

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Adaptação Preditiva de Conteúdo Multimédia

Sérgio Manuel Colaço de Sá

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major Telecomunicações

Orientador: Prof. Dra. Maria Teresa Magalhães da Silva Pinto de Andrade

Co-orientador: Prof. Dr. Jaime dos Santos Cardoso

Junho de 2009

A Dissertação intitulada


“PREDICTIVE MULTIMEDIA CONTENT ADAPTATION”

foi aprovada em provas realizadas em 17/Julho/2009

o júri



Presidente **Professor Doutor Luís António Pereira de Meneses Corte-Real**
Professor Associado da Faculdade de Engenharia da Universidade do Porto



Professor Doutor Rui Jorge da Silva Moreira
Professor Auxiliar da Faculdade de Ciência e Tecnologia da Universidade Fernando Pessoa



Professora Doutora Maria Teresa Magalhães da Silva Pinto de Andrade
Professora Auxiliar do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto



Professor Doutor Jaime dos Santos Cardoso
Professor Auxiliar Convidado do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projecto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são correctamente citados.

Autor - **SÉRGIO MANUEL COLAÇO DE SÁ**

Faculdade de Engenharia da Universidade do Porto

Resumo

Esta dissertação pretende estudar abordagens e propor e testar soluções baseadas em técnicas de aprendizagem automática, para monitorizar as condições de transmissão em rede de conteúdos multimédia. O objectivo é o de conseguir reagir antecipadamente a situações de diminuição da disponibilidade de recursos, que acarrete uma diminuição da qualidade do conteúdo com consequente degradação na qualidade da experiência do utilizador. Para tal pretende-se desenvolver um sistema automático que seja capaz de prever essas situações de variação da disponibilidade dos recursos de rede e que recomende a implementação de uma adaptação ao conteúdo, adequada às condições de transmissão previstas, minimizando dessa forma o impacto negativo na qualidade da experiência do utilizador.

O acesso a redes IP e o transporte de conteúdos multimédia sobre as mesmas tem-se vindo a intensificar cada vez mais, sendo uma boa parte da largura de banda disponível, e partilhada por todos os utilizadores, ocupada com estes tipos de conteúdos.

Tendo em mente o tipo de conteúdos, os quais têm determinados requisitos de transmissão e apresentação em tempo real e transmitidos em redes que não oferecem qualidade de serviço, a continuidade e a perceptividade que se obtém da sua visualização é afectada por problemas de rede, mesmo que por poucos milissegundos.

Uma forma de ultrapassar os problemas causados por congestão na rede e com consequente perda de qualidade do conteúdo multimédia passa pela adaptação do mesmo, por exemplo, alterando os parâmetros de codificação do vídeo. Desta forma, recorrendo à previsão das condições da rede pode-se prever com antecedência, recorrendo a algoritmos de *machine learning*, situações de diminuição de recursos disponíveis e actuar sobre os conteúdos, adaptando-os de forma a cumprirem as restrições previstas.

Alcançando o sucesso na previsão de restrições da rede e consequente adaptação de conteúdo poder-se-à prevenir situações de interrupção, por exemplo através da diminuição do componente vídeo que dessa forma cumpriria as restrições previstas.

Abstract

This thesis has as its main goal the study, analysis and implementation of a system based in techniques of automatic learning, to predict situations of failure in the transmission of continuous multimedia contents such as videos, in heterogeneous networks. The system is expected to be able to act on those contents (adapt them) in advance and transparently.

The access to IP networks and the transport of multimedia contents over these is becoming more and more intense over these last few years, being a good part of the available bandwidth, and shared by all the users, loaded with these contents.

Having in mind this type of contents, which have higher transmission requirements and real time presentation, the continuity and the perceptiveness that one gets of its visualization is affected by network problems, even if it's only for a few milliseconds.

A way to overcome the problems caused by congestion in the network and its consequent loss of quality of the multimedia content is to adapt these contents, like modifying the parameters of video coding.

This way the transmission of these contents can be made using all the available capacity of the network at the moment and the system would foresee interruption situations during the consumption of the multimedia content making use of prediction mechanisms.

Thus, even if the quality that the user is consuming the multimedia resource is being dynamically changed, the reduction of quality is check, having less negative impact in the user, in contrast with the interruption cause by network congestion in the visualization that would tremendously affect its experience.

Agradecimentos

Agradeço a todos aqueles que tornaram possível a minha chegada a este ponto do meu percurso escolar e mais concretamente à elaboração desta tese. Em especial agradeço:

Aos meus pais Fernanda Colaço e José Sá por todo o apoio e pela força que me deram ao longo destes anos.

Aos meus amigos de sempre, amigos da faculdade e fora dela por todo o apoio e ânimo em todos os momentos, em especial nesta última etapa.

Gostaria de agradecer em especial à minha orientadora, Professora Doutora Maria Teresa Andrade por todo o apoio e disponibilidade durante a realização da dissertação e por quem desenvolvi uma grande admiração.

Ao Professor Doutor Jaime Cardoso por me ter co-orientado na realização da tese e apoiado, mostrando-se sempre disponível.

Para terminar, gostaria de agradecer ao Professor Doutor Artur Pimenta Alves, que considero ter-me inspirado na escolha do ramo de comunicações multimédia, pela sua boa disposição e pela sua forma de estar e apresentar as aulas tornando-as sem dúvida muito interessantes.

Sérgio Sá

Conteúdo

1	Introdução	1
1.1	Enquadramento da Dissertação	1
1.2	Objectivos	2
1.3	Composição do Sistema	2
1.3.1	Módulos do Terminal de Consumo	3
1.4	Estrutura do Documento	5
2	Estado da Arte das Tecnologias de Previsão e Adaptação	7
2.1	Vídeo Digital	7
2.1.1	Formatos de Vídeo	7
2.1.2	Ferramentas de <i>Streaming</i>	8
2.2	Redes e Tráfego IP	10
2.2.1	Características e Parâmetros	10
2.2.2	Ferramentas Utilizadas	11
2.3	<i>Data Mining</i>	12
2.3.1	Ferramenta de <i>Machine Learning</i>	14
2.3.2	Algoritmos de <i>Machine Learning</i>	14
2.4	Adaptação de Vídeo	21
2.5	Revisão Bibliográfica de Trabalhos Publicados	21
2.5.1	Decisões de Adaptação	21
2.5.2	Classificação das Condições de Rede	22
2.5.3	Previsão de Condições de Rede	23
2.5.4	Arquitectura de Gestão de Redes	23
2.6	Análise	23
3	Planeamento do Sistema de Adaptação Preditiva	25
3.1	Introdução	25
3.2	Requisitos do Sistema	25
3.3	Recolha de Dados em Cenários de Teste	27
4	Teste de Algoritmos de <i>Machine Learning</i>	31
4.1	Tratamento de Dados	31
4.1.1	Formatação dos Ficheiros de Teste	33
4.2	Avaliação de Algoritmos de <i>Machine Learning</i>	34
4.2.1	Máquinas de Vectores de Suporte (SVM)	34
4.2.2	Redes Neurais Artificiais (RNA)	35
4.2.3	Regressão Linear	36

4.2.4	Processos Gaussianos	36
4.2.5	M5P	37
4.2.6	Locally Weighted Learning (LWL)	38
4.3	Análise	38
5	Previsão em Tempo-Real	41
5.1	Aquisição de Parâmetros em Tempo Real	41
5.2	Mecanismo de Previsão	42
5.3	Mecanismo de Adaptação	43
5.4	<i>Stream</i> de Vídeo	43
5.5	Sumário de Funcionalidades e Resultados	44
6	Conclusões e Trabalho Futuro	47
6.1	Satisfação dos Objectivos	47
6.2	Trabalho Futuro	48

Lista de Figuras

1.1	Diagrama de módulos do Sistema	2
1.2	Diagrama de módulos do Terminal de Consumo	3
2.1	Exemplo de funcionamento do DIA ?	8
2.2	Logótipo do FFmpeg	9
2.3	Logótipo do VideoLAN	9
2.4	Exemplo de aprendizagem supervisionada	13
2.5	Logótipo do WEKA	14
2.6	Princípio de funcionamento de SVM ?	15
2.7	Exemplo de funcionamento da <i>kernel</i>	16
2.8	Funcionamento de um neurónio artificial	16
2.9	Exemplo de uma rede <i>feedforward</i>	17
2.10	Gráfico (a) demonstra quatro exemplos de dados. No gráfico (b) observamos a situação após dois pontos terem sido observados. A previsão média encontra-se na linha contínua e a tracejada são quatro amostras dos dados anteriores. Em ambos os gráficos a região sombreada representa o dobro do desvio médio a cada valor de entrada, x	19
3.1	Diagrama de lógica de operação	26
3.2	Situação de congestão durante a transmissão de um vídeo.	28
4.1	Resultados do teste ao algoritmo SVM	35
4.2	Resultados do teste ao algoritmo RNA	35
4.3	Resultados do teste ao algoritmo Regressão Linear	36
4.4	Resultados do teste ao algoritmo Processos Gaussianos	37
4.5	Resultados do teste ao algoritmo Processos Gaussianos	37
4.6	Resultados do teste ao algoritmo Processos Gaussianos	38
4.7	Comparação dos resultados médios dos algoritmos.	38
5.1	Visualização dos erros de descodificação.	43
5.2	Imagem retirada de um vídeo recebido por <i>streaming</i>	44
5.3	Comparação de erros reais de descodificação vs. erros previstos pelo algoritmo Regressão Linear.	45
5.4	Comparação de erros reais de descodificação vs. erros previstos pelo algoritmo SVM.	45
6.1	Comparação de erros reais de descodificação vs. erros previstos pelo algoritmo SVM.	48

