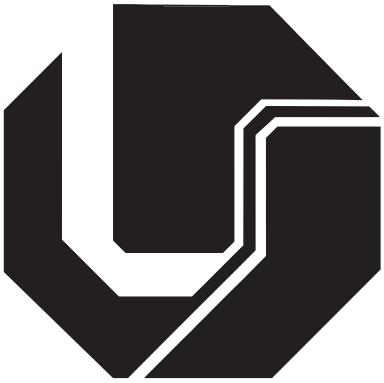
Universidade Federal de Uberlândia
Faculdade de Engenharia Elétrica
Pós-Graduação em Engenharia Elétrica



Alocação de Recursos baseada em *Clustering* com Aprendizado de Características e Orientação a QoS em Redes *LTE-Advanced*

Einar César Santos

Uberlândia - 2019

Universidade Federal de Uberlândia
Faculdade de Engenharia Elétrica
Pós-Graduação em Engenharia Elétrica

Einar César Santos

Alocação de Recursos baseada em *Clustering* com Aprendizado de Características e Orientação a QoS em Redes *LTE-Advanced*

Tese apresentada à Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia como parte dos requisitos exigidos para conclusão do curso de Doutorado em Engenharia Elétrica do programa de Pós-Graduação em Engenharia Elétrica e obtenção do título de Doutor em Ciências. Avaliada em 14/06/2019.

Membros da banca:

Prof. Dr. Paulo Roberto Guardieiro (Orientador) – UFU
Profª. Drª. Juliana Freitag Borin – UNICAMP
Prof. Dr. Márcio Andrey Teixeira – IFSP
Prof. Dr. Keiji Yamanaka – UFU
Prof. Dr. Éderson Rosa da Silva – UFU

Uberlândia
2019

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

S237 Santos, Einar César, 1981-
2019 Alocação de Recursos baseada em Clustering com Aprendizado de Características e Orientação a QoS em Redes LTE-Advanced [recurso eletrônico] / Einar César Santos. - 2019.

Orientador: Paulo Roberto Guardieiro.
Tese (Doutorado) - Universidade Federal de Uberlândia, Pós-graduação em Engenharia Elétrica.
Modo de acesso: Internet.
Disponível em: http://dx.doi.org/10.14393/ufu.te.2019.2060
Inclui bibliografia.
Inclui ilustrações.

1. Engenharia elétrica. I. Guardieiro, Paulo Roberto, 1952-, (Orient.). II. Universidade Federal de Uberlândia. Pós-graduação em Engenharia Elétrica. III. Título. CDU: 621.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:
Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074

A Deus.

Às minhas amadas, Heloiza e Olívia.

Agradecimentos

A Deus em primeiro lugar, sempre. Ele é o Princípio e o Fim, o Alfa e o Ômega. Agradeço a Ele por me conceder perdão, direção, esperança e a perseverança para seguir em frente sem desistir. Obrigado Senhor!

À minha amada esposa Heloiza. Agradeço pelo amor, pelas palavras de consolo e motivação, pela força transmitida, pelas orações e pelas lágrimas que derramamos juntos. Foram muitos momentos difíceis, e hoje podemos celebrar não apenas essa vitória mas também a chegada de nossa filha! Plantamos e colhemos. Obrigado por acreditar e me apoiar! Quero cumprir logo minha promessa! Te amo!

Ao meu orientador, professor Paulo Roberto Guardieiro. Agradeço pelo ensino, dedicação, paciência, compreensão e por motivar, direta e indiretamente, a conclusão deste trabalho. Aprendi muito contigo, apesar de nossas divergências. Muito obrigado!

Aos meus colegas da UFCat – CERCOMP: Paulo; Samuel; Guilherme; Ana Paula; e Luiz Fernando. O apoio de vocês foi fundamental! Agradeço pela compreensão em meus momentos de ausência, pela alegria, pelo ombro amigo e pelo companheirismo. Trabalhar com vocês é um privilégio para poucos! Que Deus recompense o bem que me fizeram e que nossa amizade se estreite ainda mais!

À minha mãe, Valquíria, e ao meu pai, Leonardo. Agradeço simplesmente por tê-los vivos e com saúde. É um prazer incomensurável concluir essa etapa da vida na presença de vocês! Desejo honrá-los com o título de doutor. Vocês fizeram isso acontecer! Muito obrigado!

A todos os parentes, amigos e colegas que, de alguma forma, acreditaram em mim. Mesmo sem mencionar individualmente (são muitos!), agradeço por cada gesto de apoio e suporte. Agradeço pelas orações, pelas palavras de motivação, pela torcida... Cheguei até o fim com a ajuda de todos vocês! Obrigado!

Aos técnicos administrativos e ao corpo docente da Faculdade de Engenharia Elétrica da UFU, pelo apoio técnico e acadêmico.

À UFG/UFCat, pela concessão de afastamento durantes os anos de 2015 e 2016.

"Porque dEle, e por meio dEle, e para Ele são todas as coisas. A Ele, pois a glória eternamente. Amém!" (Romanos 11:36)

Veni, vidi, Deus vicit.

Resumo

A crescente demanda por acesso a redes móveis sem fio tem elevado os custos operacionais e gerenciais assim como os níveis mínimos exigidos de Qualidade de Serviço, ou *Quality of Service* (QoS), para provisionamento às aplicações em redes *Long Term Evolution Advanced* (LTE-A). Apesar das especificações mais recentes do LTE-A oferecerem tecnologias capazes de reduzir custos, elas não são preparadas para lidar com o *Big Data*, caracterizado por um grande volume de dados constantemente produzidos em alta velocidade, variedade, veracidade e valor em sistemas de informação e comunicação. Neste sentido, o Aprendizado de Máquina, ou *Machine Learning*, é amplamente utilizado e recomendado para processar *Big Data*. A estratégia *Clustering-Based Resource Allocation* (CBRA), idealizada para alocação autônoma de recursos por meio do uso da "clusterização", ou *clustering* (uma tarefa realizada por Aprendizado de Máquina), categoriza usuários e aplicações a partir de padrões descobertos em dados do sistema de comunicação em que é empregada, fornecendo parâmetros mais adequados de QoS. Entretanto, para ser capaz de classificar adequadamente, o CBRA requer uma melhor definição das características informadas bem como parâmetros adaptados aos dados analisados durante a execução do *clustering*. Este trabalho propõe um mecanismo CBRA, especificamente desenvolvido para o LTE-A, dotado de um algoritmo para aprendizado autônomo de características e outro para desparametrização do *clustering*. A proposta é avaliada por meio de simulação computacional e comparada com outros mecanismos relacionados existentes na literatura. Os resultados apresentam melhor qualidade de classificação, em comparação com a quantidade de características analisadas, e um desempenho adequado do mecanismo para aplicações de vídeo em tempo real.

Palavras-chave: Alocação de Recursos, Aprendizado de Máquina, LTE-A, QoS, Simulação.

Abstract

The increasing demand for wireless mobile networks access has increased operational and management costs as well as the minimum levels of Quality of Service (QoS) required for applications provisioning in Long Term Evolution Advanced (LTE-A) networks. Although the latest LTE-A specifications offer cost-effective technologies, they are not prepared to deal with Big Data, characterized by a large volume of data constantly produced at high velocity, variety, veracity and value in information and communication systems. In this sense, Machine Learning is widely used and recommended to process Big Data. The Clustering-Based Resource Allocation (CBRA) strategy, designed for autonomous resource allocation by the use of clustering (a task performed by Machine Learning), categorizes users and applications from patterns discovered in communication system data in which it is employed, providing better QoS parameters. However, to be able to classify properly, the CBRA requires a better definition of features informed as well as parameters adapted to the analyzed data during the clustering execution. This work proposes a CBRA mechanism, specifically developed for LTE-A, provided with an algorithm for autonomous feature learning and another for clustering de-parameterization. The proposal is evaluated through computational simulation and compared with other related mechanisms existing in the literature. The results show better classification quality, in comparison with the number of analyzed features, and adequate performance of the mechanism for real-time video applications.

Key-words: LTE-A, Machine Learning, QoS, Resource Allocation, Simulation.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

16 QAM	*16 Point Quadrature Amplitude Modulation*
1G	Primeira Geração de Sistemas de Telefonia Móvel
2G	Segunda Geração de Sistemas de Telefonia Móvel
3G	Terceira Geração de Sistemas de Telefonia Móvel
3GPP	*3rd Generation Partnership Project*
3GPP2	*3rd Generation Partnership Project 2*
4G	Quarta Geração de Sistemas de Telefonia Móvel
5G	Quinta Geração de Sistemas de Telefonia Móvel
64QAM	*64 Point Quadrature Amplitude Modulation*
AANN	*Auto-Associative Neural Network*
AG	Algoritmo Genético
AIC	*Akaike Information Criterion*
AMC	*Adaptive Modulation and Coding*
AMPS	*Advanced Mobile Phone System*
AMR	*Adaptive Multi-Rate*
ANSI	*American National Standards Institute*
APN	*Access Point Name*
APN-AMBR	*Per APN Aggregate Maximum Bit Rate*
ARP	*Allocation and Retention Priority*
AS	*Access Stratum*
ATDSA	*Adaptive Time Domain Scheduling Algorithm*
BE	*Best Effort*
BIC	*Bayesian Information Criterion*
CAC	*Call Admission Control*
CBRA	*Clustering-Based Resource Allocation*
CBSA	*Clustering-Based Scheduling Algorithm*
CDSA	*Control/Data Plane Separation Architecture*
CEPT	*Committee of European Post and Telecommunications*
CoMP	*Coordinated Multi-Point*
CP	*Cyclic Prefix*
CQI	*Channel Quality Indicator*

CSD	*Circuit Switched Data*
D-AMPS	*Digital-Advanced Mobile Phone Service*
D2D	*Device-to-Device*
DBN	*Deep Belief Network*
DC-HSDPA	*Dual Carrier High-Speed Downlink Packet Access*
DL-SCH	*Downlink Shared Channel*
DNN	*Deep Neural Network*
DNS	*Domain Name System*
DRB	*Data Radio Bearer*
DRX	*Discontinuous Reception*
DSMIPv4	*Dual-Stack Mobile IPv4*
DSMIPv6	*Dual-Stack Mobile IPv6*
E-RAB	*Evolved Radio Access Bearer*
E-UTRAN	*Evolved Universal Terrestrial Radio Access Network*
EB	*Exabytes*
EDGE	*Enhanced Data Rates for GSM Evolution*
eNB	*Evolved Node Base*
eNodeB	*Evolved Node Base*
EPC	*Evolved Packet Core*
EPS	*Evolved Packet System*
ETSI	*European Telecommunications Standards Institute*
EVDO	*Evolution Data Optimized*
FCM	*Fuzzy C-means*
GBR	*Guaranteed Bit Rate*
GERAN	*GSM EDGE Radio Access Network*
GPRS	*General Packet Radio Service*
GSM	*Global System for Mobile Communications*
GSM	*Groupe Spéciale Mobile*
HARQ	*Hybrid Automatic Repeat Request*
HeNB	*Home eNodeB*
HSCSD	*High Speed Circuit Switched Data*
HSDPA	*High Speed Downlink Packet Access*
HSPA	*High Speed Packet Access*
HSPA+	*Evolved High Speed Packet Access*
HSS	*Home Subscriver Service*
HSUPA	*High Speed Uplink Packet Access*
IA	Inteligência Artificial
IEEE	*Institute of Eletric and Electronic Engineers*
IETF	*Internet Engineering Task Force*
IMSI	*International Mobile Subscriber Identification*
IMT-2000	*International Mobile Telecommunications-2000*
IMT-2020	*International Mobile Telecommunications 2020*

IMT-Advanced	*International Mobile Telecommunications-Advanced*
IMT-MC	*IMT Multi-Carrier*
IoT	*Internet of Things*
IP	*Internet Protocol*
IPv4	*Internet Protocol version 4*
IPv6	*Internet Protocol version 6*
IS	*Interim Standard*
ITU	*International Telecommunications Union*
ITU-R	*ITU Radiocommunication Sector*
k-NN	*k-Nearest Neighbors*
KDD	*Knowledge Discovery in Databases*
LTE	*Long Term Evolution*
LTE-A	*Long Term Evolution Advanced*
M2M	*Machine-to-Machine*
MAC	*Medium Access Control*
MBR	*Maximum Bit Rate*
MCPTT	*Mission Critical Push to Talk*
MCS	*Modulation and Codification Scheme*
MDP	*Markov Decision Process*
MIMO	*Multiple-Input and Multiple-Output*
MIPv4	*Mobile IPv4*
MIPv6	*Mobile IPv6*
NAS	*Non Access Stratum*
NGMN	*Next Generation Mobile Networks*
NMT	*Nordic Mobile Telephone*
NRT	*Non Real-Time*
OFDM	*Orthogonal Frequency Domain Multiplexing*
OFDMA	*Orthogonal Frequency Domain Multiple Access*
P-GW	*Packet Data Network Gateway*
PCA	*Principal Component Analysis*
PCRF	*Policy and Charging Rules Function*
PDCCH	*Physical Downlink Control Channel*
PDCP	*Packet Data Convergence Protocol*
PDM	Processo de Decisão de Markov
PDN *Gateway*	*Packet Data Network Gateway*
PDR	*Packet Drop Rate*
PDSCH	*Physical Downlink Shared Channel*
PDU	*Packet Data Unit*
PF	*Proportional Fair*
PLR	*Packet Loss Rate*
PMI	*Precoding Matrix Indicator*
PMIPv4	*Proxy Mobile IPv4*

PMIPv6	*Proxy Mobile IPv6*
PSTN	*Public Switched Telephone Network*
PTT	*Push to Talk*
PUCCH	*Physical Uplink Control Channel*
PUSCH	*Physical Uplink Shared Channel*
QCI	*QoS Class Identifier*
QoE	*Quality of Experience*
QoS	*Quality of Service*
QPSK	*Quadrature Phase-Shift Keying*
RB	*Resource Block*
ReLU	*Rectified Linear Unit*
RI	*Rank Indicator*
RLC	*Radio Link Control*
RR	*Round Robin*
RRC	*Radio Resource Control*
RRM	*Radio Resource Management*
RSRP	*Reference Signal Received Power*
RSRQ	*Reference Signal Received Quality*
RSSI	*Received Signal Strength Indicator*
RT	*Real-Time*
S-GW	*Serving Gateway*
S1	*Interface* física da eNB/HeNB com a EPC
S1-C	Canal de controle da S1
S1-MME	Canal de controle da S1
S1-U	Canal de dados da S1
SAE	*System Architecture Evolution*
SAP	*Service Access Points*
SB	*Scheduling Block*
SDF	*Service Data Flow*
SINR	*Signal-to-Interference plus Noise Ratio*
SLA	*Service Level Agreement*
SMS	*Short Message Service*
SOM	*Self-organizing Map*
SON	*Self-organizing Networks*
SRB	*Signaling Radio Bearer*
SVD	*Singular Value Decomposition*
SVM	*Support Vector Machine*
TACS	*Total Access Communication System*
TB	*Transport Block*
TBS	*Transport Block Size*
TCP	*Transmission Control Protocol*
TEID	*Tunneling End ID*

TFT	*Traffic Flow Template*
TOS	*Type of Service*
UE	*User Equipment*
UE-AMBR	*Per UE Aggregate Maximum Bit Rate*
UL-SCH	*Uplink Shared Channel*
UMB	*Ultra Mobile Broadband*
UMTS	*Universal Mobile Telecommunication System*
UTRAN	*Universal Terrestrial Radio Access Network*
V2X	*Vehicle to Everything*
WCDMA	*Wideband Code Division Multiple Access*
WDM	*Wavelength Division Multiplexing*
WiMAX	*Worldwide Interoperability for Microwave Access*
XB	*Xie-Beni*
XSOM	*X-means with* SOM

Sumário

Capítulo 1

Introdução

A utilização de tecnologias e serviços de acesso a redes móveis de banda larga sem fio tornou-se inegavelmente um hábito cultural na atualidade, especialmente nos grandes centros urbanos. Em decorrência deste novo hábito cultural, a crescente demanda por meios de comunicação cada vez mais ágeis e eficientes vem impulsionando o desenvolvimento das tecnologias de informação e comunicação.

Estudos recentes prevêem até 2021 um crescimento de algo em torno de 48,3 *exabytes* (EB) no volume total mensal de tráfego em sistemas de comunicação em todo o planeta [1]. Estima-se uma quantidade de 1,5 dispositivos móveis *per capita* até 2020, em meio a 3,4 dispositivos em geral conectados à *Internet*. De todo o tráfego de dados correspondente a dispositivos móveis, prevê-se que aplicações de vídeo corresponderão a 75% do total [2]. Nota-se, portanto, um enorme desafio no sentido de atender toda a demanda apresentada, principalmente quando considera-se o suporte à qualidade de serviço ou *Quality of Service* (QoS).

Como observado, a proliferação de dispositivos móveis e o crescimento praticamente exponencial da demanda em sistemas de comunicação vêm produzindo uma quantidade extraordinária de dados de usuários, de aplicações e de controle [3]. Este fenômeno caracteriza o *Big Data*, termo cunhado para referir-se ao volume excessivo de dados produzidos continuamente, em alta velocidade e com ampla variedade de informações por dispositivos e sistemas de informação e de comunicação [4].

O *Big Data* sobreveio de forma a transformar as próximas gerações de redes móveis, ou *Next Generation Mobile Networks* (NGMN)[1], de maneira considerável e permanente. Há uma grande expectativa em relação às tecnologias da NGMN e sua capacidade de proporcionar altas taxas de transferência de dados, baixa latência, custos gerenciais e operacionais reduzidos, maior confiabilidade e autonomia de serviço [5]. Entretanto, tendo em vista que o *Big Data* proporciona excessivo volume de dados e alta complexidade para processamento a *softwares* (algoritmos) tradicionais, é esperado que as abordagens rotineiramente utilizadas para gerenciamento e operacionalização das novas tecnologias de redes móveis sejam seriamente afetadas em termos de

[1] Atualmente a sigla também refere-se a tecnologias pertinentes à Quinta Geração de Sistemas de Telefonia Móvel (5G).

desempenho e controle.

Como consequência de maior volume e complexidade ocasionados pelo *Big Data*, os custos de implantação, gerenciamento e manutenção de redes móveis elevam-se naturalmente. Embora haja tecnologias recentes para automação e redução de intervenção humana, elas também são afetadas negativamente pela existência de *Big Data*. Não obstante, armazenar e dar um sentido ao *Big Data* tornou-se algo imprescindível [6]. Assim, processos de descoberta de conhecimento em bases de dados, ou *Knowledge Discovery in Databases* (KDD)[2], têm sido desenvolvidos para racionalizar o *Big Data* e mitigar seus possíveis efeitos negativos em redes móveis.

O Aprendizado de Máquina, ou *Machine Learning*, vem sendo recentemente considerado uma excelente ferramenta para auxílio ao KDD, apresentando inclusive avanços significativos para análise de *Big Data* em redes móveis com foco em otimização e controle [8]. O Aprendizado de Máquina trata basicamente em habilitar um algoritmo computacional (máquina) para execução autônoma de tarefas de alto grau de complexidade com desempenho próximo ou superior a de um ser humano, sem programação explícita. O uso de Aprendizado de Máquina em redes móveis tem proporcionado maior autonomia, melhor capacidade de gerenciamento, melhor desempenho, redução de custos operacionais e gerenciais, aumento da capacidade e da eficiência espectral, bem como ampliação da cobertura e dos níveis de QoS e *Quality of Experience* (QoE) dos usuários [9].

Além disso, uma ramificação recente do Aprendizado de Máquina, o Aprendizado Profundo, ou *Deep Learning*, vem ganhando notoriedade em decorrência de alguns casos de sucesso reportados por aplicações em sistemas de controle de tráfego em redes de computadores. A aplicação do Aprendizado Profundo vem causando uma grande ruptura na forma como os sistemas de comunicação são desenvolvidos atualmente, assim como no prospecto das próximas tecnologias relacionadas a redes móveis de banda larga sem fio [10].

Entre os diversos tipos de aplicação existentes de Aprendizado de Máquina para KDD e aperfeiçoamento de sistemas de comunicação, menciona-se a estratégia (ou mecanismo) de Alocação de Recursos baseada em *Clustering*, ou *Clustering-Based Resource Allocation* (CBRA). Trata-se de um mecanismo para descoberta de conhecimento e posterior tomada de decisões na alocação de recursos em sistemas de comunicação que recorre ao *clustering* para extrair e organizar padrões de bases de dados, caracterizadas ou não como *Big Data*. O *clustering* [11] é uma tarefa, normalmente executada por algum algoritmo de Aprendizado de Máquina, cuja finalidade é classificar ou agrupar elementos (também definidos como objetos) de acordo com suas características em comum. A aplicação somente do *clustering* em sistemas de comunicação também é útil para classificação de tráfego e caracterização de parâmetros para controle de QoS de acordo com os tipos de tráfego e serviços identificados [12–14]. Outrossim, seus benefícios podem ser explorados para diversos fins.

Apesar das especificações do *Long Term Evolution Advanced* (LTE-A) oferecerem suporte à classificação de tráfego com provisionamento de QoS, elas são constantemente atualizadas em suas definições para suporte a novas aplicações. Observa-se, por exemplo, a inclusão de

[2]KDD também é referido como Mineração de Dados ou *Data Mining* [7].

novos identificadores de classes de tráfego e parâmetros de QoS mais estritos para suporte a novas aplicações entre especificações mais antigas [15] e outras mais recentes [16]. O uso da classificação de tráfego por meio do Aprendizado de Máquina elimina a necessidade de definição e atualização de classes, uma vez que o sistema "aprende" a reconhecer novos tipos de aplicações. Desta forma, o sistema adapta-se ao ambiente e torna-se capaz de prover controle mais adequado de QoS sem a necessidade de intervenção humana na definição de especificações relacionadas ao aspecto da classificação [14].

Apesar do CBRA ser uma tecnologia conceitual, ou seja, ainda não há casos conhecidos de implementação de CBRA em sistemas reais, este apresenta algumas limitações relacionadas ao *clustering* e sua aplicação, mencionadas mais adiante neste capítulo.

Portanto, investiga-se e aprimora-se, neste trabalho, a estratégia CBRA aplicada em tecnologias de comunicação móvel sob as especificações do LTE-A, que é amplamente utilizado até o momento da escrita desta tese, atualmente compreendido na Quarta Geração de Sistemas de Telefonia Móvel (4G) e projetado como provável tecnologia base de novas especificações a serem desenvolvidas e implementadas na Quinta Geração de Sistemas de Telefonia Móvel (5G) [17].

1.1 Estado da Arte

O CBRA foi idealizado a partir de alguns estudos que propõem e investigam a aplicação de *clustering* em algoritmos de escalonamento em redes de fibra óptica baseadas na tecnologia *Wavelength Division Multiplexing* (WDM) [18–20]. Os autores alegam que algoritmos de escalonamento tradicionais são estáticos e incapazes de adaptarem-se a conteúdos dinâmicos oferecidos como serviço, motivando o uso de *clustering* para KDD relacionado ao padrão de tráfego das aplicações e usuários, auxiliando a tomada de decisão. A solução proposta pelos autores caracteriza os usuários baseando-se na demanda individual de recursos por canal. Para realização do *clustering*, os autores implementam o algoritmo *K-means* (Seção 3.4.4) com o parâmetro de quantidade de *clusters* arbitrariamente estabelecido, porém testado com diferentes valores para análise e obtenção da configuração com o melhor desempenho.

Baseado nos trabalhos em [18–20], os autores em [21–24] propõem uma arquitetura CBRA para o LTE-A organizada em três etapas: (*i*) diferenciação de tráfegos; (*ii*) escalonamento no domínio do tempo com classificação de usuários, considerando parâmetros de QoS medidos pelo sistema; (*iii*) escalonamento no domínio da frequência com *feedback* de informação sobre taxa de perdas de pacotes, ou *Packet Drop Rate* (PDR). Cada etapa do mecanismo CBRA proposto visa atender um dos três níveis de operação a seguir: usuário, serviço e sistema. O controle de distribuição de recursos entre tráfegos do tipo *Real-Time* (RT) e *Non Real-Time* (NRT)[3] é feito por meio do *Hebian Learning*[4], que estabelece a quantidade adequada para cada tipo de tráfego em um determinado instante de tempo pautando-se na quantidade de tráfego e nos valores médios de PDR. O *clustering* categoriza tráfegos RT para posterior definição de prioridade de

[3]Classificação estabelecida pelos próprios autores nos trabalhos apresentados.

[4]O *Hebian Learning* é um algoritmo tradicional cujo modelo baseia-se na dinâmica de aprendizagem realizada entre dois neurônios biológicos.

acesso aos recursos.

Considerando também um sistema *Orthogonal Frequency Division Multiplexing* (OFDM), os autores em [25; 26] propõem um esquema estruturado em duas fases. A primeira encarrega-se da obtenção e análise dos dados, enquanto a segunda aloca os recursos e atualiza a base de dados utilizada. Os dados coletados referem-se à demanda de recursos do sistema pelos usuários e contêm padrões utilizados para auxiliar o mapeamento adequado de alocação considerando as associações e correlações existentes entre os usuários e serviços. Após a obtenção dos dados, o mecanismo utiliza o algoritmo *K-means* para classificação não-supervisionada, em condições de ausência de identificação, e Máquina de Suporte Vetorial, ou *Support Vector Machine* (SVM) [27] para classificação supervisionada, em casos onde há uma identificação de classe ou categoria previamente atribuída. Na segunda fase, um Algoritmo Genético (AG), populado com os resultados da classificação realizada pelo *K-means* e SVM, é utilizado para otimizar a alocação dos recursos selecionando o "perfil" de subquadro OFDM mais adequado.

A alocação de recursos é convencionalmente tratada como um problema de otimização. Tendo em vista essa abordagem, os autores em [28] propõem um *framework* de Aprendizado de Máquina assistido por Computação em Nuvem, ou *Cloud Computing*, para coletar dados históricos relacionados a procedimentos de alocação previamente realizados e, em posse dessa informação, explorar similaridades entre etapas anteriores para aperfeiçoamento de uma etapa de alocação atual. Desta forma, o problema de alocação de recursos é transformado em um problema de classificação, solucionado por um modelo preditivo de Aprendizado de Máquina, utilizando o algoritmo *k-Nearest Neighbors* (k-NN) (Seção 3.4.3) para obtenção dos resultados desejados. A proposta beneficia-se da alta capacidade de processamento da Computação em Nuvem, viabilizando a busca por soluções ótimas ou *quasi*-ótimas. Os dados considerados referem-se à qualidade do canal, à quantidade de usuários no sistema, ao número de identificação internacional do assinante móvel ou *International Mobile Subscriber Identification* (IMSI), entre outros. Para redução da complexidade computacional e da dimensionalidade dos dados, a proposta implementa um procedimento manual de seleção de características (*feature selection*), mantendo apenas os atributos mais relevantes.

Uma Rede Neural Profunda, ou *Deep Neural Network* (DNN), é empregada em [29] com o propósito de realizar aproximação funcional em tempo real de um algoritmo de alocação de potência empregado para otimização, reduzindo assim parte da complexidade computacional necessária. O algoritmo utilizado para otimização é tratado como uma espécie de caixa preta, de maneira que a DNN é treinada para imitar seu comportamento com o mínimo de erro possível.

Em [30] os autores propõem um mecanismo de Aprendizado de Máquina multi-agente para alocação de potência e escalonamento. Um algoritmo SVM é utilizado para classificação dos subcanais para cada um dos usuários, com as possíveis classificações resultantes: não alocado; alocado com potência máxima; alocado com potência ajustável. Logo após, uma Rede de Crença Profunda, ou *Deep Belief Network* (DBN)[5], é utilizada para ajuste "fino" dos níveis de potência em cada subcanal classificado para alocação com potência ajustável.

[5]Trata-se de um tipo de rede neural profunda (com múltiplas camadas) cujas variáveis não são diretamente observadas, mas inferidas por meio de um modelo probabilístico a partir de outras variáveis relacionadas.

No aspecto apenas de classificação de tráfego por meio do Aprendizado de Máquina, entre os diversos trabalhos mais relevantes desenvolvidos com essa finalidade [12; 31–40], destacam-se:

- O trabalho em [12], em que os autores ponderam sobre os principais prejuízos causados pela alta dimensionalidade e redundância de características: degradação da precisão e da eficiência da classificação. Além disso, justificam a necessidade de manter o classificador atualizado com os dados mais recentes dos tráfegos, evitando assim impactos causados pela "má interpretação" do classificador, considerado um dos desafios mais importantes para as tecnologias de classificação baseadas em Aprendizado de Máquina. Por esse motivo, eles propõem uma abordagem baseada em Aprendizado Profundo para otimização robusta de características, cuja finalidade é a classificação aprimorada de tráfego da *Internet*. O método desenvolvido elimina características irrelevantes analisando a correlação estatística entre os atributos dos fluxos de tráfego. As características mantidas são encaminhadas para uma DBN, que encarrega-se de gerar novas características considerando as dependências mais relevantes. Finalmente, uma nova etapa de eliminação de redundâncias é realizada, porém orientada à seleção de características que beneficiem classes com menor representatividade no sistema, reduzindo o desequilíbrio de classificação entre classes "maiores" e "menores" bem como a mitigação de "má interpretação" do classificador;

- Em [31] os autores empregam *clustering* utilizando um *Auto-encoder* (Seção 4.2.2.2) para classificação não-supervisionada de tráfego da *Internet*. Embora o *Auto-encoder* seja geralmente utilizado para redução da dimensionalidade e codificação, os autores desenvolvem um método para classificação a partir de sua codificação resultante. A rotulação dos agrupamentos formados baseia-se em um método semi-automático, que considera tipos de tráfegos previamente conhecidos enquanto mantém-se "agnóstico" a novos tipos de classificações, normalmente relacionadas a aplicações recentes;

- Os autores em [35; 36] propõem um mecanismo de classificação de tráfego com extração de características, redução de dimensionalidade por meio de Análise de Componentes Principais, ou *Principal Component Analysis* (PCA), e *clustering* com uso de *K-means*. O objetivo é detectar anomalias no comportamento dos tráfegos observados para fins de segurança da rede.

Um consenso quase absoluto, existente entre os autores dos trabalhos cujo foco é a classificação de tráfego por meio do Aprendizado de Máquina, é o reconhecimento de que os métodos tradicionais de classificação (identificação por porta, cabeçalho, tipo de protocolo de transporte, entre outros) são limitados, inadequados e ineficientes para o contexto atual, caracterizado por uma enorme variedade de aplicações.

Uma aplicação diferente do *clustering* em sistemas de comunicação pode ser vista em [41]. A proposta emprega *K-means* para auxiliar a difusão de dados do tipo *server push*[6] em redes sem fio. Os agrupamentos estabelecidos definem categorias de conteúdo com maior demanda

[6] A tecnologia *server push* emprega um tipo de comunicação cuja requisição inicia-se por uma entidade denominada *publisher* ou servidor central, ao invés de um cliente.

no sistema. A quantidade de agrupamentos no *clustering* é estabelecida considerando a estrutura física do sistema, neste caso a quantidade de discos utilizados para armazenamento e disponibilização de conteúdo.

Finalmente, o *clustering* também pode ser utilizado de forma bastante diversificada no LTE-A, como na proposta apresentada em [42]. Nela os autores propõem um mecanismo de gerenciamento de mobilidade em sistemas tipo *Self-organizing Networks* (SON)[7] com a implementação de Mapa Auto-organizável, ou *Self-organizing Map* (SOM)[8], para detectar regiões onde o *handover* ocorre com mais frequência e, baseado em "experiências" anteriores, opta pela admissão ou não de um dispositivo móvel à célula, reduzindo assim sinalizações desnecessárias. A quantidade de neurônios do SOM é estimada e estabelecida de forma autônoma por meio do algoritmo *X-means* (Seção 3.4.6). Os autores denominam a técnica como *X-means with* SOM (XSOM).

1.2 Definição do Problema

A preocupação com o gerenciamento adequado de recursos permeia praticamente todos os trabalhos mencionados como estado da arte. No âmbito de sistemas de comunicação em geral, como via de regra, o gerenciamento destina-se a otimizar o uso dos recursos disponíveis visando maximizar o desempenho na comunicação. Para realizar parte desse objetivo, procedimentos como classificação de tráfego e controle de QoS são recorrentemente utilizados.

De maneira geral, o CBRA gerencia o acesso aos recursos em função de características que designam a necessidade de cada aplicação e os acordos de nível de serviço estabelecidos, se for aplicável. Entretanto, um questionamento inicial a ser feito é: como definir corretamente as características/atributos e os acordos de nível de serviço das aplicações e usuários de maneira que estes representem adequadamente todas as informações necessárias para classificação de tráfego e controle otimizado de QoS?

A abundância de informações proporcionada pelo *Big Data* pode auxiliar a obtenção de uma resposta para a pergunta anteriormente realizada. Contudo, o excesso de dados desencadeia maior complexidade computacional, especialmente se o algoritmo utilizado possui fator de complexidade vinculado à quantidade de registros. Como exemplo desse fato, menciona-se os trabalhos em [18–20] e [25; 26], que possuem algumas limitações no modelo proposto nesse sentido. Em [25; 26], o uso de AG para tratamento do *Big Data* coletado (mesmo com a população inicial reduzida) proporciona complexidade acima do convencional, pois além da quantidade de usuários é preciso considerar as iterações necessárias para convergência do algoritmo, impactando diretamente o desempenho obtido por aplicações em tempo real e também em sistemas reais, se aplicado. Em [18–20], dados dos canais são utilizados como atributos dos elementos mapeados, indicando a demanda dos usuários por capacidade de canal. Em sistemas reais, se a quantidade de canais for muito grande e seus dados forem utilizados como atributos, a com-

[7] O termo SON (redes auto-organizáveis) foi introduzido pelo *3rd Generation Partnership Project* (3GPP) na *Release* 8 e desenvolvido nas demais *releases* subsequentes.

[8] O SOM é um tipo de rede neural artificial, treinado por meio de Aprendizado Não-supervisionado, usado para criar representações, em maior ou menor escala, dos dados recebidos como entrada.

plexidade do *clustering* pode tornar-se insustentável para o mecanismo considerando o tempo disponível pelo sistema para realização dos procedimentos de alocação necessários. No caso de um sistema *Orthogonal Frequency Domain Multiple Access* (OFDMA), por exemplo, o subquadro possui uma duração limitada e o CBRA deve ser capaz de realizar todo o processamento para alocação no intervalo de tempo estabelecido.

Além do aumento da complexidade, o uso de uma quantidade muito grande de características ou atributos para representação dos elementos desencadeia um problema denominado "maldição da dimensionalidade", ou *curse of dimensionality* [43]. À medida que a quantidade de características (dimensões) aumenta, as (dis)similiaridades entre os objetos analisados tornam-se cada vez menos significativas. O excesso de características tende a deixar os dados muito dispersos no hiperplano, dificultando a formação de agrupamentos (ou *clusters*) de boa qualidade. Essa limitação impacta a caracterização adequada dos parâmetros de QoS pelo CBRA, visto que ela baseia-se nos agrupamentos produzidos [12]. Logo, no CBRA, necessita-se sobretudo uma definição adequada de características para representação dos dados com nível tolerável de complexidade e alto fator de qualidade nas classificações.

Para reduzir a quantidade de características, recorre-se muitas vezes ao procedimento de seleção de características ou *feature selection* (Seção 4.2.1), assim como realizado em [21–24] e [28], por exemplo, em que elegem-se as características mais importantes para o modelo e problema analisados. A seleção de características é bastante conveniente para melhorar a eficiência do *clustering* e também para reduzir seu tempo de processamento [44]. Entretanto, embora os diferentes tipos de seleção de características disponíveis sejam sempre orientados ao modelo empregado [45], eles podem suprimir informações relevantes para o algoritmo de Aprendizado de Máquina [46], restringindo a capacidade do CBRA.

O PCA, um algoritmo de redução de dimensionalidade amplamente utilizado na literatura, reduz automaticamente a quantidade de características de uma base de dados e captura as correlações entre os elementos analisados mantendo uma determinada quantidade de variância. No entanto, o PCA possui algumas limitações para representação apropriada da informação. A mais importante diz respeito à incapacidade em identificar correlações não-lineares, restringindo a precisão da representação e a capacidade de compressão com alto nível de variância [47].

Não obstante os desafios apresentados, a parametrização arbitrária do *clustering* é um obstáculo para tornar o procedimento totalmente autônomo e melhorar a qualidade das classes (*clusters*) produzidas. O estabelecimento arbitrário da quantidade de classes nem sempre corresponde ao ideal para o conjunto de dados selecionado e pode, até mesmo algumas vezes, desencadear erros severos de classificação. Estabelecer a quantidade ideal de classes no *clustering* reduz as perdas de similaridades entre elementos, melhorando a qualidade dos agrupamentos [48], e provê um controle mais adequado de QoS após a classificação de tráfego [12]. Para o CBRA, melhorar a qualidade das classes implica, consequentemente, em melhorar a caracterização de parâmetros de QoS para as aplicações e usuários [31; 49].

1.3 Solução Proposta

Em vista dos problemas apresentados na seção anterior, propõe-se nesta tese um mecanismo CBRA para sistemas LTE-A evidenciado por duas etapas: uma para aprendizado de características e redução da dimensionalidade, por meio de um *Auto-encoder*; outra para classificação não-supervisionada de tráfego por meio de *clustering não-paramétrico*[9], mediante o uso do algoritmo *X-means*.

A etapa inicial de um mecanismo CBRA é determinada pelo estabelecimento de características de acordo com o modelo considerado. As características estabelecidas devem ser suficientes para mapear os parâmetros de QoS mais adequados conservando baixa complexidade computacional para o mecanismo. O *Auto-encoder*, um tipo de rede neural autônoma treinada de forma não-supervisionada, realiza o aprendizado de características (*feature learning*) extraindo apenas as informações mais relevantes dos elementos a partir de qualquer dimensionalidade inicialmente considerada. Desta forma, o CBRA proposto obtém a melhor representação possível da informação para análise sem ser impactado pelos efeitos da "maldição da dimensionalidade". A "maldição da dimensionalidade" é um problema tradicional em aplicações de Aprendizado de Máquina, porém sua abordagem e implicações em estratégias CBRA ainda não foram investigadas, principalmente em sistemas LTE-A, de acordo com o levantamento bibliográfico realizado pelo autor, até o presente momento.

Se comparado com o PCA, o *Auto-encoder* é mais direto e flexível. Ele captura correlações não-lineares entre os elementos, mantendo maior nível de variância, e suporta amostras de qualquer tamanho, transformando um elemento por vez. Embora a fase de treinamento de um *Auto-encoder* possua maior complexidade computacional [51], essa fase é *offline* e realizada apenas uma única vez.

Para realização do *clustering* sem parâmetros, emprega-se o *X-means*. Trata-se de um algoritmo iterativo para particionamento cuja condição de parada satisfaz um determinado critério estipulado. Após a parada, assume-se a quantidade resultante de partições estabelecidas como o ideal para organização do conjunto de dados. Para o CBRA, o *X-means* desempenha papel importante pelo fato de que a quantidade ideal de classes não necessita ser arbitrariamente estabelecida ou fixada. Desta forma, a caracterização dos parâmetros de QoS pelo CBRA baseia-se apenas nos padrões descobertos nos dados, sem qualquer tipo de parametrização adicional, automatizando o procedimento.

1.4 Objetivos e Metodologia

O principal objetivo deste trabalho é propor e avaliar uma solução (mecanismo) para automatizar as etapas de composição de dados e classificação e agrupamento de elementos em estratégias de alocação de recursos baseadas em *clustering* (CBRA) implementadas em redes

[9]Refere-se como *clustering não-paramétrico* o método de *clustering* que não afere "suposições" sobre a quantidade ou formato dos *clusters* bem como a distribuição dos elementos [50].

móveis de banda larga sem fio, mais especificamente baseadas na tecnologia LTE-A, oferecendo suporte adequado a QoS e contornando o problema da "maldição da dimensionalidade", que pode ocorrer nas etapas iniciais do CBRA.

Investiga-se e utiliza-se o Aprendizado de Máquina Não-supervisionado como principal técnica aplicada à solução proposta. Além disso, deseja-se melhorar o nível de QoS oferecido a aplicações em tempo-real transmitidas em redes móveis, como vídeo e VoIP, por exemplo, demonstrando a capacidade da solução proposta em atender adequadamente as demandas de aplicações mais exigentes em termos de recursos do sistema. Finalmente, pretende-se também verificar a plausibilidade do Aprendizado de Máquina na autonomia e ampliação da capacidade de tomada de decisões na alocação de recursos em redes móveis.

Visto que não há conhecimento de uma estratégia de CBRA implementada em sistemas reais, juntamente com o alto custo e complexidade exigidos para análise nesses ambientes, avalia-se a solução proposta por meio de simulação computacional de redes móveis. Concorrentemente, avalia-se o desempenho e o comportamento de trabalhos relacionados, em sua implementação original, para fins de comparação (*benchmark*) sob condições experimentais controladas. Os cenários descritos são reproduzidos de forma a atender as expectativas dos usuários e operadores de rede em condições mais próximas da realidade e que também sejam capazes de diferenciar apropriadamente as propostas para avaliação.

1.5 Contribuições

Assumindo como ponto de partida todo o referencial teórico relacionado às estratégias CBRA desenvolvidas nos trabalhos mencionados na Seção 1.2, destaca-se como principais contribuições desenvolvidas pelo autor, mencionadas neste trabalho:

- Estabelecimento de uma arquitetura ou *framework* genérico para o CBRA. Apesar de existirem alguns fundamentos sobre o CBRA na literatura, até o presente momento não encontrou-se, a partir do levantamento bibliográfico realizado, um esquema ou arcabouço de CBRA comum aplicável a todo tipo de tecnologia de comunicação;

- Definição de uma arquitetura de CBRA desenvolvida especificamente para sistemas LTE-A. A arquitetura proposta descreve um esquema "ideal" para realização do CBRA em um sistema de comunicação baseado nas especificações do 3GPP pertinentes à 4G e, futuramente, à 5G;

- Desparametrização do procedimento de *clustering*. O procedimento para formação de agrupamentos (*clustering*) do CBRA é do tipo paramétrico, ou seja, requer obrigatoriamente alguns parâmetros necessários para classificação dos elementos, como a quantidade de classes, por exemplo. A proposta desenvolvida e apresentada nesta tese elimina a exigência de informação da quantidade de classes como parâmetro, recorrendo unicamente ao conjunto de dados para obtenção desse tipo de informação. Para tal, implementa-se

um algoritmo de Aprendizado Não-supervisionado específico (*X-means*) para estimar a quantidade ideal de classes a partir dos dados definidos para uso pelo CBRA;

- Implementação de mecanismo de aprendizado de características com redução da dimensionalidade e reestruturação dos dados fornecidos ao CBRA, possibilitando melhor representação da informação e inserção de inúmeras características do sistema analisado sem afetar o desempenho do CBRA ou do próprio sistema, decorrente do problema da "maldição da dimensionalidade". Essa é a principal contribuição desenvolvida: possibilitar a uma tecnologia de comunicação a utilização do máximo de informações possíveis sem sobrecargas no desempenho é um avanço significativo no sentido de ampliar sua capacidade de maneira geral;

- Extensão de um mecanismo CBRA para suporte a tráfegos em tempo-real em sistemas de comunicação *wireless*. Sugerido pelos autores do trabalho em [20];

- *Benchmark* de diferentes mecanismos de CBRA existentes na literatura (trabalhos relacionados) aplicados a redes LTE-A em modo de simulação. Tal procedimento visa auxiliar a análise e continuidade de investigação do CBRA em estudos posteriores.

