function(x) as.numeric(as.character(x))))

kml_dist_noc5 <- t(kml_dist_noc5)
kml_dist_noc3 <- t(kml_dist_noc3)
kml_dist_acf_noc3 <- t(kml_dist_acf_noc3)
kml_dist_dtw_noc5 <- t(kml_dist_dtw_noc5)
kml_dist_dtw_noc3 <- t(kml_dist_dtw_noc3)

ch_dist_noc3 <-  t(ch_dist_noc3)
ch_dist_wass_noc3 <-  t(ch_dist_wass_noc3)

c_dist_dtw_noc3 <-  t(c_dist_dtw_noc3)
c_dist_noc3 <-  t(c_dist_noc3)
c_dist_mink_noc3 <-  t(c_dist_mink_noc3)

#Cálculo da ditância de Minkowski (Foi calculada em C mas é necessária para as estatisticas)
dist_mink <- distance(t(use_data), method = "minkowski", p = 3)

#distância que se quer usar na função
ddd <- as.matrix(dist_mink)
ddd <- as.dist(t(ddd), diag = FALSE, upper = FALSE)

                                    #Resultado do clustering
estatisticas <- cluster.stats(d = ddd,          ...            , noisecluster = FALSE)

estatisticas

```

Bibliografia

[1] Snezana Cundeva, Math Bollen e Daphne Schwanz. "Hosting capacity of the grid for wind generators set by voltage magnitude and distortion levels". Em: *IET Conference Publications* 2016.CP711 (2016). DOI: 10.1049/cp.2016.1062.

[2] Ahmad Ghasemi, Seyed Saeidollah Mortazavi e Elaheh Mashhour. "Hourly demand response and battery energy storage for imbalance reduction of smart distribution company embedded with electric vehicles and wind farms". Em: *Renewable Energy* 85 (2016), pp. 124–136. ISSN: 18790682. DOI: 10.1016/j.renene.2015.06.018. URL: http://dx.doi.org/10.1016/j.renene.2015.06.018.

[3] The European Electricity Grid Initiative (EEGI). *Roadmap 2010-18 and Detailed Implementation Plan 2010-12*. 2010. URL: https://ec.europa.eu/research/participants/portal/doc/call/fp7/fp7-energy-2012-1-1stage/31280-eegi%7B%5C_%7Dimplementation%7B%5C_%7Dplan%7B%5C_%7Dmay%7B%5C_%7D2010%7B%5C_%7Den.pdf%20[Julho2020]].

[4] CEN/CENELEC/ETSI Joint Working Group on Standards for Smart Grids. "Smart Grid Reference Architecture". Em: November (2012), pp. 1–107. URL: ftp://ftp.cen.eu/EN/EuropeanStandardization/HotTopics/SmartGrids/Reference_Architecture_final.pdf.

[5] C. Gouveia et al. "Development and implementation of Portuguese smart distribution system". Em: *Electric Power Systems Research* 120 (2015), pp. 150–162.

[6] Hélder Costa et al. "Voltage control demonstration for lv networks with controllable DER-the sustainable project approach". Em: (2016).

[7] José Costa et al. "LV network control architecture: H2020 INTEGRID case study". Em: *CIRED Workshop 2018* 0455 (2018), pp. 1–4. URL: http://www.cired.net/publications/workshop2018/pdfs/Submission%200455%20-%20Paper%20(ID-21473).pdf.

[8] Konstantinos Kotsalos et al. "(ADMS4LV) – Improved observability of LV grids based on advanced analytics". Em: *25th International Conference on Electricity Distribution* 1797.June (2019), pp. 1–5. DOI: 20.500.12455/579. URL: https://www.cired-repository.org/handle/20.500.12455/579.

[9] Filipe Campos et al. “ADMS4LV–advanced distribution management system for active management of LV grids”. Em: *CIRED-Open Access Proceedings Journal* 2017.1 (2017), pp. 920–923.

[10] H. Pezeshki e P. J. Wolfs. “Consumer phase identification in a three phase unbalanced LV distribution network”. Em: *IEEE PES Innovative Smart Grid Technologies Conference Europe* July 2011 (2012), pp. 1–7. DOI: 10.1109/ISGTEurope.2012.6465632.

[11] V. Arya et al. “Phase identification in smart grids”. Em: *2011 IEEE International Conference on Smart Grid Communications, SmartGridComm 2011* (2011), pp. 25–30. DOI: 10.1109/SmartGridComm.2011.6102329.

[12] ZIV. *4SLV – Advanced low voltage supervision solution*. 2020. URL: https://www.zivautomation.com/substation-automation/advanced-low-voltage-supervision-cabinets-for-lv-grid-monitoring-increase-service-quality/lv-supervisor-node/ (acedido em 04/07/2020).

[13] Enedia.IO. *Low Voltage Network Monitoring and Optimization*. 2020. URL: http://eneida.io/deepgrid-2/ (acedido em 04/07/2020).

[14] Frederic Olivier et al. “Phase identification of smart meters by clustering voltage measurements”. Em: *20th Power Systems Computation Conference, PSCC 2018* (2018). DOI: 10.23919/PSCC.2018.8442853.

[15] Wenyu Wang et al. “Phase identification in electric power distribution systems by clustering of smart meter data”. Em: *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2016, pp. 259–265.

[16] Satya Jayadev Pappu et al. “Identifying topology of low voltage distribution networks based on smart meter data”. Em: *IEEE Transactions on Smart Grid* 9.5 (2017), pp. 5113–5122.

[17] Jeremy Donald Watson, John Welch e Neville R Watson. “Use of smart-meter data to determine distribution system topology”. Em: *The Journal of Engineering* 2016.5 (2016), pp. 94–101.