Abreviaturas e Símbolos

CSV	Comma-separated values
DIA	Digital Item Adaptation
FEC	Forward Error Correction
GOP	Group Of Pictures
GPS	Global Positioning System
IEC	International Electrotechnical Commission
IP	Internet Protocol
IPTV	Internet Protocol Television
ISO	International Organization for Standardization
LWL	Locally Weighted Learning
MGEN	Multi-Generator
PSNR	Peak Signal-to-Noise Ratio
QoE	Quality of Experience
QoS	Quality of Service
RNA	Redes neuronais artificiais
RTT	Round-Trip Time
SVM	Support Vector Machine
TRPR	TRace Plot Real-time
UDP	User Datagram Protocol

Capítulo 1

Introdução

1.1 Enquadramento da Dissertação

Os avanços tecnológicos que se tem vindo a observar nos últimos anos têm levado à proliferação de dispositivos portáteis com capacidades de acesso à Internet e de reprodução de conteúdos multimédia. No entanto, de uma forma geral, esses dispositivos multimédia apresentam diferentes características e capacidades, nomeadamente dimensão dos ecrãs e capacidade de processamento. Paralelamente, observou-se um aumento exponencial de conteúdos multimédia disponíveis em rede ?. Estes factores levaram a uma crescente procura e conseqüente consumo de conteúdos multimédia nos ambientes mais variados.

Sendo que estes dispositivos fazem o seu acesso à Internet recorrendo a redes sem fios, que se caracterizam de certo modo pelo grande grau de variabilidade da largura de banda, dependendo do número instantâneo de utilizadores e das condições ambientais, torna-se difícil seleccionar parâmetros adequados/óptimos de codificação levando-nos à inevitável consideração de que tais parâmetros têm de ser adaptados à medida que variam as condições do contexto de consumo. Torna-se portanto imperativo a criação de ferramentas que garantam a qualidade de serviço (QoS) na visualização de conteúdos multimédia.

Esta tese tem como objectivo, a oferta de garantia de qualidade na reprodução de conteúdo vídeo. Este tipo de conteúdos são caracterizados pela reprodução contínua no tempo e devido a esta característica e aos processos que ocorrem nos decodificadores para proporcionar a visualização dos mesmos, erros que possam ocorrer têm um impacto muito grande na percepção do utilizador e traduzem-se numa perda de qualidade que pode espalhar-se no tempo. Ainda mais, se o conteúdo estiver a ser transmitido em tempo real, tais erros podem traduzir-se na perda de informação.

Nos últimos anos têm-se vindo a desenvolver trabalhos nesta área, recorrendo a algoritmos de Machine Learning para previsão de alteração de condições da rede, podendo com alguma antecedência tomar decisões sobre possíveis adaptações dos conteúdos multimédia, tentando assim garantir uma melhor qualidade de serviço ??.

1.2 Objectivos

Como foi descrito na secção anterior, o acesso a conteúdos multimédia em redes que não oferecem garantias de qualidade, depara-se actualmente com uma série de desafios que dificultam a tarefa de assegurar uma qualidade de serviço constante e aceitável para o consumidor do conteúdo multimédia.

O grande grau de variabilidade no contexto de utilização, desde as capacidades dos terminais até às condições das redes, passando pelos próprios utilizadores, fazem com que seja necessário efectuar uma monitorização constante desse contexto e tomar medidas reactivas para tentar minimizar a degradação da qualidade de experiência do consumidor. Essas medidas podem passar pela adaptação do conteúdo de modo a satisfazer as novas condições ou restrições do contexto de utilização. No entanto, a latência com que essas medidas reactivas são implementadas pode ser excessiva e acarretar uma degradação da qualidade mal tolerada pelo consumidor.

Assim, o objectivo desta tese é o de estudar alternativas que permitam tomar medidas a-priori para combater os efeitos de variação das condições de contexto de utilização. Pretende-se fazer uso de técnicas de Machine Learning, as quais fazendo uso de um grande conjunto de valores de informação de contexto típicos de situações usuais de distribuição de conteúdo multimédia em redes *best-effort*, sejam capazes de treinar algoritmos de decisão de adaptação, por forma a que a decisão seja tomada antes da consequência de situações restritivas tais como congestão ou perdas na rede, diminuição de capacidade de processamento, entre outros.

1.3 Composição do Sistema

O sistema proposto para a realização do estudo de adaptação preemptiva baseada em técnicas de *Machine Learning* é composto pelos seguintes módulos funcionais, ilustrados na Figura 1.1 e descritos nos pontos seguintes.

- **Servidor de Streaming de vídeo** - Representa uma máquina na rede com capacidade de emissão de um *stream* vídeo para a rede e a possibilidade de transcodificação do mesmo. Este bloco tem ainda de possuir a capacidade de recepção e interpretação de mensagens provenientes do terminal de consumo, afim de alterar os parâmetros de codificação do vídeo.

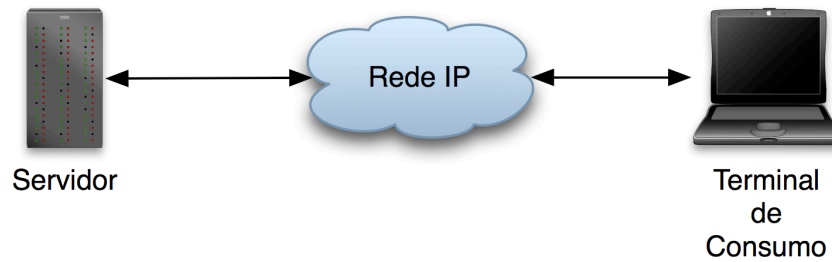


Figura 1.1: Diagrama de módulos do Sistema

- **Rede IP** - Representa a rede em que as máquinas acima referidas se encontram, podendo o acesso ao meio ser feito via rede com fios ou sem fios. Nesta rede será transportado todo o tráfego gerado pelo *stream* de vídeo e será também injectado mais tráfego afim de criar condições de saturação ou diminuição da disponibilidade de recursos.
- **Terminal de Consumo** - Representa o receptor de um *stream* de vídeo, tendo no mesmo contidas as capacidades necessárias à tomada de decisão sobre adaptações a efectuar no vídeo a ser recebido. Encontra-se o mesmo descrito nas secções seguintes.

1.3.1 Módulos do Terminal de Consumo

Para a realização das funções necessárias são implementados no terminal de consumo os seguintes módulos.

Os módulos apresentados na figura 1.2 operam da seguinte forma:

- **Leitura de parâmetros de rede**, função capaz de, de uma forma contínua, fazer medições de qualidade da rede.
- **Máquina de *Machine Learning***, através dos parâmetros fornecidos pelo módulo anterior, terá de obter uma previsão de um futuro próximo desses mesmos parâmetros.
- **Motor de tomada de decisão**, bloco que será chamado apenas no caso de uma previsão que indique a possível existência de uma quebra de qualidade no vídeo a ser recebido e estará responsável pela decisão da adaptação a ser efectuada tentando dessa forma minimizar o impacto da mesma para o utilizador.
- **Descodificador de Vídeo**, que opera de uma forma contínua descodificando o vídeo a ser recebido através da rede, sendo que o acesso ao mesmo só necessita de ser feito aquando de uma tomada de decisão de adaptação.

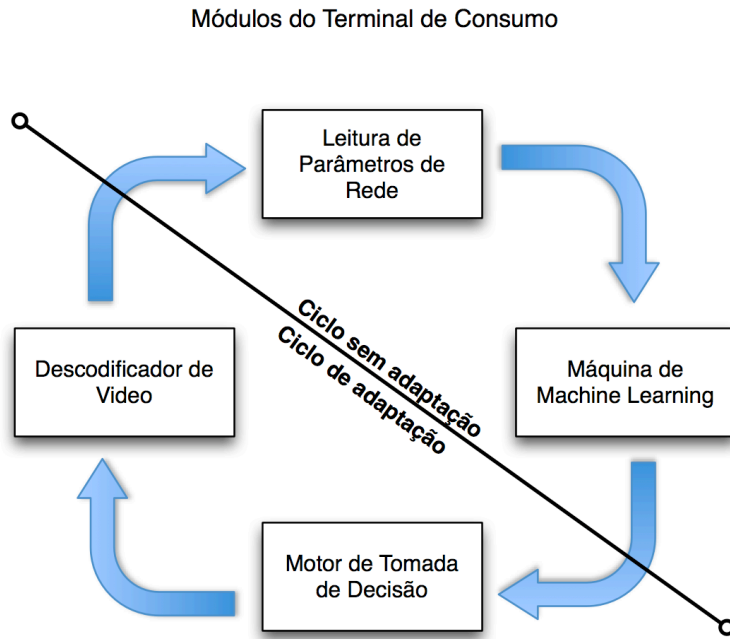


Figura 1.2: Diagrama de módulos do Terminal de Consumo

A interacção entre os blocos acima descritos poderá ocorrer de duas formas distintas, originando dois possíveis ciclos:

- **Ciclo sem adaptação**, nos quais estão a operar os blocos de leitura de parâmetros de rede e a máquina de *machine learning*. Chama-se a este ciclo "sem adaptação" uma vez que as previsões dadas pela máquina de *machine learning* não indicam a possibilidade de degradação da qualidade de vídeo, não sendo assim necessário tomar decisões de adaptação, mantendo-se o sistema a funcionar entre estes dois blocos.
- **Ciclo de adaptação**, ocorre no caso de existir uma previsão dada pela máquina de *machine learning* em que se detecta a possibilidade de uma degradação de qualidade vídeo. Assim sendo, o motor de tomada de decisão terá de escolher quais os parâmetros de vídeo a serem alterados e fornecer os mesmos, neste sistema, ao descodificador de vídeo afim da adaptação surtir efeito.