Além dos itens previamente enumerados como contribuição, ao longo do desenvolvimento do projeto apresentado nesta tese, foram escritos e publicados os trabalhos a seguir relacionados direta e indiretamente com o plano de trabalho realizado.

1.5.1 Produção técnica e científica

Santos, Einar C., and Paulo R. Guardieiro. "DRR Adaptive Quantum Scheduling Algorithm for WiMAX Multihop Relay Networks." *Telecommunications (IWT), 2015 International Workshop on*. IEEE, 2015. https://doi.org/10.1109/IWT.2015.7224572

Santos, Einar C., and Paulo R. Guardieiro. "Aprendizagem Computacional Não Supervisionada Aplicada à Alocação Autônoma de Recursos em Redes 4G LTE." *Anais do XII Encontro Anual de Computação - EnAComp*. UFG, 2015. https://doi.org/10.13140/RG.2.1.2848.6009

Santos, Einar C., and Paulo R. Guardieiro. "Upgrading LTE-Sim with a Simulation Model for Relay Type 1 Networks with QoS Support." *Proceedings of the 9th Latin America Networking Conference*. ACM, 2016. https://doi.org/10.1145/2998373.2998444

Santos, Einar C. "A Simple Reinforcement Learning Mechanism for Resource Allocation in LTE-A Networks with Markov Decision Process and Q-Learning." *arXiv preprint arXiv:1709.09312*. ArXiv.org, 2017. https://arxiv.org/abs/1709.09312[10,11]

[10] Citado em: Brand, Peter and Falk, Joachim and Sue, Jonathan Ah and Brendel, Johannes and Hasholzner, Ralph and Teich, Jürgen. "Reinforcement Learning for Power-Efficient Grant Prediction in LTE." *SCOPES'18. Proceedings of the 21st International Workshop on Software and Compilers for Embedded Systems*. ACM, 2018. https://doi.org/10.1145/3207719.3207722

[11] Citado em: Chen, Xinyu, Yu Liu, and Yumei Wang. "A Novel Downlink Scheduler Based on Q-Learning for Video Traffic in LTE Networks." *International Conference on Network Infrastructure and Digital Content (IC-NIDC)*. IEEE, 2018. https://doi.org/10.1109/ICNIDC.2018.8525617

Santos, Einar C. "Autonomous QoS-Based Mechanism for Resource Allocation in LTE-Advanced Pro Networks." *IEEE Colombian Conference on Communications and Computing - COLCOM*. IEEE, 2018. https://doi.org/10.1109/ColComCon.2018.8466714

Santos, Einar C. "A Supervised Machine Learning Mechanism for Traffic and Flow Control in LTE-A Scheduling." *Proceedings of the 10th Latin America Networking Conference*. ACM, 2018. https://doi.org/10.1145/3277103.3277121

Santos, Einar C. "Clustering-Based Resource Allocation Mechanism in Long Term Evolution Advanced Networks with Auto-encoder for Feature Learning." *Transactions on Emerging Telecommunications Technologies*. John Wiley & Sons, 2019. https://doi.org/10.1002/ett.3591

1.6 Notações

Esta seção explica as notações matemáticas utilizadas ao longo do texto desta tese.

Variáveis do tipo escalares são denotadas por uma letra em itálico com caixa baixa x ou caixa alta X, dependendo do contexto onde são empregadas. Vetores são denotados por uma letra em negrito, itálico e em caixa baixa $\boldsymbol{x}$, assim como também podem ser denotados com setas sobrescritas $\vec{\boldsymbol{x}}$.

O uso de caixa alta, negrito e itálico caracteriza uma matriz ou um conjunto de vetores representado na forma de matriz: $\boldsymbol{X}$. Conjuntos genéricos são denotados em negrito, itálico, caixa alta e cursiva: $\boldsymbol{\mathcal{X}}$.

Vetores e matrizes transpostos são sucedidos pelo símbolo † em sobrescrito, na forma $\boldsymbol{X}^{\dagger}$. A dimensão de vetores ou matrizes pode ser apresentada utilizando-se o separador $\times$ em subscrito, por exemplo: $\boldsymbol{X}_{2\times2}$; $\boldsymbol{x}_{4\times1}$.

Índices sucedem a variável de referência e são denotados em itálico no modo subscrito. Por exemplo, o índice i na variável de referência x é denotado como x_i. Mais de um índice pode ser utilizado em uma mesma variável, desde que haja separação por uma vírgula em subscrito: $x_{i,j}$.

Caracteres gregos também são empregados seguindo a mesma regra para notação apresentada. Ou seja, pode-se utilizar λ para variáveis escalares, $\boldsymbol{\lambda}$ para vetores e $\boldsymbol{\Lambda}$ para matrizes.

Os símbolos $\leftarrow$ e $:=$ denotam atribuição de valores a variáveis.

Funções são denotadas por letras em caixa alta ou baixa sucedidas por parênteses: $f()$ ou $F()$. Os índices e parâmetros de uma função são apresentados, respectivamente, sucedendo a letra da função, em subscrito, e entre os parênteses, por exemplo: $f_a(x)$; ou $F_\alpha(\boldsymbol{X})$.

1.7 Sinopse dos Capítulos

O Capítulo 2 refere-se ao embasamento teórico fundamental para compreensão de sistemas LTE-A. Nele são apresentados um histórico da evolução da tecnologia, conceitos relacionados à

especificação do sistema juntamente com sua arquitetura básica, implementações das camadas física, enlace e de rede, e alguns conceitos necessários referentes a QoS no LTE-A.

Dando continuidade ao embasamento teórico, os conceitos de Aprendizado de Máquina são apresentados no Capítulo 3. Este trata dos fundamentos do Aprendizado de Máquina de maneira mais abrangente, porém sem a pretensão de esgotar o assunto. Define-se Aprendizado de Máquina, suas categorias e as particularidades de cada uma. São apresentados alguns modelos e algoritmos de Aprendizado de Máquina pertinentes a cada categoria, selecionados de acordo com sua relevância para cada uma.

O Capítulo 4 trata da conceituação do CBRA. O capítulo descreve: os fundamentos do CBRA, idealizado a partir do conceito de Aprendizado de Máquina em ambiente de sistemas de comunicação com suporte a *Big Data*; o arcabouço genérico proposto para aplicação a qualquer tecnologia de comunicação, contemplando todas as etapas necessárias para realização do CBRA em seu nível de abstração mais elevado; uma arquitetura específica de CBRA desenvolvida para sistemas LTE-A; princípios para definição de características e composição da base de dados utilizada pela estratégia; fundamentos sobre classificação de elementos; descrição da etapa de ordenação e priorização de classes e elementos; e descrição da etapa de alocação dos recursos.

No Capítulo 5 apresenta-se a proposta de solução para o problema definido na Seção 1.2. O problema definido é pormenorizado com a finalidade de oferecer uma visão mais clara dos desafios a serem enfrentados. O mecanismo proposto como solução é então descrito em suas especificidades, juntamente com algumas discussões acerca de sua complexidade computacional. Os detalhes dos trabalhos relacionados também são apresentados com a intenção de contextualizar o mecanismo proposto e estabelecer suas principais diferenças para efeito de comparação.

O cenário, os parâmetros de simulação e as métricas para avaliação da proposta desenvolvida, bem como os resultados obtidos da avaliação conduzida são descritos no Capítulo 6. Os resultados apresentados são comentados e discutidos para ampliar a compreensão da avaliação efetuada.

O Capítulo 7 encerra o trabalho com as conclusões apropriadas a respeito. Algumas considerações importantes são dispostas como arremate da ideia geral da proposta e dos possíveis trabalhos futuros relacionados.

Capítulo 2

Redes Long Term Evolution Advanced (LTE-A)

2.1 Evolução dos Sistemas Móveis de Telecomunicação

A história dos sistemas móveis de telecomunicação na era moderna teve seu início com as especificações *Advanced Mobile Phone System* (AMPS), *Nordic Mobile Telephone* (NMT) e *Total Access Communication System* (TACS), pertencentes à chamada Primeira Geração de Sistemas de Telefonia Móvel (1G).

Empregadas comercialmente no começo da década de 1980, as tecnologias da 1G adotavam o sistema de comunicação analógico com divisão de frequência para recepção e transmissão de sinal, oferecendo suporte a apenas uma única chamada por canal. A primeira especificação AMPS foi originalmente padronizada como *Interim Standard-3* (IS-3) pelo *American National Standards Institute* (ANSI). Apesar da coexistência com outras especificações, o AMPS tornou-se na época a tecnologia dominante em todo o mundo em escala comercial.

Inaugurando a Segunda Geração de Sistemas de Telefonia Móvel (2G), já no fim da década de 1980 e início da década de 1990, o sistema *Digital-Advanced Mobile Phone Service* (D-AMPS) – também referido como IS-54, para canal de controle analógico, ou IS-136, para canal de controle digital – aperfeiçoou o AMPS com a adoção de modulação digital, além de utilizar o método *Time Division Multiple Access* (TDMA) para acesso ao meio. Pouco depois, a empresa Qualcomm desenvolveu o cdmaOne (nome comercial para a especificação IS-95), que utilizava o *Code Division Multiple Access* (CDMA), possibilitando melhor aproveitamento da frequência alocando usuários em códigos de espalhamento espectral diferentes.

Paralelamente, outra tecnologia contemporânea importante competia com o D-AMPS e o cdmaOne, colaborando para a redução de custos. Era o *Global System for Mobile Communications* (GSM), desenvolvido pelo *European Telecommunications Standards Institute* (ETSI).

O GSM, originalmente conhecido como um grupo do *Committee of European Post and Telecommunications* (CEPT) denominado *Groupe Spéciale Mobile*, tornou-se especificação para redes celulares digitais no início da década de 1990. A padronização do GSM, em sua primeira fase, buscou habilitar a implantação do sistema para a banda de 900 MHz, utilizando TDMA,

incluindo suporte a telefonia (voz), chamadas de emergência, *Short Message Service* (SMS) entre outros serviços suplementares como a transmissão de dados por comutação de circuito [52]. No início a tecnologia *Circuit Switched Data* (CSD) possibilitava a transmissão de dados no GSM em taxas de até 9,6 Kbps e era equivalente a um *modem* convencional conectado a uma rede *Public Switched Telephone Network* (PSTN). Posteriormente, novas tecnologias aperfeiçoaram a especificação GSM, melhorando a velocidade alcançada nas taxas de dados, como o *High Speed Circuit Switched Data* (HSCSD), ampliando até 57,6 Kbps. Para superar as limitações da comutação por circuito, o *General Packet Radio Service* (GPRS) acrescentou ao GSM a tecnologia celular de comutação por pacotes, oferecendo suporte transparente ao protocolo TCP/IP na rede de núcleo e melhorando as taxas de dados para até 114 Kbps.

A segunda fase de padronização do GSM teve como contribuição mais relevante uma complexa arquitetura de protocolos e camadas para o sistema, compreendendo canais de controle e dados, e o *GPRS Tunnelling Protocol* (GTP), todos utilizados nas especificações mais recentes geradas a partir do GSM e pertencentes à Quarta Geração de Sistemas de Telefonia Móvel (4G). Outro avanço do GSM foi a inclusão da tecnologia *Enhanced Data Rates for GSM Evolution* (EDGE), que praticamente triplicou a taxa máxima de dados alcançada em comparação ao GPRS.

A Terceira Geração de Sistemas de Telefonia Móvel (3G) teve início formal estabelecido pela *International Telecommunications Union* (ITU), no fim da década de 1990, ao criar o programa *International Mobile Telecommunications-2000* (IMT-2000), onde foram definidos os requisitos básicos para alinhamento tecnológico com a denominação 3G. Nesse período dois grupos de trabalho (*workgroups*) distintos foram formados por diferentes organizações de desenvolvimento de padrões de telecomunicação: o *3rd Generation Partnership Project* (3GPP), cujo foco estava no desenvolvimento das especificações da família GSM; e o *3rd Generation Partnership Project 2* (3GPP2), que direcionou seus esforços para as tecnologias 3G baseadas no IS-95. A partir de então duas novas especificações passaram a competir pelo mercado das operadoras de rede celular: o *Wideband Code Division Multiple Access* (WCDMA) e o CDMA2000.

O WCDMA caracteriza-se por um sistema com *interface* de rádio padrão, fruto da evolução do GSM com inclusão do CDMA ao sistema, inspirado no cdmaOne, na tentativa de suprir limitações e ampliar o mercado. Por vezes o WCDMA é confundido, ou até mesmo tido como sinônimo do *Universal Mobile Telecommunication System* (UMTS). Na realidade, o UMTS é composto por um sistema completo, desenvolvido a partir da inclusão da tecnologia EDGE ao GSM, ilustrado na Figura 2.1, compreendendo o WCDMA como tecnologia da rede de acesso via rádio – a *Universal Terrestrial Radio Access Network* (UTRAN) –, uma rede de núcleo, protocolos e arquiteturas para garantia de QoS. O WCDMA opera em dois modos: *Frequency Division Duplex* (FDD) e *Time Division Duplex* (TDD). O sistema alcança largura de banda de 5 MHz e as taxas de dados podem chegar até 384 Kbps (na 3GPP *Release '99*).

O CDMA2000, também conhecido como C2K ou *IMT Multi-Carrier* (IMT-MC), foi o passo seguinte de desenvolvimento do IS-95 (cdmaOne). É um sistema de multiportadoras, com largura de banda variando entre 1,25 MHz e 5 MHz, e suporte a QoS. Em seu primeiro estágio, o IS-2000, alcançava taxas de até 153 Kbps e era compatível com equipamentos do IS-95. No

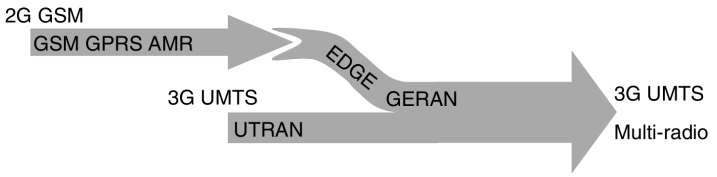


Figura 2.1: Inclusão do UMTS na transição da 2G para a 3G. Extraído de [52].

segundo estágio, conhecido como *Evolution-Data Optimized* (EVDO, EV-DO ou EV), era capaz de alcançar taxas de até 2 Mbps em áreas com alta potência e baixa interferência de sinal, oferecendo em média algo entre 400 e 700 Kbps.

A transição da 3G para a 4G foi marcada pela inclusão do *High Speed Downlink Packet Access* (HSDPA), da especificação 3GPP *Release 5*, ao WCDMA, inaugurando a era da banda larga móvel. No HSDPA as taxas de dados no canal *downlink* variavam inicialmente entre 1,8 e 3,6 Mbps. A 3GPP *Release 6* trouxe melhorias nas taxas do canal *uplink* com o *High Speed Uplink Packet Access* (HSUPA), ampliando a taxa do *uplink* para até 5,74 Mbps, além de aumentar a velocidade no *downlink* para até 14,4 Mbps e introduzir a tecnologia *Multimedia Broadcast Multicast Services* (MBMS), adequada para transmissão de sinal de TV móvel.

A fusão das tecnologias HSDPA e HSUPA tornou-se conhecida como *High Speed Packet Access* (HSPA) [53], substituindo o WCDMA como tecnologia adotada na UTRAN de redes UMTS. A 3GPP *Release 7*, conhecida como *Evolved High Speed Packet Access* (HSPA+) trouxe avanços, aumentando a quantidade de antenas utilizadas pelos dispositivos com o *Multiple-Input and Multiple-Output* (MIMO) e também com o *beamforming*, melhorando a taxa de dados experimentada pelos usuários para até 28 Mbps no *downlink* e 11 Mbps no *uplink*.

A partir da 3GPP *Release 8*, em meados de 2010, criou-se a especificação *Long Term Evolution* (LTE). Neste sentido o LTE e o HSPA+ passaram a ser desenvolvidos paralelamente, embora o LTE era tecnicamente a principal escolha para assumir o posto da 4G pelo 3GPP, apesar de ainda não atender todos os requisitos estabelecidos pelo recém estabelecido *International Mobile Telecommunications-Advanced* (IMT-Advanced) da *ITU Radiocommunication Sector* (ITU-R), visto que o 3GPP havia estabelecido seus próprios objetivos independentemente do IMT-Advanced.

Algumas características do HSPA+ foram aperfeiçoadas, como inclusão de duas portadoras no canal *downlink*, conhecida como *Dual Carrier High-Speed Downlink Packet Access* (DC-HSDPA), para dobrar a taxa de dados, e uso simultâneo de MIMO e modulação 64QAM. Os requisitos estabelecidos pelo IMT-Advanced só seriam completamente atendidos pelo 3GPP nas futuras *Releases*, com a chegada do *Long Term Evolution Advanced* (LTE-A). A Figura 2.2 apresenta a transição do HSPA e HSPA+ para o LTE e LTE-Advanced nas *Releases* do 3GPP.

É importante mencionar tecnologias semelhantes e que evoluíram paralelamente, como o *Worldwide Interoperability for Microwave Access* (WiMAX), desenvolvido pelo *Institute of Eletric and Electronic Engineers* (IEEE), e o *Ultra Mobile Broadband* (UMB) (uma *Release* poste-

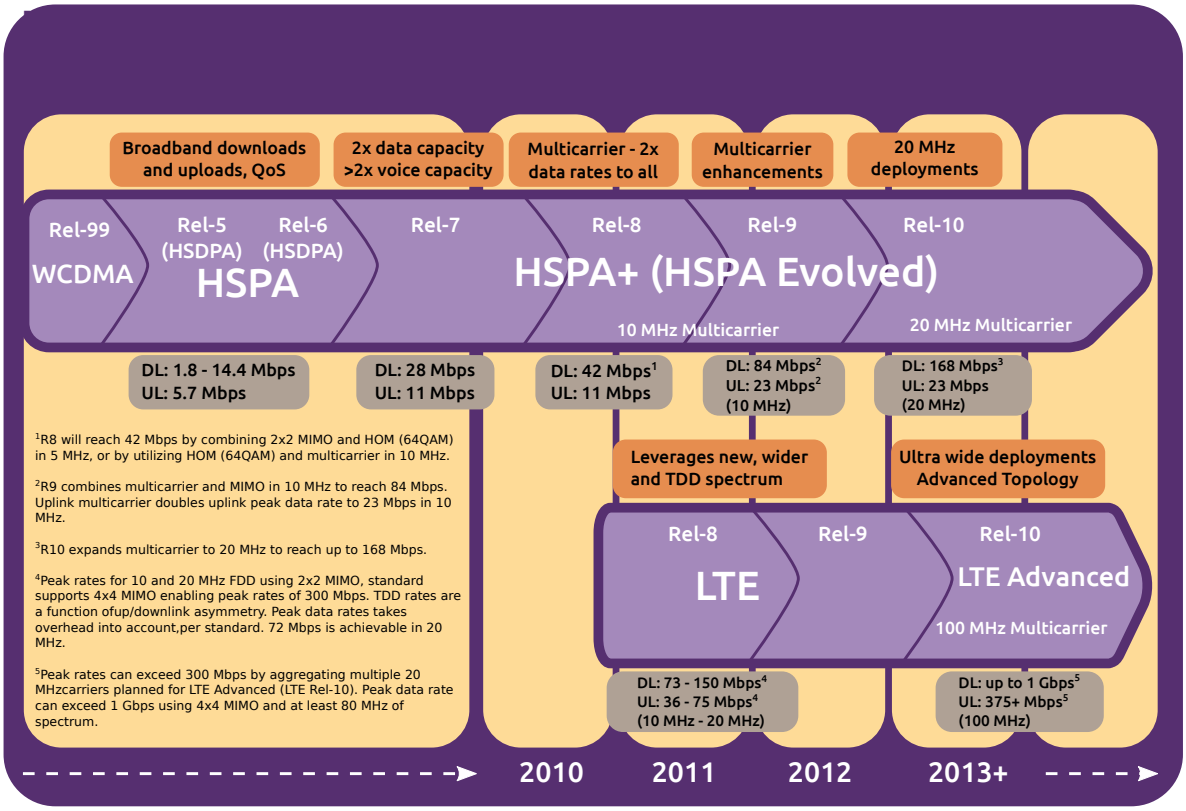


Figura 2.2: Evolução do HSPA ao LTE-Advanced. Extraído, adaptado e redesenhado de [54].

rior ao EVDO), descontinuado pelo 3GPP2.

A Figura 2.3 ilustra a evolução dos sistemas móveis de telecomunicação mais importantes, desde a 1G até a 4G.

A Quinta Geração de Sistemas de Telefonia Móvel (5G) correponde ao próximo estágio de evolução após a 4G. A 5G está em desenvolvimento, até o momento da escrita deste trabalho, e há diversos tópicos de discussões sobre suas principais características e as tecnologias que farão parte da nomenclatura. Algumas *releases* recentes, como a *Release 15* [56] por exemplo, abordam as características essenciais que devem compor a tecnologia. Outro aspecto importante também considerado é o impacto tecnológico, econômico e social causado pela chegada da 5G. A própria ITU reconhece, em alguns eventos onde é representada, a necessidade de ponderar sobre o impacto social causado pela implantação da 5G em todo o planeta.

Até a chegada da 5G são aguardadas algumas características, a saber:

- **ubiquidade:** permeará (quase) todas as regiões do planeta de maneira que seu acesso possa ocorrer em (praticamente) todos os lugares;
- **transparência:** sua utilização não deverá requerer grau elevado de conhecimento ou

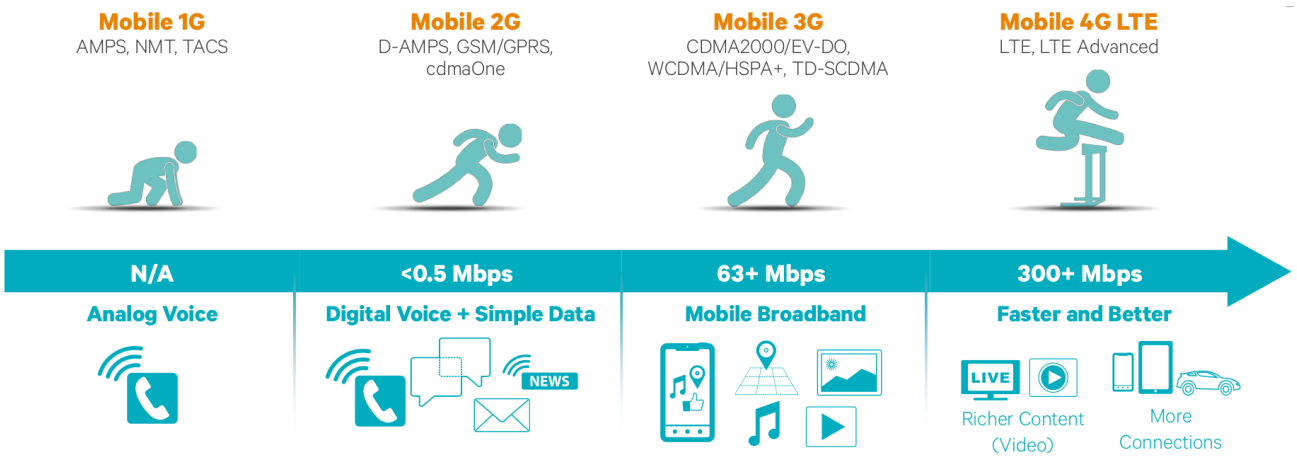


Figura 2.3: Evolução dos sistemas móveis de telecomunicação da 1G à 4G. Extraído de [55].

sequer conhecimento técnico para realizar algum tipo de acesso;

- **composição por diversas tecnologias convergentes:** a ideia é que a 5G possibilitará a coexistência e convergência de diversos tipos e tecnologias de redes móveis operando em conjunto, sob demanda e com alto fator de disponibilidade;
- **rapidez:** o desempenho real bem como a percepção de desempenho pelos usuários será ágil e eficiente;
- **"inteligência":** a incorporação de mecanismos de otimização e inteligência artificial nos dispositivos dos usuários proporcionará uso mais racional dos recursos disponíveis.

Além das características mencionadas, espera-se muito também sobre as tecnologias integradas à 5G e seu impacto na comunicação entre dispositivos e usuários. As principais tecnologias relacionadas são [57]:

- **tranceptores integrados de ondas milimétricas:** possibilitam transmissão e recepção de sinal em ondas curtas, alcançando frequências na ordem de dezenas (até mesmo centenas) de *gigahertz*, aumentando consequentemente a largura de banda disponível no sistema. Mais detalhes podem ser encontrados em [58];
- **sistema de múltiplas antenas:** também conhecido como *Massive MIMO*, proporciona mecanismos para utilização de um número ainda maior de antenas, se comparado com a 4G, instaladas nos dispositivos transmissores e receptores;
- **comunicação entre dispositivos formando redes de sensores (arquitetura *Device-to-Device* (D2D):** compreende a troca de informações, o sensoreamento e controle de dispositivos de forma autônoma utilizando a infraestrutura de rede existente, caracterizando o conceito da *Internet* das coisas ou *Internet of Things* (IoT). O conceito de comunicação entre dispositivos também é denominado *Machine-to-Machine* (M2M), embora haja algumas pequenas diferenças conceituais entre os termos.

A ITU-R estabeleceu no início de 2012 o programa *International Mobile Telecommunications 2020* (IMT-2020) com a finalidade de conduzir as atividades para desenvolvimento da 5G até a próxima década em todo o mundo. O IMT-2020 estabelece as diretrizes e requisitos necessários para suporte a redes 5G em escala global.

Diversas atividades também vem sendo realizadas por outras entidades, como a *Next Generation Mobile Networks Alliance* (NGMN *Alliance*) – composta por 24 empresas operadoras de sistemas móveis de telecomunicação – que estabeleceu em [59] alguns requisitos necessários para padronização e disponibilidade da tecnologia 5G até 2020.

Muito é esperado sobre a 5G e seu real impacto nas telecomunicações e também nas relações sociais, cada vez mais afetadas pela tecnologia. Existem diversas iniciativas dedicadas ao alinhamento da tecnologia com os requisitos necessários para atendimento às expectativas geradas até o momento. Apesar disso, a certeza sobre a 5G é que sua implantação deve superar em muito a experiência proporcionada por tecnologias pertencentes à 4G em termos de conectividade e desempenho.

2.2 Especificação LTE-A

Para compreensão, denota-se redes LTE as especificações da 3GPP estabelecidas até a *Release 8* [60], e LTE-A as especificações a partir da *Release 10* [61]. Semelhantemente ao intercâmbio dos termos WCDMA e UMTS, há uma relação pouco comum entre as terminologias do LTE/LTE-A e do UMTS melhor explicada a seguir [62]:

- os componentes do LTE/LTE-A são uma evolução da UTRAN no UMTS e compreendem toda a rede de acesso via rádio, conhecida como *Evolved Universal Terrestrial Radio Access Network* (E-UTRAN);
- a rede de núcleo, baseada no protocolo TCP/IP, é denominada *Evolved Packet Core* (EPC);
- a combinação da EPC com a E-UTRAN compõe o *Evolved Packet System* (EPS), sendo o termo correto para referir-se a todo o sistema;
- o termo mais comum sinônimo do EPS para referência ao sistema é o *System Architecture Evolution* (SAE), LTE/SAE (LTE-A/SAE) ou simplesmente LTE/LTE-A.

Incontestavelmente, o LTE-A é a tecnologia mais utilizada dentro do contexto da 4G até o momento da escrita deste trabalho. O crescente aumento na demanda por serviços móveis de telecomunicação, desde o período compreendido entre a transição da 3G para a 4G, impulsionou o desenvolvimento da especificação. Sua característica mais notável para os usuários é o suporte a aplicações avançadas com altas taxas de transmissão de dados, podendo alcançar teoricamente 100 Mbps em condições de alta mobilidade e 1 Gbps com baixa mobilidade.

A maioria das implementações LTE-A opera dentro da banda recomendada pelo IMT-Advanced, que é de 2,1 GHz. Outras bandas podem ser utilizadas, dependendo da regulação de

cada país, como o caso da faixa de 700 MHz recentemente leiloada no Brasil.

Outro fato (relativamente) recente foi a avaliação do LTE-A por 18 empresas de telecomunicações que demonstraram satisfazer completamente os critérios estabelecidos pela ITU-R no IMT-Advanced [63]. Além dos requisitos, alguns pontos-chave do IMT-Advanced incluem:

- alto grau de uniformidade de funcionalidades;
- compatibilidade entre serviços (redes móveis e fixas);
- capacidade de interconexão com outros sistemas de acesso a rádio;
- alta qualidade em mobilidade;
- adequação do uso de equipamentos de usuário para todo o mundo;
- equipamentos, serviços e aplicações "amigáveis";
- capacidade global de *roaming*.

2.2.1 Arquitetura Básica

A arquitetura do LTE-A compreende a rede de acesso via rádio (E-UTRAN) e a rede de núcleo (EPC), formando o EPS/SAE. A Figura 2.4 apresenta o modelo conceitual da arquitetura básica do sistema LTE-A (EPS/SAE) composto pelas redes EPC e E-UTRAN.

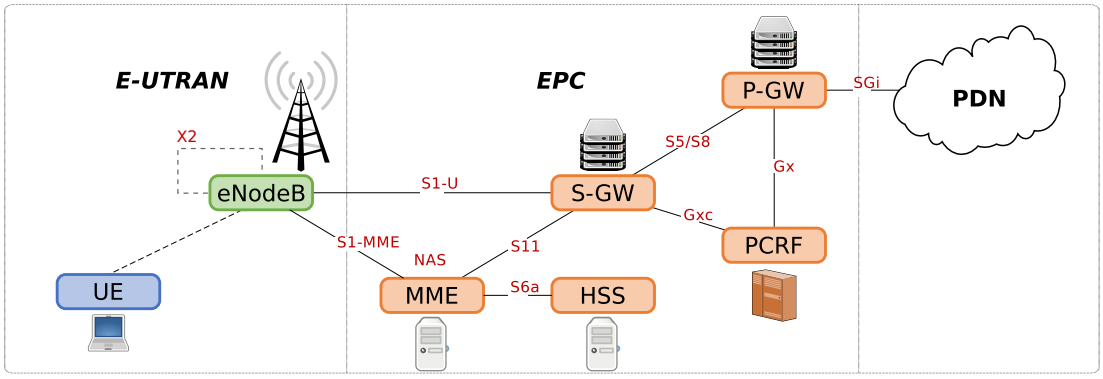


Figura 2.4: Ilustração da arquitetura básica do sistema LTE-A. Adaptado de [64].

A *Evolved Node Base* (eNB ou eNodeB), estação de rádio base pertencente à E-UTRAN, estabelece a ponte de comunicação entre as duas redes do sistema EPS/SAE, além de fornecer acesso aos usuários conectados, os *User Equipments* (UEs). Além da eNB há também estações bases locais (residenciais) conhecidas como *Home eNodeBs* (HeNBs) ou *Femtocells*, cuja finalidade é ampliar a cobertura do sistema provendo acesso a usuários situados em locais fechados como casas ou escritórios.

A *interface* física de comunicação das eNBs e HeNBs com a EPC é cabeada, conhecida como S1. O canal S1-U estabelece a comunicação da eNB/HeNB com o *Serving Gateway* (S-GW)

no plano de usuário. O canal S1-C, ou S1-MME, conduz informações sobre o plano de controle com o *Mobility Management Entity* (MME) [65].

Há ainda uma *interface* de comunicação lógica entre eNBs denominada *interface* X2. A *interface* X2 estabelece um canal de sinalização entre eNBs (plano de controle) de uma mesma E-UTRAN. A função da X2 é realizar a troca de informações dos planos de usuário e controle.

Fisicamente, a comunicação via *interface* X2 é estabelecida entre eNBs no enlace *backhaul* via canal S1-C. Entretanto, visando reduzir o *overhead* no canal S1-C, a comunicação via *interface* X2 pode ser realizada fisicamente por meio da tecnologia *Control/Data Plane Separation Architecture* (CDSA) [66]. O CDSA propõe a separação física dos planos de dados e controle para reduzir a complexidade de gerenciamento da rede, estabelecendo limites bem definidos entre os canais de comunicação, e eliminando gargalos em decorrência da sobrecarga de tráfego dos planos sobre um meio físico em comum.

O S-GW é responsável em obter informações para realização da cobrança de serviços, se for o caso, e pela "ancoragem" dos UEs em mobilidade no sistema, além do roteamento e encaminhamento dos dados para o *Packet Data Network Gateway* (PDN *Gateway* ou P-GW). O P-GW responde pela alocação de endereço IP a um terminal, conecta a EPC à *Internet* ou alguma *Packet Data Network* (PDN) externa, realiza filtragem de pacotes e ações para aplicação dos acordos de níveis de serviço estabelecidos no *Policy and Charging Rules Function* (PCRF). O PCRF encarrega-se da cobrança e atribuição das políticas de QoS para os usuários, enquanto o *Home Subscriver Service* (HSS) gerencia a base de dados de informações dos assinantes de serviços.

Normalmente, existem diversas PDNs no sistema, e esses são identificados por um *Access Point Name* (APN). Trata-se basicamente de um endereço do protocolo *Domain Name System* (DNS) para tradução do endereçamento de rede da PDN.

2.2.2 Camada Física

A tecnologia de transmissão adotada na camada física do sistema LTE-A é o *Orthogonal Frequency Domain Multiple Access* (OFDMA) – um método variante do *Orthogonal Frequency Domain Multiplexing* (OFDM) que permite alocação de múltiplos usuários nas portadoras pertencentes ao mesmo *slot* –, para o canal *downlink*, e o *Single Carrier Frequency Division Multiple Access* (SC-FDMA), para o *uplink*.

O OFDMA divide o canal em subportadoras ortogonais com espaçamento de 15 kHz, segmentadas no domínio do tempo, com duração padrão de 66,7 μs, em que cada segmento é denominado *símbolo OFDM*. Os símbolos OFDM são separados por períodos de guarda denominados como *Cyclic Prefix* (CP), utilizados para eliminar interferência causada por atraso de propagação multipercurso. A duração de um CP normal é de 4,69 μs. O agrupamento de 12 subportadoras e 7 símbolos OFDM (ou 1 *slot*) forma a menor unidade lógica de alocação de dados em redes LTE-A: o *Resource Block* (RB). O RB totaliza 180 kHz de largura de banda com resolução de 0,5 ms. A Figura 2.5 ilustra a estrutura física de um RB.

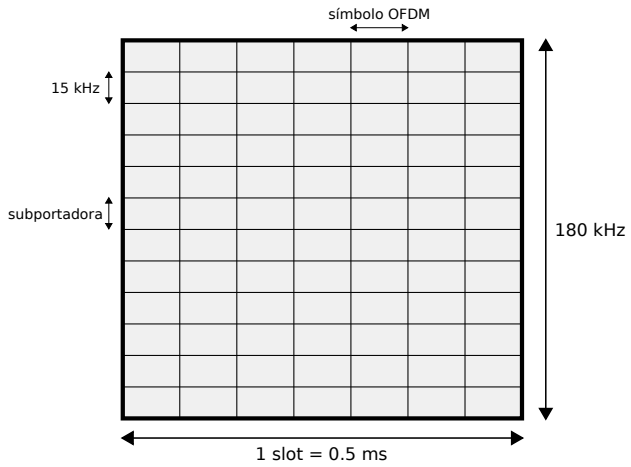


Figura 2.5: Exemplo da estrutura física de um RB.

O subquadro OFDM é formado por 2 *slots* de 0,5 ms enquanto o quadro é formado por 10 subquadros, totalizando 10 ms no domínio do tempo. No domínio da frequência o sistema suporta diferentes larguras de banda: 1,4 MHz; 3 MHz; 5 MHz; 10 MHz; 15 MHz e 20 MHz. A partir da *Release 10* implementou-se o suporte a agregação de portadoras, característica importante e necessária para ampliação da largura de banda no LTE-A, tanto no *downlink* como no *uplink*. Com esse recurso é possível obter largura de banda de até 100 MHz.

O esquema adotado pelo SC-FDMA é o de subportadoras adjacentes, onde cada símbolo é transmitido sequencialmente, ocupando 60 kHz cada no domínio da frequência. Detalhes sobre as diferenças entre o OFDMA e o SC-FDMA podem ser encontrados em [62; 65].

A camada física do LTE-A suporta os modos FDD e TDD. Os esquemas de modulação adotados são: o QPSK; o 16 QAM e o 64 QAM.

O suporte a múltiplas antenas possibilita utilização de até 8x8 MIMO no *downlink* e 4x4 MIMO no *uplink*.

Os canais mais importantes no LTE-A para comunicação no plano de usuário e de controle são: o *Physical Downlink Shared Channel* (PDSCH), para transmissão de dados no canal *downlink*, sendo compartilhado entre todos os UEs da célula; o *Physical Uplink Shared Channel* (PUSCH), correspondente ao PDSCH no *uplink*; os canais *Physical Downlink Control Channel* (PDCCH) e *Physical Uplink Control Channel* (PUCCH), utilizados para sinalização no *downlink* e *uplink*, respectivamente.

Para transmissão no *downlink* e no *uplink*, a eNB, primeiramente, deve obter conhecimento da qualidade do meio físico percebida pelos UEs. Essa informação é transmitida regularmente à eNB pelos UEs e conhecida como *Channel Quality Indicator* (CQI).

O CQI é um índice de referência com valor escalar entre 0 e 15, que indica a capacidade máxima de modulação, eficiência espectral do canal (ou subcanal), entre outras informações, sendo organizado inclusive por níveis de banda. É transmitido periodicamente, pelo PUCCH, ou aperiodicamente, pelo PUSCH. A Tabela 2.1 apresenta os tipos de modulação e eficiência

espectral correspondentes aos índices de CQI estabelecidos na especificação.

Tabela 2.1: Tabela de referência do CQI. Extraído, adaptado e traduzido de [67].

CQI	**Modulação**	**Eficiência Espectral (*bits*/símbolo)**
0	Não há	0
1	QPSK	0,1523
2	QPSK	0,2344
3	QPSK	0,3770
4	QPSK	0,6016
5	QPSK	0,8770
6	QPSK	1,1758
7	16 QAM	1,4766
8	16 QAM	1,9141
9	16 QAM	2,4063
10	64 QAM	2,7305
11	64 QAM	3,3223
12	64 QAM	3,9023
13	64 QAM	4,5234
14	64 QAM	5,1152
15	64 QAM	5,5547

O CQI é útil para estimar a relação sinal-interferência-mais-ruído ou *Signal-to-Interference plus Noise Ratio* (SINR) experimentada.

Outras informações importantes, especialmente em ambientes MIMO, são o *Precoding Matrix Indicator* (PMI) e o *Rank Indicator* (RI). Com múltiplas antenas de transmissão e recepção faz-se necessário realizar a multiplexação espacial do sinal. Desta forma, os sinais referentes às antenas são organizados em camadas. O RI indica a quantidade de camadas distinguíveis por um UE, e o PMI indica os parâmetros de precodificação utilizados para equalizar as camadas.

O principal serviço realizado em nível de camada física é o *Radio Resource Management* (RRM). O RRM, geralmente realizado na eNB, trata do gerenciamento dos mecanismos e recursos de rádio do LTE-A, e também pode ser considerado como uma arquitetura composta por vários tipos de serviços, entre os principais:

- adaptação do enlace;
- controle e alocação de potência;
- ajuste adaptativo de modulação, também conhecido como *Adaptive Modulation and Coding* (AMC);
- escalonamento;
- controle de admissão de chamadas ou *Call Admission Control* (CAC);
- gerenciamento de qualidade de serviço (QoS);

- controle e coordenação de interferência entre células ou *Inter-Cell Interference Coordination* (ICIC);
- coordenação avançada de transmissão entre múltiplas estações ou *Coordinated Multi-Point* (CoMP);
- controle de mobilidade e *handover*;
- controle de congestionamento e balanceamento de carga;
- gerenciamento de medições da camada física;
- gerenciamento de retransmissões com o mecanismo *Hybrid Automatic Repeat Request* (HARQ);
- encaminhamento do CQI;
- redução de sinalização para economia de energia por meio da recepção descontínua ou *Discontinuous Reception* (DRX).

O RRM demanda algumas medições importantes para sua operacionalização:

- ***Received Signal Strength Indicator* (RSSI):** medida de potência de um sinal recebido, somado ao ruído percebido, sobre toda a largura de banda utilizada;
- ***Reference Signal Received Power* (RSRP):** potência média do sinal considerando os valores de referência para cada subportadora;
- ***Reference Signal Received Quality* (RSRQ):** estabelece um valor de referência para indicação da qualidade do sinal, baseado no RSRP e o RSSI.

Em termos gerais, o RRM encarrega-se do transporte físico dos dados entre os dispositivos da E-UTRAN. Também compreende algumas funções na camada de enlace e é importante no ajuste de parâmetros relacionados ao controle de mobilidade, potência de transmissão, alocação de recursos, esquemas de modulação, etc.

2.2.3 Camadas de Enlace e de Rede

Os dados recebidos pela camada física são encapsulados em *Packet Data Units* (PDUs), e encaminhados para a camada de enlace do sistema. Nos sistemas UMTS e E-UTRAN (LTE-A) a camada de enlace corresponde ao *Access Stratum* (AS).

O AS é um agrupamento funcional responsável pelo gerenciamento dos recursos de rádio e o transporte de dados entre os terminais e a estação base da E-UTRAN [68]. Além disso, o AS também pode ser organizado em dois planos: controle e usuário. O plano de controle é

responsável pelo gerenciamento de conectividade dos UEs com a rede (sinalização), enquanto o plano de usuário conduz os dados dos usuários. O AS pertence ao escopo da E-UTRAN apenas.

Os protocolos embutidos na camada AS são: o *Radio Resource Control* (RRC); o *Packet Data Convergence Protocol* (PDCP); o *Radio Link Control* (RLC); e o *Medium Access Control* (MAC). No plano de usuário são utilizados apenas os protocolos PDCP, RLC e MAC. No plano de controle são utilizados todos os protocolos também implementados para plano de usuário e o RRC [69].