[18] Aravind Rajeswaran, Nirav P Bhatt, Ramkrishna Pasumarthy et al. “A novel approach for phase identification in smart grids using graph theory and principal component analysis”. Em: *2016 American Control Conference (ACC)*. IEEE. 2016, pp. 5026–5031.

[19] Frédéric Olivier, Damien Ernst e Raphaël Fonteneau. “Automatic phase identification of smart meter measurement data”. Em: *CIRED-Open Access Proceedings Journal* 2017.1 (2017), pp. 1579–1583.

[20] Roberto Brunelli e Tomaso Poggio. “Face recognition: Features versus templates”. Em: *IEEE transactions on pattern analysis and machine intelligence* 15.10 (1993), pp. 1042–1052.

[21] Richard O Duda, Peter E Hart e David G Stork. *Pattern classification and scene analysis*. Vol. 3. Wiley New York, 1973, p. 92.

[22] Rafael C Gonzales e Richard E Woods. *Digital image processing*. 2002.

[23] M S Beck. "Correlation in instruments: cross correlation flowmeters". Em: *Journal of Physics E: Scientific Instruments* 14.1 (1981), pp. 7–19.

[24] Rui Xu e Donald Wunsch. "Survey of clustering algorithms". Em: *IEEE Transactions on Neural Networks* 16.3 (2005), pp. 645–678. ISSN: 10459227. DOI: 10.1109/TNN.2005.845141.

[25] Lior Rokach. "A survey of clustering algorithms". Em: *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 269–298.

[26] Jiawei Han, Jian Pei e Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[27] Alexis Sardá-Espinosa. "Time-series clustering in R Using the dtwclust package". Em: *R Journal* 11.1 (2019), pp. 1–45. ISSN: 20734859. DOI: 10.32614/rj-2019-023.

[28] Pavel Senin. "Dynamic Time Warping Algorithm Review". Em: *Science* 2007.December (2008), pp. 1–23. ISSN: 1557170X. DOI: 10.1109/IEMBS.2007.4353810. URL: http://129.173.35.31/%7B~%7Dpf/Linguistique/Treillis/ReviewDTW.pdf.

[29] Chotirat Ann Ratanamahatana e Eamonn Keogh. "Everything you know about dynamic time warping is wrong". Em: *Third workshop on mining temporal and sequential data*. Vol. 32. Citeseer. 2004.

[30] Pedro Galeano e Daniel Pella Peña. "Multivariate Analysis in Vector Time Series". Em: *Resenhas do Instituto de Matemática e Estat{\'\i}stica da Universidade de São Paulo* 4.4 (2000), pp. 383–403.

[31] Pablo Montero e José A. Vilar. "TSclust: An R package for time series clustering". Em: *Journal of Statistical Software* 62.1 (2014), pp. 1–43. ISSN: 15487660. DOI: 10.18637/jss.v062.i01.

[32] Rosanna Verde e Antonio Irpino. "Comparing histogram data using a mahalanobis-wasserstein distance". Em: *COMPSTAT 2008 - Proceedings in Computational Statistics, 18th Symposium* (2008), pp. 77–89.

[33] Elizaveta Levina e Peter Bickel. "The earth mover's distance is the mallows distance: Some insights from statistics". Em: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vol. 2. IEEE. 2001, pp. 251–256.

[34] Vladimir Estivill-Castro e Jianhua Yang. "Fast and robust general purpose clustering algorithms". Em: *Pacific Rim International Conference on Artificial Intelligence*. Springer. 2000, pp. 208–218.

[35] Chris Fraley e Adrian E Raftery. "How many clusters? Which clustering method? Answers via model-based cluster analysis". Em: *The computer journal* 41.8 (1998), pp. 578–588.

[36] Peter H A Sneath, Robert R Sokal et al. *Numerical taxonomy. The principles and practice of numerical classification.* 1973.

[37] Sudipto Guha, Rajeev Rastogi e Kyuseok Shim. "CURE: an efficient clustering algorithm for large databases". Em: *ACM Sigmod record* 27.2 (1998), pp. 73–84.

[38] Benjamin King. "Step-wise clustering procedures". Em: *Journal of the American Statistical Association* 62.317 (1967), pp. 86–101.

[39] Joe H Ward Jr. "Hierarchical grouping to optimize an objective function". Em: *Journal of the American statistical association* 58.301 (1963), pp. 236–244.

[40] Fionn Murtagh. "A survey of recent advances in hierarchical clustering algorithms". Em: *The computer journal* 26.4 (1983), pp. 354–359.

[41] Pierre Hansen e Brigitte Jaumard. "Cluster analysis and mathematical programming". Em: *Mathematical programming* 79.1-3 (1997), pp. 191–215.

[42] James MacQueen et al. "Some methods for classification and analysis of multivariate observations". Em: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[43] Inderjit S Dhillon e Dharmendra S Modha. "Concept decompositions for large sparse text data using clustering". Em: *Machine learning* 42.1-2 (2001), pp. 143–175.

[44] Malika Charrad et al. "Nbclust: An R package for determining the relevant number of clusters in a data set". Em: *Journal of Statistical Software* 61.6 (2014), pp. 1–36. ISSN: 15487660. DOI: 10.18637/jss.v061.i06.