1.3.1.1 Leitura de Parâmetros de Rede

Consiste este bloco na aquisição de dados sobre a ligação de rede e que têm uma influência directa sobre a recepção e visualização de um *stream* de vídeo.

- **Bitrate** - Valor representativo da velocidade de recepção de dados, representados em quantidade de dados por segundo, que pode ser comparado com a velocidade

total de dados a chegarem ao terminal, afim de detecção de congestão imposta pela limitação da rede.

- **Latência** - Valor representativo do atraso dos dados medido instantaneamente pela diferença entre o tempo em que os dados foram emitidos pelo servidor até à recepção dos mesmos no terminal.
- **Jitter** - Este valor é calculado pela diferença de tempo de chegada entre dois pacotes de dados consecutivos recebidos, desta forma representa a variação do tempo de chegada dos mesmos.
- **Taxa de Erros** - Esta pode ser medida de duas formas, verificando os pacotes totais perdidos na rede, ou seja, que não chegaram ao terminal de consumo, ou de uma forma mais directa e que detecta outros tipos de erro, tais como erros de sincronismo, directamente no descodificador de vídeo.

Este bloco terá a sua utilização em tempo real, para fornecer os seus parâmetros à máquina de *machine learning* mas será utilizado numa primeira instância para a recolha de dados afim de se analisarem os dados para uma utilização *off-line*.

1.3.1.2 Máquina de *Machine Learning*

A função responsável pela operação deste módulo terá de ser capaz de desempenhar todas as funcionalidades necessárias à previsão.

Numa fase inicial do projecto estará encarregue de treinar e avaliar vários algoritmos recorrendo a dados de histórico capturados pelo módulo da secção anterior. Desta forma serão apenas seleccionados os algoritmos que apresentem os melhores resultados tendo em conta as médias de erro obtidas nos diversos testes.

Numa segunda fase, e já com a aplicação a correr em tempo real, este módulo tratará da detecção de situações em que terão de ser tomadas medidas de adaptação do conteúdo vídeo.

1.3.1.3 Motor de Tomada de Decisão

Após detectadas as situações em que uma decisão de adaptação terá de ser tomada, nomeadamente por alteração das condições da rede, cabe a este bloco a decisão sobre as adaptações a efectuar e ainda o envio com a informação da mesma ao servidor afim das mesmas reproduzirem efeito.

1.4 Estrutura do Documento

Este documento é composto por seis capítulos.

O primeiro capítulo em que este texto se insere encontra-se composto pelo enquadramento da dissertação, os seus objectivos e a composição do sistema.

O capítulo numero dois representa todo o estado da arte, ou seja a revisão bibliográfica sobre os temas abordados no desenvolvimento desta tese.

Os três capítulos seguintes intitulados de "Sistema de adaptação preditiva", "Teste de Algoritmos de Machine Learning" e "Previsão em Tempo-Real" têm como conteúdo todo o trabalho e resultados obtidos ao longo das várias fases de desenvolvimento do trabalho. A divisão e apresentação do trabalho neste quatro capítulos deve-se ao facto de cada um representar uma fase distinta do trabalho.

No capítulo sete estão apresentadas as conclusões desta dissertação, os objectivos cumpridos os pontos que ficam em aberto para trabalho futuro.

Para terminar, estão apresentadas as referências bibliográficas utilizadas no estudo desta tese.

Capítulo 2

Estado da Arte das Tecnologias de Previsão e Adaptação

Para dar início ao trabalho proposto foi necessário, em primeiro lugar, fazer um levantamento do estado da arte de codificação vídeo e suas normas, de alguns algoritmos de Machine Learning e ainda parâmetros de tráfego e qualidade em redes IP. Esse capítulo encontra-se dividido nas secções atendendo então às três grandes áreas aqui abordadas.

2.1 Vídeo Digital

Nesta secção são abordados temas relativos à codificação e reprodução de vídeo em formato digital.

Começaremos por falar de alguns formatos de vídeo digital que se encontram neste momento no mercado e com utilização em massa e ainda alguns formatos mais recentes ainda não totalmente normalizados.

Estão também descritos temas relativos à adaptação de vídeo, a difusão do mesmo sobre redes IP e ferramentas utilizadas em ambos os casos.

2.1.1 Formatos de Vídeo

Existem actualmente inúmeros formatos de vídeo normalizados e que são já utilizados no quotidiano doméstico, nomeadamente em IPTV, recorrendo a dispositivos que descodificam o sinal para aparelhos de televisão ou no simples uso da Internet em vídeos são partilhados entre utilizadores.

São aqui abordados dois formatos, escolhidos pela sua predominância e importância na utilização e investigação.

2.1.1.1 MPEG

O MPEG é um formato cuja produção inicial se deu em 1988 aquando da criação de um grupo de trabalho pertencente à ISO/IEC, tendo-se iniciado como uma norma de codificação de vídeo.

Até hoje várias versões foram normalizadas e são utilizadas nas mais diversas áreas. De grande relevância hoje em dia destacam-se as normas MPEG-2 e MPEG-4, no que diz respeito ao vídeo, sendo o MPEG-2 utilizado em larga escala em sistemas de IPTV e o MPEG-4, uma versão mais recente, focada para codificação de vídeo para dispositivos móveis e em redes cuja largura de banda é menor. A descrição detalhada destas normas de codificação ultrapassa o âmbito do trabalho, mas é relevante a referência de que ambos possuem níveis/perfis, que definem a qualidade e as funcionalidades oferecidas pelos mesmos.

De grande relevância para as adaptações estão também normalizados formatos de metadados, criados no mesmo âmbito e que fornecem valor acrescentado aos conteúdos vídeo, associando ao mesmo conteúdos descritivos. Aqui inserem-se os formatos MPEG-7 e MPEG-21.

No MPEG-21 está já normalizada uma secção denominada de *Digital Item Adaptation* (DIA), sendo que esta tem como objectivo o fornecimento de conteúdos de forma personalizada e de parâmetros para adaptação de forma a cumprir restrições impostas pelos dispositivos e pelas redes.

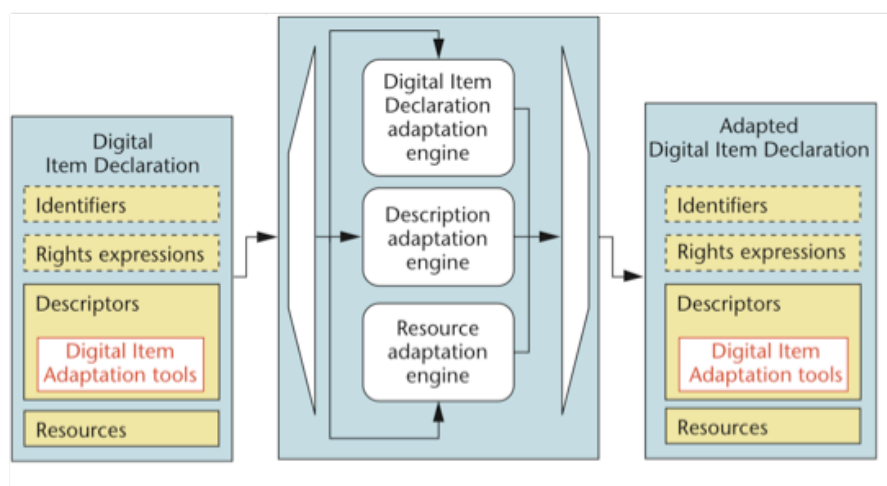


Figura 2.1: Exemplo de funcionamento do DIA

A descrição do ambiente de utilização que vem normalizada é abordada em?, onde apresenta quatro grupos descritivos, de grande importância no que diz respeito às adaptações a realizar e que serão abordados nos pontos seguintes.

2.1.2 Ferramentas de *Streaming*

Neste ponto são identificadas algumas ferramentas de *Streaming* que se encontram disponíveis na Internet e possuem licenças públicas. É feita também referência a uma ferramenta, que apesar de não ter sido possível testar é de grande importância para o assunto, pelas funcionalidades que oferece.

2.1.2.1 FFmpeg



Figura 2.2: Logótipo do FFmpeg

O FFMPEG ? é uma ferramenta que suporta as funcionalidades de gravação, conversão e streaming de áudio e vídeo. Este possui uma vasta gama de codificadores e decodificadores, permitindo a alteração dos parâmetros de vídeo a transmitir.

O FFMPEG está a ser desenvolvido em Linux, embora suporte os variados sistemas operativos.