A Figura 2.6 ilustra a pilha de protocolos adotada para o plano de usuário, enquanto a Figura 2.7 ilustra o modelo adotado para o plano de controle no AS, além de parte da pilha de protocolos para plano de controle na EPC. A camada NAS, descrita a seguir, também é apresentada na ilustração.

O canal de comunicação intercamadas ocorre por meio dos *Service Access Points* (SAPs). Normalmente, existem dois SAPs entre cada camada do sistema para encaminhamento de dados das camadas superiores para as inferiores e vice-versa.

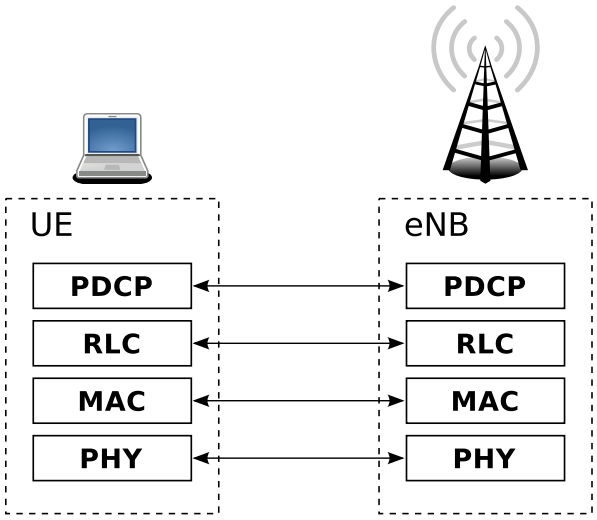


Figura 2.6: Pilha de protocolos do plano de usuário *Access Stratum*. Extraído e adaptado de [69].

A camada RRC é a camada mais alta do AS. Suas principais funcionalidades são [69]: difusão de informações do sistema (*broadcast*); estabelecimento e manutenção de conexão entre o UE e a E-UTRAN; gerenciamento de qualidade do enlace da conexão de rádio; estabelecimento, configuração e liberação de *bearers* no plano do usuário. O conceito de *bearers* será descrito mais adiante.

Na camada PDCP são realizadas as funções necessárias para suporte à RRC e camadas superiores. Realiza-se na PDCP a transferência dos dados dos planos de controle e de usuário, a compressão de cabeçalhos, proteção e integridade de dados, criptografia, entre outras funções.

Para gerenciamento do enlace de rádio a camada RLC encarrega-se da concatenação e segmentação das mensagens (pacotes), e procedimentos para retransmissão e detecção de duplici-

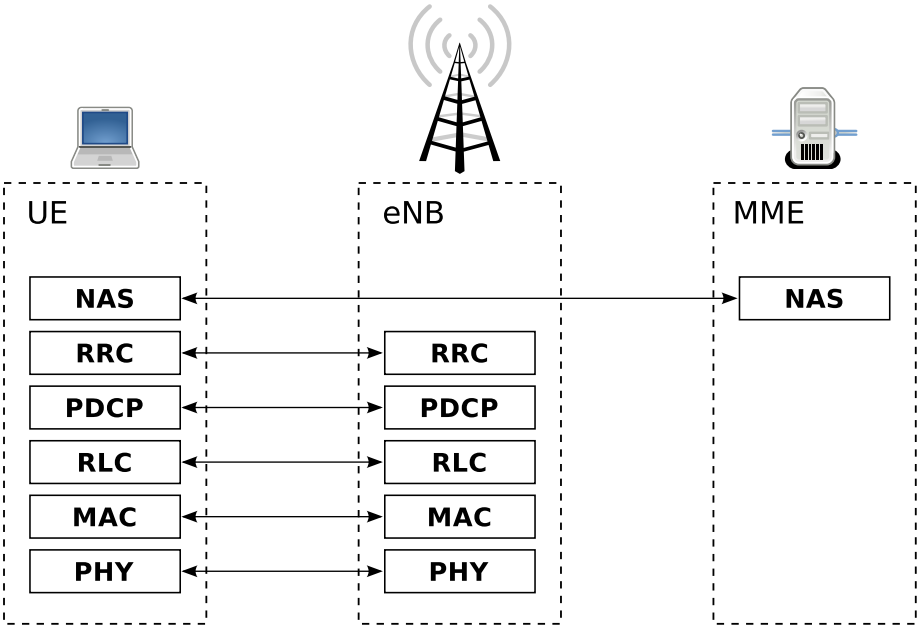


Figura 2.7: Pilha de protocolos do plano de controle do *Access Stratum* (escopo da E-UTRAN) e pilha de protocolos do plano de controle da EPC entre eNB e MME. Extraído e adaptado de [69].

dades [65].

A camada MAC é responsável em estabelecer o controle de acesso ao meio físico, a definição de prioridades no acesso aos canais lógicos, o compartilhamento e controle de alocação de recursos entre os usuários (escalonamento), o mecanismo de repetição de retransmissões de dados de sinalização sem redundância e correção de erros por meio do HARQ, o encaminhamento de informações sobre o escalonamento, entre diversas outras funções.

São estabelecidos dois canais principais para transporte de dados na camada MAC: o *Downlink Shared Channel* (DL-SCH) e o *Uplink Shared Channel* (UL-SCH). Esses são transportados em unidades de informação definidos entre a camada MAC e PHY conhecidas como *Transport Block* (TB). O tamanho do TB, ou *Transport Block Size* (TBS), é definido a partir da quantidade de RBs utilizados e também do Esquema de Modulação e Codificação ou *Modulation and Codification Scheme* (MCS) adotado para transmissão (*downlink*) ou recepção (*uplink*). O DL-SCH e o UL-SCH comunicam-se, respectivamente, com os canais da camada física: PDSCH e PUSCH.

As Figuras 2.8 e 2.9 apresentam, respectivamente, o mapeamento de canais de comunicação do LTE-A (físico, transporte e lógico) correspondente às três primeiras camadas do AS (PHY, MAC e RLC) para o *downlink* e o *uplink*. Os canais são classificados por camadas e as setas indicam o esquema de encapsulamento e comunicação dos canais das camadas superiores com as inferiores. Neste capítulo são descritos apenas os canais mais importantes para compreensão dos fundamentos do LTE-A. Para mais detalhes, recomenda-se a leitura do material em [65].

A camada de rede do LTE-A corresponde à camada *Non Access Stratum* (NAS). O NAS

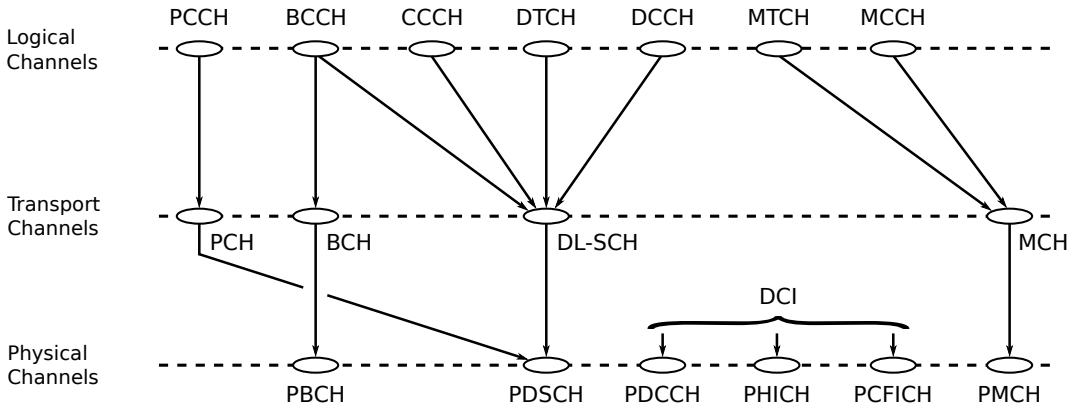


Figura 2.8: Mapeamento de canais no *downlink*. Extraído de [65].

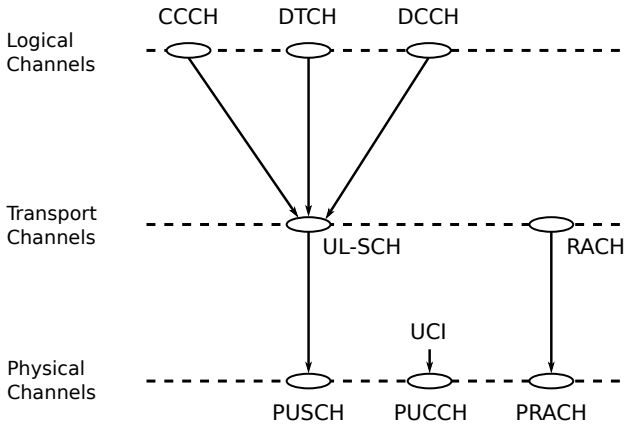


Figura 2.9: Mapeamento de canais no *uplink*. Extraído de [65].

gerencia operações de suporte a mobilidade, procedimentos para gerenciamento de sessão e manutenção de conectividade do protocolo IP entre os UEs com o P-GW [70]. Resumidamente, o NAS atua como elo funcional entre a rede de núcleo (EPC) e a rede de acesso *wireless* (E-UTRAN).

Com relação à arquitetura de protocolos adotada nas *interfaces* da EPC, são utilizados diversos modelos. Cada modelo apresenta uma pilha diferente de protocolos, sendo que cada protocolo é implementado para suprir algumas funcionalidades específicas no sistema. As pilhas utilizadas na EPC são todas orientadas ao protocolo IP e seus principais protocolos são:

- GTP, responsável pelo tunelamento de dados nos planos de usuário e controle;
- *Mobile IPv4* (MIPv4) ou *Mobile IPv6* (MIPv6), dependendo da implementação desejada;
- *Dual-Stack Mobile IPv4* (DSMIPv4) ou *Dual-Stack Mobile IPv6* (DSMIPv6), permitindo utilização simultânea das versões IPv4 e IPv6 na pilha de protocolos quando necessário;
- *Proxy Mobile IPv4* (PMIPv4) ou *Proxy Mobile IPv6* (PMIPv6), encarregado de possibilitar mobilidade aos UEs mantendo o endereçamento de rede.

Não faz parte do escopo deste trabalho abordar todo o aspecto de implementação dos protocolos utilizados na EPC. Mais detalhes sobre a implementação de pilha de protocolos na EPC podem ser encontrados em [71].

2.3 Qualidade de Serviço

A qualidade de serviço (QoS) é definida como a capacidade da rede em transmitir dados assegurando parâmetros quantitativamente mensuráveis (métricas) em relação à demanda de uma determinada aplicação (serviço) [72]. As medidas de desempenho normalmente consideradas são: vazão, taxa de erros, atraso de transmissão, variação do atraso (*jitter*), disponibilidade e taxa de perdas.

Diversos mecanismos podem ser aplicados para assegurar QoS: gerenciamento de filas, moldagem de tráfego, escalonamento, etc. O mecanismo mais comum é a priorização de tráfego, onde um tipo de serviço tem preferência sobre outro no acesso aos recursos da rede. Adicionalmente são estabelecidos os Acordos de Níveis de Serviço, ou *Service Level Agreement* (SLA), que definem os requisitos mínimos necessários para atendimento aos serviços fornecidos na rede. Apesar disso, a forma mais prática ainda é a prevenção e o controle de congestionamento.

Cabe destacar que, na garantia de QoS, o escalonador exerce fundamental influência na prevenção e controle de congestionamento na rede, estabelecendo a ordem e a quantidade de recursos a serem disponibilizados para um determinado serviço.

Em sistemas LTE-A, o suporte a QoS é uma característica essencial, devendo ser proporcionado fim a fim, ou seja, tanto no escopo da EPC como na E-UTRAN. Para o caso da E-UTRAN, o mecanismo de priorização de tráfego pode ser aplicado por meio de um parâmetro do sistema conhecido como *QoS Class Identifier* (QCI), que define a ordem ideal de prioridade de acordo com o tipo de serviço e seus requisitos mínimos.

Além do QCI, há também ainda outros parâmetros aplicados para oferecer suporte a QoS em sistemas LTE-A:

- ***Allocation and Retention Priority* (ARP):** parâmetro utilizado para definição de prioridade no estabelecimento de novos SLAs para um serviço;
- ***Guaranteed Bit Rate* (GBR):** parâmetro que estabelece uma taxa mínima garantida para o tráfego associado;
- ***Maximum Bit Rate* (MBR):** indica a taxa máxima permitida para o tráfego associado, assegurando que não sejam alocados mais recursos do que o necessário;
- ***Per APN Aggregate Maximum Bit Rate* (APN-AMBR):** indica a taxa máxima agregada permitida para um APN;
- ***Per UE Aggregate Maximum Bit Rate* (UE-AMBR):** indica a taxa máxima agregada permitida para um UE.

A Tabela 2.2, definida pelo 3GPP em [15], apresenta os valores de QCI para cada tipo de serviço, sua prioridade, requisitos necessários e exemplos de aplicações que atendem os parâmetros estabelecidos. É importante mencionar que a referida tabela corresponde aos serviços descritos até a *Release 15* da especificação.

Tabela 2.2: Valores padronizados de QCI para mapeamento de características dos serviços. Extraído e traduzido de [16].

QCI	**Tipo de Recurso**	**Taxa de erro**	**Limite de atraso (ms)**	**Prioridade do QCI**	**Exemplo de serviço**
1	GBR	10^{-2}	100	2	Voz em tempo real
2	GBR	10^{-3}	150	4	Vídeo em tempo real
3	GBR	10^{-3}	50	3	Jogos em tempo real
4	GBR	10^{-6}	300	5	Vídeo armazenado
65	GBR	10^{-2}	75	0.7	Plano de usuário p/ serviço *Push to Talk* (PTT) do tipo *Mission Critical*[1]
66	GBR	10^{-2}	100	2	Plano de usuário p/ serviço PTT do tipo *Non-Mission Critical*
75	GBR	10^{-2}	50	2.5	Mensagens do tipo V2X[2]
5	*Non*-GBR	10^{-6}	100	1	Sinalização
6	*Non*-GBR	10^{-6}	300	6	Vídeo armazenado e transferência de arquivos sobre TCP
7	*Non*-GBR	10^{-3}	100	7	Voz, vídeo e jogos em tempo real
8	*Non*-GBR	10^{-6}	300	8	Vídeo armazenado e transferência de arquivos sobre TCP
9	*Non*-GBR	10^{-6}	300	9	Vídeo armazenado e transferência de arquivos sobre TCP. *Bearer* padrão
69	*Non*-GBR	10^{-6}	60	0.5	Sinalização p/ serviço PTT do tipo *Mission Critical*
70	*Non*-GBR	10^{-6}	200	5.5	Dados do tipo *Mission Critical*
79	*Non*-GBR	10^{-2}	50	6.5	Mensagens do tipo V2X

Para realizar o tratamento de serviço e o mapeamento dos parâmetros descritos a fim de oferecer suporte a QoS durante o transporte dos dados, o LTE-A implementa uma ferramenta denominada *bearer*, abordada na próxima subseção.

[1] O *Mission Critical Push to Talk* (MCPTT) é uma funcionalidade inserida a partir da *Release 13* do 3GPP que possibilita o uso de sistemas LTE para serviços de emergência, alerta e segurança pública [73].

[2] V2X – *Vehicle to Everything* – Trata-se de um tipo de sistema de comunicação veicular.

2.3.1 *Bearers*

Bearer é um *pipeline* ou "duto lógico" que conecta dois ou mais pontos de comunicação, ao qual são vinculados os parâmetros necessários para garantia de QoS (QCI, ARP, GBR e MBR, por exemplo). Todo fluxo de tráfego existente na EPS deve ser transportado por meio de uma *bearer*, sendo que um UE pode possuir diversas *bearers* associadas.

No LTE-A uma *bearer* pode ser classificada por tipo e escopo. Existem dois tipos de *bearers*: padrão e dedicado. Com relação ao escopo a *bearer* pode ser classificada em [69]:

- ***Radio Bearer*:** utilizada para estabelecer a comunicação entre a eNB e um UE no escopo da E-UTRAN. É denominada *Signaling Radio Bearer* (SRB), se for utilizada para o plano de controle, e *Data Radio Bearer* (DRB), se for para o plano de usuário [74];
- ***S1 Bearer*:** transporta dados entre a eNB e a EPC até o S-GW, no canal S1-U;
- ***S5/S8 Bearer*:** estabelece comunicação entre o S-GW e o P-GW;
- ***Evolved Radio Access Bearer* (E-RAB):** conhecida como uma junção da *Radio Bearer* e a *S1 Bearer*. Utilizada apenas como forma de referir-se ao escopo das duas *bearers* que a compõe;
- ***EPS Bearer*:** compreende o escopo mais amplo de uma *bearer* no sistema, abrangendo a E-UTRAN e a EPC até o P-GW. A *EPS Bearer* é composta de todas as outras *bearers* mencionadas anteriormente (*Radio Bearer*, *S1 Bearer* e *S5/S8 Bearer*).

A Figura 2.10 ilustra o escopo das *bearers* no LTE-A, auxiliando a compreensão da abrangência de cada uma. Para simplicidade e foco no tema deste trabalho, o texto refere-se apenas a *bearer* daqui em diante, independente de seu escopo.

Para fins de controle de QoS toda *bearer* dedicada deve, obrigatoriamente, possuir um *Traffic Flow Template* (TFT) associado. O TFT é utilizado para identificar a *bearer*, mapear informações necessárias para transporte dos dados e realização da filtragem de pacotes ou *packet filtering*. O TFT também é utilizado para conduzir o *Tunneling End ID* (TEID), usado para identificar as *bearers* em cada nível no escopo. A Figura 2.11 ilustra o tunelamento de *bearers*, identificadas pelos seus respectivos TEIDs, com o TFT no canal associado (DL ou UL).

O TFT vincula a *bearer* a um (ou mais) *Service Data Flow* (SDF), utilizado para representar um ou mais fluxos de tráfego (pacotes IP) relacionados a um serviço de usuário (aplicação): navegação *Web*, *e-mail*, vídeo, etc. No LTE-A, os operadores de rede definem as políticas de delimitação dos SDFs por meio de um TFT, associado a uma ou mais *bearers* específicas, contendo informações de mapeamento e filtragem de pacotes [74]. Um TFT pode possuir diversos parâmetros, entretanto os parâmetros mínimos obrigatórios são:

- Endereço IP de origem;
- Endereço IP de destino;

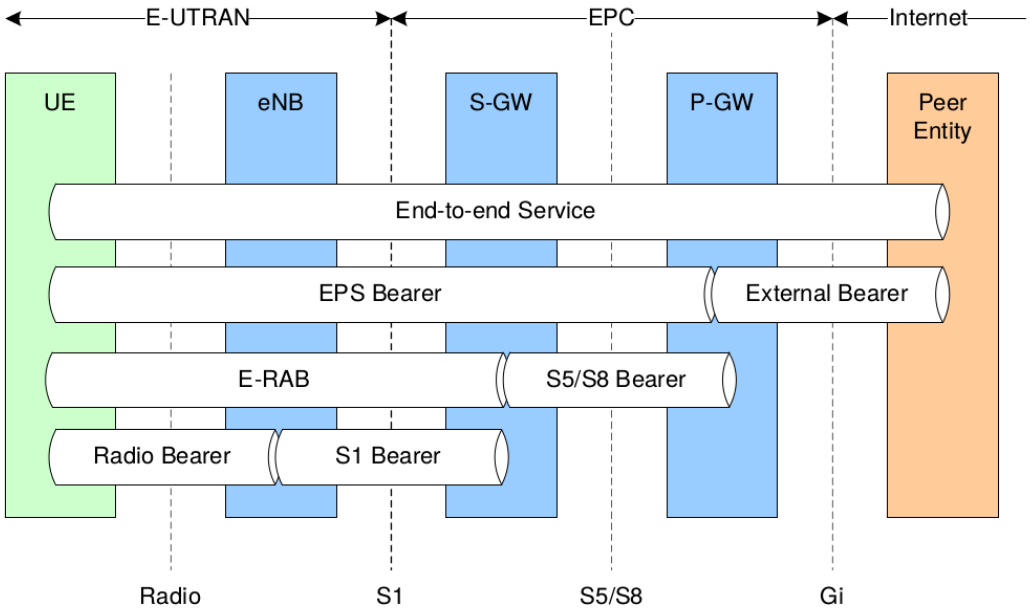


Figura 2.10: Arquitetura de serviço da *EPS Bearer*. Extraído de [75].

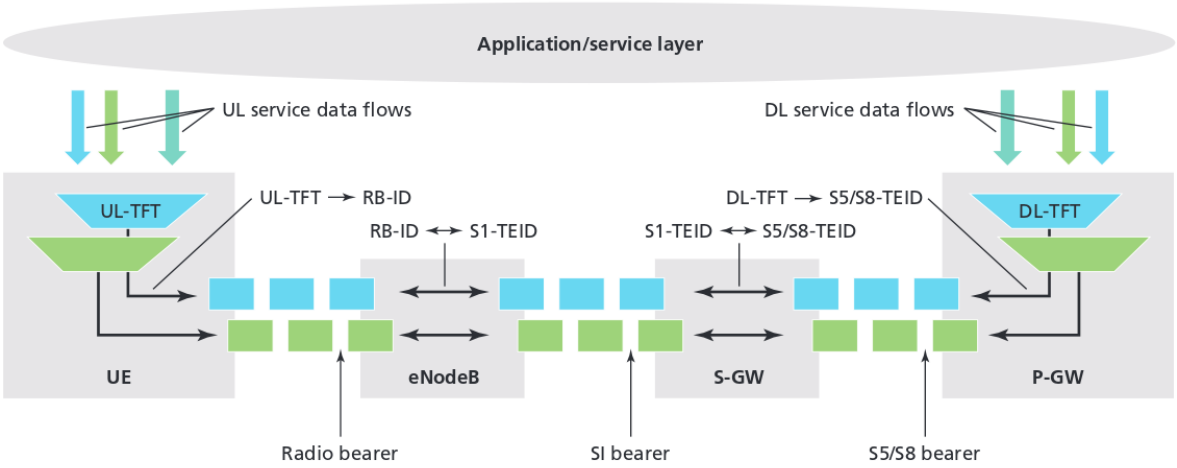


Figura 2.11: *EPS Bearer* no LTE-A/SAE entre as diferentes *interfaces*. Extraído de [76].

- Número de porta de origem;
- Número de porta de destino;
- Identificação do protocolo de transporte: TCP ou UDP.

Outros parâmetros opcionais podem ser acrescentados, como o campo para tipo de serviço ou *Type of Service* (TOS), por exemplo, ou qualquer outro que possa ser incluído em cabeçalhos de pacotes do protocolo IP e esteja em conformidade com os padrões e normas do *Internet Engineering Task Force* (IETF).

O estabelecimento de uma *bearer* ocorre de formas distintas para os tipos padrão e dedicado.

Quando um UE realiza pareamento com a eNB, uma *bearer* padrão é automaticamente criada para aquele UE no sistema. Isso ocorre na operação conhecida como *Initial Context Setup Request* dentro do processo de gerenciamento de sessão do EPS, responsável pela configuração da requisição do UE para pareamento [71].

A *bearer* padrão permanece ativa enquanto houver conexão do UE com a PDN e há apenas uma única *bearer* padrão por UE no sistema. Os parâmetros de QoS associados à *bearer* padrão são básicos e atribuídos pela rede baseado nas informações de assinatura ou contrato do usuário com o provedor de serviço [74]. Toda *bearer* padrão é do tipo *Non*-GBR, e por esse motivo, assume-se que o canal de dados proporcionado por uma *bearer* padrão opere na estratégia de "melhor esforço", ou *Best Effort* (BE).

A *bearer* dedicada é criada sob demanda, de acordo com os requisitos de QoS da aplicação, podendo ser do tipo GBR ou *Non*-GBR. Diferentemente da *bearer* padrão, a dedicada pode ser desativada a qualquer momento ainda que haja conexão do UE com a PDN. A sinalização para criação da *bearer* dedicada origina-se no P-GW, que recebe os parâmetros de QoS gerados e encaminhados pelo PCRF.

As *bearers* são ferramentas importantes para o escalonamento no sistema. Elas viabilizam, por meio do TFT, informação sobre o fluxo de dados das aplicações, auxiliando o provisionamento de QoS em nível mais alto de abstração.

2.4 Considerações sobre o Capítulo 2

Este capítulo apresentou os principais conceitos relacionados à tecnologia LTE-A. Abordou-se o histórico até a concepção do LTE-A e o contexto atual com a chegada de novas tecnologias pertencentes à 5G com menção a ações de órgãos, como o ITU-R e NGMN, no sentido de desenvolver e padronizar a 5G até 2020. Uma parte da especificação do LTE-A foi apresentada brevemente com intuito de proporcionar os fundamentos da especificação, arquitetura do sistema e camadas. Aspectos de qualidade de serviço (QoS) inerentes ao sistema foram descritos incluindo a conceituação e o suporte a *bearers*, fundamentais para transportar os parâmetros necessários para assegurar QoS aos tráfegos da rede.

O LTE-A será, muito provavelmente, predecessor da próxima tecnologia que deverá compor a maior parte da 5G. Compreender os aspectos básicos da tecnologia LTE-A, bem como um pouco de sua evolução, é fundamental para o desenvolvimento de novas tecnologias e produtos. Além disso, os conceitos apresentados neste capítulo fornecem um embasamento teórico necessário para prosseguimento na compreensão da proposta deste trabalho.

Capítulo 3

Aprendizado de Máquina – Conceitos, Modelos e Algoritmos

A Aprendizagem Computacional ou Aprendizado de Máquina (*Machine Learning*) é uma das diversas áreas do conhecimento que compõem a Inteligência Artificial (IA) dentro da Ciência da Computação. Duas definições mais populares sobre Aprendizado de Máquina são apresentadas a seguir.

A primeira definição, mais clara e objetiva para Aprendizado de Máquina, de acordo com Arthur Samuel [77], é "a área de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados".

Outra definição, mais moderna e completa, apresentada por Tom Mitchell [78], sobre Aprendizado de Máquina, afirma "que um programa de computador aprende em uma experiência $\boldsymbol{\mathcal{E}}$ a partir de algum conjunto de tarefas $\boldsymbol{\mathcal{T}}$ com medida de desempenho $\boldsymbol{\mathcal{P}}$, se seu desempenho em tarefas $\boldsymbol{\mathcal{T}}$ medido por $\boldsymbol{\mathcal{P}}$ melhora com a experiência $\boldsymbol{\mathcal{E}}$".

Basicamente, o Aprendizado de Máquina trata do aperfeiçoamento de habilidades executadas de forma autônoma por algum algoritmo computacional [79]. Para tal, assume-se que tanto a execução das habilidades como seu aperfeiçoamento sejam "guiados" por um ou mais conjuntos de dados. O conceito de aprendizagem a partir da experiência também pode ser substituído, sem prejuízo das definições iniciais, pela aprendizagem a partir de dados.

O Aprendizado de Máquina trata portanto em tornar computadores capazes de modificarem ou adaptarem suas ações de maneira que estas tornem-se cada vez mais precisas, onde a precisão é considerada como a medida do quão bem as ações são realizadas de forma correta [80].

O tipo de habilidade a ser empregada caracteriza o método de Aprendizado de Máquina a ser utilizado. A saber existem três métodos ou categorias principais:

- **Aprendizado Supervisionado;**
- **Aprendizado Não-supervisionado;**
- **Aprendizado por Reforço.**

Para efeito de delimitação do escopo não serão abordados conceitos, modelos, algoritmos ou aplicações sobre *Deep Learning* ou Aprendizado Profundo [81], embora este ramo de estudo é digno de menção, visto que vem se desenvolvendo rapidamente e suas aplicações mais recentes têm produzido resultados significativos [82–84]. Tecnicamente o *Deep Learning* baseia-se em pelo menos um dos princípios das três categorias principais de Aprendizado de Máquina. Sugere-se a leitura do trabalho em [85] como referência inicial acerca dos conceitos do *Deep Learning* e sua aplicação em redes móveis de banda larga sem fio.

As seções a seguir apresentam os fundamentos de cada uma das categorias de Aprendizado de Máquina acima listadas.

3.1 Aprendizado Supervisionado

Entende-se por Aprendizado Supervisionado a aprendizagem a partir de exemplos. Neste caso, é fornecido ao computador um conjunto de dados acompanhado das respostas correspondentes e deseja-se, a partir dos dados informados, condicionar o algoritmo a responder corretamente todas as possíveis novas entradas. O "condicionamento do algoritmo" normalmente é conhecido como generalização, e obtida após a etapa de treinamento.

O modelo pode ser representado por um conjunto de dados $\boldsymbol{X}$ composto por variáveis "de entrada" denotadas por x_i, em que [86]:

$$\boldsymbol{X} = (x_1, x_2, \ldots, x_n)^{\dagger}, \quad i \in \{1, 2, \ldots, n\}. \quad \boldsymbol{X} \in \mathbb{R}^n \tag{3.1}$$

Sendo que n indica a quantidade de elementos ou amostras do conjunto $\boldsymbol{X}$. A resposta correspondente equivale ao conjunto $\boldsymbol{Y}$, composto por variáveis "de saída" denotadas por y_i e conhecidas como variáveis objetivo ou simplesmente *target*. Há uma correspondência sobrejetiva dos elementos do conjunto $\boldsymbol{X}$ para os do conjunto $\boldsymbol{Y}$ normalmente representada pelo par (x_i,y_i), denominado como amostra de treinamento.

As variáveis de entrada e saída também podem ser representadas na forma de vetores:

$$\vec{\boldsymbol{x}}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,m}), \quad \vec{\boldsymbol{y}}_i = (y_{i,1}, y_{i,2}, \ldots, y_{i,p}) \tag{3.2}$$

Onde m indica a quantidade de atributos ou características analisadas nos conjuntos e p indica a quantidade de variáveis ou parâmetros de saída.

O objetivo é aprender uma função $h_{\boldsymbol{\theta}}(\boldsymbol{x})$, conhecida como função *hipótese*, tal que $h_{\boldsymbol{\theta}}(\boldsymbol{x}) : \boldsymbol{X} \rightarrow \boldsymbol{Y}$, ou seja, dado um novo valor $\boldsymbol{x}$, onde $\boldsymbol{x} \notin \boldsymbol{X}$, busca-se encontrar uma hipótese que preveja uma saída $\boldsymbol{y}$, onde $\boldsymbol{y} \in \boldsymbol{Y}$, obedecendo a correspondência entre os conjuntos $\boldsymbol{X}$ e $\boldsymbol{Y}$ informados.

Para encontrar os parâmetros do vetor $\boldsymbol{\theta}$ de uma função hipótese que generalize o modelo, é necessário minimizar uma função de custo $J(\boldsymbol{\theta})$ tal que:

$$\arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \tag{3.3}$$

A equação utilizada para a função de custo $J(\boldsymbol{\theta})$ depende do método e da aplicação utilizados para generalização do modelo. Neste caso, a função de custo é diferente para aplicações de regressão linear e não-linear por exemplo, entretanto o objetivo é sempre minimizar $J(\boldsymbol{\theta})$.

A Figura 3.1 ilustra a representação do modelo descrito de aprendizado supervisionado para um caso genérico.

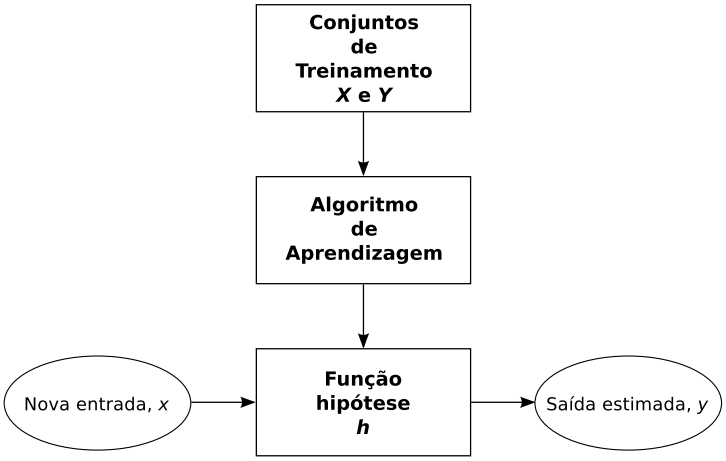


Figura 3.1: Modelo genérico de aprendizado supervisionado. Extraído, traduzido e adaptado de [86].

3.2 Aprendizado Não-supervisionado

O Aprendizado Não-supervisionado caracteriza-se basicamente pela realização de inferências sobre um conjunto de dados de maneira inteiramente autônoma e sem qualquer tipo de informação adicional. As inferências são aplicadas com a finalidade de descobrir novas estruturas ou características sobre os dados, sendo utilizadas, principalmente, para classificação, representação da informação, codificação/decodificação e redução de dimensionalidade.

Para fins de delimitação de escopo, esta seção mantém enfoque apenas na classificação de dados, de maneira que estes não possuem nenhum tipo de rotulação ou mapeamento a outros conjuntos. O enfoque de Aprendizado Não-supervisionado voltado para codificação/decodificação e redução de dimensionalidade é melhor explorado na Seção 4.2.2.

O objetivo do Aprendizado Não-supervisionado na classificação é encontrar padrões existentes nos dados apresentados, possibilitando a representação organizada de tal informação na forma de agrupamentos. A maneira mais comum para encontrar padrões é utilizando medidas de similaridade ou dissimilaridade entre atributos de cada um dos elementos ou amostras do conjunto de dados.

A produção de agrupamentos a partir da classificação não-supervisionada também é conhecida como *clustering*, e realizada quando não há informação de pertinência dos elementos a categorias [87], isto é, quando não há conjunto correspondente que defina qualquer representação dos dados informados.

O modelo do *clustering* é representado por um conjunto de dados $\boldsymbol{X}$ informado em que o objetivo é particionar $\boldsymbol{X}$ em k agrupamentos ou *clusters* $\boldsymbol{\mathcal{C}}_i$. As seguintes condições devem ser satisfeitas:

$$i \in \{1, 2, \ldots, k\} \tag{3.4}$$

- Todo *cluster* deve possuir pelo menos um único elemento do conjunto $\boldsymbol{X}$:

$$\boldsymbol{\mathcal{C}}_i \neq \emptyset \tag{3.5}$$

- A união de todos os *clusters* deve ser igual ao conjunto $\boldsymbol{X}$:

$$\boldsymbol{\mathcal{C}}_1 \cup \boldsymbol{\mathcal{C}}_2 \cup \cdots \cup \boldsymbol{\mathcal{C}}_k = \boldsymbol{X} \tag{3.6}$$

- Dois *clusters* não podem conter o mesmo elemento:

$$\boldsymbol{\mathcal{C}}_i \cap \boldsymbol{\mathcal{C}}_j = \emptyset, \quad i \in \{1, 2, \ldots, k\}\,. \quad j \in \{1, 2, \ldots, k\}\,. \quad i \neq j. \tag{3.7}$$

O agrupamento dos elementos considera suas características em comum. Para isso, utiliza-se alguma medida de similaridade $d(\boldsymbol{x}_i, \boldsymbol{\mu}_j)$ entre os elementos analisados e os centroides, sendo o centroide $\boldsymbol{\mu}_j$ um elemento, pertencente ou não ao conjunto $\boldsymbol{X}$ (dependendo do algoritmo empregado), que representa as características de um *cluster* $\boldsymbol{\mathcal{C}}_j$.

Existem diversas medidas de similaridade, as mais comuns adotadas no Aprendizado Não-supervisionado são [88]:

- Distância Euclidiana:

$$d(\boldsymbol{x}_i, \boldsymbol{\mu}_j) = \sqrt{\sum_{l=1}^{m} (x_{i,l} - \mu_{j,l})^2} \tag{3.8}$$

- Distância *Mahalanobis*:

$$d(\boldsymbol{x}_i, \boldsymbol{\mu}_j) = \sqrt{\sum_{l=1}^{m} \frac{(x_{i,l} - \mu_{j,l})}{s_l^2}^2} \tag{3.9}$$

  Em que s_l^2 é o desvio padrão do l-ésimo atributo em todo o conjunto $\boldsymbol{X}$.

- *Manhattan* ou *Taxicab*:

$$d(\boldsymbol{x}_i, \boldsymbol{\mu}_j) = \sum_{l=1}^{m} |x_{i,l} - \mu_{j,l}| \tag{3.10}$$

- Distância *Minkowski*:

$$d(\boldsymbol{x}_i, \boldsymbol{\mu}_j) = \left(\sum_{l=1}^{m} |x_{i,l} - \mu_{j,l}|^p \right)^{\frac{1}{p}} \tag{3.11}$$

  Que acrescenta o parâmetro p equivalente à Distância *Manhattan*, se $p = 1$, ou Euclidiana se $p = 2$.

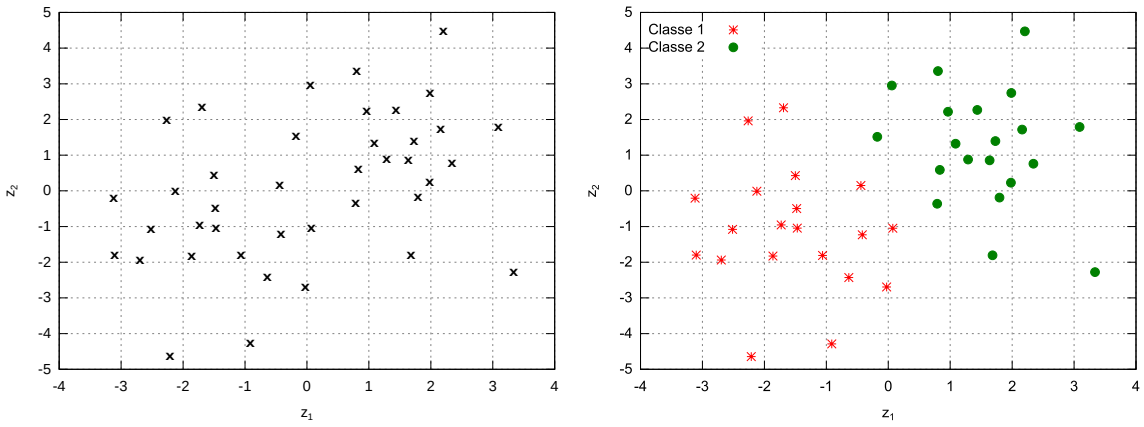
- Similaridade de Cossenos:

$$d(\boldsymbol{x}_i, \boldsymbol{\mu}_j) = \frac{\boldsymbol{x}_i \cdot \boldsymbol{\mu}_j}{\|\boldsymbol{x}_i\| \left\|\boldsymbol{\mu}_j\right\|} = \frac{\sum_{l=1}^{m} x_{i,l}\mu_{j,l}}{\sqrt{\sum_{l=1}^{m} x_{i,l}^2}\sqrt{\sum_{l=1}^{m} \mu_{j,l}^2}} \tag{3.12}$$

$$i \in \{1, 2, \ldots, n\}\,. \quad j \in \{1, 2, \ldots, k\}\,. \tag{3.13}$$

Sendo n a quantidade de elementos do conjunto $\boldsymbol{X}$, m é a quantidade de atributos ou características analisadas no conjunto e k é a quantidade estabelecida de agrupamentos.

A Figura 3.2 representa um conjunto de dados classificado por meio de um algoritmo de Aprendizado Não-supervisionado. São definidos dois agrupamentos (classes) no exemplo apresentado. Cada elemento do conjunto possui dois atributos z_1 e z_2.



(a) Sem classificação (b) Com classificação

Figura 3.2: Exemplo de um conjunto de dados classificado por meio de algum algoritmo de Aprendizado Não-supervisionado.

3.3 Aprendizado por Reforço

As abordagens realizadas nos processos de Aprendizado Supervisionado e Não-supervisionado são muito específicas e, em alguns casos, sofrem com algumas limitações. O Aprendizado Supervisionado, por exemplo, não é muito adequado para aplicações interativas se for utilizado isoladamente. Geralmente, a obtenção de exemplos a partir de um comportamento desejado

é impraticável do ponto de vista de representação de todas as ações que o computador possa realizar [89].

Considerando o problema de representação de ações, a ausência de exemplos torna-se um ponto positivo no Aprendizado por Reforço e que assemelha-se bastante à abordagem Não-supervisionada. Entretanto, no Aprendizado Não-supervisionado, um problema comum é a ausência de otimização dos resultados produzidos pelo algoritmo. Visto que não há resposta a uma ação realizada, informando se houve ou não sucesso, alguns algoritmos de Aprendizado Não-supervisionado podem não alcançar resultados ideais dependendo da forma como a informação de entrada está estruturada.

O Aprendizado por Reforço aborda um problema enfrentado por um algoritmo de aprendizagem, denominado agente, que busca aprender a realização de uma atividade orientando-se por respostas produzidas em um ambiente dinâmico [90]. O agente interage com o ambiente, modificando-o por meio das ações realizadas, possuindo objetivos claros e bem definidos. O agente interage continuamente com o ambiente, sendo que esta interação pode ser em um horizonte de tempo finito, infinito ou indeterminado.

O problema pode ser representado, tradicionalmente, da forma a seguir [89]:

- Um agente interage com o ambiente em uma sequência de fases discretizadas no domínio do tempo $t \in \{0, 1, 2, 3, \dots\}$;
- O agente recebe a representação de um estado $s_t \in \boldsymbol{\mathcal{S}}$ sendo $\boldsymbol{\mathcal{S}}$ é o conjunto de todos os estados possíveis;
- O agente realiza uma ação $a_t \in \boldsymbol{\mathcal{A}}(s_t)$ sendo $\boldsymbol{\mathcal{A}}(s_t)$ o conjunto de todas as ações possíveis a partir do estado s_t;
- Um valor de recompensa $r_{t+1} \in \mathbb{R}$ é devolvido pelo ambiente ao agente no próximo instante de tempo $t + 1$ e o agente assume o próximo estado s_{t+1};
- A política do agente, denotada por π_t, estabelece o comportamento do agente, em termos numéricos, em relação à escolha das ações (decisão) a serem realizadas para transição entre estados.

A Figura 3.3 mostra o diagrama do modelo de Aprendizado por Reforço apresentado.

Apesar da representação formal anterior, o modelo de Aprendizado por Reforço apresentado pode ser adaptado de diversas formas. A sequência de fases para realização de uma ação, por exemplo, não necessita ser precisamente em unidades no domínio do tempo ou em períodos fixos e discretizados. A representação por estados também pode ser ajustada. De fato, há várias adaptações que podem ser realizadadas de acordo com o modelo ou o algoritmo desenvolvido.