2.1.2.2 VideoLAN



Figura 2.3: Logótipo do VideoLAN

O VideoLAN ? é uma ferramenta para descodificação de vídeo já utilizada em grande escala, possui também a possibilidade de instalação de um servidor para streaming com capacidades avançadas de transcodificação.

Tal como a ferramenta anterior, suporta as várias arquitecturas de sistemas operativos. Embora ambas sejam muito semelhantes, este último possui um interface gráfico de configuração.

2.1.2.3 ISABEL-lite

Apesar de não estarem ainda disponíveis aplicações que suportem a adaptação, existem muitas em desenvolvimento, encontrando-se em testes relacionados com o tema desta tese, sendo utilizadas pelos grupos que as desenvolvem.

Um dos exemplos é a aplicação ISABEL-lite, utilizada pelo grupo de trabalho responsável pelo lançamento do artigo ?.

Para a realização do trabalho este grupo adaptou uma aplicação denominada de ISABEL ?, utilizada para tele-conferência de forma a que esta adapte o seu comportamento dinamicamente e em tempo real aos recursos disponíveis.

Esta aplicação permite de uma forma dinâmica e em tempo real a adaptação do conteúdo a ser transmitido, nomeadamente:

- **Esquemas de codificação.** A aplicação pode interagir com conteúdos codificados por várias normas, tais como MPEG/H.261/H.263/MJPEG para vídeo, ou GSM/G.722/G.711 para áudio. A aplicação é capaz de escolher entre estes formatos à escolha do utilizador ou baseada em informação recebida de camadas mais baixas.
- **Sampling rate**, consiste na diminuição do número de *frames* a serem transmitidos por segundo, resultando numa diminuição de qualidade temporal mas muitas vezes bem suportada.
- **Tamanho do componente**, obtendo ganhos substanciais aquando de uma transmissão ajustada ao ecrã do dispositivo consumidor.
- **Utilização de componentes**, dependendo do *codec* em utilização, alguns fornecem a capacidade de enviar ou não todos os componentes de uma imagem, oferecendo assim a possibilidade de enviar apenas elementos relevantes da imagem.
- **Buffering**, representa a capacidade de adaptação do tamanho do *buffer*, aliviando assim efeitos provocados pelo *jitter*.

Observando as funcionalidades oferecidas por esta aplicação, podemos observar os ganhos que existiram, na procura da qualidade de rede aquando da disponibilidade de ferramentas deste género ao público em geral.

2.2 Redes e Tráfego IP

Nesta secção ficam descritos os parâmetros de rede pertinentes e identificados na composição do sistema e ainda ferramentas utilizadas na captura destes valores.

2.2.1 Características e Parâmetros

Neste ponto ficam apresentados alguns dos parâmetros de qualidade da rede que são afectados e afectam por sua vez todo o tráfego na mesma.

Sendo a rede IP a mais utilizada hoje em dia e que fornece acesso à Internet, na sua generalidade não fornece garantias de serviço tornando por isso todos os parâmetros de uma determinada ligação, variáveis cuja previsão traria grandes benefícios na procura da qualidade de serviço.

Para a realização do estudo de adaptação preditiva será necessário proceder ao levantamento de uma elevada quantidade de dados de rede afim de alimentar o treino de algoritmos de *machine learning*, desta forma pretende-se que os algoritmos sejam capazes de prever o comportamento da rede num futuro próximo.

Esta previsão, para além de fundamental na busca de uma melhor qualidade de serviço para consumidores de elementos multimédia poderá contribuir para um melhor funcionamento das redes, evitando situações de congestão tanto nos servidores, como em qualquer outro dispositivo que faça parte dessas mesmas redes.

De acordo com a revisão bibliográfica descrita mais à frente são destacados os seguintes parâmetros para recolha:

- **Largura de Banda**, que representa a velocidade máxima de transmissão, representado em *bits* por segundo.
- **Atraso**, representa o tempo desde que uma informação é transmitida até à sua chegada ao terminal de destino, medido em milissegundos.
- **Jitter**, é uma medida de variação do atraso, sendo um parâmetro que afecta em especial aplicações em tempo real, como o streaming de vídeo.
- **Perdas de Pacotes**, este é um dos principais parâmetros de qualidade de uma rede, e como o próprio nome indica representa os pacotes que não chegaram ao terminal de destino, representado em valores percentuais.

2.2.2 Ferramentas Utilizadas

Neste ponto apresenta-se uma breve descrição sobre as ferramentas utilizadas na componente do trabalho relacionada com os parâmetros de rede. Temos ferramentas com capacidade tanto de teste das redes como de captura de dados e análise dos mesmos.

- **WIRESHARK ?**, esta ferramenta consiste num analisador de protocolos de rede, sendo mesmo a mais famosa e utilizada como *standard* em variadas indústrias e instituições educacionais.

Esta ferramenta funciona nos variados sistemas operativos existentes no mercado, é capaz de identificar os mais variados protocolos de rede, e suporta tecnologias de transporte desde *Ehternet* a *Bluethooth*.

Apresenta ainda como grande vantagem a capacidade de exportar os dados para variados formatos, incluindo CSV, afim de criar gráficos para análise.

- **TCPDUMP ?**, esta aplicação oferece ao utilizador a capacidade de observar e capturar os cabeçalhos de pacotes num interface de rede em tempo real. Fazendo passar estes dados a uma outra aplicação, variados parâmetros de rede podem ser calculados.

Das principais opções que esta aplicação fornece é o filtro, que a torna capaz de fornecer dados sobre uma específica e determinada ligação ignorando todas as restantes.

A vantagem desta ferramenta em relação à anterior é a possibilidade de aceder à mesma em tempo real para ser utilizada pelo programa desenvolvido no âmbito desta tese.

- **Multi-Generator (MGEN) Network Test Tool ?**, esta aplicação permite a realização de testes e medições de performance em redes IP recorrendo à injeção de tráfego UDP.

Através da criação de *scripts* permite recriar padrões de tráfego na rede afim de testar as redes com cargas variadas.

O tráfego gerado pode ser visualizado em tempo real, ou armazenado em ficheiro possibilitando assim análises em tempo real ou mais tarde por análise dos ficheiros de dados.

Os resultados da geração de tráfego podem estar a ser visualizados em tempo real ou armazenados em ficheiro, o que é uma grande vantagem para o treino dos algoritmos de *machine learning*.

Uma limitação encontrada nesta aplicação é a incapacidade de, em tempo real, fornecer mais do que um parâmetro supra referido. Isto significa que se estivermos a realizar uma análise ao atraso da comunicação não é possível verificar ao mesmo tempo a perda de pacotes. Para a obtenção dos vários parâmetros de rede é necessário o armazenamento dos dados e a realização de uma análise posterior.

- **TRace Plot Real-time (TRPR) ?**, este programa é utilizado em conjunto com o MGEN, e executa a medição dos parâmetros de rede, ou seja, faz a leitura do cabeçalho dos pacotes capturados e utiliza esses dados para fornecer valores exactos sobre o parâmetro a ser analisado.

Esta pode também ser utilizada em tempo real, recolhendo dados capturados tanto pelo MGEN como pelo TCPDUMP, ou fazendo a análise de dados armazenados em ficheiros.

Desta forma conclui-se a apresentação de ferramentas disponíveis para geração e monitorização de tráfego em redes IP.

2.3 Data Mining

Data Mining é definido como o processo de análise de dados que tem por objectivo identificar padrões ou relações entre os dados recolhidos mediante a análise de grandes volumes de informação, o *data mining* visa encontrar relações que permitam estabelecer modelos até então desconhecidos.

Descobertos os padrões, será então possível determinar regras que permitam prever resultados ou comportamentos futuros. Para o conseguir, estas aplicações servem-se de técnicas de matemática e estatística, bem como de variadas tecnologias de reconhecimento de padrões.

Relativamente aos métodos de aprendizagem dos algoritmos de machine learning agrupam-se em dois grandes tipos, a aprendizagem **supervisionada** e **não supervisionada**.

- A **aprendizagem supervisionada** consiste no mapeamento de valores de entrada com as desejadas saídas, ou respostas para esse conjunto. Neste modo de aprendizagem tem de ser escolhido um conjunto de dados medidos ou recolhidos do mundo real e as respectivas respostas (ver Figura 2.4). Após a recolha destes dados, é necessário escolher a forma de representação dos mesmos (um novo espaço de representação); este procedimento é muito importante, pois os mesmos dados representados de formas diferentes podem criar resultados muito distintos.
- No modo de **aprendizagem não supervisionada** são apenas fornecidos os dados de entrada aos algoritmos. Não conhecendo a resposta pretendida, cabe ao algoritmo em utilização tentar reconhecer a distribuição e comportamento dos dados de entrada. O objectivo aqui não é fazer previsões mas sim reconhecer como se agrupam os dados.

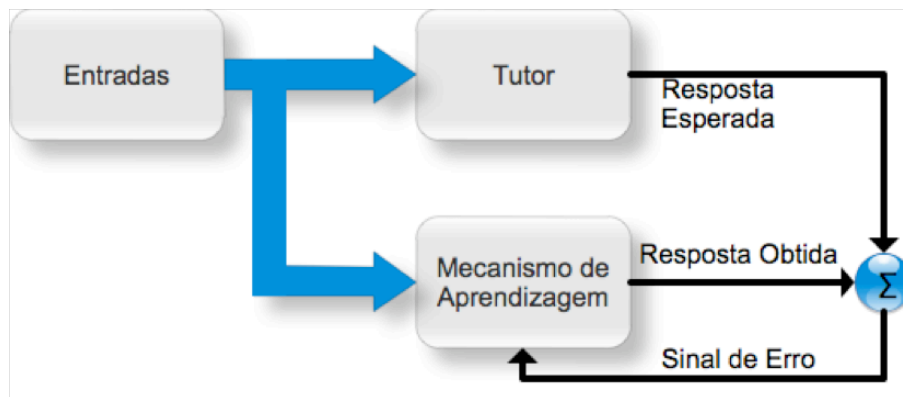


Figura 2.4: Exemplo de aprendizagem supervisionada

Os algoritmos de *aprendizagem supervisionada* dividem-se em dois grupos principais, os algoritmos de **classificação** e os de **regressão**. A distinção faz-se ao nível da variável a prever.

- Os **modelos de regressão** são modelos que assentam geralmente em análise estatística. Em regressão a variável a prever pelo modelo toma valores em R . O processo de desenho do modelo consiste na minimização de uma função de custo pré-definida, geralmente baseada no erro quadrático médio.
- Os **modelos de classificação** tentam prever o resultado de uma variável que pode tomar apenas um número finito de valores. Caso a variável a prever possa tomar apenas dois valores temos um problema de classificação binária; caso o número de valores possíveis seja superior a dois temos um problema de classificação multi-classe. O problema multi-classe pode ainda ser categorizado em classificação multi-classe ordinal caso existe uma ordem natural entre os valores possíveis a prever (por exemplo, quando os valores possíveis de saída são "excelente", "bom", "razoável" e "mau") ou em classificação multi-classe nominal quando essa ordem não existe.

Uma vez que o objectivo desta tese é a previsão de condições de rede, que se assume poder tomar qualquer valor real, será utilizado o modelo de regressão com uma aprendizagem supervisionada. Desta forma serão recolhidos dados que depois de modelados serão apresentados ao algoritmo de teste.

Nas seguintes secções será descrita uma ferramenta para aprendizagem e teste de algoritmos de *machine learning* e o funcionamento de alguns algoritmos.

2.3.1 Ferramenta de *Machine Learning*

De entre várias ferramentas de treino destes algoritmos foi revista a plataforma Weka [?], esta ferramenta é implementada em Java, além de conter vários algoritmos de Machine

Learning possui ainda ferramentas de pré-processamento, classificação e visualização dos dados. Vejamos em seguida alguns dos algoritmos a serem estudados.



Figura 2.5: Logótipo do WEKA

2.3.2 Algoritmos de *Machine Learning*

Nesta secção encontram-se brevemente descritos alguns algoritmos utilizados em *machine learning*. Foram escolhidos apenas seis algoritmos mas de forma que fossem utilizados os vários grupos que distinguem a execução dos algoritmos:

- **Funções** - Regressão Linear, Redes Neurais Artificiais, Máquina de Vectores de Suporte e Processos Gaussianos
- **Árvores** - M5P
- **Lazy** - *Locally Weighted Learning*

A descrição não é muito aprofundada, em especial em determinados algoritmos que têm procedimentos muito complexos uma vez que o funcionamento dos mesmos não se enquadra no âmbito desta tese, ficando portanto apenas uma ligeira abordagem a cada um destes algoritmos.

2.3.2.1 Máquinas de Vectores de Suporte (SVM)

O algoritmo de vectores de suporte é uma generalização não linear de um algoritmo denominado *Generalized Portrait* desenvolvido na Rússia nos anos sessenta.

Na sua forma actual foi desenvolvido na *AT&T Bell Laboratories* e devido ao seu contexto industrial foi orientado a aplicações do "mundo real".

Este algoritmo é um método de aprendizagem supervisionada, na qual conhecendo a classe que gerou cada padrão de treino, o classificador é treinado para replicar a decisão correcta para todos os padrões de treino. Os dados são vistos como dois conjuntos de vectores num espaço de n dimensões. Vejamos o funcionamento do SVM de regressão.

Neste modelo o objectivo é encontrar uma função que apresente no máximo um desvio ϵ entre os resultados dados pelo modelo e os valores reais no conjunto de treino. Desta forma, desde que os valores a prever se encontrem dentro da banda limitada por ϵ o erro

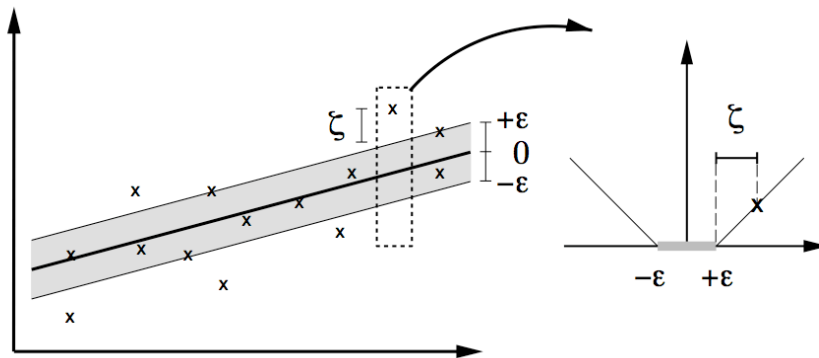


Figura 2.6: Princípio de funcionamento de SVM ?

é considerado irrelevante (ϵ -insensitivity zone). Podemos observar este funcionamento na figura 2.6.

Como podemos observar na figura 2.6, apenas os pontos localizados fora da região a sombreado são considerados como um erro, sendo os mesmos descartados do modelo de aprendizagem.

Durante o processo de desenho do modelo, em que a função de regressão vai ser definida, o processo de otimização tenta minimizar o número de exemplos de treino incorrectamente previsto pelo modelo (pontos que caem fora da zona- ϵ) e minimizar a complexidade do próprio modelo. O processo de otimização (quadrática) tenta conciliar este objectivos contraditórios ?.

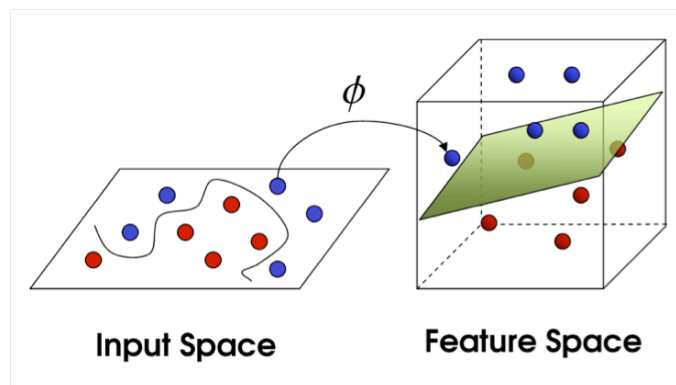


Figura 2.7: Exemplo de funcionamento da kernel

Quando estamos interessados em modelos de regressão não linear é possível transformar o espaço original dos dados num espaço apropriado (geralmente de dimensão superior), onde o modelo de regressão desejado seja transformado num modelo linear. Esta abordagem é genérica e pode ser adoptada com qualquer classe de algoritmos de regressão, como ilustrado na Figura 2.7. Os SVMs permitem-nos evitar fazer esta transformação de forma explícita, recorrendo ao *kernel trick* ?. Recorrendo a funções-núcleo

apropriadas, é possível capturar a não-linearidade pretendida sem nunca sair do espaço original dos dados.

2.3.2.2 Redes Neurais Artificiais (RNA)

Na gênese das redes neuronais artificiais está uma tentativa de reproduzir um modelo simplificado do comportamento do cérebro humano. Este tema tem sido alvo de inúmeros trabalhos científicos fruto fundamentalmente da diversidade de áreas de aplicação das redes neuronais. A título de exemplo pode-se mencionar áreas como a robótica, auxílio ao diagnóstico médico, segmentação de imagens, reconhecimento de padrões (faces, gestos, OCR, voz, etc.), previsão de acontecimentos, etc.

Estabelecendo a analogia com o funcionamento cérebro humano, as redes neuronais recebem estímulos e tomam decisões procurando deduzir correlações entre os padrões de entrada e resultados à saída.

As suas principais vantagens são a tolerância a falhas (resultante da sua estrutura em paralelo que permite que um valor que seja assumido como um erro tenha uma menor influência), a sua capacidade para actuar em situações de decisões em tempo real e uma grande capacidade de adaptação.

As RNA têm vindo a ser estudadas nas mais diversas áreas de aplicação mas a título de exemplo fica apenas a sua referência em aplicações de predição ??.