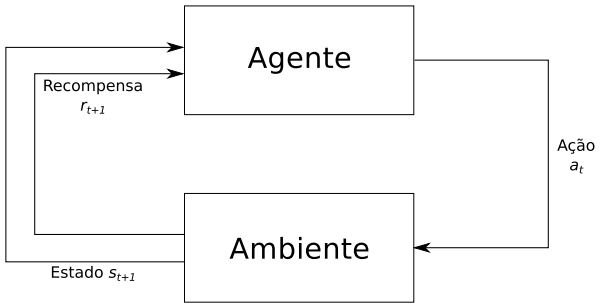


Figura 3.3: Modelo de aprendizado por reforço. Extraído, traduzido e adaptado de [89].

3.4 Modelos e Algoritmos de Aprendizado de Máquina

Esta seção dedica-se à apresentação dos modelos e algoritmos de Aprendizado de Máquina mais importantes encontrados em cada categoria. O escopo desta seção abrange os conceitos básicos visando apenas facilitar a introdução do leitor ao tema abordado. Alguns desses modelos e algoritmos foram adotados e implementados na proposta deste trabalho.

3.4.1 Descida de Gradiente ou Método do Gradiente

Em problemas de regressão linear ou regressão logística o método mais comum para encontrar os parâmetros de uma função hipótese para generalização do modelo é o Método do Gradiente ou Descida de Gradiente. Este algoritmo caracteriza-se pela estimativa de parâmetros da função hipótese que minimizem a função de custo ou aproximem-se de seu valor mínimo global. Pode ser empregado na solução de problemas de Aprendizado Supervisionado ou por Reforço.

Suponha um modelo de regressão linear aplicado sobre os conjuntos de dados $\boldsymbol{X}$ e $\boldsymbol{Y}$ com n elementos e apenas um atributo para cada i-ésimo elemento. Neste caso, têm-se, respectivamente, as seguintes funções hipótese e de custo para o modelo:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \theta_0 + \theta_1 x_{i,1} \tag{3.14}$$

$$J(\boldsymbol{\theta}) = J(\theta_0, \theta_1) = \frac{1}{2n} \sum_{i=1}^{n} (h_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - \boldsymbol{y}_i)^2 \tag{3.15}$$

Dada a função de custo em um ponto de partida, isto é, com valores de θ_0 e θ_1 arbitrariamente atribuídos no início, o algoritmo deve "mover-se" em direção ao declive da reta tangente (derivada) que passa pelo ponto de partida desta função [86]. Portanto, são realizados passos de descida, com ajustes dos parâmetros em $\boldsymbol{\theta}$, a cada iteração do algoritmo em direção ao valor mínimo da função. O "tamanho" do passo é determinado por um parâmetro α conhecido como taxa de aprendizagem. Os ajustes dos parâmetros são dados pela atribuição:

$$\theta_j := \theta_j - \alpha \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_j} \tag{3.16}$$

O Algoritmo 1 apresenta o funcionamento da Descida de Gradiente:

Algoritmo 1 Descida de Gradiente

```
Inicializar parâmetros θ
repita
  para j = 0 até m faça
    Ajustar parâmetro θ_j de acordo com (3.16)
  fim para
até Obter convergência
```

A convergência do algoritmo pode ser verificada a partir da diferença entre os valores da função de custo do passo atual e do passo anterior. Se essa diferença estiver abaixo de um valor de tolerância estabelecido o algoritmo deve ser finalizado.

Deve-se contudo estabelecer um valor adequado para o parâmetro α. Um valor muito grande resulta em passos maiores, implicando na possibilidade do algoritmo não convergir. Por outro lado, um valor muito pequeno resulta em passos menores, implicando, portanto, em uma convergência mais lenta.

3.4.2 *Support Vector Machine*

A Máquina de Suporte Vetorial, ou *Support Vector Machine* (SVM), é uma técnica de Aprendizado de Máquina Supervisionado utilizada originalmente para classificação binária. O SVM é caracterizado por duas etapas: treinamento e avaliação.

A etapa de treinamento tem por objetivo estimar uma fronteira entre classes a partir de amostras extraídas do conjunto de dados. A etapa de avaliação busca classificar um novo elemento como pertinente a uma das regiões estabelecidas e separadas pela fronteira estimada.

Suponha um vetor de características $\boldsymbol{x} \in \mathbb{R}^m$ (elemento) e um rótulo de classificação escalar $y \in \{-1, +1\}$. Considerando todos os elementos utilizados para treinamento do SVM, este deve produzir um *hiperplano ótimo* definido como a margem máxima de separação entre duas classes, ou seja, a distância máxima entre os elementos mais próximos pertinentes a classes distintas, denominados *vetores de suporte*. Para tal, utiliza-se uma equação do hiperplano, definida como $\boldsymbol{wx} + b = 0$ em que $\boldsymbol{wx} = \sum_{i=1}^{m} w_i x_i$. Os parâmetros $\boldsymbol{w}$ e b são obtidos após treinamento.

A etapa de avaliação realiza a classificação após a obtenção do hiperplano ótimo. Para um novo elemento $\boldsymbol{x}$ aplica-se a fórmula [27; 91]:

$$y = f(\boldsymbol{x}) = \begin{cases} +1, & \boldsymbol{wx} + b \geq 0 \\ -1, & \boldsymbol{wx} + b < 0 \end{cases} \tag{3.17}$$

Evidentemente, a equação do hiperplano realiza apenas uma separação linear entre os elementos. Para solução de problemas de classificação não-linear, um mapeamento não-linear pode ser aplicado nas amostras de treinamento e nos elementos utilizados na avaliação. O mapeamento é realizado por uma função $\phi(\boldsymbol{x}) : \mathbb{R}^m \mapsto \boldsymbol{\mathcal{F}}$, em que $\boldsymbol{\mathcal{F}}$ refere-se ao espaço de características, criado para tornar a classificação linearmente separável.

A Figura 3.4 ilustra um exemplo de mapeamento não-linear realizado pela função ϕ para um SVM. Embora o SVM tenha sido originalmente desenvolvido para classificação binária, há algumas adaptações que possilitam a classificação multiclasses.

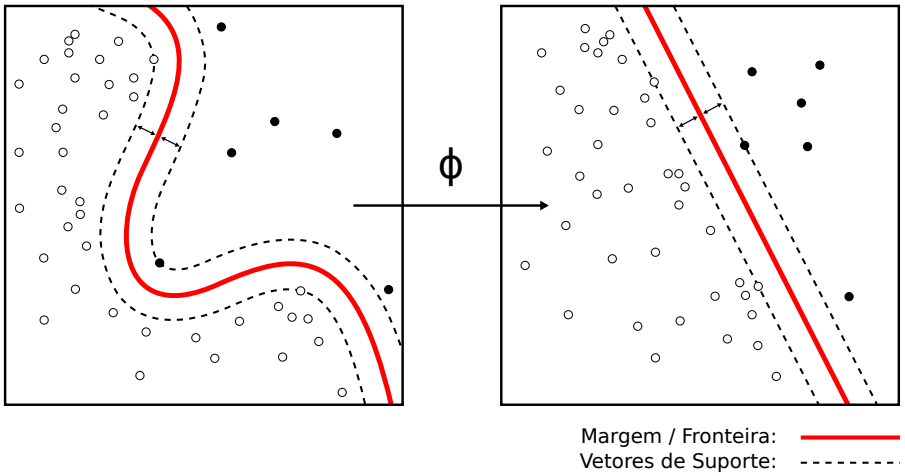


Figura 3.4: Ilustração de mapeamento não-linear pela função ϕ para classificação por um SVM. Extraído e adaptado de [92].

3.4.3 *k-Nearest Neighbors*

O *k-Nearest Neighbors* (k-NN) é um algoritmo simples de Aprendizado de Máquina Supervisionado utilizado para classificar elementos em um conjunto de dados sem a necessidade de definição da função *hipótese*, e por esse motivo é conhecido como algoritmo de aprendizagem "preguiçoso" [93].

O k-NN consiste basicamente na classificação de um novo elemento a partir da rotulação de k elementos mais próximos (vizinhos) no conjunto analisado. Neste caso, a classificação é realizada a partir de instâncias do conjunto, sendo que o rótulo pertencente à maioria dos vizinhos é atribuído ao novo elemento. A classificação realizada pelo k-NN é simples, direta e conveniente para situações onde existem muitas classes envolvidas.

Dado um conjunto com n pares $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, $i \in \{1, 2, \ldots, n\}$, onde $\boldsymbol{x}_i$ corresponde a um elemento com m atributos e $\boldsymbol{y}_i$ indica seu rótulo associado, o algoritmo deve descobrir o valor do rótulo $\boldsymbol{y}_l$ de uma nova amostra de entrada $\boldsymbol{x}_l$ [94; 95].

Os vizinhos mais próximos são obtidos a partir do cálculo de alguma medida de similaridade considerada. A mais comum, adotada no k-NN, é a distância euclidiana (3.8).

Considerando $k = 1$, a nova amostra de entrada $\boldsymbol{x}_l$ receberá o rótulo do elemento $\boldsymbol{x}_i$ que satisfaça a regra $\min\{d(\boldsymbol{x}_i, \boldsymbol{x}_l)\}$. Para uma política $k > 1$ a nova amostra é rotulada com os valores correspondentes à maioria dos vizinhos pertencentes à mesma classe. Neste caso, deve-se escolher um valor ímpar para k de maneira que não haja possibilidade de empate na seleção do rótulo.

Dada a complexidade computacional do algoritmo, equivalente a $\mathcal{O}(knm)$, o valor de k não pode ser muito grande, de maneira que o algoritmo torne-se computacionalmente inviável. Além disso, um valor muito grande para k pode eventualmente incluir elementos de outras classes mais distantes, poluindo a classificação produzida. Por outro lado, o valor de k também não pode ser muito pequeno, implicando em uma rotulação muito sensível a ruídos (elementos próximos à nova amostra analisada, mas que não pertencem à mesma classe da maioria dos demais elementos no escopo). O Algoritmo 2 define o funcionamento do k-NN.

Algoritmo 2 *k-Nearest Neighbors*

Obter n pares de elementos $(\boldsymbol{x}_i, \boldsymbol{y}_i)$. $i \in \{1, 2, \ldots, n\}$
Obter p amostras de entrada $\boldsymbol{x}_l$. $l \in \{1, 2, \ldots, p\}$
para $l = 1$ **até** p **faça**
  Definir o conjunto $\boldsymbol{N}_l$ como os k vizinhos mais próximos de $\boldsymbol{x}_l$
  para $i = 1$ **até** n **faça**
    Calcular $d(\boldsymbol{x}_i, \boldsymbol{x}_l)$
    se $d(\boldsymbol{x}_i, \boldsymbol{x}_l) = \min\{d(\boldsymbol{x}_i, \boldsymbol{x}_l)\}$ **e** $(\boldsymbol{x}_i, \boldsymbol{y}_i) \notin \boldsymbol{N}_l$ **então**
      Colocar o par corrrespondente $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ no conjunto $\boldsymbol{N}_l$
    fim se
  fim para
  Obter o rótulo $\boldsymbol{y}_i$ que pertence à maioria dos elementos no conjunto $\boldsymbol{N}_l$ para a l-ésima amostra de entrada
fim para

A Figura 3.5 ilustra a classificação supervisionada de um elemento por meio do algoritmo k-NN. No exemplo apresentado tem-se o parâmetro $k = 5$. A "fronteira" refere-se ao raio máximo de distância utilizado para encontrar os elementos vizinhos mais próximos.

3.4.4 *K-means*

O *K-means* é o algoritmo mais simples em termos de implementação e complexidade computacional existente na categoria de Aprendizado de Máquina Não-supervisionado [96]. Trata-se de um algoritmo iterativo utilizado para formar agrupamentos em um conjunto informado. A quantidade k de agrupamentos é o principal parâmetro a ser informado ao algoritmo, e este deve atribuir os elementos do conjunto a cada agrupamento produzido, retornando como saída a identificação (rótulo) dos agrupamentos de cada elemento. É historicamente um dos algoritmos mais importantes em mineração de dados [93].

Dado n elementos com m dimensões (ou atributos) em um conjunto $\boldsymbol{X}$, o *K-means* encarrega-se de particionar este conjunto em k *clusters*. Cada *cluster* é representado por um elemento

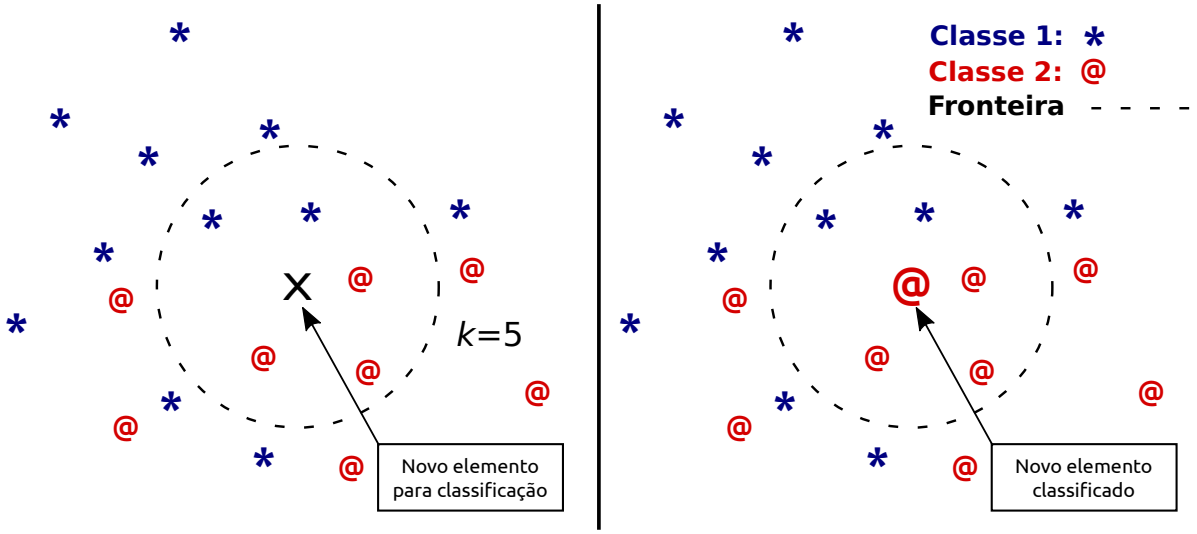


Figura 3.5: Classificação supervisionada de um elemento por meio do algoritmo *k-Nearest Neighbors* (k-NN).

denominado centroide $\boldsymbol{c}_j$, em que $j \in \{1, 2, \ldots, k\}$, pertencente ao conjunto de centroides $\boldsymbol{\mathcal{C}} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k\}$ e usado como referência para cálculo da similaridade dos elementos do conjunto $\boldsymbol{X}$ com os *clusters*. De maneira geral, o algoritmo é realizado em duas etapas até a convergência: atribuição dos elementos aos *clusters* mais próximos; e ajuste dos valores de centroides de cada *cluster*. O Algoritmo 3 descreve o *K-means* detalhadamente.

Algoritmo 3 *K-means*

Obter conjunto de dados $\boldsymbol{X}$
Obter quantidade de *clusters* k
Estabelecer aleatoriamente k centroides $\boldsymbol{c}$ no conjunto $\boldsymbol{X}$ informado
repita
  para $j = 1$ **até** k **faça**
    para $i = 1$ **até** n **faça**
      Calcular medida de similaridade $d(\boldsymbol{x}_i, \boldsymbol{c}_j)$
      se $d(\boldsymbol{x}_i, \boldsymbol{c}_j) = \min\{d(\boldsymbol{x}_i, \boldsymbol{C})\}$ **então**
        $y_i \leftarrow j$
      fim se
    fim para
  fim para
  para $j = 1$ **até** k **faça**
    Atualizar valor do centroide $\boldsymbol{c}_j$ (média dos elementos $\boldsymbol{x}_i$ cujo rótulo $y_i = j$)
  fim para
até Obter convergência
Retornar $\boldsymbol{C}$ e $\boldsymbol{Y}$

A medidade de similaridade adotada no *K-means* é a Distância Euclidiana (3.8). A convergência ocorre quando não há mais alteração dos rótulos obtidos para os elementos do conjunto e também dos valores dos centroides durante uma determinada quantidade de iterações.

É possível a ocorrência de *clusters* vazios, isto é, quando a convergência do algoritmo resulta em um ou mais *clusters* com nenhum elemento $\boldsymbol{x}_i$ associado. Isso normalmente ocorre quando são informados valores muito altos de k para o algoritmo em relação ao conjunto $\boldsymbol{X}$ ou também quando a quantidade de atributos m é alta. Neste caso, uma solução padrão é manter a execução do algoritmo reduzindo a quantidade de *clusters* em uma unidade até que não haja nenhum vazio.

3.4.5 *Fuzzy C-means*

O *Fuzzy C-means* (FCM) é um algoritmo desenvolvido para *clustering* cuja técnica atribui graus de pertinência para os elementos relacionados a cada um dos agrupamentos produzidos [97]. O FCM é apropriado para conjuntos de dados em que há incertezas ou imprecisões na formação ideal de agrupamentos. É menos sensível a ruídos[1] e ideal para estabelecimento de agrupamentos cuja fronteira entre eles não é muito bem definida.

No FCM, o conjunto de graus de pertinência $\boldsymbol{U}$ é definido como:

$$\boldsymbol{U} = (\vec{\boldsymbol{u}}_1, \vec{\boldsymbol{u}}_2, \ldots, \vec{\boldsymbol{u}}_i, \ldots, \vec{\boldsymbol{u}}_k)^\dagger, \quad i \in \{1, 2, \ldots, k\} \tag{3.18}$$

$$\vec{\boldsymbol{u}}_i = (u_{i,1}, u_{i,2}, \ldots, u_{i,j}, \ldots, u_{i,n}), \quad j \in \{1, 2, \ldots, n\}, \quad u_{i,j} \in [0, 1] \quad \forall i, j \tag{3.19}$$

Os graus de pertinência em $\boldsymbol{U}$ são calculados considerando a medida de similaridade entre os elementos $\boldsymbol{X}$ e os centroides $\boldsymbol{M}$ estabelecidos para cada agrupamento, da forma a seguir:

$$u_{i,j} = \left[\sum_{c=1}^{k} \left(\frac{\|\boldsymbol{x}_j - \boldsymbol{\mu}_i\|}{\|\boldsymbol{x}_j - \boldsymbol{\mu}_c\|} \right)^{\frac{2}{(\beta-1)}} \right]^{-1} \tag{3.20}$$

Em que β é o expoente de ponderação utilizado para estabelecer um fator de "fuzzyficação" (*fuzziness*), ou seja, um valor para o grau de foco da pertinência dos elementos a cada agrupamento formado.

O objetivo geral do FCM é minimizar a função:

$$J_\beta(\boldsymbol{U}, \boldsymbol{\mu}) = \sum_{j=1}^{n} \sum_{i=1}^{k} (u_{i,j})^\beta \|\boldsymbol{x}_j - \boldsymbol{\mu}_i\|^2 \tag{3.21}$$

Considerando que a atualização dos centroides $\boldsymbol{M}$ é estabelecida da forma:

$$\boldsymbol{\mu}_i = \frac{\sum_{j=1}^{n} (u_{i,j})^\beta \boldsymbol{x}_j}{\sum_{j=1}^{n} (u_{i,j})^\beta} \tag{3.22}$$

[1] O termo *ruído*, utilizado no contexto de *clustering*, refere-se a alguma instância (ou característica) que "polui" um conjunto de dados, ou seja, que agrega informação improdutiva.

O algoritmo para minimizar a função $J_\beta(\boldsymbol{U}, \boldsymbol{\mu})$, encontrando a configuração ótima dos parâmetros, é dado a seguir:

Algoritmo 4 *Fuzzy C-means*

Definir β e k
Inicializar tolerância ε
Inicializar $\boldsymbol{U}$ e $\boldsymbol{M}$ aleatoriamente
$t \leftarrow 0$
repita
  $t \leftarrow t + 1$
  Atualizar $\boldsymbol{U}^t$ de acordo com (3.20)
  Atualizar $\boldsymbol{M}^t$ de acordo com (3.22)
até $\left\|\boldsymbol{U}^t - \boldsymbol{U}^{t-1}\right\| < \varepsilon$

3.4.6 *X-means*

Um problema recorrente em algoritmos de classificação é encontrar e definir a quantidade ideal de *clusters* existentes no conjunto informado. Quando a estrutura dos dados e do problema de classificação são conhecidos essa quantidade pode ser previamente estabelecida pelo algoritmo. Entretanto, há casos em que isso não é possível, demandando, portanto, autonomia na obtenção e definição desse valor.

O *X-means* é um algoritmo utilizado para estimar a quantidade ideal de k agrupamentos, em que o valor k é procurado em um espaço amostral que busque maximizar dois critérios medidos, a escolher: o *Bayesian Information Criterion* (BIC) ou o *Akaike Information Criterion* (AIC) [98]. O BIC e o AIC são pontuações obtidas para um conjunto de dados X por meio do cálculo da máxima verossimilhança em relação a um modelo com distribuição gaussiana [99].

O conjunto de dados $\boldsymbol{X}$ é definido a seguir:

$$\boldsymbol{X} = (\vec{\boldsymbol{x}_1}, \vec{\boldsymbol{x}_2}, \ldots, \vec{\boldsymbol{x}_n})^\dagger, \quad i \in \{1, 2, \ldots, n\} \tag{3.23}$$

$$\vec{\boldsymbol{x}_i} = (x_{i,1}, x_{i,2}, \ldots, x_{i,m}) \tag{3.24}$$

Onde $\boldsymbol{X} \in \mathbb{R}^m$ e n é a quantidade de elementos em um espaço m-dimensional.

Para o *X-means*, o critério mais utilizado é o BIC, obtido pela equação a seguir:

$$\mathrm{BIC}(m_j) = \hat{l}_j(\boldsymbol{X}) - \frac{p_j}{2} \cdot \log n \tag{3.25}$$

Sendo $\hat{l}_j(\boldsymbol{X})$ a log-verossimilhança obtida no ponto j de máxima verossimilhança do conjunto de dados $\boldsymbol{X}$, m é a quantidade de dimensões ou atributos e p_j é a quantidade de parâmetros a ser estimada em m_j.

A quantidade de parâmetros p_j é calculada por [42]:

$$p_j = (k_j - 1) + (m \cdot k_j) + k_j \tag{3.26}$$

Em que k_j é o valor de k testado na j-ésima iteração do algoritmo.

A log-verossimilhança dos dados pertencentes ao centroide $\boldsymbol{\mu}_c(\hat{l}_j(\boldsymbol{X}_c))$, para $\boldsymbol{X}_c$ e n_c respectivamente definidos como subconjunto de $\boldsymbol{X}$ e quantidade de elementos do conjunto associados ao centroide $\boldsymbol{\mu}_c$, incluindo a máxima verossimilhança estimada é calculada por:

$$\hat{l}_j(\boldsymbol{X}_c) = -\frac{n_c}{2}\log(2\pi) - \frac{n_c \cdot m}{2}\log(\hat{\sigma}^2) - \frac{n_c - k_j}{2} + n_c \log n_c - n_c \log n \tag{3.27}$$

Os centroides $\boldsymbol{M}$ são um conjunto em que cada um de seus elementos é atribuído e define o ponto central de um agrupamento. Define-se $\boldsymbol{M}$ como:

$$\boldsymbol{M} = (\vec{\boldsymbol{\mu}}_1, \vec{\boldsymbol{\mu}}_2, \ldots, \vec{\boldsymbol{\mu}}_k)^\dagger \tag{3.28}$$

$$\vec{\boldsymbol{\mu}}_c = (\mu_{c,1}, \mu_{c,2}, \ldots, \mu_{c,m}), \quad c \in \{1, 2, \ldots, k\} \tag{3.29}$$

Os centroides $\boldsymbol{\mu}_c$ são obtidos pela equação a seguir:

$$\boldsymbol{\mu}_c = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i, \quad \boldsymbol{x}_i \in \boldsymbol{X}_c, \quad \boldsymbol{X}_c \subset \boldsymbol{X}, \quad c \in \{1, 2, \ldots, k\} \tag{3.30}$$

A variância $\hat{\sigma}^2$ estimada para cálculo da máxima verossimilhança é dada por:

$$\hat{\sigma}^2 = \frac{1}{n - k_j}\sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu}_c)^2 \tag{3.31}$$

Sendo $\boldsymbol{x}_i$ os elementos do conjunto $\boldsymbol{X}$ e k_j o valor de k estabelecido no algoritmo para a j-ésima iteração.

Basicamente, o *X-means* é realizado em sucessivas iterações do algoritmo *K-means* [96], com valores de k incrementados gradualmente, dado que o *K-means* necessita sempre do valor de k informado como parâmetro. Os passos para execução do *X-means* são descritos no Algoritmo 5, a seguir [98]:

Normalmente, inicializa-se os parâmetros $k_{\min} = 2$ e $k_{\max}$ a critério do operador. Entretanto, considerando que a quantidade de elementos contida no conjunto de dados informados é desconhecida, é desejável atribuir um valor $k_{\max}$ proporcional à quantidade real de classes a serem descobertas, evitando assim a geração de agrupamentos com poucos ou nenhum elemento associado. Pode-se utilizar a regra de ouro ou *Rule of Thumb* [100]:

Algoritmo 5 *X-means*

$k_{\text{selecionado}} \leftarrow 0$
$k_{\min} \leftarrow 2$
$k_j \leftarrow k_{\min}$
repita
  $i \leftarrow k_{\min}$
  repita
    Particionamento dos dados em k_j centroides
    Selecionar os pontos mais próximos aos centroides
    Calcular novos centroides
  até Obter convergência (semelhante ao *K-means*)
  repita
    $i \leftarrow i + 1$
    Particionamento do i-ésimo agrupamento em dois sub-agrupamentos
    Determinar $BIC(1)$ para agrupamento de acordo com (3.25)
    Determinar $BIC(2)$ para sub-agrupamentos de acordo com (3.25)
    se $BIC(2) > BIC(1)$ **então**
      $k_{\text{selecionado}} \leftarrow k_{\text{selecionado}} + 1$
    fim se
  até $i = k_j$
  $k_j \leftarrow k_j + 1$
até $k_j = k_{\max}$
Obtenção do valor $k_{\text{selecionado}}$

$$k_{\max} = \left\lceil \sqrt{\frac{n}{2}} \right\rceil \tag{3.32}$$

3.4.7 Processo de Decisão de Markov

Processo de Decisão de Markov (PDM) ou *Markov Decision Process* (MDP) é um modelo de processo, que enquadra-se na categoria de Aprendizado por Reforço, utilizado para controle de problemas destinados à tomada de decisão para otimização dos resultados obtidos pelo agente a longo prazo [101; 102].

O PDM é modelado como uma tupla $\langle \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{P}}, \boldsymbol{\mathcal{R}} \rangle$ em que:

- $\boldsymbol{\mathcal{S}}$ é o conjunto de estados do processo;
- $\boldsymbol{\mathcal{A}}$ é o conjunto de ações possíveis de serem realizadas;
- $\boldsymbol{\mathcal{P}}$ é o conjunto contendo as probabilidades de transição entre estados $\boldsymbol{\mathcal{S}}$;
- $\boldsymbol{\mathcal{R}}$ é o conjunto de valores de recompensa para o processo após a realização de uma ação $a \in \boldsymbol{\mathcal{A}}$.

O PDM pode ser representado por uma cadeia de Markov, caracterizada por estados discretos, onde a transição entre estados ocorre em uma *época de decisão*. A época de decisão é a unidade de tempo que serve como referência para tomada de decisões. Cada decisão implica em uma ação realizada, correspondente à escolha de um estado (transição) no qual o PDM assumirá na próxima unidade de tempo. A realização de uma ação resulta no retorno de um valor escalar de recompensa (ou punição) ao PDM pelo ambiente investigado. Além disso, a escolha das ações a serem realizadas são definidas a partir das probabilidades do PDM.

A Figura 3.6 ilustra a cadeia de um PDM contendo seus estados e ações. As probabilidades de transição e os valores de recompensa não são apresentados, visando simplificar a ilustração. Entretanto, o diagrama da Figura 3.3 auxilia a compreensão do mecanismo de obtenção de valores de recompensa do processo.

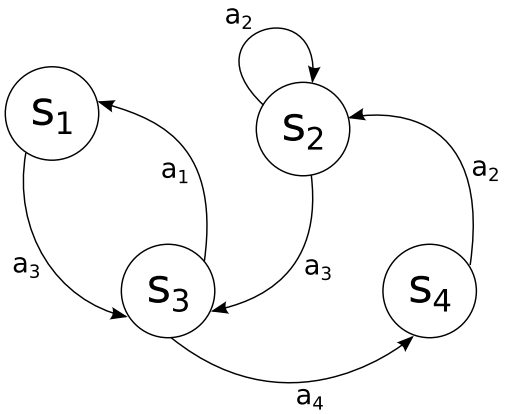


Figura 3.6: Cadeia de Markov de um PDM com estados finitos.

As ações em um PDM são tomadas a partir de regras de decisão d em uma época de decisão k tal que $d_k(s) : \boldsymbol{\mathcal{S}} \mapsto \boldsymbol{\mathcal{A}}$. O conjunto de regras de decisão é denominado política. O objetivo principal em um PDM é encontrar uma política π que otimize a longo prazo o valor ótimo da função de recompensa total $u_k^*(s)$, dado por:

$$u_k^*(s_k) = \max_{d_k(s) \in \boldsymbol{\mathcal{A}}} \left\{ r_k(s_k, d_k(s)) + \gamma \sum_{s' \in \boldsymbol{\mathcal{S}}} p(s_k, d_k(s), s') u_{k+1}^*(s') \right\} \tag{3.33}$$

Lembrando que $\pi \in \{d_0, d_1, \ldots, d_{Z-1}\}$, em que Z é a quantidade total de épocas de decisão em um horizonte de eventos finito. A variável γ é um fator de desconto de recompensas imediatas e s' equivale ao próximo estado do PDM na época de decisão considerada. A probabilidade de transição p é dada a partir do estado corrente s_k e a regra de decisão d_k (sendo $d_k = a$), que conduzirá ao estado s'.

Em um PDM as informações de recompensa imediata são obtidas por uma função *estado-ação* denominada *função Q*, denotada por:

$$Q(s, a) = r(s, a) + \gamma \max_{a'} Q(s', a') \tag{3.34}$$

3.4.7.1 *Q-Learning*

Em situações em que os valores de probabilidade de transição ou recompensa são desconhecidos emprega-se algum algoritmo para descoberta dos valores da *função Q*. O *Q-Learning* [103; 104], além de ser um algoritmo com a finalidade de descobrir valores para o conjunto $\boldsymbol{\mathcal{P}}$, também é utilizado para encontrar uma política de otimização de PDMs. Trata-se de um algoritmo livre de modelo ou *model-free*, ou seja, capaz de encontrar políticas sem a necessidade de conhecer previamente o modelo no qual está sendo aplicado. Sua implementação é simples e garante a convergência do PDM para uma política ótima se o conjunto de estados e ações forem finitos [105].

Para cada par de *estado-ação* (s, a) visitado em um passo t, tendo α como fator de aprendizagem do algoritmo, o *Q-Learning* atualiza os valores na *tabela Q* seguindo a equação (3.35):

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right] \tag{3.35}$$

O Algoritmo 6 apresenta o funcionamento do *Q-Learning*.

Algoritmo 6 *Q-Learning*

Inicializar Q_0 e α
para $t = 0$ **até** $T - 1$ **faça**
  Selecionar estado s_t
  Selecionar ação a_t
  Enviar informação sobre a_t e s_t ao ambiente
  Obter valor de recompensa r_{t+1}
  Calcular $Q(s_t, a_t)$ de acordo com (3.35)
fim para

O PDM com *Q-Learning* pode ser aplicado a processos determinísticos e estocásticos. Além disso, o treinamento pelo *Q-Learning* pode ser feito de maneira *online* ou *offline*. Para processos determinísticos, cujo valor de recompensa é determinado para uma ação correspondente, a etapa de treinamento do PDM realizado pelo *Q-Learning*, normalmente, deve acontecer antes de sua execução (treinamento *offline*), com os valores de probabilidade de transição atualizados após essa etapa.

Processos estocásticos, por sua vez, apresentam valores de recompensa incertos após a escolha de uma ação. Por esse motivo, é mais adequado a realização do treinamento de maneira *online*. Isso implica na atualização dos valores de probabilidade de transição do PDM durante o treinamento do algoritmo *Q-Learning*.

No treinamento *offline*, o par de *estado-ação* (s, a) a ser visitado em um passo t do algoritmo é selecionado aleatoriamente, sendo que o valor do parâmetro T, que indica a quantidade total de passos do algoritmo, deve ser adequado o suficiente para possibilitar que todos os pares sejam visitados e calculados, conduzindo à convergência.

No treinamento *online* é necessário tratar e encontrar um equilíbrio adequado para o para-

digma conhecido como *exploration versus exploitation*. Este paradigma estabelece a necessidade de selecionar pares de *estado-ação* aleatoriamente (*exploration*), visando descobrir novos estados e valores de recompensa que possam conduzir a um valor ótimo global, em contraste com a seleção de pares de *estado-ação* comprovadamente úteis (maior valor) para otimização do sistema no contexto ótimo local (*exploitation*). Entre diversas estratégias na literatura existem duas amplamente utilizadas para tratar o paradigma: o ϵ-*greedy* e o *softmax* [89].

No ϵ-*greedy*, o algoritmo obedece a duas regras de seleção: uma gananciosa (*greedy*) e outra aleatória. A regra gananciosa seleciona sempre os pares de *estado-ação* (s, a) cujo valor equivalha ao maior correspondente na *tabela Q*. A regra aleatória normalmente é selecionada com probabilidade ϵ enquanto a *greedy* é escolhida com probabilidade $(1 - \epsilon)$, onde $\epsilon \in [0, 1]$. É importante neste caso observar um valor de ϵ adequado para otimização do PDM a longo prazo.

O *softmax*, ou função exponencial normalizada, considera a probabilidade de seleção de um par *estado-ação* proporcional aos valores obtidos na *tabela Q*. Neste caso, os pares nos quais é esperado o retorno de melhor recompensa possuem maior probabilidade de serem selecionados. A probabilidade $P_t(s, a)$ de seleção de uma ação a em um estado s, em um total de n ações disponíveis a partir deste estado, é calculada da seguinte forma:

$$P_t(s, a) = \frac{e^{\frac{Q_{t-1}(s,a)}{\tau}}}{\sum_{b=1}^{n} e^{\frac{Q_{t-1}(s,b)}{\tau}}} \tag{3.36}$$

O parâmetro τ é uma constante com valor real positivo denominada *temperatura*. Um valor de τ muito próximo de zero faz a regra de seleção atribuir maior probabilidade às ações cuja recompensa esperada seja maior. Em contrapartida, um valor de τ muito grande ($\tau \to \infty$) assume todas as ações equiprováveis.

É importante observar que tanto na estratégia ϵ-*greedy* como no *softmax* a *tabela Q* deve ser inicializada com valores aleatórios.

Para auxiliar a compreensão do *Q-Learning*, a Tabela 3.1 apresenta uma *tabela Q* hipotética, baseada no PDM ilustrado na Figura 3.6, contendo valores atribuídos após execução do algoritmo em um passo t executando qualquer uma das regras anteriormente descritas. O par de *estado-ação* $Q(2, 3)$, por exemplo, possui valor igual a 3,11. É importante observar que entre estados onde não há ação que resulte em transição direta, ou seja, onde não há uma ação que conecte um estado a outro diretamente, são atribuídos valores muito pequenos (igual a -100, por exemplo) para evitar seleção e cálculo de transições inexistentes no modelo, quando for o caso. Assume-se, portanto, a *tabela Q* como uma representação da experiência adquirida pelo agente no modelo utilizado.

Tabela 3.1: *Tabela Q* representando a experiência de um agente.

	a_1	a_2	a_3	a_4
s_1	-100	-100	2,54	-100
s_2	-100	3,03	3,11	-100
s_3	2,85	-100	-100	2,81
s_4	-100	2,76	-100	-100

3.5 Considerações sobre o Capítulo 3

Neste capítulo foram apresentados os fundamentos do Aprendizado de Máquina, suas principais categorias e alguns modelos e algoritmos para conhecimento. Foram abordados os principais conceitos das categorias e mencionadas as tecnologias mais comuns da área. Presume-se que as informações contidas neste capítulo, acrescidas das informações do Capítulo 4, auxiliem o leitor na compreensão da proposta deste trabalho, a ser apresentada no Capítulo 5.

O Aprendizado de Máquina é indiscutivelmente uma área do conhecimento em franco desenvolvimento. Sua aplicação estende-se a diversos domínios da ciência e também da vida cotidiana. Não obstante, o Aprendizado de Máquina tem corroborado e ampliado a compreensão de alguns conceitos e fenômenos em diversas áreas graças à sua capacidade de aprimorar e acelerar a execução de habilidades com elevado grau de complexidade para automatização.

Capítulo 4

Estratégia de Alocação de Recursos baseada em *Clustering*

Este capítulo conceitua a estratégia ou mecanismo de Alocação de Recursos baseada em *Clustering*, ou *Clustering-Based Resource Allocation* (CBRA). A arquitetura (que apresenta um tipo de fluxograma) e características da estratégia são pormenorizadas com a finalidade de descrever sua atividade. Realiza-se, conjuntamente, uma abordagem sobre a aplicação da estratégia em sistemas LTE-A a fim de conduzir o leitor na compreensão da proposta apresentada mais adiante.

4.1 Fundamentos

O CBRA pode ser definido como uma estratégia diferenciada de alocação de recursos em sistemas de comunicação. Trata-se basicamente de um mecanismo que recorre à classificação e (re)organização de elementos (dispositivos, fluxos de tráfego, etc.) para melhorar o desempenho de um sistema de comunicação. Sua arquitetura é orientada a quatro princípios fundamentais: composição de dados; representação de elementos e características do sistema; mapeamento de padrões; e estabelecimento de prioridades para acesso aos recursos disponíveis.

A composição de dados trata especificamente da definição e estruturação de uma base de dados, conjunto de dados, ou simplesmente *dataset*.

A representação de elementos e características do sistema busca estabelecer uma correspondência entre elementos reais do sistema de comunicação com elementos virtuais, de maneira que características reais em cada correspondência são expressas numericamente e incluídas em um modelo a ser utilizado. A composição de dados está diretamente relacionada com a representação de elementos, uma vez que os dados estabelecidos contém valores que correspondem às características representadas. Basicamente, a base de dados reflete as condições do sistema de comunicação examinado.

O mapeamento de padrões ocorre na forma de classificação ou agrupamento de elementos.

Numericamente, a tarefa de classificação é realizada por meio do *clustering* [11; 106], considerado o núcleo da estratégia.

O estabelecimento de prioridades encarrega-se da ordenação dos elementos representados para alocação de acordo com as características de cada elemento e dos agrupamentos formados. Algumas características podem apresentar maior relevância para efeito de ordenação, ou todas podem ser consideradas igualmente relevantes para a estratégia.

Todos os princípios relatados tem como finalidade o auxílio à tomada de decisão na alocação dos recursos disponíveis.

4.1.1 Modelo

O modelo do CBRA define, essencialmente, a representação de um fluxo de tráfego f do sistema de comunicação por um elemento $\boldsymbol{x}_f \in \mathbb{R}^n$. Cada elemento $\boldsymbol{x}_f$ possui uma quantidade n de características que expressam atributos do fluxo de tráfego em nível de rede, como por exemplo: vazão média; atraso médio; etc. Assim, um elemento $\boldsymbol{x}_f$ é representado por um vetor na forma: $\boldsymbol{x}_f = (x_{f,1}, x_{f,2}, \ldots, x_{f,n})$.

Para um total de F fluxos de tráfego no sistema, estabelece-se um conjunto de dados (matriz) $\boldsymbol{X} \in \mathbb{R}^{F \times n}$ para realização da classificação, definido na forma a seguir:

$$\boldsymbol{X} = \begin{vmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_f \\ \vdots \\ \boldsymbol{x}_F \end{vmatrix} = \begin{vmatrix} x_{1,1}, & x_{1,2}, & \cdots & x_{1,n} \\ x_{2,1}, & x_{2,2}, & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{f,1}, & x_{f,2}, & \cdots & x_{f,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{F,1}, & x_{F,2}, & \cdots & x_{F,n} \end{vmatrix} \tag{4.1}$$

Após o estabelecimento do conjunto de dados, atribui-se cada elemento a um conjunto $\boldsymbol{\mathcal{C}}_j$ mais similar, em uma quantidade total J de conjuntos cujo índice $j \in \{1, 2, \ldots, J\}$. Semelhantemente ao procedimento de Aprendizado Não-supervisionado, descrito na Seção 3.2, a quantidade de conjuntos J indica a quantidade de classificações ou *clusters* a serem produzidos. Cada conjunto $\boldsymbol{\mathcal{C}}_j$ é representado por um centroide $\boldsymbol{\mu}_j$ constituído pelos valores médios dos elementos pertinentes ao conjunto representado.

A regra mais comum para atribuição de um elemento a um conjunto é [20; 24]:

$$j = \underset{j \in \{1,2,\ldots,J\}}{\arg\min}\ d(\boldsymbol{x}_f, \boldsymbol{\mu}_j) \tag{4.2}$$

A função $d(\boldsymbol{x}_f, \boldsymbol{\mu}_j)$ obtém a medida de similaridade entre o elemento $\boldsymbol{x}_f$ e o centroide $\boldsymbol{\mu}_j$. A regra habitualmente utilizada para obtenção da medida é a distância euclidiana (3.8).

O CBRA atualiza, portanto, os centroides e atribui conjuntos aos elementos de maneira iterativa até a convergência, observada pela minimização da função de custo a seguir:

$$G = \sum_{j=1}^{J} \sum_{\boldsymbol{x}_f \in \boldsymbol{\mathcal{C}}_j}^{F} d(\boldsymbol{x}_f, \boldsymbol{\mu}_j) \tag{4.3}$$

Após organizar os elementos em conjuntos, o CBRA realiza ordenação de todos os *clusters* e elementos. Por exemplo, para dois conjuntos $\boldsymbol{\mathcal{C}}_1$ e $\boldsymbol{\mathcal{C}}_2$ ($J = 2$), estabelece-se uma relação do tipo $\boldsymbol{\mathcal{C}}_1 \succ \boldsymbol{\mathcal{C}}_2$ ou $\boldsymbol{\mathcal{C}}_1 \prec \boldsymbol{\mathcal{C}}_2$. Para os elementos, suponha $\boldsymbol{x}_1$ e $\boldsymbol{x}_2$, tal que $\boldsymbol{x}_1 \in \boldsymbol{\mathcal{C}}_1$ e $\boldsymbol{x}_2 \in \boldsymbol{\mathcal{C}}_1$. Como ambos elementos pertencem ao mesmo conjunto $\boldsymbol{\mathcal{C}}_1$, deve-se estabelecer uma relação do tipo $\boldsymbol{x}_1 \succ \boldsymbol{x}_2$ ou $\boldsymbol{x}_1 \prec \boldsymbol{x}_2$, assim como realizado para os conjuntos $\boldsymbol{\mathcal{C}}_1$ e $\boldsymbol{\mathcal{C}}_2$.