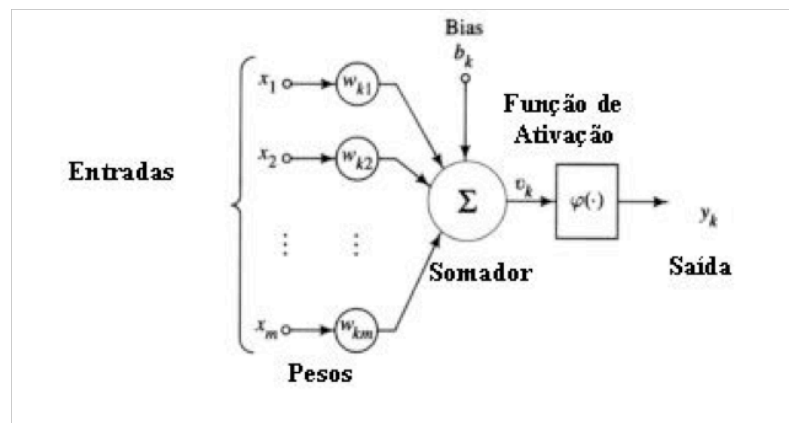


Figura 2.8: Funcionamento de um neurónio artificial

O modelo apresentado na figura 2.8 apresenta a estrutura mais simples deste algoritmo, um neurónio, que é composto pelas seguintes partes:

- **Entradas**, que têm como origem as saídas de outros neurónios ou entradas externas, tais como os dados a serem avaliados.
- **Combinação de Entradas**, representado na figura pelo somatório, que faz a respectiva soma das entradas tendo em conta pesos associados, representados por w .

- **Função de Activação**, que determina a saída, ou estado do neurónio.

A estrutura apresentada anteriormente na figura 2.8, utilizada neste algoritmo pode ser estendida, ou melhor podem ser ligadas várias destas estruturas de forma a realizar uma rede mais complexa como se pode ver na figura 2.9.

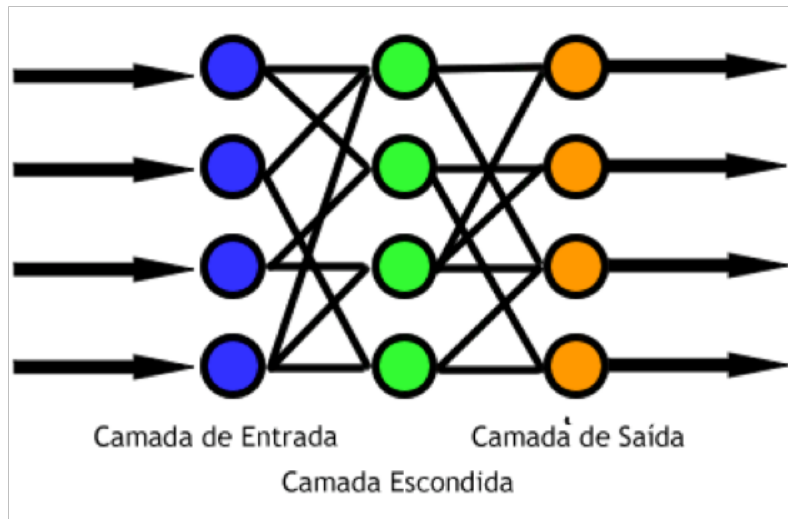


Figura 2.9: Exemplo de uma rede *feedforward*

Este modelo é conhecido por tipo de rede feedforward e torna-se útil em situações de não-linearidade entre classes, um modelo conhecido por **perceptrão multicamada** aquando da existência de pelo menos três camadas.

A capacidade de adaptação fornecida por estes algoritmos é de extrema importância no âmbito deste problema uma vez que é impossível treinar o mecanismo de adaptação para todas as situações com que se irá deparar.

Durante a fase de treino os pesos w_i da rede são otimizados para que a rede aprenda a relação entre padrões de entrada e saída. No algoritmo de treino mais comum, o algoritmo de retropropagação, os pesos sinápticos são aleatoriamente inicializados com valores pequenos devido à ausência de conhecimento. Depois são fornecidos à rede exemplos de entrada e a respectiva saída supervisionada. Com base nesta informação, a rede aplica a entrada recebida e determina a sua estimativa para a saída. O erro entre a estimativa e a saída desejada é retropropagado até à entrada para ajustar iterativamente os coeficientes das conexões de modo a minimizar os erros entre a estimativa e os exemplos supervisionados. Casos os pesos sejam ajustados apenas depois da rede ver todos os exemplos de treino temos o modo de aprendizagem em **Batch**, desta forma os pesos atribuídos em cada neurónio só são definidos após o teste de todos os valores de entrada e saída. No modo **Incremental** os pesos são actualizados após cada caso de treino ser alimentado à rede, ou seja, antes do final do processamento de todos os dados disponíveis para a aprendizagem do algoritmo.

Na utilização do modelo **perceptrão multicamada**, uma primeira fase de aprendizagem supervisionada é imperativa, para que os coeficientes de conexões sejam ajustados automaticamente, processo que podemos observar no fluxo correspondente à figura anterior.

2.3.2.3 Regressão Linear ?

O modelo de regressão linear tem sido vastamente utilizado em aplicações estatísticas há décadas pela sua simplicidade na predição numérica embora estejam restritos à linearidade dos dados a prever.

Assumamos que temos um conjunto de treino com N observações de x , em conjunto com as saídas correspondentes y . O objectivo é explorar este conjunto de treino para fazer estimativas \hat{y} da variável alvo para novos casos x da variável de entrada. Isto envolve tentar descobrir um modelo para gerar os valores y , que no caso linear toma a forma

$$y_n(x) = w_0 + \sum_{d=1}^D w_d x_{n,d} \quad (2.1)$$

($x_{n,d}$ é a coordenada d da observação n)

Para ajustar o modelo linear aos dados de treino e encontrar os parâmetros óptimos w_i , iremos minimizar uma função de erro pré-definida. Uma escolha simples para a função de erro é a soma dos quadrados dos erros entre a estimativa dada pelo modelo e o valor observado no conjunto de treino:

$$Erro\{w_0, w_1, \dots, w_D\} = \frac{1}{2} \sum_{n=1}^N \left\{ w_0 + \sum_{d=1}^D w_d x_{n,d} - y_n \right\}^2 \quad (2.2)$$

Adoptando uma notação matricial podemos re-escrever a expressão anterior como:

$$\mathbf{y} = \text{matriz } N \times 1 \quad (2.3)$$

$$\mathbf{X} = \text{matriz } N \times D \quad (2.4)$$

$$\mathbf{w} = \text{matriz } (D+1) \times 1 \quad (2.5)$$

O erro pode ser re-escrito como:

$$Erro = (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w}) \quad (2.6)$$

O valor óptimo para \mathbf{w} vem igual a:

$$\mathbf{w} = (X^T X)^{-1} X^T \mathbf{y} \quad (2.7)$$

2.3.2.4 Processos Gaussianos ?

Os processos Gaussianos consistem numa generalização da função de probabilidade de distribuição Gaussiana, onde a probabilidade de distribuição descreve variáveis aleatórias que podem ser escalares ou vectores.

De uma forma simplificada podemos pensar numa função como sendo um vector muito longo, em que cada entrada no vector especifique o valor da função $f(x)$ para uma determinada entrada x .

O objectivo deste algoritmo, tal como os restantes utilizados em *machine learning*, é a de fornecer valores de resposta baseados em aproximações de outros valores de resposta que se conheçam à priori. A aproximação criada por este algoritmo consiste então em atribuir probabilidades mais elevadas a funções consideradas mais prováveis, a título de exemplo, podemos atribuir probabilidades mais elevadas a funções que se apresentem mais "suaves", em termos de variações, que outras.

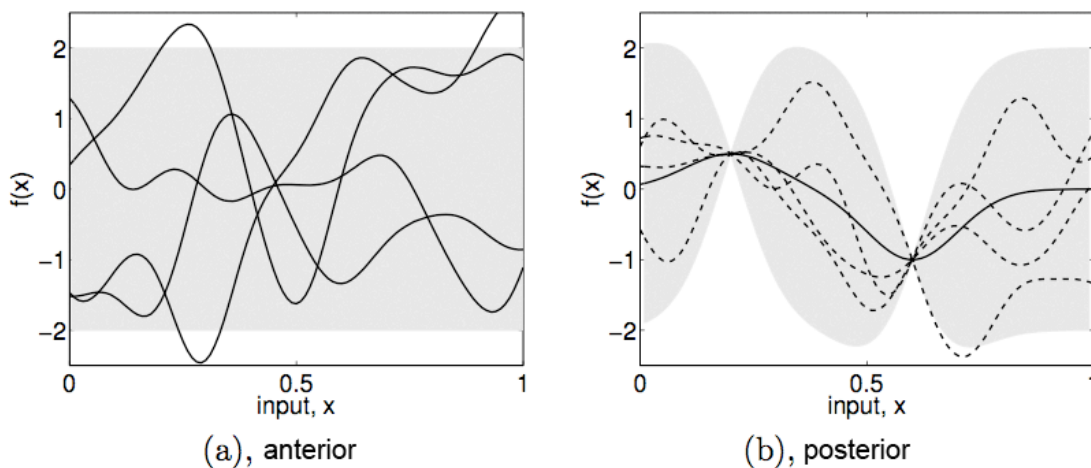


Figura 2.10: Gráfico (a) demonstra quatro exemplos de dados. No gráfico (b) observamos a situação após dois pontos terem sido observados. A previsão média encontra-se na linha contínua e a tracejada são quatro amostras dos dados anteriores. Em ambos os gráficos a região sombreada representa o dobro do desvio médio a cada valor de entrada, x .

Na figura 2.10 podemos observar uma abordagem deste método a um conjunto de dados.

Este algoritmo consiste então na utilização do método gaussiano, processo descrito acima. Desta forma é criada uma generalização, ponderada, e escolha de valores a utilizar de forma a aproximar uma função aos dados existente, dotando-se desta forma da capacidade de fornecer valores de resposta sobre valores que à partida não possui informação.

2.3.2.5 MP5 ?

O algoritmo M5P é baseado numa ideia simples que assenta na construção de árvores de decisão. As árvores de decisão são sistemas hierárquicos em que certas condições são sequencialmente testadas até que um valor seja previsto para a saída. As questões fulcrais no desenho de árvores de regressão são:

- a selecção do teste a ser realizado em cada nó. Cada teste corresponde a uma divisão binária em dois nós descendentes. Cada nó tem associado um dado subconjunto S do conjunto de treino. A divisão do nó equivale à divisão do subconjunto S em dois subconjuntos disjuntos.
- o critério de divisão, que permite seleccionar a melhor divisão do conjunto de teste possíveis.
- o critério de paragem que controla o crescimento da árvore e a declaração de um nó como nó-folha.
- a regra da atribuição de um valor aos nós-folha.

Por vezes, evita-se a paragem do crescimento da árvore, alterando critério para o mesmo, deixando a árvore crescer até atingir um valor excessivo. Posteriormente efectua-se a "poda" da árvore, podando-se a árvore de acordo com algum critério que entre em consideração com a complexidade da árvore e o erro no conjunto de treino. Este método vem desde Breiman et al. (1984) e Quinlan (1986) e geralmente conduz a melhores resultados do que tentar controlar o crescimento da árvore.

2.3.2.6 Locally Weighted Learning ?

Os métodos de aprendizagem denominados de *Lazy* atrasam o processo de treino do conjunto de dados até que seja pedida uma previsão.

Desta forma, este algoritmo requer guardar os dados de treino em memória e aquando de um pedido de avaliação/previsão esses dados são pesquisados afim de fornecer uma resposta. A relevância dada a cada conjunto de dados é dada recorrendo a funções de distância, sendo que os pontos do conjunto de treino mais próximos do valor a ser pesquisado são aqueles que obtêm mais pontos de relevância.

Este algoritmo fornece várias funções de medição de peso/relevância de dados, tais como:

- **Kernel Gaussiana;**
- **Exponencial;**
- **Linear;**

- **Tricúbica.**

Cada um destes métodos, representa a função que desenha a curva de aproximação, ou seja, a relevância dada aos dados pertencentes ao conjunto de treino. Assim, pedida a avaliação de um novo valor, consegue-se extrapolar, de forma ponderada e baseada numa função, os valores conhecidos afim de obter a previsão/avaliação.

2.4 Adaptação de Vídeo

Vários estudos têm vindo a ser realizados com o objectivo de fornecer melhor qualidade de serviço no fornecimento de vídeo pela Internet ????. Estes têm vindo a focar-se em ferramentas de machine learning de forma a prever comportamentos da rede, e existem também estudos de forma a identificar as melhores opções de adaptação, visto que como referido anteriormente, cumprir restrições impostas pela rede ou pelos terminais não leva a uma solução única de adaptação ?.

Apresentam-se em baixo vários parâmetros de definição de um vídeo digital, sobre os quais o mecanismo de adaptação poderá actuar resultando assim na adaptação do conteúdo do vídeo, obtendo assim diferentes resultados de qualidade mas também de capacidade de rede necessária à transmissão dos mesmos. Tendo em conta que o foco do trabalho se encontra no *timing* em que a adaptação deve ocorrer, ficam descritos apenas alguns dos muitos parâmetros que definem um vídeo digital, resultando em adaptações **temporais e espaciais**.

- **Frames por segundo**, que tal como próprio nome indica, define o número de imagens por segundo a serem apresentadas.
- **Tamanho dos frames**, afectando a qualidade final, mas que se encontra directamente relacionado com o terminal em que o vídeo vai ser visualizado, permitindo ganhos muito significativos aquando da sua alteração.
- **GOP (Group Of Pictures)**, esta é uma variável intrínseca dos codificadores MPEG e define o número de imagens entre imagens do tipo I, estas imagens são importantes por fornecerem pontos de acesso ao vídeo, mas são também responsáveis pelo aumento dos dados transferidos.

2.5 Revisão Bibliográfica de Trabalhos Publicados

Nesta secção são analisados alguns trabalhos relevantes para a realização desta tese. Estes trabalhos têm em comum a procura da oferta de uma melhor qualidade de experiência para os utilizadores de recursos multimédia.

2.5.1 Decisões de Adaptação

No artigo ? é feito um estudo sobre as decisões de adaptação de conteúdos multimédia.

Este grupo propõe uma nova abordagem à procura de qualidade de serviço (QoS), tomando as decisões de adaptação não apenas a pensar em cumprir as restrições impostas pela rede mas adicionando um "parâmetro" que é a percepção que o utilizador experimenta ou QoE.

Como se sabe, num cenário de adaptação em que têm de ser cumpridas as restrições impostas pela rede, existem inúmeros factores ou conjuntos de factores que podem ser alterados de forma a cumprir essas restrições (p.ex. diminuir o tamanho do vídeo, alterar o formato do áudio ou vídeo, entre outros). E a maioria dos sistemas apenas selecciona um conjunto de opções que satisfaça as condições impostas pela rede.

Do ponto de vista dos autores do artigo acima referido, nem todos os conjuntos de opções que satisfazem a mesma restrição, ou conjunto de restrições, oferecem a mesma qualidade. Assim, realizando testes com utilizadores reais foram avaliadas as várias formas de adaptação possíveis, atribuindo pontuações às mesmas.

Uma vez que a forma de avaliar a qualidade perceptível pelos utilizadores não pode ser modelada analiticamente, as pontuações dadas pelos utilizadores são fornecidas como entrada num algoritmo de *machine learning* baseado em regras, denominado de SLIPPER ?.

O resultado deste trabalho demonstrou que a adaptação dinâmica pode beneficiar em muito com a utilização de algoritmos de *machine learning* na tomada de decisão em como adaptar um *stream* de forma a cumprir as restrições impostas no momento e fornecer assim uma melhor experiência ao utilizador que pode inclusivamente passar pela não utilização máxima dos recursos disponíveis.

Uma outra abordagem é descrita no artigo ?.

Aqui, os autores identificam que a adaptação deve também ter em atenção o conteúdo do vídeo. Desta forma utilizam algoritmos de *machine learning*, não para tomar a decisão de adaptação, mas sim para classificar esse conteúdo.

Do ponto de vista da adaptação recorrem a um processo simples, em que para cada *bitrate* disponível fazem corresponder três tipos de adaptação, temporais, nomeadamente a taxa de imagens por segundo total, metade e um quarto da qualidade temporal inicial do vídeo.

Como o conteúdo dos vídeo pode ser classificado com complexidade baixa, média e alta, a cada uma destas classificações pode corresponder qualquer uma das adaptações descritas acima e tal adaptação é realizada com base em preferências dadas por utilizadores e recolhidas previamente.

2.5.2 Classificação das Condições de Rede

Num trabalho publicado em 2006 ? são utilizadas Máquinas de Vectores de Suporte (SVM) para classificar eventos de congestão ou saturação em redes sem fios.

O grupo de trabalho responsável pela publicação consegue com sucesso detectar e classificar tanto o grau de congestão da rede como a perda de sinal do canal sem fios. Para tal são utilizadas métricas de rede simples, tais como o atraso, o *jitter*, a taxa de transferência e a taxa de perda de pacotes.

2.5.3 Previsão de Condições de Rede

No ano 2005 existiam já algumas tentativas de previsão da qualidade da rede em comunicações multimédia.

No artigo ?, um grupo de trabalho da Austrália é capaz de prever a condição oferecida por uma rede sem fios recorrendo à posição geográfica dos utilizadores.

Após uma primeira recolha de dados e mapeamento das informações, é criada uma base de dados correlacionando qualidade de serviço com posições geográficas.

Desta forma, não só são capazes de prever a qualidade de serviço que um utilizador vai "experimentar" aquando da tua movimentação, como também conseguem fornecer informação aos utilizadores de que caminho devem seguir afim de melhorarem as suas ligações.

As métricas utilizadas na recolha de dados inicial são as mais comuns e referidas já noutros trabalhos como o atraso e perda de pacotes, mas com duas adições de vital importância para o sucesso da sua aplicação, sendo estas a potência de sinal recebido no dispositivo terminal e informação GPS dos mesmos.

Mais uma vez, foram também utilizados métodos de *machine learning* para efectuar as previsões, sendo o algoritmo escolhido por este grupo as **Redes Neurais Artificiais**.

2.5.4 Arquitectura de Gestão de Redes

Em 2008, um grupo belga desenvolveu toda uma arquitectura de optimização de recursos descrita no artigo ?.

A arquitectura desenvolvida faz uso da perda de pacotes numa rede, antes e depois da utilização de um código corrector de erros conhecido como FEC, da largura de banda e na avaliação da qualidade de vídeo recorrem ao PSNR, uma forma objectiva de avaliação da qualidade vídeo.

A relação potência de sinal-ruído (PSNR) é utilizada na fase de testes, fazendo corresponder a este cada conjunto de dados de métrica de qualidade na rede.