Após ordenação, encaminha-se todos os *clusters* e elementos para alocação de recursos. A Seção 4.5 descreve com mais detalhes o procedimento de alocação do CBRA.

4.1.2 Arquitetura Genérica de CBRA

Embora os mecanismos CBRA encontrados na literatura sejam diferentes entre si em diversos aspectos, todos compartilham características em comum em sua operação. Pretende-se aqui capturar e ilustrar essas propriedades a fim de apresentar uma arquitetura genérica da estratégia CBRA aplicável a qualquer tipo de sistema de comunicação.

Basicamente, o CBRA possui quatro etapas ou fases fundamentais. Embora tais etapas possam ser mescladas ou divididas, suas atividades continuam sendo realizadas de alguma forma. As quatro etapas mencionadas são:

- **Composição da base de dados;**
- **Classificação/agrupamento dos elementos;**
- **Ordenação dos elementos / priorização das classes e elementos;**
- **Alocação de recursos.**

A Figura 4.1 ilustra a organização e sequenciamento das etapas mencionadas.

Inicialmente, uma base de dados deve ser constituída para realização da etapa de classificação/agrupamento dos elementos representados. Os itens para representação mais comuns são os fluxos de tráfego, entretanto é possível empregar no CBRA a representação de outros elementos do sistema de comunicação. Após a classificação, uma etapa de ordenação e estabelecimento de prioridades de classes e elementos assume a incumbência de decidir, a partir dos padrões encontrados na classificação, quais elementos terão acesso aos recursos bem como a ordem de acesso a ser prescrita.

A última etapa trata da alocação de recursos propriamente dita. Os parâmetros de ordenação dos elementos devem ser estritamente obedecidos nesta fase. Contudo, a seleção de quais recursos serão alocados é de encargo exclusivo desta etapa.

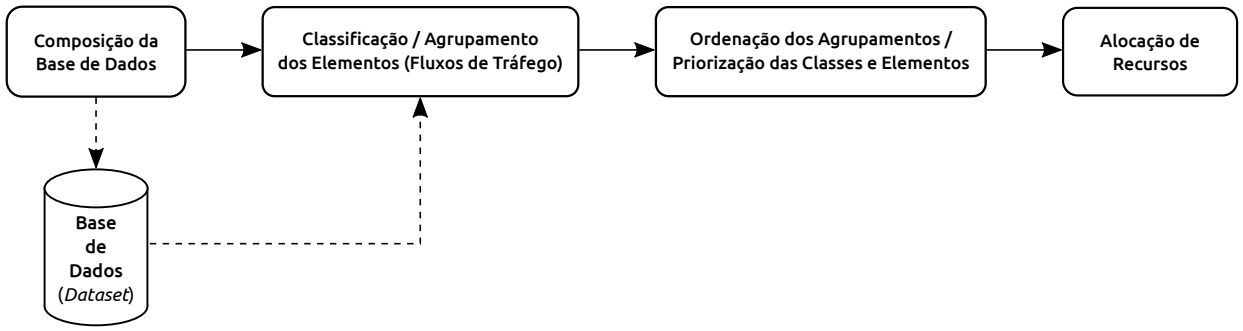


Figura 4.1: Arquitetura genérica de um CBRA.

4.1.3 Arquitetura de CBRA para Sistemas LTE-A

Para descrever a arquitetura CBRA em sistemas LTE-A é preciso, inicialmente, compreender como as etapas estão relacionadas com o sistema. A Figura 4.2 ilustra, em nível de abstração mais elevado, como o CBRA relaciona-se com uma rede do tipo LTE-A. É importante observar que as especificações do 3GPP não são invalidadas, mas incorporadas ao mecanismo CBRA.

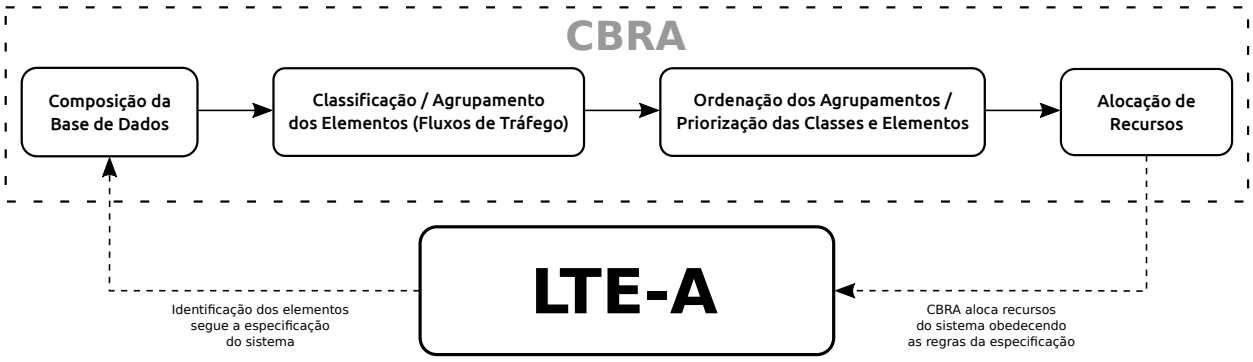


Figura 4.2: Arquitetura de mecanismo CBRA para um sistema LTE-A.

Especificamente, o CBRA é empregado na E-UTRAN, tendo a eNB como a entidade controladora. Entretanto, há possibilidade de uso da estratégia em toda a EPS, caso haja interesse em que alguma outra entidade controle a alocação de recursos no sistema, como o P-GW por exemplo.

Os elementos do LTE-A representados pelo CBRA na arquitetura apresentada são os fluxos de tráfego. Em sistemas LTE-A, os fluxos de tráfego são transportados por meio das *bearers*, de maneira que cada *bearer* é vinculada a um ou mais SDFs (Seção 2.3.1). Para identificação de fluxos de tráfego, levando em conta a *bearer*, o CBRA requer apenas a vinculação de um fluxo de tráfego IP externo a um SDF, e o estabelecimento de um SDF por *bearer*, o que pode ser facilmente configurado pelo operador da rede na política de filtragem de pacotes dos TFTs [74; 107]. Desta forma, qualquer fluxo de tráfego pode ser identificado e ter seus respectivos dados coletados a qualquer instante pelo sistema, inclusive na rede de acesso, na eNB.

A eNB deve possuir uma base de dados "acoplada" para realizar o mapeamento de UEs e fluxos de tráfego e armazenar a representação deles como elementos para o CBRA. Para possibilitar o armazenamento dos dados em uma base de dados, a eNB deve oferecer suporte à tecnologia de *cache* de conteúdo [108].

Após a etapa de ordenação dos agrupamentos e elementos, a eNB deve realizar uma consulta na base de dados para verificar qual fluxo de tráfego e UE é representado por cada elemento. Em posse da ordem dos fluxos de tráfego e UEs, a eNB realiza a alocação de recursos.

Para alocação de recursos do tipo RB, o LTE-A recorre ao escalonamento, de maneira que as especificações não estabelecem um algoritmo específico para utilização, deixando a escolha livre para os operadores de rede. No LTE-A o escalonamento restringe-se a dois domínios: tempo e frequência. No domínio do tempo estabelece-se os UEs selecionados para alocação em um determinado TTI. No domínio da frequência selecionam-se os RBs mais adequados para cada UE. No CBRA, as três primeiras etapas correspondem logicamente ao escalonamento no domínio do tempo, enquanto a última (alocação de recursos) corresponde especificamente ao domínio da frequência.

A Figura 4.3 ilustra a primeira e a última etapa do mecanismo CBRA aplicadas a um sistema LTE-A de acordo com a descrição informada. As referidas etapas foram mencionadas para detalhar a integração do mecanismo com o LTE-A.

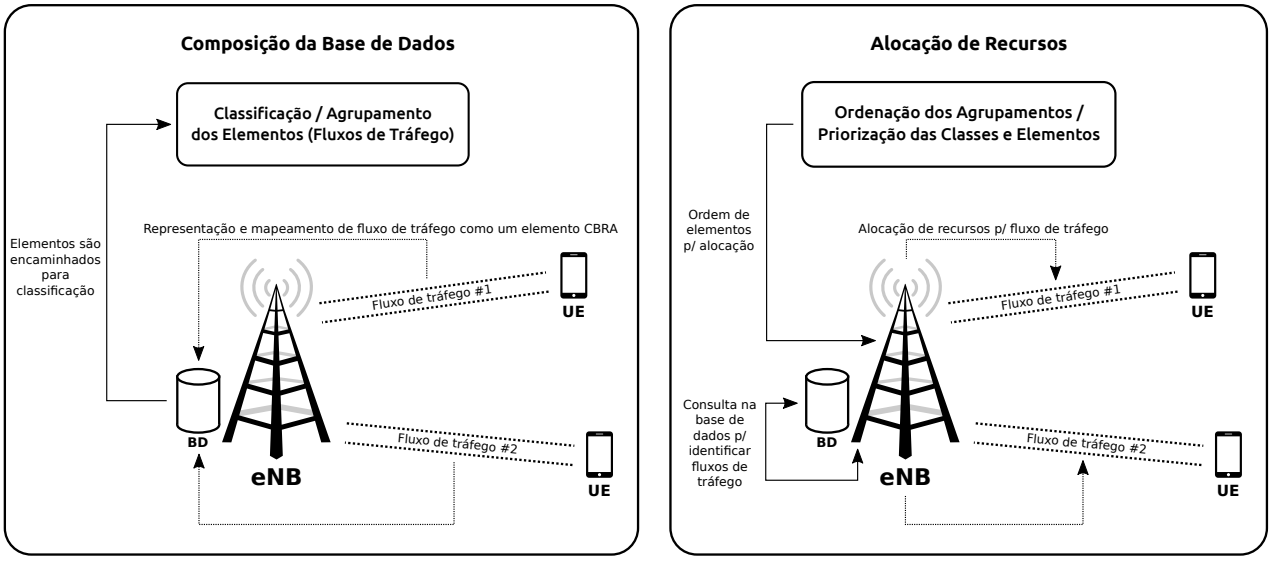


Figura 4.3: Etapas de composição de dados e alocação de recursos em um mecanismo CBRA implementado em sistemas LTE-A.

4.2 Composição da Base de Dados

Esta etapa trata de um dos aspectos mais importantes do CBRA: a composição dos dados utilizados pelo mecanismo. As decisões de alocação do CBRA refletem indiretamente a forma como a base de dados está estruturada. Portanto, estabelecer uma base de dados adequada

para análise é fundamental para definição dos padrões extraídos e, consequentemente, para seleção das decisões a serem tomadas a partir do conhecimento produzido pelo algoritmo de Aprendizado de Máquina implementado no CBRA.

Entre as inúmeras definições existentes para o termo *dado*, a mais importante para o escopo deste trabalho é: "dado é uma observação de algum fenômeno ou evento do mundo real" [45]. Tais observações são interpretadas, registradas e armazenadas em algum meio físico. Geralmente, as observações realizadas são organizadas e estruturadas para auxiliar a inteligibilidade e adequação a algum tipo de modelo.

A interpretação de algum evento define uma característica (*feature*) [109]. Características são representações numéricas de dados "brutos" ou não tratados (*raw data*), sendo que os termos "bruto" e "não tratado" referem-se a observações que ainda não possuem uma interpretação ou significado claramente definidos.

O processo de formular as melhores características para o modelo, a partir de um determinado domínio do conhecimento, é denominado engenharia de características, ou *feature engineering*. Logo, o objetivo inicial na etapa de composição da base de dados é realizar uma engenharia de características eficiente.

Para melhor compreensão, alguns conceitos relacionados à engenharia de características são apresentados a seguir:

- a seleção de características (*feature selection*) é um processo (manual ou automatizado) para definição inicial de um subconjunto de características;
- o aprendizado de características (*feature learning*) é uma técnica automatizada de representação (ou transformação) de características empregada para facilitar a extração de informação inteligível [46];
- o mapa de características (*feature map*) é a utilização de alguma função para redefinir a forma como uma característica é apresentada, sendo que essa mesma função pode receber uma ou mais características como parâmetro. Pode ser empregado tanto para aumentar a inteligibilidade como para acelerar a execução de algum algoritmo;
- o dimensionamento de características (*feature scaling*) é a realização de ajustes na escala de valor de uma ou mais características;
- a extração de características (*feature extraction*) é um termo sinônimo de *feature learning*. Trata-se de um processo de extração (ou refinamento) de características a partir de um conjunto de características estabelecido *a priori*.

Cabe observar que alguns conceitos relacionados à engenharia de características, como a seleção ou aprendizado de características, podem ser eventualmente confundidos com a redução de dimensionalidade, por exemplo. A redução de dimensionalidade refere-se à transformação dos dados com uma dimensão maior (maior quantidade de características ou atributos) em uma nova representação com dimensão menor (menor quantidade de características ou atributos),

mantendo a informação principal [110]. Idealmente, a dimensionalidade da representação transformada é igual à dimensionalidade interna (ou *intrínseca*) dos dados. Trata-se do número mínimo de variáveis necessárias para expressar todas as características possíveis sem perdas significativas de representatividade da informação original [111].

Desta forma, qualquer procedimento de seleção ou aprendizado de características que transforme os dados originais em uma nova representação com quantidade menor de características é considerado um mecanismo de redução de dimensionalidade.

4.2.1 Seleção de Características

A forma mais simples de estabelecer uma base de dados é pela seleção de características. Dado um conjunto inicial de características obtidas a partir de observações, realiza-se um procedimento supervisionado de separação entre as características mais e menos úteis para o modelo considerado, descartando-se as menos úteis [87].

O procedimento de seleção pode ser categorizado em um dos três tipos a seguir [45]:

- **Filtragem:** emprega técnicas de pré-processamento para remoção de características dispensáveis. Por exemplo, um cálculo de correlação pode ser realizado entre as características, e aquelas que estiverem fora de um limiar pré-estabelecido são removidas;
- ***Wrapper* ou *Empacotamento*:** organiza as características em subconjuntos e realiza testes no modelo de maneira que, para cada subconjunto testado, o modelo retorne uma pontuação de qualidade (*score*). Desta forma, as características são refinadas até encontrar um subconjunto ideal;
- ***Embedded* ou *Embutido*:** a seleção de características é realizada como parte do processo de treinamento do modelo. Isso é muito comum em modelos de aprendizagem do tipo *Árvores de Decisão*[1], em que uma determinada decisão pode referir-se a um subconjunto de características selecionadas.

4.2.2 Aprendizado de Características

Também definido como aprendizado de representação. Emprega algoritmos para descobrir características de mais alto nível, sendo realizado, portanto, de forma inteiramente autônoma. Entre as inúmeras técnicas existentes na literatura menciona-se a seguir apenas as duas consideradas mais importantes para o escopo deste trabalho: PCA e *Auto-encoder*.

[1]Trata-se de uma ferramenta para suporte à decisão baseada em um modelo de árvore, em que os ramos representam possíveis consequências e resultados de uma ação realizada.

4.2.2.1 Principal Component Analysis (PCA)

O PCA é uma técnica de análise estatística multivariada utilizada para extrair informação de um conjunto de dados, transformando-o em um novo conjunto contendo variáveis ortogonais não correlacionadas definidas como componentes principais [112–114]. A quantidade de componentes extraídos é menor ou igual do que a quantidade de variáveis do conjunto original.

O objetivo do PCA é extrair uma determinada quantidade de componentes de maneira que a variância total de *projeção* das variáveis originais seja maximizada e o conjunto de dados original seja representado com um erro mínimo de reconstrução.

Para uma matriz $\boldsymbol{X}_{m\times n}$ normalizada (veja Seção 4.2.4), com uma quantidade m de elementos, n variáveis originais e uma quantidade p de componentes a serem extraídos, tal que $p \leq n$, realiza-se inicialmente sua decomposição em valores singulares ou *Singular Value Decomposition* (SVD) na forma a seguir:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\dagger} \tag{4.4}$$

em que $\boldsymbol{U}_{m\times m}$ é uma matriz unitária que contém os autovetores de $\boldsymbol{X}\boldsymbol{X}^{\dagger}$ (matriz de covariância). $\boldsymbol{S}_{m\times n}$ é uma matriz retangular diagonal com números reais não negativos na diagonal, e $\boldsymbol{V}^{\dagger}$ é matriz conjugada transposta (ou transposta usual) de $\boldsymbol{V}_{n\times n}$, outra matriz unitária, que contém os autovetores de $\boldsymbol{X}^{\dagger}\boldsymbol{X}$ e atende a propriedade $\boldsymbol{V}^{\dagger}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^{\dagger} = \boldsymbol{I}$ (matriz identidade).

Assim, a decomposição dos componentes principais $\boldsymbol{C}$ de $\boldsymbol{X}$ é calculada como:

$$\begin{aligned} \boldsymbol{C} &= \boldsymbol{X}\boldsymbol{V} && (4.5)\\ &= \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\dagger}\boldsymbol{V} && (4.6)\\ &= \boldsymbol{U}\boldsymbol{S} && (4.7) \end{aligned}$$

De maneira que, para reduzir a dimensionalidade de $\boldsymbol{X}$ em uma quantidade p de componentes, seleciona-se a matriz $\boldsymbol{U}$, limitada apenas às p primeiras colunas, e multiplica-se com a $p \times p$-ésima parte superior esquerda da matriz $\boldsymbol{S}$:

$$\boldsymbol{C}_{m\times p} = \boldsymbol{U}_{m\times p}\boldsymbol{S}_{p\times p} \tag{4.8}$$

Para verificação do erro de representação, a matriz original $\boldsymbol{X}$ pode ser reconstruída em $\hat{\boldsymbol{X}}$ a partir da matriz de componentes $\boldsymbol{C}$ e dos autovetores de $\boldsymbol{X}^{\dagger}\boldsymbol{X}$ em $\boldsymbol{V}$, na forma:

$$\hat{\boldsymbol{X}} = \boldsymbol{C}\boldsymbol{V}^{\dagger}_{n\times p} \tag{4.9}$$

Em que apenas as p primeiras colunas em $\boldsymbol{V}$ são utilizadas para efeito de reconstrução.

A Figura 4.4 ilustra a redução de um conjunto com três variáveis (x_1, x_2, x_3) em um novo conjunto contendo dois componentes, apresentando uma projeção ótima do PCA (sobre os vetores $\boldsymbol{u}_1$ e $\boldsymbol{u}_2$) com máxima variância.

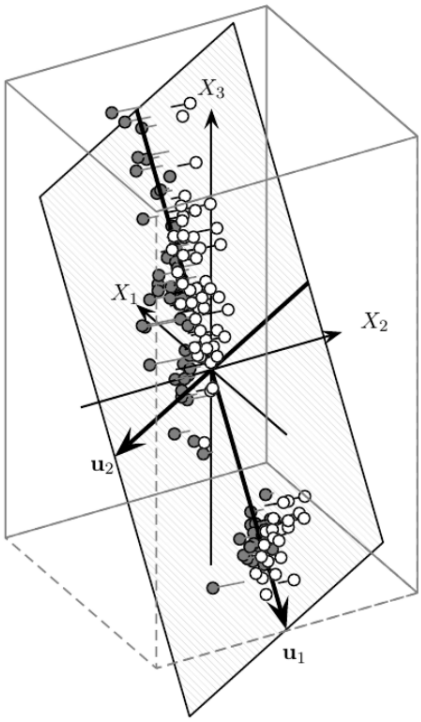


Figura 4.4: Ilustração de redução de dimensionalidade de um conjunto de dados com PCA. Extraído de [115].

4.2.2.2 *Auto-encoder*

O *Auto-encoder* é um tipo alternativo de rede neural artificial auto-associativa, ou *Auto-Associative Neural Network* (AANN) [47; 116], utilizada especificamente para codificação e aprendizado de características. É basicamente organizado em três camadas fundamentais: entrada, ou *input layer*; saída, ou *output layer*; e uma camada interna denominada camada de codificação, *gargalo* ou *code layer*. Tradicionalmente, o *Auto-encoder* apresenta duas camadas internas, ou *hidden layers*, com a finalidade de intermediar as camadas de entrada e saída com a camada de codificação. Entretanto, pode-se recorrer a um maior número de camadas internas ou com estruturas diferentes do tradicional.

As camadas de entrada e saída possuem a mesma quantidade de neurônios, enquanto a camada de codificação possui uma quantidade diferente de neurônios de acordo com o objetivo desejado: compressão ou difusão/dispersão dos dados. Para compressão, utiliza-se uma quantidade menor de neurônios, e para difusão, uma quantidade maior.

As primeiras camadas do *Auto-encoder* (entrada e codificação) compõem a parte de codificação, *codificador* ou *encoder*, enquanto as últimas camadas (codificação e saída) compõem a parte de decodificação dos dados, *decodificador* ou *decoder*. O codificador gera uma representação reduzida dos dados, enquanto o decodificador retorna a representação "original" dos dados a partir de sua versão *codificada*. O codificador e decodificador sobrepõem-se na região da camada de codificação, onde define-se o *espaço latente* ou *latent space*, que é onde a representação original dos dados é retida em sua forma codificada.

Considere um elemento $\boldsymbol{x}$ com n características. O objetivo do *Auto-encoder* é produzir uma representação $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ codificada desse elemento de maneira que tal representação possa ser reconstruída pelo decodificador e assemelhe-se ao máximo com o elemento original na saída, incorrendo apenas em um erro mínimo tolerável de reconstrução. Se a finalidade for compressão dos dados, a quantidade m de características deve ser $m < n$, caso contrário, se for utilizado para difusão, $m > n$. Para isso, o *Auto-encoder* deve ser treinado para realizar a aproximação de uma função $h_{\boldsymbol{\theta}}$ tal que $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \approx \boldsymbol{x}$. O treinamento do *Auto-encoder* caracteriza-se pela minimização da função de custo a seguir, denominada *função de erro quadrático de reconstrução* [117]:

$$J(\boldsymbol{\theta}) = \frac{1}{2}\|h_{\boldsymbol{\theta}}(\boldsymbol{x}) - \boldsymbol{x}\|^2 \tag{4.10}$$

O parâmetro $\boldsymbol{\theta}$ refere-se aos pesos sinápticos da rede neural do *Auto-encoder*. Visto que os rótulos para classificação da saída correspondem exatamente aos mesmos parâmetros de entrada, assume-se que o treinamento do *Auto-encoder* ocorra de maneira não-supervisionada. O treinamento pode ser realizado por meio do método de gradiente (Seção 3.4.1) ou por meio de retropropagação (*backpropagation*), e técnicas como *dropout* [118] e regularização também podem ser utilizadas.

A Figura 4.5 ilustra o *Auto-encoder* de acordo com a descrição realizada. Cabe observar que, assim como qualquer outra rede neural artificial, as funções de ativação nos neurônios são definidas a critério visando atender o modelo selecionado.

4.2.3 Mapa de Características

As redefinições realizadas pela seleção e pelo aprendizado de características estabelecem um subconjunto de características a partir do conjunto inicialmente informado. No *feature map*, o conjunto inicial é mantido, porém as características são modificadas, ou novas características derivadas das originais são acrescentadas, de maneira que a apresentação do conjunto seja redefinida.

As mudanças ocorrem por meio do uso de funções de qualquer natureza. Por exemplo, suponha um conjunto de dados $\boldsymbol{X}$ no qual cada elemento $\boldsymbol{x}$ possua apenas duas características: x_1 e x_2. Para todos os efeitos do exemplo, não há mais como realizar uma nova observação para obtenção de novas características. Com o *feature map*, a característica x_1 pode ser modificada para $x_1 = f(x_1) = \sqrt{x_1}$, ou uma "nova" característica x_3 pode ser acrescentada ao conjunto baseando-se nas características já existentes. Neste caso, pode-se definir $x_3 = f(x_1, x_2) = \frac{x_1}{x_2}$, por ora. Enfim, qualquer tipo de cálculo para constituição de características por um mapa pode ser considerado.

É importante salientar que a finalidade é aumentar a inteligibilidade e acelerar a execução do algoritmo utilizado para processamento do conjunto, observando sempre o modelo empregado. Se utilizado sem parcimônia, o *feature map* pode desencadear redundância e até mesmo dificultar a obtenção de padrões.

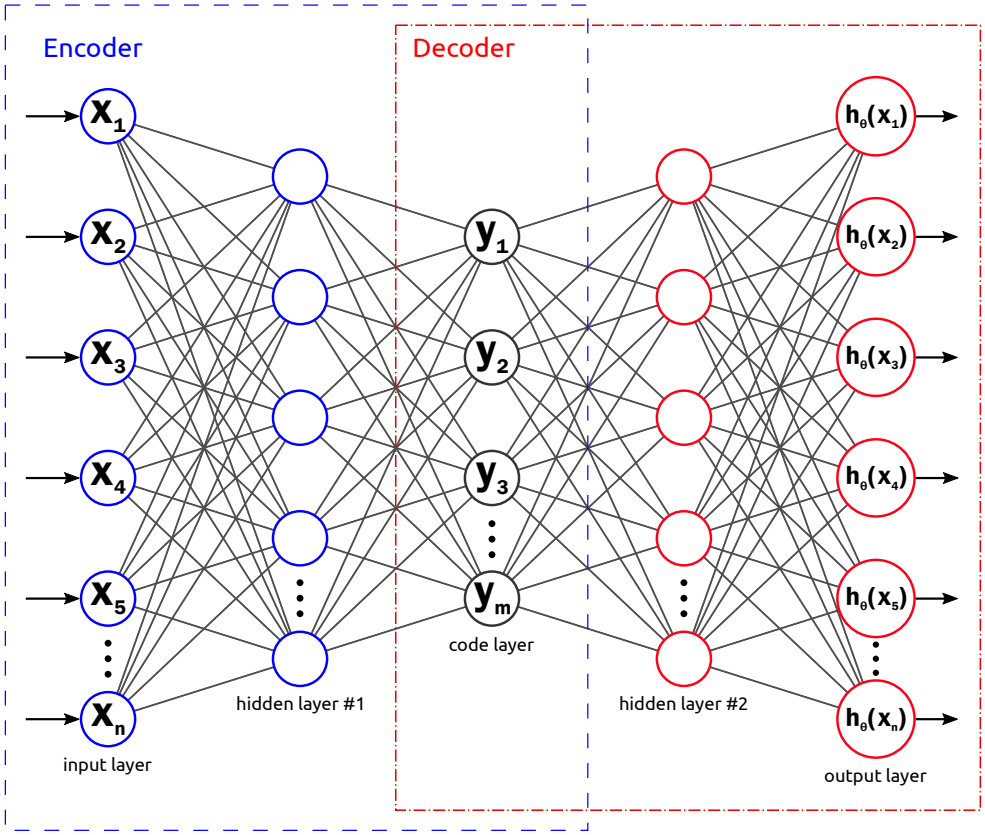


Figura 4.5: *Auto-encoder*

4.2.4 Dimensionamento de Características

Se o modelo for sensível à diferença de grandeza de valores entre características, seu desempenho, precisão ou eficiência podem ser seriamente afetados. Por esse motivo, o dimensionamento ou normalização de características (*feature scaling*) é algo importante a ser realizado em um conjunto de dados.

O *feature scaling* trata da regulação dos intervalos de valores das características do conjunto. Quando realizado, o dimensionamento deve considerar cada característica individualmente. Desta forma, o conjunto apresentará todos os valores de características dentro de um intervalo comum, facilitando tarefas de classificação, por exemplo. A seguir são apresentadas algumas técnicas de dimensionamento amplamente utilizadas.

4.2.4.1 Dimensionamento *Min-Max*

Considere um valor de característica individual x qualquer e, para todos os elementos do conjunto, sobre a característica considerada, um valor mínimo $\min(x)$ e máximo $\max(x)$ observados. O dimensionamento *Min-Max* ajusta cada valor individual x para um novo valor $\tilde{x}$ considerado na forma a seguir:

$$\tilde{x} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4.11}$$

Desta forma, os valores de características são redefinidos para acomodarem-se em um intervalo entre 0 e 1.

4.2.4.2 Dimensionamento Médio

O dimensionamento médio é semelhante ao dimensionamento *Min-Max*, porém o numerador apresenta a diferença de x (valor a ser dimensionado) com o valor médio $\bar{x}$ dos elementos observados para a referida característica no conjunto:

$$\tilde{x} = \frac{x - \bar{x}}{\max(x) - \min(x)} \tag{4.12}$$

No dimensionamento médio os valores das características ajustadas acomodam-se em um intervalo entre -1 e 1.

4.2.4.3 Uniformização ou Dimensionamento de Variância

Utilizada quando deseja-se características com média zero e variância unitária. É geralmente utilizada como normalização na maioria dos algoritmos de Aprendizado de Máquina [119]. Para tal, calcula-se primeiramente a média $\bar{x}$ e o desvio padrão σ dos valores de elementos observados para a característica x em questão. O dimensionamento é feito da forma:

$$\tilde{x} = \frac{x - \bar{x}}{\sigma} \tag{4.13}$$

4.2.4.4 Normalização Euclidiana

A normalização Euclidiana, também denominada *normalização* L^p, é utilizada para dimensionamento dos elementos de maneira que o comprimento do vetor de características seja igual a 1. Para tal, divide-se os valores de cada característica x_i pela *norma Euclidiana* L^p do elemento:

$$\tilde{x_i} = \frac{x_i}{\|\boldsymbol{x}\|_p} = \frac{x_i}{\left(|x_1|^p + |x_2|^p + \cdots + |x_m|^p\right)^{1/p}}, \quad i \in \{1, 2, \ldots, m\} \tag{4.14}$$

em que $p \in \mathbb{R}, p \geq 1$ é um parâmetro arbitrário para satisfação da função de comprimento da norma, e m é a quantidade de características do vetor ou elemento.

4.3 Classificação de Elementos

Após estabelecimento da base de dados, a próxima etapa a ser realizada no CBRA é a classificação de elementos. A classificação, propriamente dita, é uma tarefa antiga e extremamente útil, aplicável a qualquer área do conhecimento. Basicamente, trata-se da habilidade em formar agrupamentos (*clusters*) de objetos similares entre si. Por meio da classificação é possível abstrair detalhes, organizar conceitos e pensamentos, e principalmente, auxiliar a tomada de decisões em uma determinada atividade [120].

Como mencionado inicialmente, a análise de *clusters*, *cluster analysis* ou simplesmente *clustering*, é um método numérico para realização da classificação [106]. Assim como apresentado na Seção 4.1.1, representa-se um objeto ou elemento como um arranjo numérico $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, por exemplo, de maneira que seus atributos são armazenados nas variáveis indexadas do arranjo. Para um conjunto de elementos, define-se uma matriz $\boldsymbol{X}$, também conhecida como conjunto de dados ou *dataset* – veja a Equação (4.1) – em que cada índice de linha indica um elemento e os índices de coluna indicam os atributos capturados (características). Busca-se então realizar o particionamento do conjunto $\boldsymbol{X}$ em uma determinada quantidade de *clusters* J, cujo valor pode ou não ser previamente informado, dependendo do algoritmo utilizado.

Matematicamente, não há uma definição clara e objetiva para o conceito de *cluster* [106]. Além disso, diferentes algoritmos empregados para realização do *clustering* podem oferecer diferentes definições e significados para o conceito de *cluster*. Sabe-se somente que um *cluster* deve apresentar as seguintes propriedades [121]:

- ***Compactness* ou "compactação"**: relacionada à densidade do agrupamento, apontando um fator de aglomeração dos elementos. Nem sempre é uma propriedade obrigatória, visto que podem existir *clusters* com estruturas mais complexas;
- ***Connectedness* ou conectividade**: trata-se da relação de compartilhamento de um mesmo *cluster* por elementos vizinhos;
- ***Spatial separation* ou separação espacial**: estabelece um critério inicial para soluções triviais. Via de regra, *clusters* tendem a estar espacialmente separados uns dos outros.

A Figura 4.6 ilustra um exemplo de um conjunto de dados que apresenta as propriedades anteriormente mencionadas. O conjunto no exemplo possui apenas duas características, ilustradas graficamente. Note que há uma compactação e separação espacial em ambos os *clusters*. Mesmo que o *cluster* da direita apresente uma estrutura pouco comum, seus elementos apresentam um determinado nível de conectividade.

O *clustering* é um procedimento iterativo de descoberta de conhecimento. Pode ser concebido como um problema de otimização multi-objetivo, visto que busca organizar os elementos mais similares entre si, enquanto mantém simultaneamente uma separação homogênea entre elementos dissimilares. Ao longo do procedimento, uma função de custo é utilizada para "guiar" o

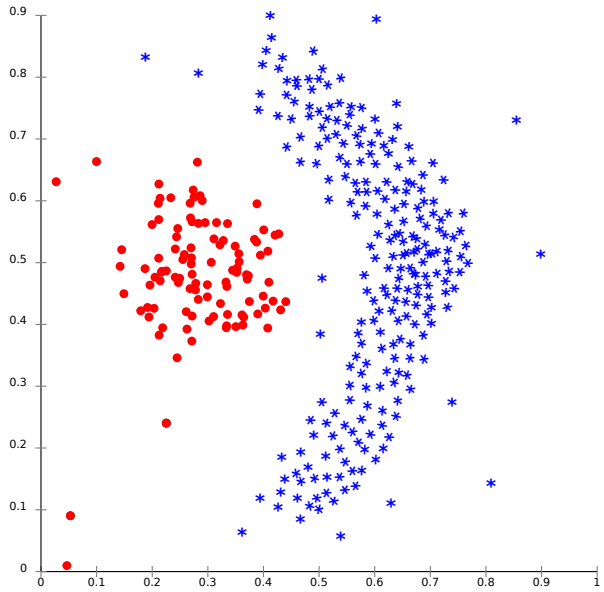


Figura 4.6: Exemplo de *clustering* formado em um conjunto de dados bidimensional. Extraído e adaptado de [122].

algoritmo na busca da solução do problema. Os conceitos essenciais do *clustering* para o CBRA são vistos na Seção 3.2.

Para o CBRA genérico, o *clustering* classifica os elementos com as seguintes finalidades:

- caracterizar parâmetros de QoS dinamicamente;
- reduzir *gaps* de recursos, como *slots* não alocados por exemplo, que podem ocorrer em decorrência de uma demanda irregular e não organizada [20];
- auxiliar a tomada de decisões para alocação em função das características comuns dos elementos.

Embora os elementos usualmente representados sejam os fluxos de tráfego, não há necessidade em rotular[2] as aplicações relacionadas aos fluxos de tráfego transmitidos na rede. Desta maneira, após a classificação, a rotulação de classes e elementos não é fundamental para o mecanismo.

4.4 Ordenação de Classes e Elementos

Após a classificação dos elementos, cada classe gerada possui um elemento virtual que a representa (centroide), e esse elemento também possui características contendo valores numéricos

[2]Define-se como *rotulação* a identificação de todos os atributos possíveis para a aplicação correspondente ao fluxo de tráfego considerado: nome; tipo; protocolo da camada de transporte (TCP ou UDP); número da porta de comunicação; *etc.*

assim como os demais elementos do CBRA. O objetivo nesta etapa é definir, a partir dos valores numéricos, uma relação de ordem entre as classes e seus respectivos elementos de maneira que essa ordem reflita uma prioridade das classes e elementos na etapa de alocação de recursos.

A estratégia CBRA proposta na literatura não define uma abordagem ou sequer algum algoritmo de ordenação padrão para uso nesta etapa. Logo, qualquer abordagem ou algoritmo de ordenação podem ser considerados, desde que a complexidade computacional seja adequada para o procedimento.

Quando um conjunto de dados apresenta elementos com apenas uma característica, por exemplo, a ordenação é realizada de forma simples. Entretanto, a maioria dos casos apresenta conjuntos com múltiplas características analisadas. Neste caso, deve-se adotar alguma estratégia para estabelecimento de prioridades em condições de multiplicidade de características. Esta seção apresenta, portanto, algumas considerações e possíveis abordagens para ordenação de classes nessas condições.

Para um conjunto de dados $\boldsymbol{X}$ classificado em J *clusters* distintos, deseja-se estabelecer uma relação de ordem entre os *clusters*, inicialmente, e aos elementos pertinentes a cada *cluster*, posteriormente. A ordem a ser estabelecida deve considerar, para todos os efeitos, as características dos elementos e as regras para avaliação de cada uma delas.

Para um elemento $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, deve-se obter um valor escalar que o represente em um procedimento de ordenação. O referido elemento deve possuir, portanto, um índice para identificação e diferenciação dos demais elementos dentro do conjunto de dados, e uma representação numérica. Pode-se descrever três abordagens típicas para representação numérica de um elemento:

- **seleção de uma característica/atributo para representação**: entre os diversos tipos de dados que caracterizam um elemento, seleciona-se o mais relevante para um determinado domínio. Por exemplo, para um fluxo de tráfego caracterizado por vazão, atraso e *jitter*, define-se a vazão como a característica que representa numericamente o elemento para efeito de ordenação. Desta maneira os elementos no conjunto são ordenados apenas de acordo com os valores apresentados relacionados à vazão, desconsiderando os demais valores de outras características;

- **soma para representação**: o valor resultante da soma dos valores de todas as características representa o elemento numericamente;

- **ponderação de características para representação**: em todo o conjunto, aplicam-se pesos $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$ aos valores das características, de maneira que os pesos indiquem fatores de relevância de cada característica para efeito de ordenação. Pesos maiores indicam maior relevância para a característica no índice referenciado, e vice-versa. A ponderação pode ser dinâmica ou estática, e considerar no final um valor v para representação do elemento, gerado a partir da soma (ou média) ponderada dos valores, na forma $v = \sum_{i=1}^{n} w_i x_i$.

Outras abordagens podem ser propostas e utilizadas para aperfeiçoar mecanismos CBRA. Acredita-se que a implementação de uma abordagem de ordenação adequada possa otimizar a alocação de recursos em uma estratégia CBRA. Entretanto, essa alegação carece de investigações futuras.

4.5 Alocação de Recursos

A última etapa do CBRA trata da alocação de recursos propriamente dita. Após estabelecimento de prioridades para acesso aos recursos, definem-se nesta etapa os tipos de recursos a serem alocados bem como a sua quantidade, seguindo a ordem estabelecida de cada classe e elemento. É uma etapa simples e com poucas considerações. Direciona-se parte do foco desta seção para a alocação de RBs em sistemas LTE-A por meio de um CBRA.

Como observado anteriormente, as três primeiras etapas do CBRA, implementado para LTE-A, correspondem ao escalonamento no domínio do tempo. Obtidas as ordens dos elementos, a eNB deve identificar os UEs e fluxos de tráfego correspondentes aos elementos para alocação.

Seguindo a ordem dos UEs e fluxos de tráfego, realiza-se a alocação dos RBs com melhor qualidade em termos de SINR para os UEs selecionados que, teoricamente, possuem acesso irrestrito a todos os RBs no canal. Após a ordenação das classes e elementos não há uma limitação de recursos disponíveis aos elementos ordenados. Em consequência disso, problemas relacionados à "monopolização" dos recursos podem ocorrer. Se a demanda para o primeiro UE na fila de alocação for muito grande, os demais UEs perdem a oportunidade de recepção ou transmissão de dados durante o TTI considerado. Assim, somente no próximo TTI, com um novo ciclo de execução do CBRA, os elementos serão novamente classificados e ordenados para tornarem-se eleitos à uma nova etapa de alocação.

Embora algumas medidas possam ser realizadas para adaptar a ordenação adequada dos elementos a cada TTI, como uma definição mais adequada das características na etapa de composição de dados, é importante que existam estratégias para regulação na última etapa. Um exemplo é o trabalho desenvolvido pelo autor desta tese, não relacionado a um mecanismo CBRA e apresentado em [123], que propõe a limitação de recursos de acordo com métricas para avaliação de todo o sistema, calculadas regularmente, visando controlar a quantidade de recursos disponíveis aos elementos na fila de alocação.

4.6 Considerações sobre o Capítulo 4

Neste capítulo apresentou-se os conceitos básicos da estratégia CBRA e as etapas que compõem sua operacionalização. Cada etapa foi conceituada e descrita em nível de detalhamento um pouco mais avançado. Embora o CBRA seja uma abordagem relativamente recente na literatura, não há até o presente momento, material bibliográfico que ofereça um embasamento mais amplo para a estratégia, razão que motivou a escrita deste capítulo.

Assim, espera-se que o conteúdo apresentado sirva para auxiliar e ampliar a compreensão da estratégia, bem como da proposta desenvolvida e apresentada a seguir.

Capítulo 5

Proposta de Mecanismo de Alocação de Recursos baseado em *Clustering* para LTE-A

Este capítulo descreve a proposta de mecanismo de alocação de recursos baseado em *clustering* em sistemas LTE-A. Inicialmente, averigua-se o problema abordado na Seção 1.2 de forma mais detalhada, seguido do modelo de um sistema LTE-A, apresentado para definição do ambiente em que o problema é tratado e também para embasar os conceitos necessários a fim de facilitar a compreensão do mecanismo proposto, apresentado a seguir. Finalmente, realiza-se uma breve análise sobre a complexidade computacional do mecanismo proposto e os trabalhos relacionados.

5.1 Definição do Problema

A demanda por serviços móveis de telecomunicação tem crescido rapidamente nos últimos anos. Esse crescimento é causado pela evolução e diversidade das aplicações, pelo aumento dos equipamentos utilizados para acesso (*smartphones*, *tablets* e *notebooks*) e sua ampla disseminação, resultando em maior volume de tráfego e expectativa de qualidade por parte dos usuários [17]. Tal constatação expõe a necessidade de ampliação da capacidade e modernização dos sistemas de comunicação concernentes a redes móveis de banda larga sem fio.

Em resposta ao atual cenário, a estratégia CBRA propõe uma abordagem diferenciada para lidar com a alocação de recursos em condições de volume excessivo de tráfego e dados de controle. O CBRA realiza mapeamento de padrões sobre dados de fluxos de tráfego (e outros elementos da rede, se for o caso), em nível de rede, para auxílio à tomada de decisão sobre a alocação de recursos em sistemas de comunicação, considerando sobretudo o controle congruente de QoS das aplicações.

Apesar das propostas mais recentes de CBRA na literatura apresentarem soluções relevantes, elas falham em considerar de maneira apropriada a natureza vasta e diversificada que o *Big Data* apresenta. Exemplo disso pode ser observado no trabalho desenvolvido pelos autores em [18–

20] e também no trabalho em [25; 26]. Nos trabalhos mencionados, cada usuário apresenta uma determinada demanda de dados[1] por canal disponível, de maneira que o CBRA categoriza os usuários e estabelece um esquema adaptável de alocação em consonância com a demanda apresentada. As características coletadas correspondem diretamente à demanda individual por canal, que podem ser maiores do que o esperado se a demanda for muito alta e a capacidade do sistema em quantidade de canais também for muito grande.

A ocorrência de uma quantidade muito grande de características proporciona ao *clustering* dificuldades para estabelecimento de agrupamentos com capacidade adequada de diferenciação dos elementos. Isso ocorre devido ao problema da "maldição da dimensionalidade", detalhada a seguir na Seção 5.1.1. Para o CBRA, agrupamentos mal produzidos na classificação, independentemente do motivo, proporcionam má caracterização dos parâmetros necessários para provisionamento adequado de QoS às aplicações, além de dificultarem o gerenciamento eficiente da rede [12; 13].

Para solução do problema da "maldição da dimensionalidade", uma das alternativas mais utilizadas é a redução da dimensionalidade dos dados. Esta pode ser realizada por meio de algum método de seleção de características – como empregado em [21–24] e [28], por exemplo – ou aprendizado de características, sendo este último amplamente utilizado. Neste sentido, utiliza-se usualmente o PCA como mecanismo de aprendizado, que transforma um determinado conjunto de dados em um novo conjunto com quantidade reduzida de atributos, porém retendo uma representação da informação original com o máximo de variância possível. Entretanto, o PCA não é capaz de capturar correlações não-lineares entre as variáveis, limitando a capacidade da técnica das seguintes maneiras [47]: (i) a variância do novo conjunto produzido pode ser maximizada somente até um limiar específico, restrito a sistemas lineares; (ii) a capacidade de compressão, ou redução da dimensionalidade, torna-se reduzida com variância limitada, ou seja, um nível menor de variância apresentado na representação ocasiona o aumento da quantidade mínima de componentes a serem extraídos, impossibilitando uma redução ainda maior de dimensões para melhoria da classificação.

Outra limitação do PCA é o mau desempenho de representação, retratada quando há quantidade insuficiente de amostras ou elementos no conjunto [124]. Embora isso não seja um problema ao considerar *Big Data*, reconhecido por uma grande quantidade de elementos para análise, há situações em que a demanda em um sistema de comunicação pode tornar-se reduzida, caracterizando uma quantidade inferior de elementos e dificultando, portanto, uma redução adequada de dimensões pelo PCA. Isso resulta em má classificação do CBRA e, consequentemente, em uma definição inapropriada dos parâmetros de QoS e dos procedimentos para alocação de recursos.

Finalmente, um problema especificamente relacionado à classificação de elementos no CBRA diz respeito à parametrização da quantidade de *clusters* no procedimento de *clustering*. O valor do parâmetro que indica a quantidade de *clusters* para classificação é arbitrariamente informado nos mecanismos CBRA desenvolvidos em [18–26], impactando diretamente a qualidade dos agrupamentos produzidos pelos CBRAs propostos em cada trabalho. Nos trabalhos menci-

[1] Define-se *demanda de dados* as solicitações de dados realizadas por um usuário, por meio de um dispositivo, em um determinado sistema de comunicação.

onados, além de não haver procedimentos para validação[2], a quantidade informada de *clusters* é sempre fixa e arbitrária. Desta maneira, não há como saber se os agrupamentos formados correspondem ao ideal para o conjunto estabelecido e se as decisões tomadas para alocação de recursos seriam beneficiadas com uma melhor qualidade de agrupamentos produzidos.

5.1.1 Maldição da Dimensionalidade

O *clustering* é a principal atividade do CBRA. Por esse motivo os dados coletados para análise são estruturados para atender um modelo de classificação e agrupamento. Além disso, como observado anteriormente, as decisões de alocação realizadas pelo CBRA refletem indiretamente a estruturação dos dados compostos na primeira etapa do mecanismo assim como os padrões extraídos. É fundamental, portanto, estabelecer o máximo possível de informações que auxiliem a tomada de decisões no sentido de otimizar e ampliar a capacidade do sistema de comunicação. Contudo, apesar do excesso de informação trazer alguns benefícios, também resulta em "maldição da dimensionalidade" para o modelo do CBRA.

O problema da "maldição da dimensionalidade" é observado tipicamente em modelos em que a similaridade entre elementos é obtida pelo método dos "vizinhos mais próximos", ou *nearest neighbors*. Pode ser definido da seguinte forma: à medida em que a dimensionalidade de um conjunto de dados aumenta, a distância de seu elemento mais próximo $d_{\min}$ tende a aproximar-se da distância de seu elemento mais distante $d_{\max}$ [125]. Desta maneira, tem-se que [126]:

$$\lim_{n\to\infty}\left(\frac{d_{\max}(n)-d_{\min}(n)}{d_{\min}(n)}\right)\to 0 \tag{5.1}$$

$$\lim_{n\to\infty}\left(\frac{d_{\max}(n)}{d_{\min}(n)}\right)\to 1 \tag{5.2}$$

Para uma dimensão n, ou seja, para uma quantidade n de características, à medida em que seu valor tende ao infinito, a razão entre as distâncias $d_{\max}$ e $d_{\min}$ tende a um. Isso implica em maior dispersão dos elementos no hiperplano e, consequentemente, em maior dificuldade na distinção de agrupamentos durante sua formação.

Além disso, para o *clustering*, dependendo do conjunto de dados considerado, há um limiar de dimensões implicitamente estabelecido em que incluir uma dimensão adicional, ou seja, inserir uma característica adicional, não acrescenta informação útil ao modelo, mas apenas ruído[3] [43].

[2]Refere-se como validação de *cluster*, ou *cluster validation*, técnicas empregadas para avaliação da qualidade interna e externa dos *clusters* [120].

[3]O termo *ruído*, utilizado no contexto de *clustering* em um conjunto de dados, refere-se a alguma instância (ou característica) que "polui" o conjunto, ou seja, que agrega informação improdutiva.

5.2 Modelo do Sistema

Analisa-se um sistema tipo OFDMA sob as especificações do LTE-A. Considera-se uma rede de acesso via rádio (RAN) com uma eNB n responsável pela seleção e atribuição de RBs apenas no enlace *downlink*, em um total de K RBs ou subcanais, durante um TTI t a uma quantidade total I de UEs. Cada RB k é selecionado e alocado por uma eNB a um UE i e, de acordo com os princípios de teoria da informação, possui uma capacidade máxima definida por:

$$C_{i,k}^{n} = b \log_2 \left(1 + \gamma_{i,k}^{n}\right) \tag{5.3}$$

em que b refere-se à largura de banda do subcanal, e a variável $\gamma_{i,k}^{n}$ refere-se à SINR que o i-ésimo UE constata, após o recebimento do sinal, no referido RB (subcanal) k quando transmitido pela eNB n. A SINR é calculada, de maneira simplificada, por:

$$\gamma_{i,k}^{n} = \frac{\left|\mathbf{h}_{i,k}^{n}\right|^2 p_k^n}{\sum_{m=1,m\neq n}^{N} \left|\mathbf{h}_{i,k}^{m}\right|^2 p_k^m + \sigma^2} \tag{5.4}$$

O termo $\mathbf{h}_{i,k}^{n}$ refere-se ao vetor de coeficientes de canal entre a n-ésima eNB e o i-ésimo UE. Cada coeficiente é uma variável complexa aleatória com distribuição normal, média zero, variância unitária, independente e identicamente distribuída. p_k^n refere-se à potência estabelecida pela eNB n no subcanal k. O termo denominador apresentado refere-se ao sinal interferente, de maneira que σ^2 caracteriza o ruído de fundo.

A taxa máxima de transferência de dados alcançável por um UE i é definida por:

$$\hat{R}_i = \sum_{k=1}^{K} a_{i,k} C_{i,k}^{n} \tag{5.5}$$

Neste caso, $a_{i,k} \in \{0, 1\}$ é uma variável binária e indica a alocação ou não de um recurso do tipo RB ao UE considerado. Cabe à eNB estabelecer os valores mais adequados de $a_{i,k}$ de maneira que a seguinte restrição seja respeitada:

$$\sum_{i=1}^{I} \sum_{k=1}^{K} a_{i,k} \leq K \tag{5.6}$$

Embora as especificações do LTE-A considerem um UE i como elemento de última granularidade[4], o modelo pode ser estendido, sem qualquer perda de generalização, e considerar o escalonamento de um fluxo de tráfego f, ao invés do UE, como elemento de última granularidade. Isso ocorre pelo fato de que um UE pode executar múltiplas aplicações, e há pelo menos um fluxo de tráfego associado para cada aplicação em um dado canal (*downlink* ou *uplink*). Desta maneira, a variável a é modificada em seu índice e expressa como $a_{f,k}$. Isso implica em

[4]Granularidade é a extensão à qual um sistema é dividido em partes pequenas, ou o sistema propriamente dito em sua descrição ou observação [93].

afirmar que a alocação de recursos do tipo RB será específica para os fluxos de tráfego, embora para todos os efeitos isso não invalide a especificação, visto que um fluxo de tráfego pertence obrigatoriamente a um UE. Como referência, a subseção 2.3.1 relembra alguns conceitos de definição de fluxos de tráfego e sua operacionalização em um sistema LTE-A pelas especificações do 3GPP.

A estratégia CBRA proposta e apresentada na seção a seguir busca, indiretamente, selecionar os valores $a_{f,k}$ apropriados para otimização de desempenho do sistema assegurando parâmetros de QoS dos fluxos de tráfego, tais como vazão, atraso e taxa de perdas de pacotes, por exemplo.

5.3 Solução Proposta

Para atender os problemas mencionados na Seção 5.1, propõe-se um mecanismo CBRA para sistemas LTE-A contendo os aspectos a seguir, inéditos em estratégias CBRA, levando-se em conta todo o levantamento bibliográfico realizado pelo autor:

- implementação de um *Auto-encoder* para aprendizado de características durante a etapa de composição de dados, possibilitando a redução de dimensionalidade de um conjunto de dados inicial contendo todas as características possíveis de serem coletadas dos fluxos de tráfego, bem como dos UEs, em nível de rede[5]. Pretende-se, com o uso do *Auto-encoder*, contornar o problema da "maldição da dimensionalidade", reduzir a complexidade computacional do *clustering* com a consequente diminuição da quantidade de características, capturar correlações não-lineares entre os elementos e, com isso, estabelecer maior nível de compressão para conjuntos de dados assegurando variância elevada para representação. Além disso, o *Auto-encoder* é capaz de lidar com menor quantidade de elementos, garantindo uma boa redução de dimensionalidade mesmo em condições de baixa demanda no sistema;

- adaptação da etapa de classificação de elementos por meio da desparametrização da quantidade de classes informadas. Para tal, implementa-se o algoritmo de *clustering X-means*, que elimina a necessidade de informação da quantidade de *clusters* sobre o conjunto de dados retornado pelo *Auto-encoder*. Neste caso, o CBRA proposto dispensa o estabelecimento arbitrário do parâmetro relacionado à quantidade de classes, recorrendo apenas ao conjunto de dados estabelecido.

Importante observar que a combinação das técnicas *Auto-encoder* e *X-means* em um mesmo mecanismo CBRA proporciona uma condição exclusiva para a tomada de decisões. Enquanto o *Auto-encoder* é capaz de extrair automaticamente as características mais significativas dos elementos assegurando um alto fator de representação da informação original, o *X-means* obtém

[5]O termo *nível de rede*, ou mais precisamente *network-level*, refere-se ao contexto em que características são coletadas. Ou seja, ao invés de coletar características em nível de aplicação executada por um UE (*app-level*), coletam-se características em nível de rede para processamento de *Big Data* por meio do Aprendizado de Máquina [85; 127].

automaticamente a quantidade ideal de *clusters* a partir do novo conjunto de dados transformado. Dessa forma, pressupõe-se que a classificação resultante seja capaz de mapear os padrões mais relevantes dos dados originais possibilitando uma maior autonomia e eficiência.

A Figura 5.1 apresenta um diagrama da solução proposta, auxiliando a compreensão do mecanismo CBRA proposto para redes LTE-A. O mecanismo é executado a cada TTI.

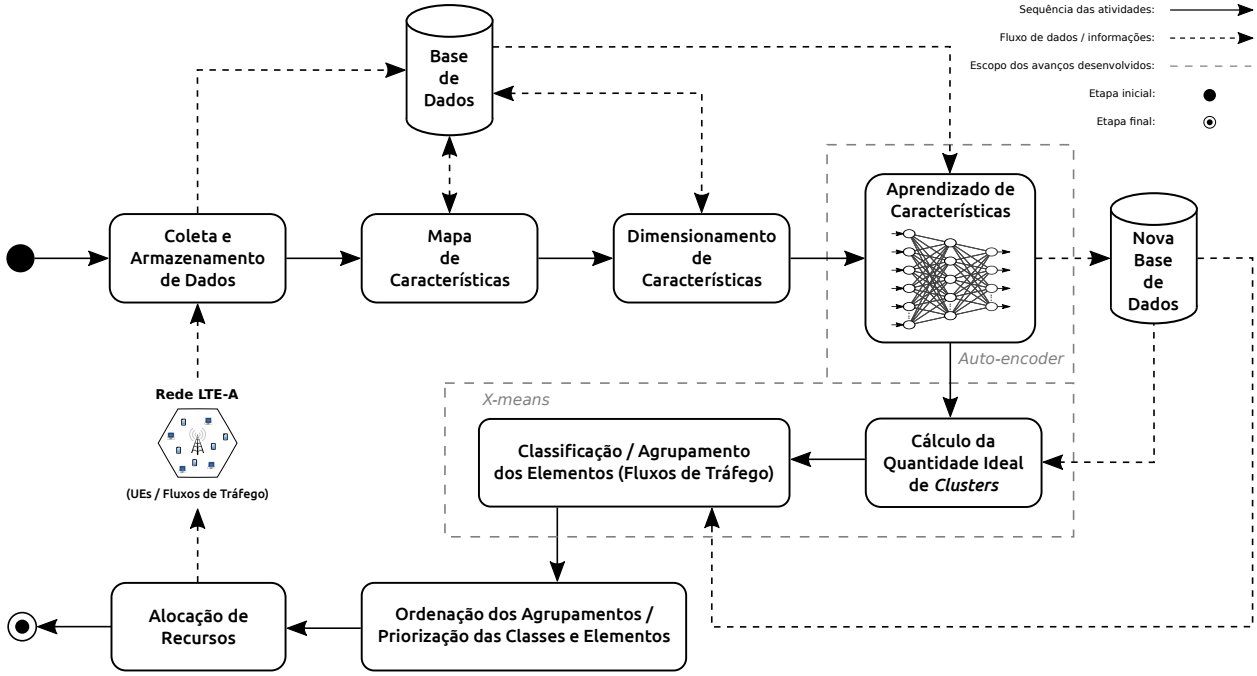


Figura 5.1: Ilustração do mecanismo de CBRA proposto compreendendo o aprendizado de características com *Auto-encoder* e o *clustering* não-paramétrico com *X-means*.

Inicialmente, realiza-se a coleta e armazenamento dos dados dos UEs e fluxos de tráfego em nível de rede. No LTE-A a coleta de dados é executada pela eNB que, além de receber informações de estado do canal referentes aos UEs, também gerencia os fluxos de tráfego transportados. As seguintes características são coletadas: CQI; SINR; atrasos dos pacotes; MBR; parâmetros de QCI como GBR, limite de perda, limite de atraso, entre outros; vazão média; atraso médio; *jitter* médio; taxa de perda de pacotes, ou *Packet Loss Rate* (PLR); taxa de descarte de pacotes (PDR); eficiência espectral; tamanho médio do *buffer*; etc. Importante observar que os dados coletados são restritos ao domínio da análise, como o tipo de enlace (*downlink* ou *uplink*), por exemplo. Ou seja, alguns tipos de dados obtidos no *downlink* podem ser (in)dispensáveis quando considera-se um enlace do tipo *uplink*.

Para a solução proposta considerou-se as características a seguir:

- **GBR:** unidade em *bits* por segundo (bps); estabelecido de acordo com o tipo da aplicação, em consonância com a Tabela 2.2;
- **Vazão média da aplicação:** unidade em *bits* por segundo (bps);

- **Atraso do pacote IP:** obtido pela diferença entre os *timestamps* de recebimento e de criação de um pacote IP a ser alocado; unidade em milissegundos;
- **Atraso médio do fluxo de tráfego:** obtido pela média móvel dos atrasos dos pacotes IP referentes a uma mesma aplicação; unidade em milissegundos;
- **Atraso tolerável:** estabelece o limite de atraso para pacotes IP referentes a um mesmo tipo de aplicação; unidade em milissegundos; valor estabelecido de acordo com a Tabela 2.2;
- **CQI:** indica a qualidade do subcanal para um UE; não possui unidade; varia de 0 a 15 (vide Tabela 2.1);
- **CQI médio:** obtido pela média móvel dos CQIs referentes a um mesmo subcanal e UE.

Após coleta e armazenamento dos dados, emprega-se um mapa de características para capturar algumas propriedades pouco evidentes, importantes para o aprendizado de características implementado e também para o mapeamento dos padrões necessários para classificação e posterior tomada de decisões. Tratam-se das funções a seguir:

$$\delta_f = \log\left(\frac{h_f}{D_f}\right) \tag{5.7}$$

$$\varrho_f = \log\left(\frac{R_f}{\bar{r}_f}\right) \tag{5.8}$$

h_f refere-se ao atraso do pacote *Head of Line* (HOL) de um fluxo de tráfego f, com limite de atraso D_f para a aplicação correspondente. R_f é a taxa mínima de *bits* (GBR) requerida para o fluxo de tráfego e $\bar{r}_f$ a taxa média de *bits* medida para o tráfego em questão. As variáveis δ_f e ϱ_f são definidas como novas características do mapa adicionadas ao elemento $\boldsymbol{x}_f$ considerado. A função log(), também denominada *log transformation*, é utilizada para "pré-normalizar" os valores bem como aproximar uma distribuição gaussiana dos elementos, em relação à característica considerada [45], auxiliando a classificação.

As funções definidas servem, respectivamente, para avaliar as condições proporcionais de atraso e vazão da aplicação. Um atraso acima do tolerável pela aplicação resultará em valores mais altos para a nova característica δ_f criada. Igualmente, uma vazão média abaixo do requerido acarretará em valores maiores observados para a característica ϱ_f.

Um procedimento de dimensionamento *Min-Max* (Seção 4.2.4.1) é realizado logo após a definição de novas características capturadas pelo mapa de características. O dimensionamento encarrega-se do ajuste de escala para facilitar o procedimento de aprendizado. Para aprendizado de características, utiliza-se um *Auto-encoder* com a mesma estrutura apresentada na Seção 4.2.2.2, em que a camada de codificação (*code layer*) implementa função de ativação do tipo *Softplus* [128] e as demais implementam a função *Leaky Rectified Linear Unit* (*Leaky* ReLU)

[129]. As quatro primeiras etapas no diagrama da proposta compreendem a etapa genérica de composição da base de dados, ilustrada na Figura 4.1 e descrita na Seção 4.2.

Encerrada a etapa de composição da base de dados, realiza-se o *clustering* no CBRA proposto. A quantidade de *clusters* para classificação é estabelecida automaticamente por meio do algoritmo *X-means*, que também encarrega-se do *clustering* na etapa de classificação de elementos.

Para ordenação dos elementos e *clusters*, consideram-se dois vetores $\boldsymbol{a}$ e $\boldsymbol{b}$ contendo, respectivamente, apenas a soma dos valores das m características dos elementos e dos centroides, na forma a seguir:

$$\boldsymbol{a} = (v_1, v_2, \ldots, v_f, \ldots, v_F), \quad v_f = \sum_{l=1}^{m} x_{f,l} \quad f \in \{1, 2, \ldots, F\} \tag{5.9}$$

$$\boldsymbol{b} = (w_1, w_2, \ldots, w_j, \ldots, w_J), \quad w_j = \sum_{l=1}^{m} \mu_{j,l} \quad j \in \{1, 2, \ldots, J\} \tag{5.10}$$

Cabe lembrar que F refere-se à quantidade total de fluxos de tráfego analisados no CBRA proposto enquanto J indica a quantidade de *clusters* estimados pelo algoritmo *X-means*. O algoritmo utilizado para ordenação na solução proposta é o *Heapsort* [130]. Escolheu-se o *Heapsort* pelo fato deste apresentar uma das menores ordens de complexidade computacional, para o pior caso, entre os inúmeros algoritmos de ordenação disponíveis na literatura, além de ser o algoritmo padrão para ordenação de vetores na linguagem C++[6].

A ordenação, realizada de forma decrescente na solução proposta, define a sequência de *clusters* e elementos que terão acesso aos recursos na etapa de alocação. Para simplificação da proposta, considera-se apenas a alocação de recursos do tipo RB. Entretanto, o CBRA proposto pode ser utilizado para alocação de outros tipos de recursos, como potência e fatores de precodificação[7]. A Figura 5.2 ilustra o esquema de alocação de RBs considerando o modelo do sistema LTE-A descrito e a ordem estabelecida na penúltima etapa. Dois RBs adjacentes formam uma unidade de alocação do tipo *Scheduling Block* (SB).

5.4 Complexidade Computacional

Pelo fato do mecanismo CBRA proposto lidar eventualmente com *Big Data*, a complexidade computacional passa a ser um critério fundamental para avaliação da capacidade e da qualidade da solução proposta. Quanto maior a complexidade computacional, maior o tempo de processamento das regras do CBRA e, em consequência disso, todo o sistema de comunicação gerenciado

[6]No C++ o *Heapsort* é o algoritmo utilizado na função *std::stable_sort()*, chamada para ordenação de estruturas de dados do tipo *std::vector*.

[7]Fatores de precodificação são valores numéricos utilizados para processamento espacial de sinal com a finalidade de mitigar interferência em cenários com múltiplos transmissores e receptores. A precodificação é utilizada para a formação de feixes de sinal (*beamforming*) com o intuito de direcionar o sinal para uma determinada região no espaço.

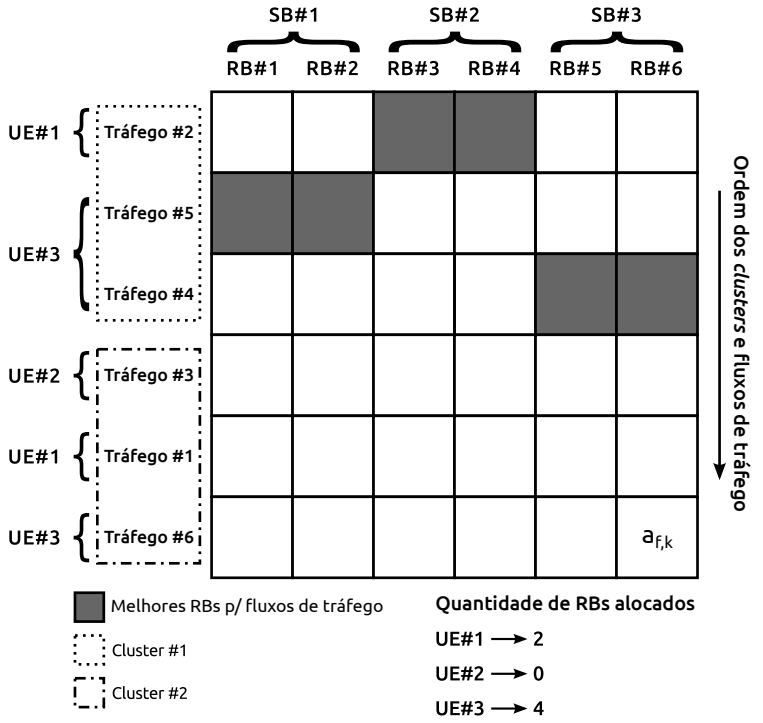


Figura 5.2: Ilustração de alocação de recursos do tipo RB em uma estratégia CBRA para sistemas LTE-A.

pelo CBRA é impactado de maneira geral. Apresenta-se aqui uma breve análise e consideração acerca da complexidade computacional no mecanismo CBRA proposto.

Para convencionamento das variáveis nesta seção, assume-se c como a quantidade de *clusters*, n como a quantidade de elementos e m a quantidade inicial de características ou atributos analisados no CBRA proposto.

5.4.1 Análise

A primeira etapa trata da coleta e armazenamento dos dados. É uma etapa simples. Logo, por intuição, a complexidade apresentada para o pior caso seria $\mathcal{O}(nm)$.

O mapa de características da proposta (vide Seção 5.3) aborda apenas duas variáveis ou características acrescentadas para cada elemento, calculadas por meio das equações (5.7) e (5.8). Neste caso, tem-se $\mathcal{O}(2n)$.

O dimensionamento de características também assume uma complexidade semelhante à etapa de coleta e armazenamento na ordem de $\mathcal{O}(nm)$.

A etapa de aprendizado de características possui complexidade relacionada ao algoritmo empregado. Se o aprendizado for realizado pelo PCA, a complexidade apresentada está na ordem de $\mathcal{O}(nm^2 + m^3)$ [51; 131], com $\mathcal{O}(nm^2)$ correspondente ao cálculo da matriz de covariância e $\mathcal{O}(m^3)$ ao procedimento SVD. Caso seja empregado um *Auto-encoder*, a ordem apresentada no

pior caso, durante o treinamento, é $\mathcal{O}(inw^5 + inw^4)$ [51], com i equivalente à quantidade de iterações para convergência e w equivalente à quantidade total de neurônios da rede neural, sendo $\mathcal{O}(inw^5)$ correspondente ao *backpropagation* e $\mathcal{O}(inw^4)$ ao *forward propagation*. Para um *Auto-encoder* treinado, a complexidade apresentada seria $\mathcal{O}(n\tilde{w}^4)$, com $\tilde{w}$ correspondente à quantidade de neurônios utilizados apenas para a codificação, eliminando-se as iterações e o *backpropagation* necessários para a fase de treinamento.

Via de regra, o *clustering* (etapa de classificação) é a tarefa que normalmente possui maior complexidade em uma estratégia CBRA. Neste sentido, selecionar um algoritmo para *clustering* que seja ágil e eficiente é fundamental. Nesta análise, menciona-se apenas a complexidade do algoritmo *X-means*, implementado na proposta. A ordem de complexidade para o pior caso do *X-means* é $\mathcal{O}(n \log k_{\max})$ [98], sendo $k_{\max}$ o parâmetro que indica a quantidade máxima de agrupamentos a ser testada pelo algoritmo – veja a Seção 3.4.6. Assume-se que o *X-means* execute uma quantidade $k_{\max}$ de iterações sucessivas do *K-means*, e o *K-means* possua uma complexidade reduzida na ordem de $\mathcal{O}(n)$ em sua versão otimizada [132], considerada na proposta.

A ordenação é realizada tanto para a quantidade de *clusters* c como para a quantidade de elementos n. O algoritmo selecionado, *Heapsort*, apresenta portanto uma complexidade de $\mathcal{O}(c \log c)$ para ordenação de *clusters* e $\mathcal{O}(n \log n)$ para ordenação de elementos [130].

Finalmente, a etapa de alocação de recursos apresenta uma complexidade $\mathcal{O}(nK)$, com K equivalente à quantidade total de subcanais disponíveis para alocação no sistema LTE-A.

5.4.2 Considerações

A partir da análise de complexidade realizada, tem-se que a complexidade computacional total do mecanismo CBRA proposto apresenta a ordem a seguir:

$$\mathcal{O}(nm) + \mathcal{O}(2n) + \mathcal{O}(nm) + \mathcal{O}(inw^5 + inw^4) + \mathcal{O}(n \log k_{\max}) + \mathcal{O}(c \log c) + \mathcal{O}(n \log n) + \mathcal{O}(nK) \tag{5.11}$$

Por propriedade, termos constantes não são considerados para efeito da representação reduzida da ordem de complexidade [133]. Semelhantemente, a representação reduzida de uma ordem composta considera somente o termo de maior ordem, visto que este tende a crescer mais rapidamente. Desta forma, as considerações a seguir são realizadas:

- a quantidade inicial de características m é constante, e geralmente $m < n$. Via de regra, não há um limite máximo definido para o valor de m, que pode, inclusive, ser $m > n$. Uma vez estabelecidas as características a serem coletadas, essas não podem ser modificadas no mecanismo a não ser que uma nova versão seja desenvolvida. Portanto, os termos $\mathcal{O}(nm)$ e $\mathcal{O}(2n)$ são reduzidos para $\mathcal{O}(n)$;
- a complexidade de treinamento do *Auto-encoder* pode ser reduzida para $\mathcal{O}(inw^5)$ (termo de maior ordem);

- para o *X-means*, o pior caso seria considerar a quantidade máxima de classes analisadas $k_{\max}$ igual à quantidade de elementos n, ou seja, com uma classe caracterizada por um elemento apenas. Nessas condições a complexidade considerada pode apresentar ordem $\mathcal{O}(n \log n)$ (*quasilinear*);

- para ordenação, se a quantidade ideal de *clusters* estimada pelo *X-means* for igual à quantidade de elementos, tem-se $c = n$ e uma complexidade resultante de $\mathcal{O}(n \log n)$;

- a complexidade necessária para a etapa de alocação de recursos do tipo RB é basicamente a mesma de um esquema de escalonamento convencional de RBs – $\mathcal{O}(nK)$ – sendo que a quantidade de subcanais K no LTE-A é constante. Desta forma, tem-se $\mathcal{O}(n)$ como ordem de complexidade para a última etapa.

Assim, dadas as considerações até então realizadas, a complexidade é inicialmente reduzida a:

$$\mathcal{O}(n) + \mathcal{O}(n) + \mathcal{O}(n) + \mathcal{O}(inw^5) + \mathcal{O}(n \log n) + \mathcal{O}(n \log n) + \mathcal{O}(n \log n) + \mathcal{O}(n) \tag{5.12}$$

e consequentemente:

$$\mathcal{O}(n) + \mathcal{O}(inw^5) + \mathcal{O}(n \log n) \tag{5.13}$$

Entre as três maiores ordens de complexidade apresentadas, destacam-se as relacionadas às etapas de aprendizado de características com *Auto-encoder* e *clustering* com o *X-means*.

Embora o *Auto-encoder* utilizado apresente quantidade constante de neurônios w, proporcionando pouco impacto computacional para o termo w^5, há que se considerar ainda a quantidade de iterações i para convergência de treinamento da rede neural. Esta quantidade é indeterminada, assim como a quantidade de elementos n, de maneira que passa a ser uma das variáveis de maior impacto para efeito da análise de complexidade.

Entretanto, pelo fato de reproduzir os parâmetros de entrada como parâmetros de saída, entre outros motivos, o *Auto-encoder* apresenta alta capacidade de generalização [134–136]. Logo, o treinamento de um *Auto-encoder* pode ser suficientemente realizado apenas uma única vez, desde que a convergência alcance um *erro quadrático de reconstrução* abaixo de uma tolerância estabelecida. Além disso, o treinamento pode ser realizado de maneira *offline*, antes da execução do mecanismo CBRA proposto, ou seja, os parâmetros do *Auto-encoder* empregado são estabelecidos *a priori* via treinamento externo, executado antes da implementação no CBRA proposto e fora do escopo do sistema de comunicação considerado.

Portanto, para o *Auto-encoder* empregado na proposta, considera-se somente a complexidade relacionada à codificação em uma rede neural treinada: $\mathcal{O}(n\tilde{w}^4)$. Na codificação, tem-se uma quantidade de neurônios reduzida praticamente pela metade ($\tilde{w} \approx \frac{w}{2}$) e igualmente constante. Assim, a complexidade computacional de codificação é reduzida para $\mathcal{O}(n)$.

Por conseguinte, a complexidade reduzida do mecanismo CBRA proposto é $\mathcal{O}(n) + \mathcal{O}(n) + \mathcal{O}(n \log n)$, com $\mathcal{O}(n \log n)$ como termo de maior ordem e, finalmente, como complexidade resultante na forma reduzida.

5.5 Trabalhos Relacionados

5.5.1 CBSA para Redes WDM

Nas referências em [18–20], os autores desenvolvem um algoritmo de escalonamento baseado em *clustering*, denominado *Clustering-Based Scheduling Algorithm* (CBSA), para redes de fibra óptica do tipo WDM. Inicialmente, o algoritmo utiliza um modelo de predição de valores para construção de uma matriz de demanda $\boldsymbol{D}$. A matriz é caracterizada por uma quantidade n de linhas e w colunas, em que n indica a quantidade de *nós* (dispositivos) e w a quantidade de canais disponíveis para alocação. A matriz é preenchida com valores numéricos que quantificam a demanda de cada *nó* por canal.

Logo após a construção da matriz de demanda $\boldsymbol{D}_{n \times w}$, o algoritmo aplica *clustering* com *K-means* sobre a matriz produzida com uma quantidade arbitrária de *clusters* k. Obtidos os *clusters*, uma ordenação é realizada por meio do algoritmo *Quicksort* [137] considerando a soma dos valores dos centroides formados. Finalmente, uma matriz de escalonamento $\boldsymbol{S}_{w \times t}$ é produzida por um algoritmo específico desenvolvido para o sistema de comunicação considerado a partir dos *clusters* e nós ordenados, de maneira que t denota o comprimento dos *slots* de tempo para alocação. Desta forma, organiza-se a cada quadro f um esquema de alocação de recursos considerando a demanda gerada no sistema.

A Figura 5.3 ilustra parte do esquema desenvolvido pelos autores de acordo com a descrição realizada.

5.5.2 ATDSA

Com foco em sistemas LTE-A no enlace *downlink*, os autores em [21–24] desenvolvem uma arquitetura dinâmica e adaptativa de escalonamento compreendendo os domínios do tempo e da frequência, bem como o gerenciamento de filas (*buffers*). Denominado *Adaptive Time Domain Scheduling Algorithm* (ATDSA), o trabalho desenvolvido recorre inicialmente a um esquema pré-estabelecido de classificação de tráfego em quatro tipos: controle; tempo-real; *streaming*; e *background*. Cada tráfego inicialmente classificado é enfileirado em um *buffer* específico para o tipo estabelecido. Visto que o LTE-A oferece mecanismos para classificação de tráfego, utiliza-se tais recursos do sistema para realizar a classificação inicial.

Um escalonamento inicial dos fluxos de tráfego na fila de controle é realizado por meio de um algoritmo *Round Robin* (RR) tradicional. Semelhantemente, para as filas de tráfegos em tempo-real (RT) e *streaming* (NRT), os autores desenvolvem um algoritmo do tipo QoS-*aware* para escalonamento, detalhado especificamente em [21]. Finalmente, para a fila de tráfegos do

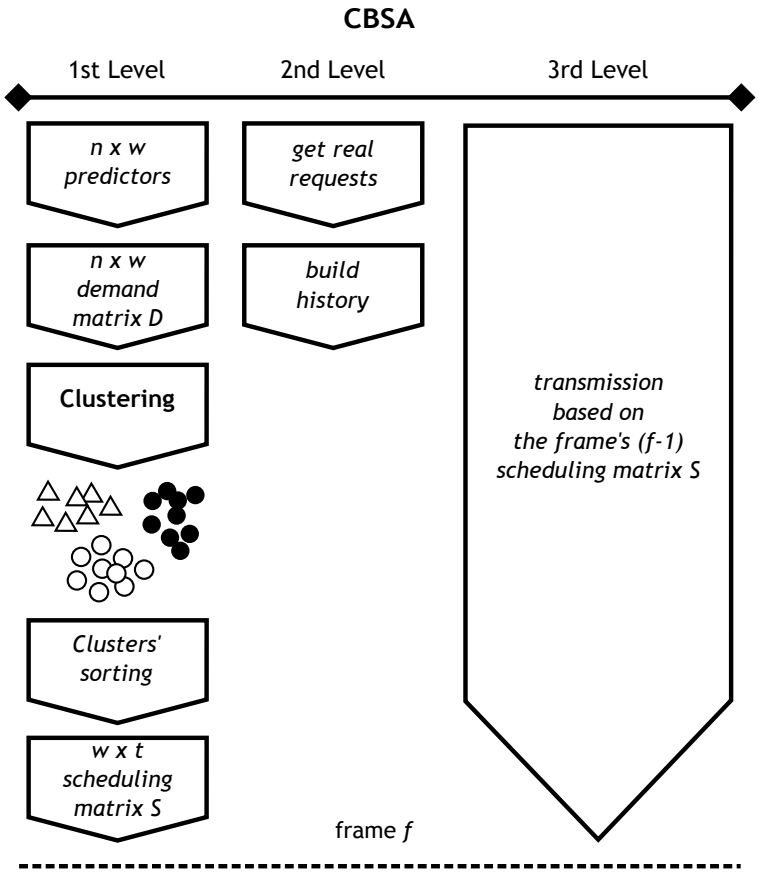


Figura 5.3: Ilustração do CBSA para sistemas WDM. Extraído e adaptado de [20].

tipo *background*, um algoritmo *Proportional Fair* (PF) é empregado.

O escalonamento do tipo QoS-*aware* é aprimorado por meio de uma regra dinâmica denominada *Hebian Learning*, que regula a proporção de recursos entre os tráfegos RT e NRT alocados pelo algoritmo responsável. Para finalizar o escalonamento no domínio do tempo, utiliza-se *clustering* com *K-means*, nos mesmos moldes da proposta descrita na Seção 5.5.1, para tráfegos do tipo RT, caracterizados apenas pelos valores médios de PDR.

Por fim, os recursos são alocados por meio de escalonamento no domínio da frequência, em que os melhores RBs são alocados para os fluxos de tráfego seguindo a ordem dos RBs com melhor qualidade (CQI mais alto), as filas com maior prioridade e, finalmente, a ordem estabelecida para as filas após o escalonamento no domínio do tempo. Nesta etapa também realizam-se medições dos parâmetros de QoS utilizados para caracterização dos tráfegos RT no *clustering*. A Figura 5.4 apresenta o desenho da arquitetura desenvolvida.

5.5.3 AG com *K-means* e SVM

Em [25; 26], os autores produzem uma solução para lidar com a alocação de RBs em sistemas LTE-A com *Big Data*. Para tal são utilizados algoritmos de Aprendizado de Máquina, organizados em fases, para capturar os padrões da demanda e otimizar o sistema a partir de um

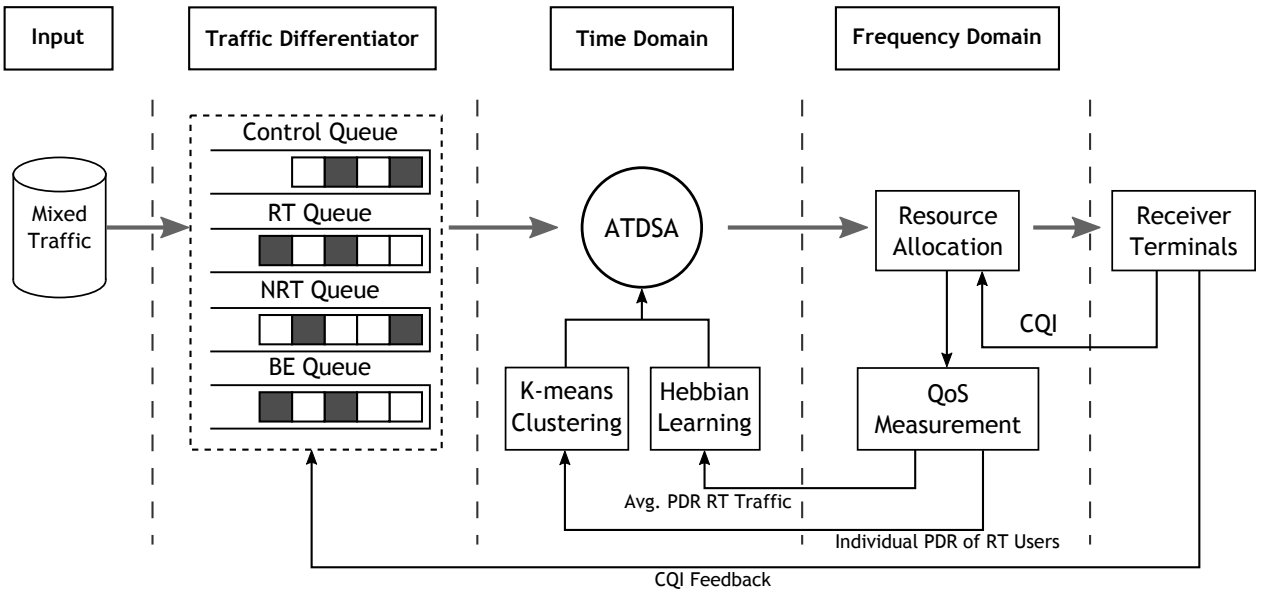


Figura 5.4: Arquitetura do mecanismo ATDSA. Extraído e adaptado de [22; 23].

problema multi-objetivo, cujo foco é maximizar a vazão total enquanto assegura os requisitos mínimos de QoS para as aplicações.

Inicialmente, o mecanismo constrói uma base de dados de demanda $\boldsymbol{D}_{n\times m} = (\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_n)^\dagger$ em que cada vetor $\boldsymbol{d}$ da matriz é caracterizado pelas demandas dos m usuários para um determinado RB, em um total de n RBs disponíveis no sistema. Depois da estruturação da base de dados, um algoritmo *K-means* arbitrariamente parametrizado com $k = 2$ classifica os elementos da base e encaminha os dois centroides resultantes para construção de uma segunda base de dados $\boldsymbol{X}$, utilizada para treinamento de um SVM. A matriz de treinamento $\boldsymbol{X}$ possui um tamanho limitado e é atualizada a cada TTI.

O treinamento do SVM condiciona o algoritmo à classificação de uma nova demanda, cujo resultado de classificação corresponde a um novo indivíduo (vetor $\boldsymbol{z}$) inserido na população de um AG, utilizado para otimizar a seguinte função:

$$\max\left[w_1 f_1 + w_2 f_2\right] \tag{5.14}$$

sendo que:

$$f_1 = \max \sum_{i=1}^{m}\sum_{j=1}^{n} C_{i,j} a_{i,j} \quad \forall a_{i,j} \in \{0,1\} \tag{5.15}$$

$$f_2 = \min \sum_{i=1}^{m_{\mathrm{GBR}}}\sum_{j=1}^{n} \left(r_{i,j} a_{i,j} - R_{i,j}\right) \tag{5.16}$$

$C_{i,j}$ refere-se à eficiência espectral para o i-ésimo usuário no j-ésimo RB, $a_{i,j}$ é uma variável

binária que indica a alocação ou não de um RB para um usuário, $r_{i,j}$ indica a vazão instantânea para o RB referido ao usuário, $R_{i,j}$ corresponde à vazão mínima requerida para o fluxo de tráfego pertinente ao usuário i, e m_{GBR} refere-se à quantidade total de usuários do tipo GBR, tal que $m_{\text{GBR}} + m_{\text{NonGBR}} = m$. Os pesos w_1 e w_2 são ajustados para atender à regra $w_1 + w_2 = 1$ e $w_2 = \frac{m_{\text{GBR}}}{m}$.

A Figura 5.5 demonstra a arquitetura do mecanismo desenvolvido.

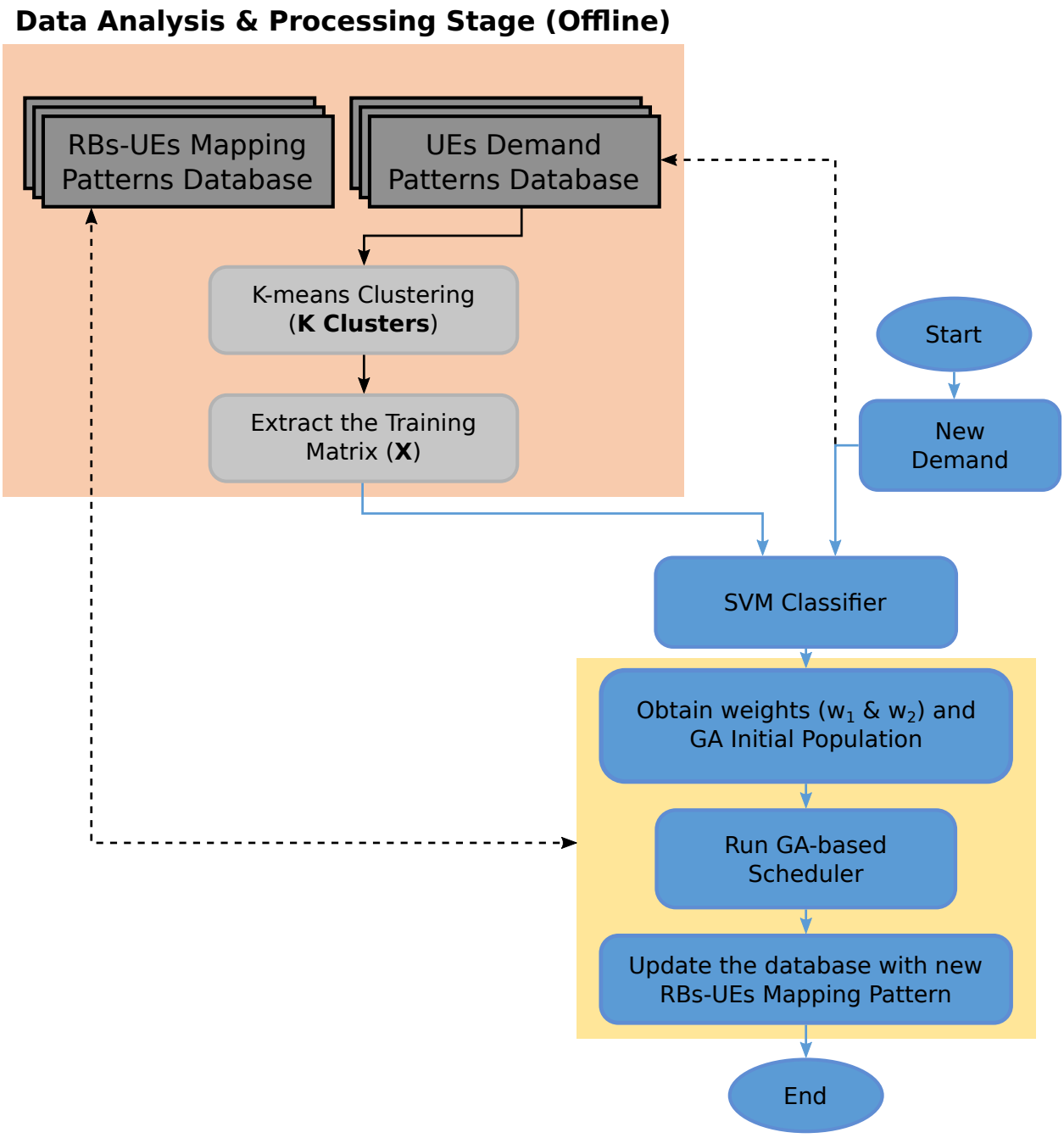


Figura 5.5: Arquitetura do mecanismo AG com *K-means* e SVM. Extraído e adaptado de [26].

A população inicial do AG é caracterizada por uma matriz $\boldsymbol{Z}$, inicializada com parâmetros

aleatórios, e posteriormente atualizada com os vetores classificados pelo SVM, cujos valores de cada elemento $\boldsymbol{z} = (z_1, z_2, \ldots, z_n)$ correspondem aos índices de UEs do sistema. Trata-se basicamente de uma matriz para seleção e alocação de UEs em cada um dos n RBs disponíveis. A Figura 5.6 ilustra o mapeamento de um elemento $\boldsymbol{z}$ na sua forma transposta, enquanto a Figura 5.7 ilustra o cruzamento (*crossover*) de dois indivíduos realizado pelo AG, no mecanismo.

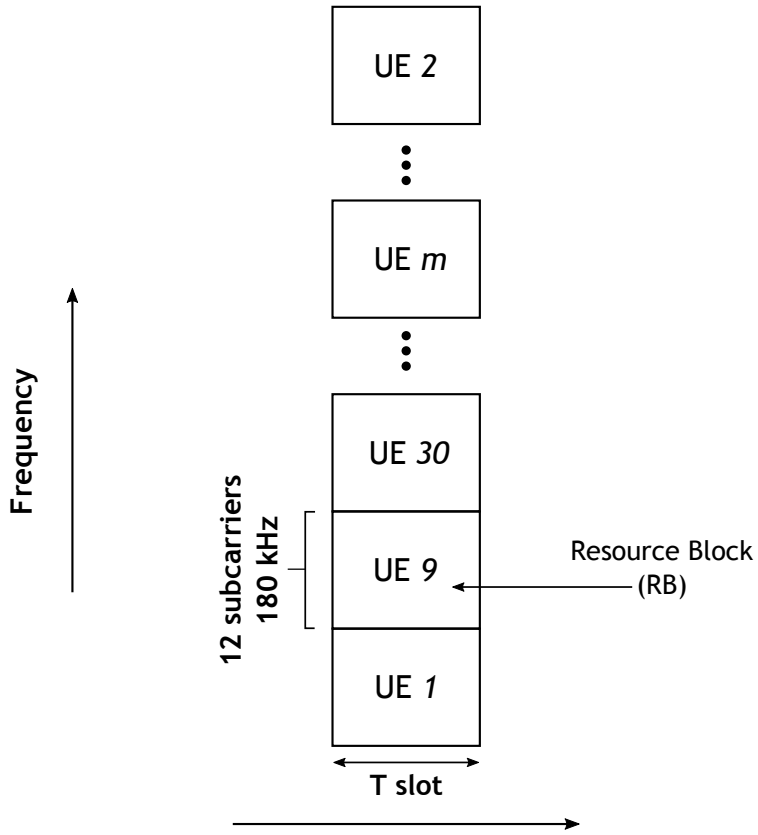


Figura 5.6: Ilustração do mapeamento de um indivíduo para inclusão em AG. Extraído e adaptado de [26].

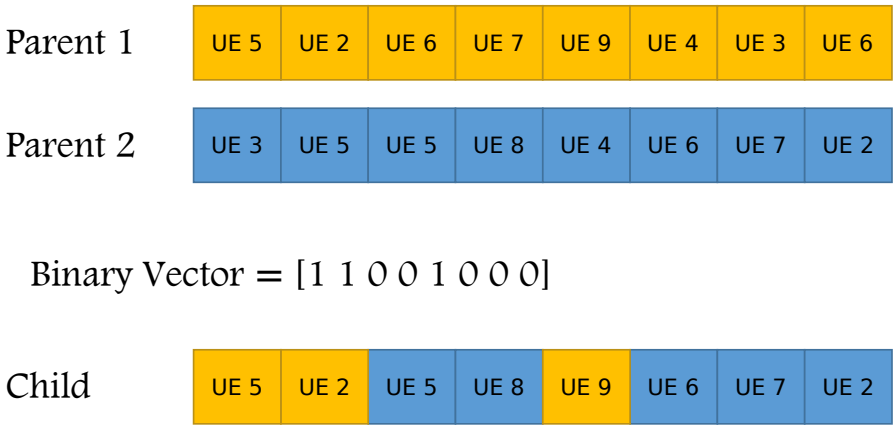


Figura 5.7: Ilustração do cruzamento de indivíduos, sem a ocorrência de mutação, no AG desenvolvido. Extraído e adaptado de [26].

5.5.4 Observações

A principal diferença entre a solução proposta e os trabalhos relacionados diz respeito às características coletadas. Enquanto os autores em [18–20; 25; 26] limitam-se a coletar apenas a demanda apresentada nos canais e subcanais, e em [21–24] consideram somente o PDR, a proposta apresentada nesta tese possibilita a obtenção de todos os dados possíveis para caracterização de um elemento, seja ele um fluxo de tráfego ou dispositivo de usuário. Obviamente, em decorrência do excesso de características, emprega-se o aprendizado de características com *Auto-encoder* para aprimorar a representação de um elemento, reduzindo sua dimensionalidade.

O *clustering* não-paramétrico também é um aspecto notável na solução proposta nesta tese. Enquanto todos os demais trabalhos relacionados assumem um parâmetro arbitrário para a quantidade de classes, o mecanismo proposto estima o parâmetro ideal a partir do conjunto de dados estabelecido, agregando maior autonomia ao procedimento.

No aspecto da complexidade computacional, a Tabela 5.1 apresenta, na forma reduzida, as complexidades relacionadas a cada um dos trabalhos mencionados nesta seção.

Tabela 5.1: Complexidade relacionada ao mecanismo proposto e aos trabalhos correlatos

Mecanismo Proposto	**Ordem de Complexidade**	**Observação**
CBRA c/ *Auto-encoder*	$\mathcal{O}(n \log n)$	Seção 5.4
CBRA p/ WDM	$\mathcal{O}(n)$	Complexidade descrita em [20]
ATDSA	$\mathcal{O}(n)$	Ordem de complexidade inferida a partir da descrição do mecanismo
AG c/ *K-means* e SVM	$\mathcal{O}(n^2)$	Análise de complexidade realizada nos trabalhos em [26; 138]

5.6 Considerações sobre o Capítulo 5

Este capítulo descreveu o mecanismo CBRA proposto para sistemas LTE-A, os detalhes de sua implementação e as considerações necessárias referentes aos trabalhos relacionados. O mecanismo proposto coleta uma ampla e variada escala de valores que representam diversos tipos de características do sistema em nível de rede. Para evitar que o excessivo volume de dados coletados afete o desempenho do sistema e a capacidade de classificação dos fluxos de tráfego, o mecanismo recorre ao aprendizado de características por meio de um *Auto-encoder*. Não obstante, o mecanismo dispensa a parametrização da quantidade de classes para o procedimento de *clustering*, usando o algoritmo *X-means* para estimar o valor ideal a partir da estrutura apresentada nos dados coletados.

O *Big Data* oferece diversas vantagens, mas também inúmeros desafios para gerenciamento de sistemas de comunicação, especialmente as tecnologias de redes móveis de banda larga sem fio, como o LTE-A, por exemplo. Neste sentido, prover soluções que atuem no atendimento aos

desafios hodiernamente apresentados é fundamental. Razão pela qual propostas como a descrita neste capítulo são desenvolvidas e apresentadas.

Capítulo 6

Avaliação do Mecanismo Proposto

Neste capítulo são definidos os parâmetros e cenários utilizados para avaliação da proposta apresentada no Capítulo 5. As métricas para avaliação são estabelecidas e os resultados obtidos são posteriormente apresentados com a discussão apropriada. Finalmente, algumas considerações importantes são feitas para fechamento da análise conduzida neste capítulo.

6.1 Cenário e Parâmetros

Como observado anteriormente no Capítulo 1, visto que não há conhecimento de mecanismos CBRA implementados em sistemas reais, o custo e complexidade exigidos para análise em ambientes reais são altos, e deseja-se avaliar o mecanismo CBRA proposto em condições experimentais controladas, definiu-se a simulação computacional de redes como critério de avaliação, sob as especificações do LTE-A. Para tal, selecionou-se um simulador de redes LTE-A em nível de sistema (*system-level*) para implementação da solução proposta no Capítulo 5, bem como dos trabalhos relacionados na literatura.

A ferramenta selecionada é o *LTE-Sim* [139]. Trata-se basicamente de um simulador de eventos para redes LTE, discretizado no domínio do tempo, desenvolvido em C++ e com suporte a orientação a objetos. Selecionou-se o *LTE-Sim* pelo fato de apresentar uma curva de aprendizado mais fechada, ou seja, ele possui um código-fonte mais fácil de ser utilizado e modificado ao longo do uso, o que possibilitou modificações expressivas para suporte ao desenvolvimento e avaliação do mecanismo CBRA proposto.

O *Auto-encoder* e o *X-means*, utilizados respectivamente para aprendizado de características e *clustering* não-paramétrico, foram implementados no *LTE-Sim* utilizando a mesma linguagem base do simulador.

O *Auto-encoder* implementado na plataforma de simulação do *LTE-Sim* é usado apenas para codificação (somente *encoder*) e caracterizado por parâmetros copiados de um mesmo *Auto-encoder* com exatamente a mesma estrutura, porém previamente implementado e treinado com o *Keras* [140], uma biblioteca de Aprendizado Profundo desenvolvida em *Python*. O motivo

é simplesmente acelerar o treinamento *offline* do *Auto-encoder* proposto bem como obter um menor erro de representação na codificação, visto que o *Keras* oferece melhores ferramentas e condições para implementação, convergência de treinamento e otimização de redes neurais.

A Figura 6.1 ilustra o cenário avaliado, que apresenta um *layout* de célula hexagonal contendo uma eNB com apenas um único setor e uma antena para transmissão. Apenas o enlace *downlink* é avaliado. A carga de tráfego é caracterizada pela quantidade de UEs no sistema, em que cada UE executa somente uma instância de cada aplicação considerada. Os UEs são distribuídos aleatoriamente e movem-se livremente na célula seguindo o modelo de mobilidade apresentado como parâmetro.

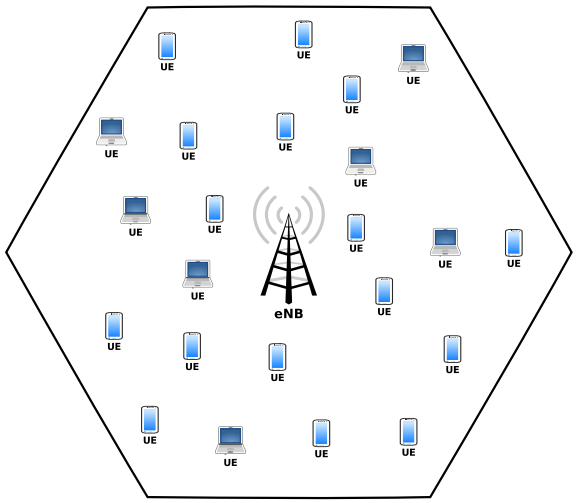


Figura 6.1: Ilustração de cenário utilizado para avaliação da solução proposta em um sistema LTE-A.

A Tabela 6.1 apresenta os valores dos parâmetros utilizados para simulação do modelo selecionado. Buscou-se aproximar os valores de referência nas análises conduzidas em [24; 25; 141], mantendo o suporte à mobilidade dos dispositivos em uma velocidade razoável. Adicionalmente, avaliam-se, além da solução proposta no Capítulo 5 (referida como CBRA com *Auto-encoder*), os seguintes mecanismos CBRA no cenário apresentado:

- *Adaptive Time Domain Scheduling Algorithm* (ATDSA) [24];
- Algoritmo Genético com *K-means* e SVM – (AG c/ *K-means*) [25];
- CBRA aprimorado para Vídeo (CBRA – Video) [141];
- CBRA aprimorado para VoIP (CBRA – VoIP) [141];

Os mecanismos CBRA mencionados em [141] para avaliação, também foram desenvolvidos pelo autor desta tese e, embora não façam parte da solução proposta, são caracterizados especificamente por:

- seleção manual e mapa de características com a finalidade de otimizar o desempenho de aplicações de vídeo e VoIP, respectivamente;
- *clustering* realizado por meio dos algoritmos *X-means* e FCM (Seção 3.4.5).

O mecanismo CBRA proposto em [18–20] não foi avaliado pelo fato de não ter sido desenvolvido para sistemas LTE.

Para simplificar a proposta e delimitar o escopo da pesquisa, o uso de uma ferramenta para estimação da dimensionalidade intrínseca[1] não foi considerado. Visto que um estimador de dimensionalidade intrínseca não é utilizado, a quantidade de características a serem extraídas deve ser estática e arbitrária. Mais detalhes relacionados à estimação da dimensionalidade intrínseca podem ser encontrados em [142; 143].

A implementação de um estimador de dimensionalidade, com atribuição dinâmica da quantidade de características a serem extraídas, demandaria treinamento *online* do *Auto-encoder* na solução proposta, tornando seu uso inviável em decorrência da alta complexidade computacional requerida durante a execução do mecanismo CBRA proposto (discutido na Seção 5.4). Desta forma, a quantidade de características extraídas[2] pelo CBRA com *Auto-encoder* foi arbitrariamente estabelecida como $m = 3$. Tal valor foi selecionado pelos seguintes motivos: (*i*) facilitar a visualização dos elementos para análise, uma vez que, graficamente, a representação de conjuntos de dados acima de três dimensões não é trivial; (*ii*) manter a variância do conjunto em um limiar mínimo tolerável[3]; (*iii*) assegurar o máximo possível de redução de dimensionalidade a fim de produzir melhor qualidade de *clusters* no CBRA.

No modelo de propagação utilizado, o parâmetro d refere-se à distância entre a unidade transmissora e receptora.

A simulação é realizada de acordo com o método de *Monte Carlo*, em que executam-se sucessivas rodadas de simulação considerando o cenário apresentado. Os geradores de números aleatórios utilizados no método são semeados com diferentes valores a cada rodada executada, a fim de assegurar melhor amostragem para as variáveis consideradas no modelo.

6.1.1 Comparação entre Mecanismos CBRA

Adicionalmente, a Tabela 6.2 apresenta uma breve comparação relacionada à quantidade de características inicialmente inserida e posteriormente extraída/selecionada entre os mecanismos CBRA considerados para avaliação. Observações relacionadas aos mecanismos também foram acrescentadas para facilitar a análise de parametrização de cada um.

[1] Define-se como *dimensionalidade intrínseca* a menor quantidade de variáveis necessárias para representação apropriada dos dados.

[2] A quantidade de características a serem extraídas (m) corresponde diretamente à quantidade de neurônios da camada de codificação do *Auto-encoder* empregado.

[3] Nas análises preliminares, a variância média dos conjuntos de dados analisados manteve-se acima de 98,5% para um *Auto-encoder* com $m = 3$ neurônios na camada de codificação. Com $m < 3$ o valor de variância média cai drasticamente.

Tabela 6.1: Parâmetros de Simulação

Parâmetro	Valor
Modulação	OFDMA
Banda	2.1 GHz
Largura de Banda	20 MHz
Raio da Célula	500 m
Tipo de Cenário	*Single-Cell*
Duração do Quadro	10 ms
Duração do Subquadro (TTI)	1 ms
Estrutura do Quadro	FDD
Modo de Antena	SISO 1x1
Tipo da Antena	Omnidirecional
Potência - eNB	46 dBm
Potência - UE	23 dBm
Ruído	-174 dBm/Hz
Modelo de Propagação	Urbano ($P_L = 128.1 + 37.6 \log d$)
Tempo de Simulação	60 s
Tempo de Tráfego	54 s
Quantidade de UEs	10 – 120
Velocidade dos UEs	30 Km/h
Modelo de Mobilidade	*Manhattan*
Aplicações	Vídeo, VoIP e BE
Tipo de Tráfego de Vídeo	H.264 440 Kbps
Tipo de Tráfego VoIP	G.711 64 Kbps
Modelo de Tráfego BE	*Web* – Pareto
Qtde de Características Extraídas (m)	3

Nota-se que o CBRA com *Auto-encoder* é a única proposta que preconiza o estabelecimento das características a critério do operador da rede.

Tabela 6.2: Quantidade de características consideradas nos mecanismos CBRA avaliados

Mecanismo CBRA	Qtde Inicial	Qtde Extraída/Selecionada	Observação
ATDSA	1	–	Apenas o PDR definido como característica
AG c/ *K-means*	Indeterminada	–	Quantidade de características depende do número de usuários e subcanais
CBRA c/ *Auto-encoder*	Indeterminada	3	Características e quantidade são estabelecidas a critério do operador da rede
CBRA – Video	6	3	Aprendizado de características realizado com PCA
CBRA – VoIP	5	2	Mapa para definição das características selecionadas

6.2 Métricas para Avaliação

Estabelece-se aqui alguns critérios relevantes para avaliação da solução proposta. Inicialmente, para avaliação da qualidade interna dos *clusters* produzidos pelo *clustering*, utiliza-se o índice *Xie-Beni* (XB) [144; 145], que basicamente estabelece uma pontuação para a relação existente entre a compactação e separação dos *clusters*. Embora existam outras métricas para avaliação de qualidade interna, selecionou-se o índice XB por apresentar valores mais estáveis

sob diferentes condições de *clustering* nas avaliações preliminares. O índice XB é calculado na forma a seguir:

$$I_{\mathrm{XB}} = \frac{\sum_{i=1}^{J} \sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{C}}_i} d^2(\boldsymbol{x}, \boldsymbol{c}_i)}{n \cdot \min_{i \neq j} d^2(\boldsymbol{c}_i, \boldsymbol{c}_j)} \tag{6.1}$$

De maneira que n indica a quantidade total de elementos do conjunto de dados, $d(\boldsymbol{x}, \boldsymbol{c}_i)$ a distância do elemento $\boldsymbol{x}$ ao centroide $\boldsymbol{c}_i$ pertinente, e $d(\boldsymbol{c}_i, \boldsymbol{c}_j)$ é a distância entre o centroide $\boldsymbol{c}_i$ de seu vizinho $\boldsymbol{c}_j$. Um valor de índice I_{XB} baixo indica melhor qualidade interna de *clustering*.

Para verificação de equilíbrio na distribuição dos recursos, avalia-se o índice de justiça de *Jain* [146], calculado da forma a seguir:

$$F_i = \frac{\left(\sum_{i=1}^{n} \bar{r}_i\right)^2}{n \sum_{i=1}^{n} \bar{r}_i^2} \tag{6.2}$$

A variável n indica, igualmente, a quantidade total de elementos do conjunto de dados e $\bar{r}_i$ a vazão média individual obtida para o fluxo de tráfego representado pelo i-ésimo elemento em questão.

Para compreender melhor o desempenho das aplicações em termos de vazão média individual, avalia-se a distribuição acumulada ou *Cumulative Distribution Function* (CDF) obtida em relação ao parâmetro em questão. As curvas no gráfico da CDF ajudam a compreender a probabilidade de uma aplicação selecionada encontrar-se em uma determinada faixa de vazão média.

A vazão média individual, o atraso médio e a taxa de perda de pacotes, ou *Packet Loss Rate* (PLR), também são analisados para efeito de verificação de desempenho e suporte a QoS pelos mecanismos avaliados.

As métricas selecionadas são úteis para avaliar o desempenho dos mecanismos CBRA em diversos contextos. Por exemplo, enquanto o índice de justiça busca verificar se o sistema está sendo igualitário na alocação dos recursos, a vazão média individual informa se os tráfegos GBR estão sendo atendidos de acordo com sua demanda.

6.3 Resultados de Simulação

Inicialmente, para elucidação da classificação na estratégia, apresenta-se na Figura 6.2 uma ilustração de *clustering* realizado pelo mecanismo CBRA – Video, capturado em um TTI aleatório durante uma das simulações conduzidas, em que os elementos são classificados em duas classes distintas ($k = 2$) e distribuídos em um gráfico de dispersão cujos eixos representam os componentes extraídos pelo PCA. Tem-se especificamente três componentes, de maneira que é plotado primeiramente o gráfico referente aos dois primeiros componentes e, posteriormente, todos os três. Observa-se a separação visual entre os *clusters* e elementos apresentados, caracterizando uma classificação dinâmica de fluxos de tráfego para o TTI considerado.

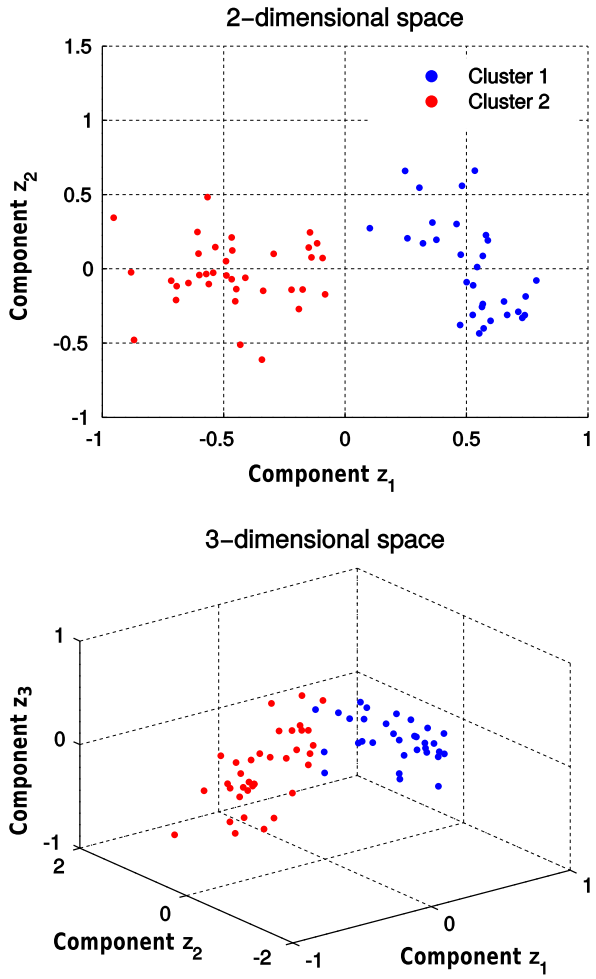
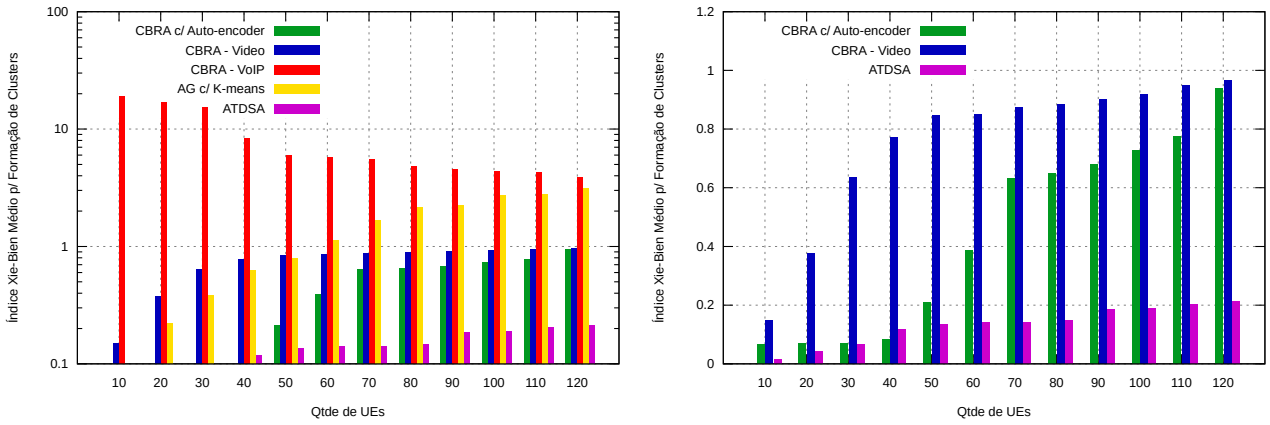


Figura 6.2: *Clustering* de elementos no CBRA capturado em um TTI aleatório, com $k = 2$. Extraído de [141].

Os primeiros resultados apresentados são referentes aos índices XB obtidos no *clustering* realizado em cada um dos mecanismos CBRA avaliados. A Figura 6.3 apresenta os gráficos contendo os valores médios obtidos na avaliação. Dada a diferença de grandeza entre alguns valores, apresenta-se inicialmente, na Figura 6.3a, uma escala logarítmica para auxiliar a referência, e na Figura 6.3b, uma escala normal contendo apenas os resultados dos mecanismos que apresentaram menor índice.

A tendência inicial, exceto para o mecanismo CBRA – VoIP, é de que os mecanismos apresentem baixo índice XB com baixa carga de tráfego e esse valor aumente à medida em que a carga de tráfego também aumentar. Entretanto, com o aumento da carga de tráfego, observa-se uma convergência nos valores dos índices XB obtidos para um valor comum, próximo a 1. O motivo não é evidente e pode ser analisado posteriormente de maneira mais acurada.

Nota-se que a quantidade de características consideradas no *clustering* não afeta diretamente o índice XB obtido. Embora o ATDSA utilize apenas uma única característica para representar



(a) Escala logarítmica (b) Escala normal

Figura 6.3: Índice médio XB para formação de *clusters*.

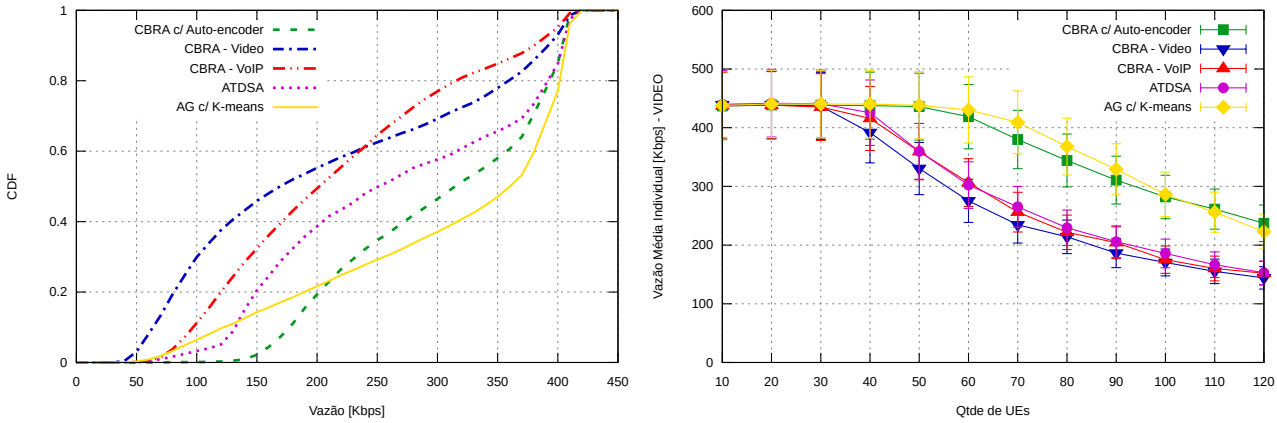
os elementos (vide Tabela 6.2), ele também apresenta os valores mais baixos de índice XB. Isso explica-se pelo fato de que o ATDSA realiza *clustering* de apenas uma parcela dos elementos, ou seja, o ATDSA considera para *clustering* apenas os fluxos de tráfego em tempo real (RT) [24], reduzindo a quantidade de elementos normalmente classificados e modificando a qualidade interna obtida. Desta forma, para que o ATDSA possa apresentar resultados mais congruentes com os demais mecanismos avaliados, sua implementação deveria ser modificada para considerar o *clustering* de todos os elementos, ou então todos os demais mecanismos CBRA deveriam ser avaliados sob as mesmas condições de implementação do ATDSA, classificando apenas fluxos de tráfego RT.

A questão é que, para avaliar os demais mecanismos CBRA sob as mesmas condições de implementação do ATDSA, ou seja, considerando *clustering* apenas para tráfegos em tempo real, seria necessário modificar os mecanismos CBRA ora analisados. O contrário também seria válido, ou seja, para que o ATDSA seja capaz de realizar *clustering* para todos os tipos de tráfego, semelhantemente aos demais mecanismos CBRA, o ATDSA deveria ser descaracterizado de sua implementação original. Visto que o objetivo é avaliar os mecanismos em suas implementações originais (vide Seção 1.4), a descaracterização dos mecanismos prejudicaria uma análise coerente.

Ainda assim, nas condições atuais, o CBRA com *Auto-encoder* apresenta baixos valores de índice XB e próximos aos valores obtidos pelo ATDSA em cenários com baixo volume de tráfego, indicando maior qualidade para formação de *clusters*. Cabe observar que, diferentemente do ATDSA, o mecanismo CBRA com *Auto-encoder* realiza *clustering* de todos os elementos (fluxos de tráfego) do sistema. Atribui-se ao aprendizado de características um melhor desempenho em termos de qualidade interna do *clustering* e maior capacidade de representar os dados e suas particularidades, em comparação à seleção de características.

As Figuras 6.4a, 6.5a e 6.6a apresentam, respectivamente, a CDF relacionada ao parâmetro da vazão para as aplicações de vídeo, VoIP e *Web*, obtidas pelos mecanismos avaliados. Percebe-se que o CBRA com *Auto-encoder*, juntamente com o AG com *K-means*, possui uma maior probabilidade de apresentar bom desempenho para o tráfego de vídeo em termos de vazão.

Por outro lado, as aplicações VoIP e *Web* apresentam um resultado aquém do desejável para o mecanismo proposto.



(a) CDF – Vídeo – Parâmetro: Vazão.

(b) Vazão média individual.

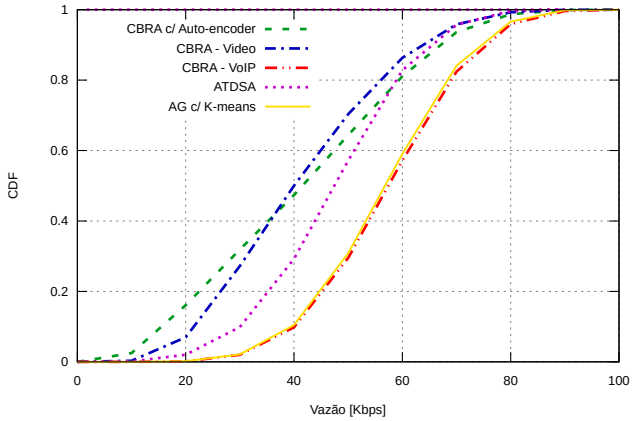
Figura 6.4: CDF e vazão média individual obtidos para aplicação de Vídeo.

A queda de desempenho relacionada à vazão das aplicações VoIP e *Web* pelo CBRA com *Auto-encoder* também é demonstrada, respectivamente, nas Figuras 6.5b e 6.6b, que apresentam a vazão média individual obtida. Em contrapartida, a Figura 6.4b expõe taxas de vazão similares entre os mecanismos CBRA com *Auto-encoder* e AG com *K-means*, que apresentam-se entre as maiores no gráfico.

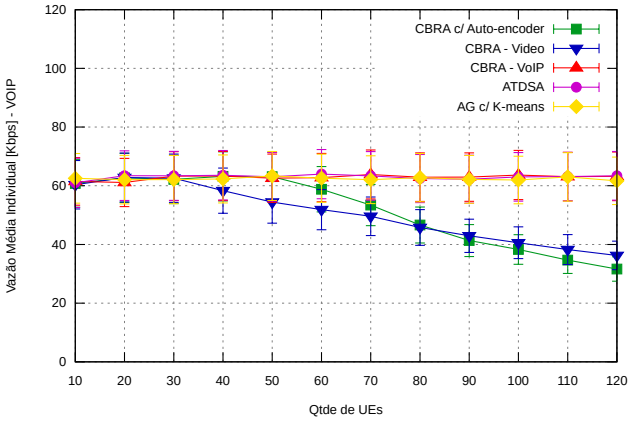
Justifica-se a queda relacionada às aplicações VoIP e *Web* para o CBRA com *Auto-encoder* pelo fato de que as características coletadas e mapeadas pelo mecanismo (vide Seção 5.3) tendem a apresentar valores abaixo das condições desejadas para maior priorização dos fluxos de tráfego relacionados às aplicações mencionadas. Por exemplo, os parâmetros de GBR requeridos para aplicações VoIP e *Web* normalmente são menores do que aplicações de vídeo, assim como os parâmetros de atraso tolerável tendem a ser maiores, em contrapartida. Desta forma, o CBRA com *Auto-encoder* tende a conceder menor prioridade para fluxos de tráfego de aplicações VoIP e *Web*, resultando em uma queda repentina de desempenho assim que um determinado nível de saturação de carga de tráfego é atingido.

O mecanismo AG com *K-means* consegue alcançar altas taxas para praticamente todas as aplicações consideradas na avaliação. Entretanto, tal feito é realizado a custo de maior esforço computacional (vide Tabela 5.1). O CBRA com *Auto-encoder*, por sua vez, não é explicitamente orientado à otimização de nenhuma aplicação (apesar das características acrescentadas via mapa) e, além disso, possui complexidade computacional relativamente baixa. Ainda assim, a proposta desenvolvida alcança bom desempenho para aplicação de vídeo, que apresenta-se como a aplicação com maior demanda no cenário avaliado.

Para confirmar um melhor desempenho do CBRA com *Auto-encoder* especificamente para a aplicação de vídeo, a Figura 6.7 apresenta o desempenho relacionado ao índice de justiça. Observa-se que, embora o AG com *K-means* alcance bom desempenho relacionado à vazão para vídeo, ele não distribui os recursos entre as instâncias da aplicação tão bem quanto o CBRA

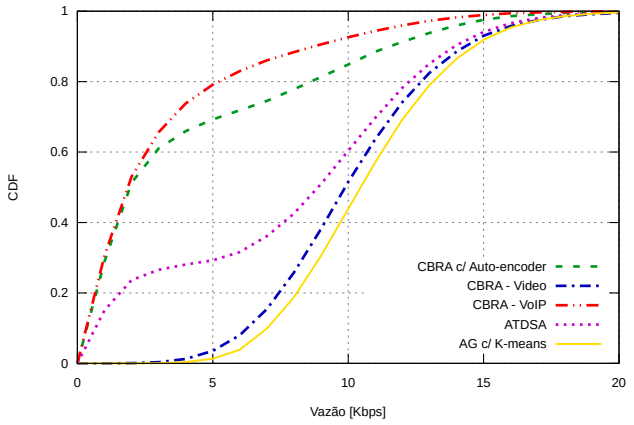


(a) CDF – VoIP – Parâmetro: Vazão.

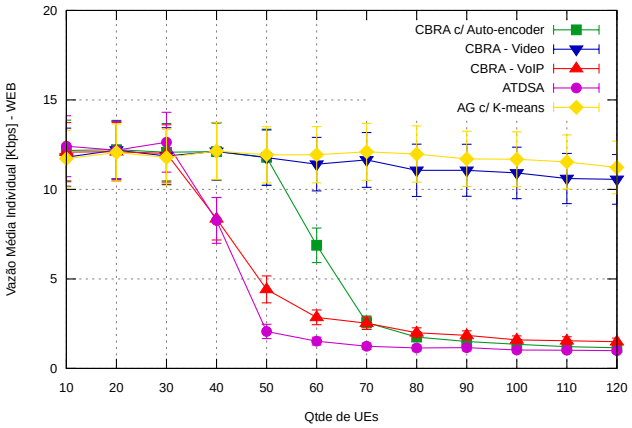


(b) Vazão média individual.

Figura 6.5: CDF e vazão média individual obtidos para aplicação VoIP.



(a) CDF – *Web* – Parâmetro: Vazão.



(b) Vazão média individual.

Figura 6.6: CDF e vazão média individual obtidos para aplicação *Web*.

com *Auto-encoder*.

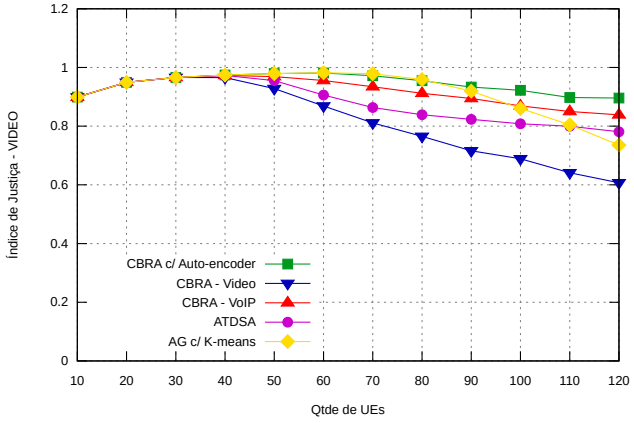


Figura 6.7: Índice de justiça obtido para aplicação de Vídeo.

Para as demais aplicações no entanto, VoIP e *Web*, o índice de justiça apresentado nas Figuras 6.8 e 6.9, respectivamente, ilustram uma diferença significativa entre o AG com *K-means* e o CBRA com *Auto-encoder*, de maneira que nota-se algum tipo de compensação na distribuição dos recursos. Enquanto o AG com *K-means* tende a distribuir melhor os recursos para aplicações VoIP e *Web* em detrimento da aplicação de vídeo, o CBRA com *Auto-encoder* faz exatamente o oposto. Isso ocorre pelo fato de que o CBRA com *Auto-encoder* caracteriza e classifica melhor as aplicações com maior demanda no sistema, observado por um maior índice de justiça medido para aplicações como o vídeo, por exemplo.

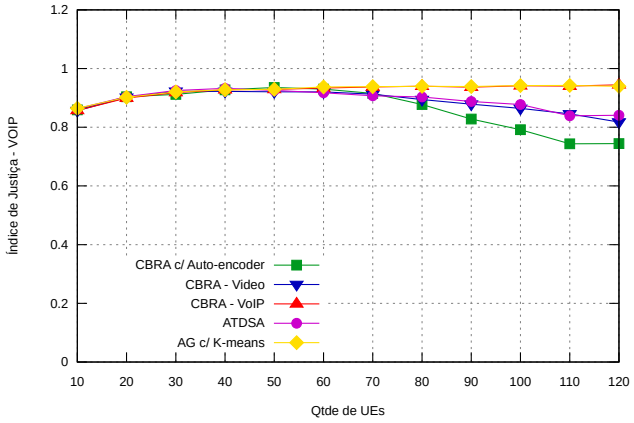


Figura 6.8: Índice de justiça obtido para aplicação VoIP.

Continuando, as Figuras 6.10a e 6.10b apresentam, respectivamente, o atraso médio e o PLR obtidos para aplicação de vídeo. Nota-se um desempenho similar entre o CBRA com *Auto-encoder* e o AG com *K-means*, com uma ligeira vantagem do AG com *K-means* relacionada ao PLR dependendo da carga de tráfego considerada.

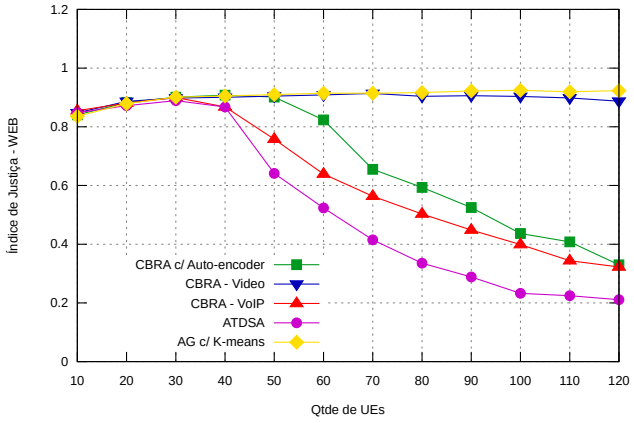
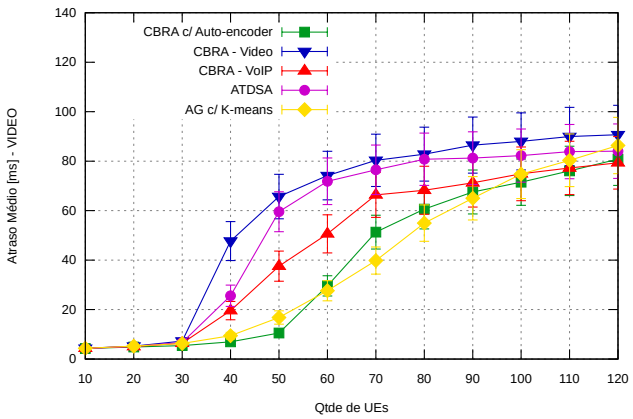


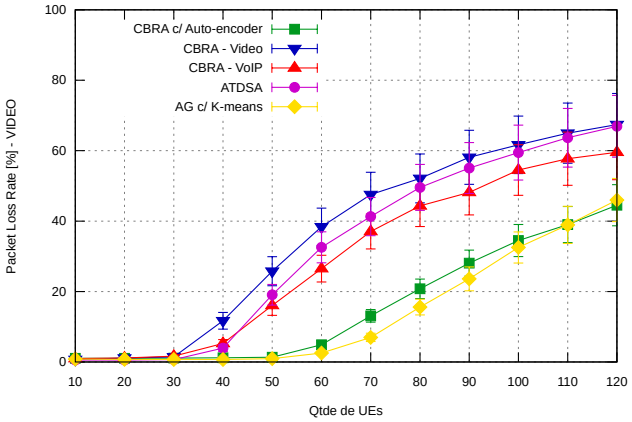
Figura 6.9: Índice de justiça obtido para aplicação *Web*.

Com relação ao VoIP, embora os resultados apresentados para atraso médio e PLR nas Figuras 6.11a e 6.11b não apresentem um desempenho tão promissor para o CBRA com *Auto-encoder*, observa-se contudo uma boa manutenção de QoS para a aplicação considerando uma carga de tráfego abaixo de 60 UEs na célula. A queda de desempenho relacionado a atraso

médio e PLR para VoIP também justifica-se pelas características coletadas e mapeadas pelo mecanismo proposto, que tendem a privilegiar fluxos de tráfego pertinentes a aplicações mais exigentes em termos de QoS, como o vídeo em tempo real. Cabe reiterar que cada UE executa uma instância de cada uma das três aplicações analisadas: vídeo, VoIP e *Web*. Além disso, os requisitos de atraso máximo estabelecidos para as aplicações de vídeo e VoIP, de maneira a assegurar o provisionamento de QoS, são apresentados na Tabela 2.2.



(a) Atraso médio – Vídeo.



(b) PLR – Vídeo.

Figura 6.10: Atraso médio e PLR para aplicação de Vídeo.

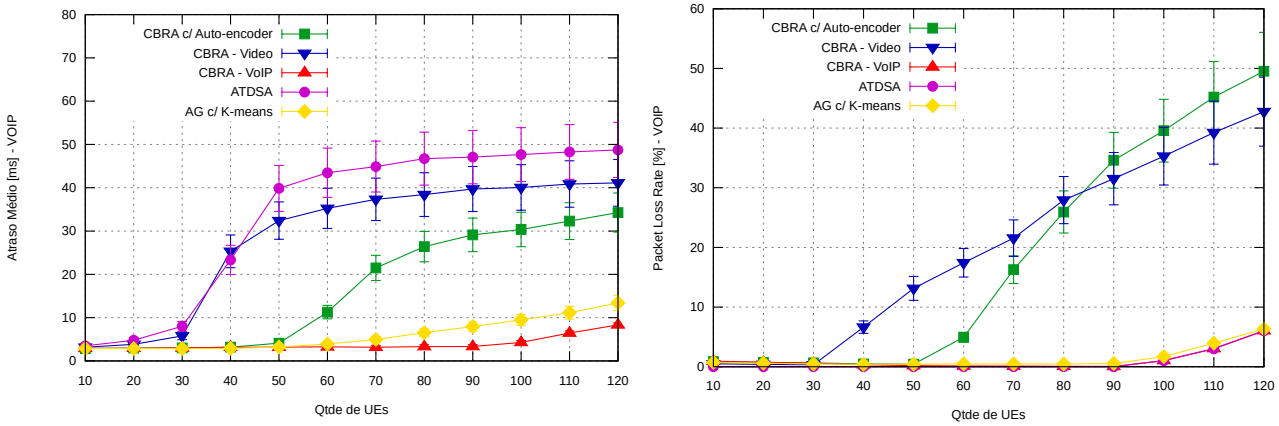
Registra-se também que, para cargas de tráfego até 50 UEs no sistema, o CBRA com *Auto-encoder* apresenta notável desempenho em todas as métricas e aplicações avaliadas, o que não é observado para todos os mecanismos, exceto o AG com *K-means*, desenvolvido específica e explicitamente para otimização.

De maneira geral, os resultados apresentados confirmam que o aprendizado de características com *Auto-encoder* e o *clustering* não-paramétrico com *X-means*, incluídos na proposta, ajudam de fato a caracterização adequada de parâmetros de QoS e a classificação dinâmica de fluxos de tráfego em um mecanismo CBRA.

6.4 Considerações Finais

Encerra-se aqui a apresentação da proposta de um novo mecanismo CBRA com aprendizado de características e *clutering* não-paramétrico para redes LTE-A. Este capítulo, em particular, apresentou o cenário, os parâmetros e as métricas para avaliação do mecanismo proposto juntamente com os resultados obtidos por meio de simulação e os comentários a respeito de cada um.

Inicialmente, os resultados apresentaram melhor classificação, observada a partir de uma melhor qualidade interna de *clusters* produzidos, apontando uma maior vantagem do procedimento de aprendizado de características sobre o método tradicional de seleção de características. Para o aprendizado de características, entre os algoritmos avaliados (CBRA com *Auto-encoder* e



(a) Atraso médio – VoIP.

(b) PLR – VoIP.

Figura 6.11: Atraso médio e PLR para aplicação VoIP.

CBRA – Video), notou-se uma vantagem do *Auto-encoder* sobre o PCA no sentido de capturar informações que possibilitassem posteriormente uma melhor formação de agrupamentos pelo *clustering*.

Uma vez que o suporte ao aprendizado de características é capaz de, a partir da inserção do máximo possível de características, extrair as informações mais relevantes para representação de um fluxo de tráfego no CBRA, não há necessidade em realizar variadas combinações de características para avaliar e observar a adaptabilidade do mecanismo, assim como era realizado nas propostas de CBRA aprimoradas para vídeo e VoIP [141].

O mecanismo CBRA proposto apresenta um bom desempenho obtido para a aplicação de vídeo em todas as métricas consideradas. Entretanto, embora o desempenho do mecanismo proposto não proporcione uma vantagem para aplicações como VoIP e *Web*, por exemplo, ele não permite que essas mesmas aplicações experimentem um desempenho muito ruim em relação aos demais mecanismos CBRA avaliados nas simulações.

A realização de *clustering* no CBRA a cada TTI, apesar de não ser essencial para alocação de recursos em um quadro (ou subquadro), ou seja, o *clustering* pode ser realizado em um intervalo de tempo maior, é interessante para que o mecanismo ofereça uma adaptação mais "afinada" com as características apresentadas. Normalmente, embora as características dos tráfegos não sofram mudanças expressivas a cada TTI, a qualidade do canal experimentada entre a eNB e os UEs pode variar significativamente neste intervalo de tempo, fazendo-se necessária a realização de *clustering* caso deseja-se considerar a qualidade de canal para efeito de alocação de recursos.

Finalmente, é importante registrar que a avaliação de mecanismos CBRA por si só também é significativa. Visto que novas áreas interdisciplinares, como Sistemas de Controle de Tráfego com Aprendizado de Máquina, por exemplo, vêm emergindo [10], é conveniente que haja análises e referências correlatas na literatura de antemão.

Capítulo 7

Conclusões Gerais

O crescente volume de dados produzidos em sistemas de comunicação, desencadeado por uma ampla e gradual demanda de serviços de acesso, tem afetado significativamente o desenvolvimento das novas tecnologias de comunicação. Embora o *Big Data* proporcione uma grande variedade de informações para obtenção de conhecimento, ele também promove enormes desafios. Neste sentido, o Aprendizado de Máquina apresenta-se como uma notável ferramenta para extração de conhecimento do *Big Data* e mitigação de alguns problemas relacionados.

O trabalho descrito nesta tese apresentou uma nova estratégia de alocação de recursos baseada em *clustering* (CBRA), um tipo de aplicação de Aprendizado de Máquina Não-supervisionado, implementado em redes móveis de banda larga sem fio do tipo LTE-A. A estratégia CBRA proposta implementa um *Auto-encoder* para aprendizado de características dos fluxos de tráfego da rede, na etapa de composição da base de dados, e um algoritmo *X-means* para realização de *clustering* não-paramétrico, na etapa de classificação dos elementos.

O *Auto-encoder* captura as informações mais relevantes dos fluxos de tráfego para suporte a QoS no CBRA e contorna o problema da "maldição da dimensionalidade", responsável por dificultar a classificação adequada de tráfego em mecanismos CBRA com grande volume e variedade de dados. O *X-means*, por sua vez, é capaz de efetuar o *clustering* dispensando a informação prévia e arbitrária da quantidade de *clusters*, tornando o mecanismo mais dinâmico e orientado ao modelo dos dados.

Após avaliação por meio de simulação computacional, os resultados obtidos para o CBRA proposto apresentaram boa qualidade interna de *clusters*, indicando uma melhor dinâmica de classificação de tráfegos levando-se em conta a quantidade total inicial de características analisadas. Além disso, houve um bom desempenho para o mecanismo proposto relacionado à aplicação de vídeo, que é caracterizada por parâmetros mais rígidos de QoS solicitados no cenário estabelecido, compensado entretanto por um desempenho aquém do esperado para as aplicações VoIP e *Web*, que solicitaram parâmetros de QoS menos rigorosos no mesmo cenário utilizado para avaliação.

De maneira geral, o mecanismo CBRA proposto é capaz de caracterizar e oferecer suporte aprimorado de QoS a aplicações em tempo-real que apresentam alta demanda de recursos no

sistema, denotando custo computacional relativamente baixo, maior capacidade de abstração da informação e maior autonomia para a tomada de decisões.

Adicionalmente, tem-se uma tecnologia como alternativa ao modelo convencional de classificação de tráfegos para estabelecimento de parâmetros de QoS em sistemas LTE-A. Com o mecanismo CBRA proposto, a classificação de tráfego para fins de controle de QoS pode ser realizada dinamicamente.

Não obstante os benefícios oferecidos, o desenvolvimento da estratégia de CBRA proposta proporcionou algumas reflexões importantes sobre aspectos relevantes para a tecnologia porém não abordados neste trabalho. Tais aspectos possuem alto potencial no sentido de produzir avanços ainda mais significativos, caso sejam desenvolvidos e implementados. Menciona-se:

- a investigação adequada acerca das implicações decorrentes do cálculo da dimensionalidade intrínseca em um conjunto de dados no CBRA, juntamente com o desenvolvimento de um algoritmo para estimar o parâmetro ideal de dimensionalidade a ser extraída na etapa de aprendizado de características. Supõe-se que o uso de um estimador de dimensionalidade seja capaz de agregar maior autonomia e precisão da classificação em um CBRA;

- o controle de ordenação de classes e elementos orientado ao desempenho do sistema de maneira geral, tornando o mecanismo capaz de otimizar a tomada de decisões em função de métricas pré-estabelecidas. Trata-se basicamente em realizar uma abordagem dinâmica de ponderação de características na etapa de ordenação de classes e elementos. Para tal, sugere-se por exemplo, a implementação de mecanismos de Aprendizado por Reforço capazes de analisar métricas de desempenho no sistema, como a vazão agregada, e em posse dessa informação sejam capazes de ajustar automaticamente os fatores de ponderação utilizados para ordenação e priorização de fluxos de tráfego;

- a implementação e avaliação de soluções para controle de acesso aos recursos na etapa final do CBRA, responsável pela alocação de recursos disponíveis. Com isso, verifica-se se o controle de acesso aos recursos é adequado para aperfeiçoamento de um mecanismo CBRA.

Certamente, há uma grande expectativa gerada direta e indiretamente pela sociedade e comunidade técnico-científica sobre os avanços relacionados a mecanismos de controle de tráfego, alocação de recursos, entre diversos outros procedimentos, baseados em Aprendizado de Máquina e executados em sistemas de comunicação. Este trabalho contribuiu, portanto, para suprir parte dessa expectativa.

Referências Bibliográficas

[1] C. V. Networking Index, "Cisco Forecast and Methodology, 2016-2021 – White Paper," *San Jose, CA, USA*, 2016.

[2] G. Mobile and D. Traffic, "Cisco Visual Networking Index (VNI): Forecast and Methodology, 2015-2020 – White Paper," no. February, 2016.

[3] A. Bora and K. K. Sarma, "Big Data and Deep Learning for Stochastic Wireless Channel," in *Computational Intelligence in Sensor Networks*, pp. 307–334, Springer, 2019. https://doi.org/10.1007/978-3-662-57277-1_13.

[4] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A Survey of Machine Learning for Big Data Processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 67, 2016. https://doi.org/10.1186/s13634-016-0382-7.

[5] S. J. Nawaz, S. K. Sharma, S. Wyne, M. N. Patwary, and M. Asaduzzaman, "Quantum Machine Learning for 6G Communication Networks: State-of-the-Art and Vision for the Future," *IEEE Access*, vol. 7, pp. 46317–46350, 2019. https://doi.org/10.1109/ACCESS.2019.2909490.

[6] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From Data Mining to Knowledge Discovery in Databases," *AI magazine*, vol. 17, no. 3, p. 37, 1996. https://doi.org/10.1609/aimag.v17i3.1230.

[7] M.-S. Chen, J. Han, and P. S. Yu, "Data Mining: An Overview from a Database Perspective," *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 6, pp. 866–883, 1996. https://doi.org/10.1109/69.553155.

[8] K. Zheng, Z. Yang, K. Zhang, P. Chatzimisios, K. Yang, and W. Xiang, "Big Data-driven Optimization for Mobile Networks toward 5G," *IEEE Network*, vol. 30, no. 1, pp. 44–51, 2016. https://doi.org/10.1109/MNET.2016.7389830.

[9] J. Moysen and L. Giupponi, "From 4G to 5G: Self-organized Network Management Meets Machine Learning," *Computer Communications*, 2018. https://doi.org/10.1016/j.comcom.2018.07.015.

[10] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "State-of-the-art Deep Learning: Evolving Machine Intelligence toward Tomorrow's Intelligent Network Traffic Control Systems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017. https://doi.org/10.1109/COMST.2017.2707140.

[11] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data Clustering: a Review," *ACM Computing Surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999. https://doi.org/10.1145/331499.331504.

[12] H. Shi, H. Li, D. Zhang, C. Cheng, and X. Cao, "An Efficient Feature Generation Approach based on Deep Learning and Feature Selection techniques for Traffic Classification," *Computer Networks*, vol. 132, pp. 81–98, 2018. https://doi.org/10.1016/j.comnet.2018.01.007.

[13] M. Shafiq, X. Yu, A. K. Bashir, H. N. Chaudhry, and D. Wang, "A Machine Learning approach for Feature Selection Traffic Classification using Security Analysis," *The Journal of Supercomputing*, pp. 1–26, 2018. https://doi.org/10.1007/s11227-018-2263-3.

[14] S. Rezaei and X. Liu, "Deep Learning for Encrypted Traffic Classification: An Overview," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 76–81, 2019. https://doi.org/10.1109/MCOM.2019.1800819.

[15] 3GPP, *3GPP TS 23.203 v11.6.0, Technical Specification Group Services and System Aspects; Policy and charging control architecture (Release 11)*. 3GPP, jun 2012.

[16] 3GPP, *3GPP TS 23.203 v15.4.0, Technical Specification Group Services and System Aspects; Policy and charging control architecture (Release 15)*. 3GPP, sep 2018.

[17] M. Patzold, "Countdown for the Full-Scale Development of 5G New Radio [Mobile Radio]," *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, pp. 7–13, 2018. https://doi.org/10.1109/MVT.2018.2811912.

[18] S. G. Petridou, P. G. Sarigiannidis, G. I. Papadimitriou, and A. S. Pomportsis, "On the Use of Clustering Algorithms for Message Scheduling in WDM Star Networks," *Journal of Lightwave Technology*, vol. 26, no. 17, pp. 2999–3010, 2008. https://doi.org/10.1109/JLT.2008.926913.

[19] S. G. Petridou, P. G. Sarigiannidis, G. I. Papadimitriou, and A. S. Pomportsis, "Clustering Based Scheduling: A New Approach to the Design of Scheduling Algorithms for WDM Star Networks," in *2007 14th IEEE Symposium on Communications and Vehicular Technology in the Benelux*, pp. 1–5, IEEE, 2007. https://doi.org/10.1109/SCVT.2007.4436255.

[20] S. G. Petridou, P. G. Sarigiannidis, G. I. Papadimitriou, and A. S. Pomportsis, "Clustering-based Scheduling: A New Class of Scheduling Algorithms for Single-Hop Lightwave Networks," *International Journal of Communication Systems*, vol. 21, no. 8, pp. 863–887, 2008. https://doi.org/10.1002/dac.929.

[21] R. Kausar, Y. Chen, and K. K. Chai, "Service Specific Queue Sorting and Scheduling Algorithm for OFDMA-based LTE-Advanced Networks," in *2011 International Conference on Broadband and Wireless Computing, Communication and Applications*, pp. 116–121, IEEE, 2011. https://doi.org/10.1109/BWCCA.2011.22.

[22] R. Kausar, Y. Chen, and K. K. Chai, "An Intelligent Scheduling Architecture for Mixed Traffic in LTE-Advanced," in *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications-(PIMRC)*, pp. 565–570, IEEE, 2012. https://doi.org/10.1109/PIMRC.2012.6362849.

[23] R. Kausar, *QoS Aware Packet Scheduling in the Downlink of LTE-Advanced Networks*. PhD thesis, Queen Mary, University of London, 2013.

[24] R. Kausar, Y. Chen, and K. Chai, "Clustering Based Packet Scheduling Adaptive to the Network Load in LTE-Advanced Networks," *Proceedings Appeared on IOARP Digital Library*, 2016. http://ioarp.org/ioarp-admin-panel/upload/articles/1460357645_IDL-ICCN15-010.pdf.

[25] W. S. Taie, A. H. Badawi, and A. F. Shalash, "Adaptive Closed Loop OFDM-Based Resource Allocation Method using Machine Learning and Genetic Algorithm," *arXiv preprint arXiv:1607.07494*, 2016. https://arxiv.org/pdf/1607.07494.

[26] W. Taie, *Adaptive OFDM-Based Resource Allocation Method using Machine Learning and Genetic Algorithm*. PhD thesis, Faculty of Engineering, Cairo University, 2015.

[27] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support Vector Machines," *IEEE Intelligent Systems and Their Applications*, vol. 13, no. 4, pp. 18–28, 1998. https://doi.org/10.1109/5254.708428.

[28] J.-B. Wang, J. Wang, Y. Wu, J.-Y. Wang, H. Zhu, M. Lin, and J. Wang, "A Machine Learning Framework for Resource Allocation assisted by Cloud Computing," *IEEE Network*, vol. 32, no. 2, pp. 144–151, 2018. https://doi.org/10.1109/MNET.2018.1700293.

[29] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to Optimize: Training Deep Neural Networks for Wireless Resource Management," in *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–6, IEEE, 2017. https://doi.org/10.1109/SPAWC.2017.8227766.

[30] X. Cao, R. Ma, L. Liu, H. Shi, Y. Cheng, and C. Sun, "A Machine Learning Based Algorithm for Joint Scheduling and Power Control in Wireless Networks," *IEEE Internet of Things Journal*, 2018. https://doi.org/10.1109/JIOT.2018.2853661.

[31] J. Höchst, L. Baumgärtner, M. Hollick, and B. Freisleben, "Unsupervised Traffic Flow Classification using a Neural Autoencoder," in *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*, pp. 523–526, IEEE, 2017. https://doi.org/10.1109/LCN.2017.57.

[32] H. R. Loo, S. B. Joseph, and M. N. Marsono, "Online Incremental Learning for High Bandwidth Network Traffic Classification," *Applied Computational Intelligence and Soft Computing*, vol. 2016, p. 1, 2016. https://doi.org/10.1155/2016/1465810.

[33] J. Zhang, X. Chen, Y. Xiang, W. Zhou, and J. Wu, "Robust Network Traffic Classification," *IEEE/ACM Transactions on Networking (TON)*, vol. 23, no. 4, pp. 1257–1270, 2015. https://doi.org/10.1109/TNET.2014.2320577.

[34] J. Zhang, Y. Xiang, W. Zhou, and Y. Wang, "Unsupervised Traffic Classification using Flow Statistical Properties and IP Packet Payload," *Journal of Computer and System Sciences*, vol. 79, no. 5, pp. 573–585, 2013. https://doi.org/10.1016/j.jcss.2012.11.004.

[35] A. Juvonen and T. Sipola, "Adaptive Framework for Network Traffic Classification using Dimensionality Reduction and Clustering," in *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems*, pp. 274–279, IEEE, 2012. https://doi.org/10.1109/ICUMT.2012.6459678.

[36] T. Sipola, A. Juvonen, and J. Lehtonen, "Anomaly Detection from Network Logs using Diffusion Maps," in *Engineering Applications of Neural Networks*, pp. 172–181, Springer, 2011. https://doi.org/10.1007/978-3-642-23957-1_20.

[37] R. Yuan, Z. Li, X. Guan, and L. Xu, "An SVM-based Machine Learning Method for Accurate Internet Traffic Classification," *Information Systems Frontiers*, vol. 12, no. 2, pp. 149–156, 2010. https://doi.org/10.1007/s10796-008-9131-2.

[38] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, "Offline/Realtime Traffic Classification using Semi-supervised Learning," *Performance Evaluation*, vol. 64, no. 9-12, pp. 1194–1213, 2007. https://doi.org/10.1016/j.peva.2007.06.014.

[39] J. Erman, M. Arlitt, and A. Mahanti, "Traffic Classification using Clustering Algorithms," in *Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data*, pp. 281–286, ACM, 2006. https://doi.org/10.1145/1162678.1162679.

[40] S. Zander, T. Nguyen, and G. Armitage, "Automated Traffic Classification and Application Identification using Machine Learning," in *The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)*, pp. 250–257, IEEE, 2005. https://doi.org/10.1109/LCN.2005.35.

[41] C. K. Liaskos, S. G. Petridou, G. I. Papadimitriou, P. Nicopolitidis, A. S. Pomportsis, and M. S. Obaidat, "Clustering-Driven Wireless Data Broadcasting," *IEEE Wireless Communications*, vol. 16, no. 6, pp. 80–87, 2009. https://doi.org/10.1109/MWC.2009.5361182.

[42] N. Sinclair, D. Harle, I. A. Glover, J. Irvine, and R. C. Atkinson, "An Advanced SOM Algorithm applied to Handover Management within LTE," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 1883–1894, 2013. https://doi.org/10.1109/TVT.2013.2251922.

[43] I. Assent, "Clustering High Dimensional Data," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 4, pp. 340–350, 2012. https://doi.org/10.1002/widm.1062.

[44] D. Mohan and G. M. Amalanathan, "A Survey on Long Term Evolution Scheduling in Data Mining," *Wireless Personal Communications*, vol. 102, no. 3, pp. 2363–2387, 2018. https://doi.org/10.1007/s11277-018-5909-9.

[45] A. C. Alice Zheng, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly Media, 2018. https://dl.acm.org/citation.cfm?id=3239815.

[46] M. Längkvist, L. Karlsson, and A. Loutfi, "A Review of Unsupervised Feature Learning and Deep Learning for Time-Series Modeling," *Pattern Recognition Letters*, vol. 42, pp. 11–24, 2014. https://doi.org/10.1016/j.patrec.2014.01.008.

[47] M. A. Kramer, "Nonlinear Principal Component Analysis using Autoassociative Neural Networks," *AIChE Journal*, vol. 37, no. 2, pp. 233–243, 1991. https://doi.org/10.1002/aic.690370209.

[48] H. Harb, A. Makhoul, D. Laiymani, A. Jaber, and R. Tawil, "K-Means Based Clustering Approach for Data Aggregation in Periodic Sensor Networks," *2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 434–441, 2014. https://doi.org/10.1109/WiMOB.2014.6962207.

[49] T. T. Nguyen and G. Armitage, "A Survey of Techniques for Internet Traffic Classification using Machine Learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008. https://doi.org/10.1109/SURV.2008.080406.

[50] K. Efimov, L. Adamyan, and V. Spokoiny, "Adaptive Nonparametric Clustering," *IEEE Transactions on Information Theory*, 2019. https://doi.org/10.1109/TIT.2019.2903113.

[51] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality Reduction: a Comparative Review," *Journal of Machine Learning Research*, vol. 10, pp. 66–71, 2009. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.125.6716rep=rep1type=pdf.

[52] T. Halonen, J. Romero, and J. Melero, *GSM, GPRS and EDGE Performance: Evolution towards 3G/UMTS*. John Wiley & Sons, 2004. https://doi.org/10.1002/0470866969.

[53] H. Holma and A. Toskala, *WCDMA for UMTS: HSDPA Evolution and LTE*. John Wiley, 2007. https://doi.org/10.1002/9780470512531.

[54] Qualcomm, *LTE Advanced: A Parallel Evolution Path to HSPA+*, jan 2010. http://www.androidauthority.com/hspa-vs-lte-which-one-is-better-78120.

[55] Qualcomm, *The Evolution of Mobile Technologies: 1G to 2G to 3G to 4G LTE*, jul 2014. https://www.qualcomm.com/documents/evolution-mobile-technologies-1g-2g-3g-4g-lte.

[56] 3GPP, *3GPP TS 21.915 v15.6.0, Technical Specification Group Services and System Aspects; Release 15 Description: Summary of Rel-15 Work Items (Release 15)*. 3GPP, feb 2019.

[57] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five Disruptive Technology Directions for 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, 2014. https://doi.org/10.1109/MCOM.2014.6736746.

[58] A. M. Niknejad and H. Hashemi, *mm-Wave Silicon Technology: 60 GHz and Beyond*. Springer Science & Business Media, 2008. https://doi.org/10.1007/978-0-387-76561-7.

[59] N. Alliance, "5G White Paper," *Next Generation Mobile Networks, White Paper*, 2015.

[60] 3GPP, *3GPP TS 22.278, Technical Specification Group Services and System Aspects; Service Requirements for Evolution of the 3GPP System (Release 8)*. 3GPP, 2009.

[61] 3GPP, *3GPP TS 21.902, Technical Specification Group Services and System Aspects; Evolution of 3GPP System (Release 10)*. 3GPP, 2011.

[62] C. Cox, *An Introduction to LTE: LTE, LTE-Advanced, SAE and 4G Mobile Communications*. John Wiley & Sons, 2012. https://doi.org/10.1002/9781119942825.

[63] *3GPP Strategies for IMT-Advanced*, 2010. http://goo.gl/MHPu4S.

[64] 3GPP, *3GPP TS 23.002 v11.6.0, Technical Specification Group Services and System Aspects; Network architecture (Release 11)*. 3GPP, jun 2013.

[65] E. Dahlman, S. Parkvall, and J. Skold, *4G, LTE-Advanced Pro and the Road to 5G*. Academic Press, 2016.

[66] A. Mohamed, O. Onireti, M. A. Imran, A. Imran, and R. Tafazolli, "Control-Data Separation Architecture for Cellular Radio Access Networks: A Survey and Outlook," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 446–465, 2016. https://doi.org/10.1109/COMST.2015.2451514.

[67] 3GPP, *3GPP TS 36.213, Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 11)*. 3GPP, 2012.

[68] 3GPP, *3GPP TS 23.101, Technical Specification Group Services and System Aspects; General Universal Mobile Telecommunications System (UMTS) Architecture (Release 9)*. 3GPP, 2009.

[69] M. Rumney *et al.*, *LTE and the Evolution to 4G Wireless: Design and Measurement Challenges*. John Wiley & Sons, 2013. https://doi.org/10.1002/9781118799475.

[70] 3GPP, *3GPP TS 24.301, Technical Specification Group Core Network and Terminals; Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3 (Release 9)*. 3GPP, 2011.

[71] M. Olsson and C. Mulligan, *EPC and 4G Packet Networks: Driving the Mobile Broadband Revolution*. Academic Press, 2012.

[72] S. Shenker, "RFC 2212 – Specification of Guaranteed Quality of Service," *Internet Engineering Task Force - IETF*, 1997. https://doi.org/10.17487/rfc2212.

[73] GSMA, "Network 2020: Mission Critical Communications - GSMA – White Paper," no. February, 2019.

[74] S. Ahmadi, *LTE-Advanced: A Practical Systems Approach to Understanding 3GPP LTE Releases 10 and 11 Radio Access Technologies*. Academic Press, 2013.

[75] 3GPP, *3GPP TS 36.300, Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (Release 10)*. 3GPP, 2010.

[76] A. Lucent, "The LTE Network Architecture – A Comprehensive Tutorial," *Strategic White Paper*, 2009.

[77] P. Simon, *Too Big to Ignore: The Business Case for Big Data*, vol. 72. John Wiley & Sons, 2013. https://doi.org/10.1002/9781119204039.

[78] T. M. Mitchell, *Machine Learning*, vol. 1. McGraw-Hill Education, 1997.

[79] J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, "Machine Learning: A Historical and Methodological Analysis," *AI Magazine*, vol. 4, no. 3, p. 69, 1983.

[80] S. Marsland, *Machine Learning: An Algorithmic Perspective*. CRC press, 2015. https://doi.org/10.1201/b17476.

[81] Y. Bengio, I. J. Goodfellow, and A. Courville, "Deep Learning," *Nature*, vol. 521, pp. 436–444, 2015. https://doi.org/10.1038/nature14539.

[82] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-Level Control through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, p. 529, 2015. https://doi.org/10.1038/nature14236.

[83] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the Game of Go with Deep Neural Networks and Tree Search," *Nature*, vol. 529, no. 7587, p. 484, 2016. https://doi.org/10.1038/nature16961.

[84] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.*, "Mastering the Game of Go without Human Knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2017. https://doi.org/10.1038/nature24270.

[85] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Communications Surveys & Tutorials*, 2019. https://doi.org/10.1109/COMST.2019.2904897.

[86] A. Y. Ng, "Lecture 1, Coursera: Machine Learning," *Stanford University*, 2015.

[87] M. Usama, J. Qadir, A. Raza, H. Arif, K.-L. A. Yau, Y. Elkhatib, A. Hussain, and A. Al-Fuqaha, "Unsupervised Machine Learning for Networking: Techniques, Applications and Research Challenges," *IEEE Access*, vol. 7, pp. 65579–65615, 2019. https://doi.org/10.1109/ACCESS.2019.2916648.

[88] M. Kirk, *Thoughtful Machine Learning: A Test-Driven Approach*. O'Reilly Media, Inc., 2014.

[89] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, vol. 1. MIT Press Cambridge, 1998.

[90] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement Learning: A Survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996. https://doi.org/10.1613/jair.301.

[91] A. Este, F. Gringoli, and L. Salgarelli, "Support Vector Machines for TCP Traffic Classification," *Computer Networks*, vol. 53, no. 14, pp. 2476–2490, 2009. https://doi.org/10.1016/j.comnet.2009.05.003.

[92] SciLab, *Machine Learning - Classification with SVM*, mar 2019. https://bit.ly/2IIgE3q.

[93] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, *et al.*, "Top 10 Algorithms in Data Mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008. https://doi.org/10.1007/s10115-007-0114-2.

[94] T. Cover and P. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967. https://doi.org/10.1109/TIT.1967.1053964.

[95] L. E. Peterson, "K-Nearest Neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009. https://doi.org/10.4249/scholarpedia.1883.

[96] S. P. Lloyd, "Least Squares Quantization in PCM," *Information Theory, IEEE Transactions on*, vol. 28, no. 2, pp. 129–137, 1982. https://doi.org/10.1109/TIT.1982.1056489.

[97] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The Fuzzy C-Means Clustering Algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984. https://doi.org/10.1016/0098-3004(84)90020-7.

[98] D. Pelleg, A. W. Moore, *et al.*, "X-means: Extending K-means with Efficient Estimation of the Number of Clusters," in *ICML*, pp. 727–734, 2000.

[99] R. H. Jones, "Bayesian Information Criterion for Longitudinal and Clustered Data," *Statistics in Medicine*, vol. 30, no. 25, pp. 3050–3056, 2011. https://doi.org/10.1002/sim.4323.

[100] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate Analysis*. Academic Press, 1979.

[101] O. Sigaud and O. Buffet, *Markov Decision Processes in Artificial Intelligence*. John Wiley & Sons, 2013. https://doi.org/10.1002/9781118557426.

[102] O. Alagoz, H. Hsu, A. J. Schaefer, and M. S. Roberts, "Markov Decision Processes: A Tool for Sequential Decision Making Under Uncertainty," *Medical Decision Making*, vol. 30, no. 4, pp. 474–483, 2010. https://doi.org/10.1177/0272989X09353194.

[103] C. J. Watkins and P. Dayan, "Q-Learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992. https://doi.org/10.1023/A:1022676722315.

[104] C. J. C. H. Watkins, *Learning from Delayed Rewards*. PhD thesis, University of Cambridge England, 1989.

[105] T. Jaakkola, M. I. Jordan, and S. P. Singh, "On the Convergence of Stochastic Iterative Dynamic Programming Algorithms," *Neural Computation*, vol. 6, no. 6, pp. 1185–1201, 1994. https://doi.org/10.1162/neco.1994.6.6.1185.

[106] B. Everitt, S. Landau, M. Leese, and D. Stahl, *Cluster Analysis*. Wiley, 2011. https://doi.org/10.1002/9780470977811.

[107] Netmanias, *LTE QoS: SDF and EPS Bearer QoS*, mar 2019. https://bit.ly/2FMwl8o.

[108] Y. He, Z. Zhang, F. R. Yu, N. Zhao, H. Yin, V. C. Leung, and Y. Zhang, "Deep Reinforcement Learning-based Optimization for Cache-enabled Opportunistic Interference Alignment Wireless Networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10433–10445, 2017. https://doi.org/10.1109/TVT.2017.2751641.

[109] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[110] MQL5, *Redes Neurais Profundas (Parte III). Seleção da Amostra e Redução de Dimensionalidade*, mar 2019. https://www.mql5.com/pt/articles/3526.

[111] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Elsevier, 2013. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.105.4702rep=rep1type=pdf.

[112] H. Abdi and L. J. Williams, "Principal Component Analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010. https://doi.org/10.1002/wics.101.

[113] I. Jolliffe, *Principal Component Analysis*. Springer, 2011. https://doi.org/10.1007/978-3-642-04898-2_455.

[114] R. Bro and A. K. Smilde, "Principal Component Analysis," *Analytical Methods*, vol. 6, no. 9, pp. 2812–2831, 2014. https://doi.org/10.1039/C3AY41907J.

[115] M. J. Zaki, W. Meira Jr, and W. Meira, *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press, 2014. https://doi.org/10.1017/CBO9780511810114.

[116] M. A. Kramer, "Autoassociative Neural Networks," *Computers & Chemical Engineering*, vol. 16, no. 4, pp. 313–328, 1992. https://doi.org/10.1016/0098-1354(92)80051-A.

[117] Y. Wang, H. Yao, and S. Zhao, "Auto-encoder based Dimensionality Reduction," *Neurocomputing*, vol. 184, pp. 232–242, 2016. https://doi.org/10.1016/j.neucom.2015.08.104.

[118] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014. http://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf.

[119] J. Grus, *Data Science from Scratch: First Principles with Python*. O'Reilly Media, Inc., 2015.

[120] C. Hennig, M. Meila, F. Murtagh, and R. Rocci, *Handbook of Cluster Analysis*. CRC Press, 2015. https://doi.org/10.1201/b19706.

[121] J. Handl and J. Knowles, "Multi-objective Clustering and Cluster Validation," in *Multi-Objective Machine Learning*, pp. 21–47, Springer, 2006. https://doi.org/10.1007/11399346_2.

[122] E. E. for Developing KDD-Applications Supported by Index-Structures, *Ilustração de Clustering sobre um Conjunto de Dados*, mar 2019. https://elki-project.github.io.

[123] E. C. Santos, "A Simple Reinforcement Learning Mechanism for Resource Allocation in LTE-A Networks with Markov Decision Process and Q-Learning," *arXiv preprint arXiv:1709.09312*, 2017. https://arxiv.org/abs/1709.09312.

[124] J. W. Osborne and A. B. Costello, "Sample Size and Subject to Item Ratio in Principal Components Analysis," *Practical Assessment, Research & Evaluation*, vol. 9, no. 11, p. 8, 2004. https://pareonline.net/htm/v9n11.htm.

[125] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "Nearest Neighbor" Meaningful?," in *International Conference on Database Theory*, pp. 217–235, Springer, 1999. https://doi.org/10.1007/3-540-49257-7_15.

[126] A. Zimek, E. Schubert, and H.-P. Kriegel, "A Survey on Unsupervised Outlier Detection in High-dimensional Numerical Data," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 5, no. 5, pp. 363–387, 2012. https://doi.org/10.1002/sam.11161.

[127] D. Z. Yazti and S. Krishnaswamy, "Mobile Big Data Analytics: Research, Practice, and Opportunities," in *2014 IEEE 15th International Conference on Mobile Data Management*, vol. 1, pp. 1–2, IEEE, 2014. https://doi.org/10.1109/MDM.2014.73.

[128] J. Wang, J. Zhuang, L. Duan, and W. Cheng, "A Multi-Scale Convolution Neural Network for Featureless Fault Diagnosis," in *2016 International Symposium on Flexible Automation (ISFA)*, pp. 65–70, IEEE, 2016. https://doi.org/10.1109/ISFA.2016.7790137.

[129] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving Deep Neural Networks for LVCSR using Rectified Linear Units and Dropout," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8609–8613, IEEE, 2013. https://doi.org/10.1109/ICASSP.2013.6639346.

[130] R. Schaffer and R. Sedgewick, "The Analysis of Heapsort," *Journal of Algorithms*, vol. 15, no. 1, pp. 76–100, 1993. https://doi.org/10.1006/jagm.1993.1031.

[131] M. Partridge and R. A. Calvo, "Fast Dimensionality Reduction and Simple PCA," *Intelligent Data Analysis*, vol. 2, no. 1-4, pp. 203–214, 1998. https://doi.org/10.1016/S1088-467X(98)00024-9.

[132] M. K. Pakhira, "A Linear Time-complexity K-means Algorithm using Cluster Shifting," in *2014 International Conference on Computational Intelligence and Communication Networks*, pp. 1047–1051, IEEE, 2014. https://doi.org/10.1109/CICN.2014.220.

[133] D.-Z. Du and K.-I. Ko, *Theory of Computational Complexity*. John Wiley & Sons, 2014.

[134] I. S. Msiza and T. Marwala, "Autoencoder Networks for Water Demand Predictive Modelling," in *2016 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH)*, pp. 1–8, IEEE, 2016. https://doi.org/10.5220/0005977202310238.

[135] G. Kerschen and J.-C. Golinval, "Feature Extraction using Auto-associative Neural Networks," *Smart Materials and Structures*, vol. 13, no. 1, p. 211, 2003. https://doi.org/10.1088/0964-1726/13/1/025.

[136] A. Rios and M. Kabuka, "Image Compression with a Dynamic Autoassociative Neural Network," *Mathematical and Computer Modelling*, vol. 21, no. 1-2, pp. 159–171, 1995. https://doi.org/10.1016/0895-7177(94)00202-Y.

[137] C. A. Hoare, "Quicksort," *The Computer Journal*, vol. 5, no. 1, pp. 10–16, 1962. https://doi.org/10.1093/comjnl/5.1.10.

[138] O. Chapelle, "Training a Support Vector Machine in the Primal," *Neural Computation*, vol. 19, no. 5, pp. 1155–1178, 2007. https://doi.org/10.1162/neco.2007.19.5.1155.

[139] G. Piro, L. A. Grieco, G. Boggia, F. Capozzi, and P. Camarda, "Simulating LTE Cellular Systems: An Open-Source Framework," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 2, pp. 498–513, 2011. https://doi.org/10.1109/TVT.2010.2091660.

[140] F. Chollet *et al.*, *Keras: Deep Learning Library for Theano and Tensorflow*, 2015. https://keras.io.

[141] E. C. Santos, "Autonomous QoS-Based Mechanism for Resource Allocation in LTE-Advanced Pro Networks," in *2018 IEEE Colombian Conference on Communications and Computing (COLCOM)*, pp. 1–6, IEEE, 2018. https://doi.org/10.1109/ColComCon.2018.8466714.

[142] K. W. Pettis, T. A. Bailey, A. K. Jain, and R. C. Dubes, "An Intrinsic Dimensionality Estimator from Near-neighbor Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 1, pp. 25–37, 1979. https://doi.org/10.1109/TPAMI.1979.4766873.

[143] M. Hein and J.-Y. Audibert, "Intrinsic Dimensionality Estimation of Submanifolds in R d," in *Proceedings of the 22nd International Conference on Machine Learning*, pp. 289–296, ACM, 2005. https://doi.org/10.1145/1102351.1102388.

[144] X. L. Xie and G. Beni, "A Validity Measure for Fuzzy Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841–847, 1991. https://doi.org/10.1109/34.85677.

[145] M. Hassani and T. Seidl, "Using Internal Evaluation Measures to Validate the Quality of Diverse Stream Clustering Algorithms," *Vietnam Journal of Computer Science*, vol. 4, no. 3, pp. 171–183, 2017. https://doi.org/10.1007/s40595-016-0086-9.

[146] H. T. Cheng and W. Zhuang, "An Optimization Framework for Balancing Throughput and Fairness in Wireless Networks with QoS Support," *IEEE Transactions on Wireless Communications*, vol. 7, no. 2, pp. 584–593, 2008. https://doi.org/10.1109/TWC.2008.060507.