Com estes dados, e uma arquitectura algo complexa, recorrendo à ajuda de uma rede neuronal artificial (RNA) tomam decisões sob duas formas, podendo ser feita uma alteração ao *bitrate* do vídeo e/ou à quantidade de FEC a utilizar de modo a corrigir erros evitando a perda de pacotes por esse motivo.

2.6 Análise

Tendo em vista o estudo efectuado nas áreas explicadas nas secções anteriores, compreende-se que a adaptação de conteúdo multimédia tem aumentado de relevância nestes últimos anos.

Muito do estudo a ser feito prende-se com a adaptação de vídeo mas numa situação tardia, em que a experiência do utilizador já foi afectada de uma forma negativa. Pretende-se então com esta tese estudar a possibilidade de **detecção antecipada** de situações comprometedoras da qualidade perceptível pelo utilizador (QoE), matéria ainda muito pouco explorada na bibliografia disponível e em que os benefícios seriam de extrema importância numa altura em que existe uma grande procura pela qualidade uma vez que os mesmos recursos são disponibilizados por diversas fontes.

Capítulo 3

Planeamento do Sistema de Adaptação Preditiva

Neste capítulo são abordados os aspectos práticos iniciais do trabalho, ou seja, o planeamento e recolha de dados que permitam o desenvolvimento da aplicação em si.

3.1 Introdução

Tendo realizado a revisão de literatura, podemos identificar a extrema importância no estudo da identificação antecipada de situações de alteração nas condições de uma rede, onde se estejam a transferir conteúdos multimédia para visionamento em tempo real, alterações essas que afectam a qualidade percebida pelo utilizador de uma forma geralmente muito negativa.

Embora o estudo sobre que decisões tomar aquando da realização de uma adaptação de codificação de um vídeo esteja já relativamente avançada, a verdade é que um motor que identifique quando tais alterações devem ocorrer ainda se encontra em fases muito iniciais, sendo os estudos realizados na área sujeitos a situações estáticas em que se conhece a arquitectura da rede e do espaço físico onde a mesma se insere.

Pretendemos assim, realizar um estudo de viabilidade de detecção dos momentos em que as condições da rede se alterem ao ponto de ser necessário uma adaptação no vídeo a ser transmitido, utilizando os dados que possam ser "lidos" do próprio terminal ou de um nó na rede.

As vantagens de um sistema capaz de detectar com alguma antecedência condições de saturação ou congestão na rede são inegáveis, não só na transmissão de conteúdos vídeo em tempo real, mas na realidade, para qualquer tipo de transmissão em que a "palavra-chave" seja "tempo real".

3.2 Requisitos do Sistema

Nesta secção estão identificadas as funcionalidades básicas necessárias à operação do sistema.

Tendo em vista o objectivo de identificar de forma contínua/recorrente condições de saturação da rede apresenta-se na figura 3.1 um diagrama da lógica de operação, identificando os ciclos e caminhos necessários para atingir a resposta à necessidade ou não de alteração dos parâmetros de transcodificação do vídeo.

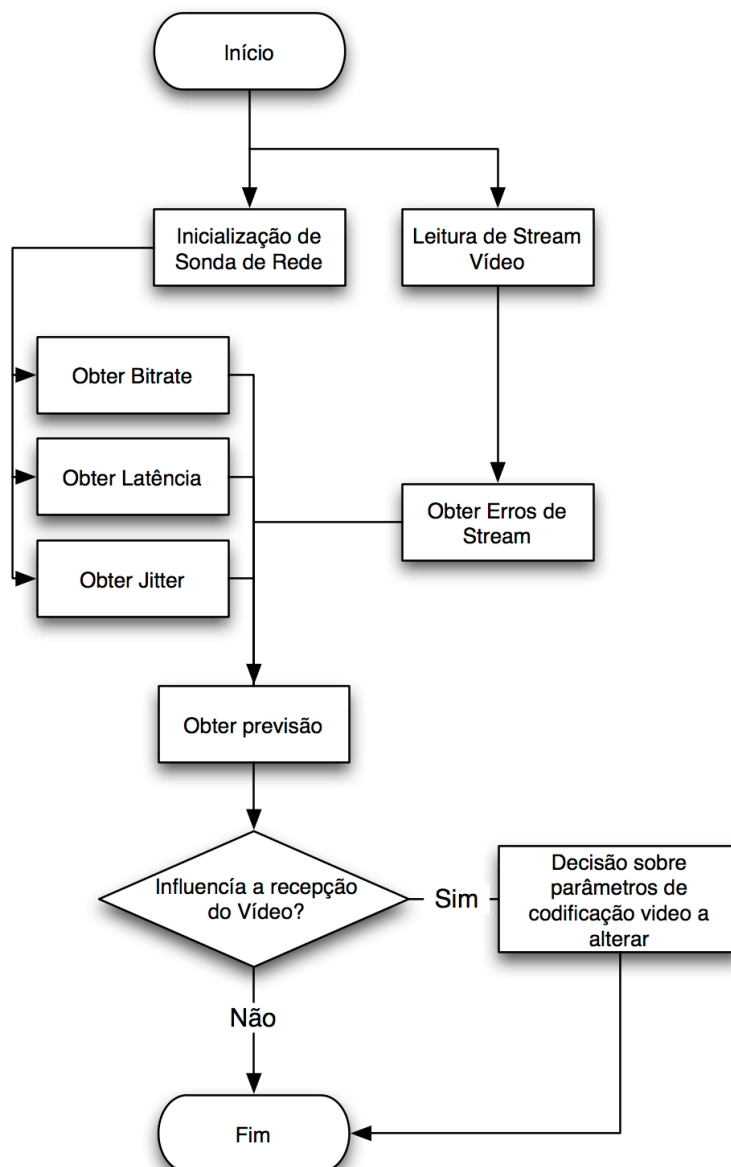


Figura 3.1: Diagrama de lógica de operação

A sequência aqui apresentada terá de ser capaz de correr ininterruptamente, enquanto

o vídeo esteja a ser visionado, evitando assim situações de descontinuidades na recepção maximizando sempre a QoE.

Os blocos apresentados na figura 3.1 podem ser divididos em quatro áreas distintas:

- **Visualização de vídeo**, composto por uma aplicação de descodificação e apresentação de vídeo, que possua capacidade de fornecer informações em tempo real sobre erros ou falhas na recepção e reprodução do elemento vídeo.
- **Sonda de rede**, elemento que terá de ser capaz de adquirir, também em tempo real os valores pertinentes e descritos na bibliografia, terá de ser também capaz de armazenar valores de histórico proporcionando uma análise *off-line* desses dados.
- **Algoritmo de *machine learning***, que alimentado pelos valores de histórico de parâmetros de rede e erros de descodificação e visualização do vídeo, terá de ser capaz, atendendo ao seu treino de fornecer uma previsão sobre uma situação em que o vídeo será ou não afectado.
- **Motor de decisão**, está responsável pela redefinição dos parâmetros de transcodificação de vídeo e um possível envio dos mesmos para o servidor que os deverá aplicar, alterando então o *stream* que está a ser transmitido.

Tendo definido alguns dos requisitos necessários para a realização do trabalho da tese, inicia-se o trabalho fazendo um estudo de condições de rede em situação de teste.

3.3 Recolha de Dados em Cenários de Teste

Por forma a dar início ao desenvolvimento prático propriamente dito, foi necessária a avaliação de aplicações de difusão de vídeo sobre redes IP disponíveis no mercado. Apesar de durante o estudo do estado da arte terem sido apresentadas aplicações com capacidade de adaptação de vídeo em tempo real, as mesmas não se encontram disponíveis ao público geral uma vez que são propriedade de grupos ou instituições de investigação.

O trabalho foi então desenvolvido recorrendo a uma ferramenta disponível no mercado, o VLC ? em que, apesar de aquando de uma ocorrência de alteração nos parâmetros de codificação de vídeo exista uma quebra no stream, é capaz de dar início automático à recepção do novo stream minimizando assim essa descontinuidade.

Embora esta não seja a solução final pretendida numa aplicação com o objectivo de adaptar vídeo em tempo real, tendo em vista que as adaptações ocorrem para prevenir situações de erros grosseiros de apresentação de vídeo, esta pequena quebra de reprodução que tem uma duração de milissegundos é muito bem suportada pelos utilizadores ?, ficando o desenvolvimento de uma aplicação ou protocolo capaz de suportar essas adaptações para um trabalho futuro.

Tendo a transmissão de vídeo operacional, foram feitos vários levantamentos de parâmetros da rede durante transmissões tanto de um-para-um (*unicast*) como um-para-muitos (*multicast*).

Com a análise destes dados exclui-se dos testes as transferências *multicast*, embora as mesmas consigam aliviar tráfego da rede devido às suas características, ou seja, uma vez que apenas existe um *stream* na rede, independentemente do número de utilizadores a receberem esse mesmo *stream*. Tal exclusão deve-se ao facto de se pretender criar situações de congestão na rede afim de afectar negativamente o transporte do vídeo sobre a rede. Outro facto entrou em consideração para essa exclusão prende-se com a característica deste trabalho, ou seja, se um utilizador está a receber um vídeo numa qualidade abaixo do aceitável pretende-se alterar os parâmetros de codificação do vídeo, o que ocorreria para todos os utilizadores que "subscreveram" o *stream multicast* e que poderiam não estar perante situações de congestão.

Com o **servidor** de vídeo a operar numa máquina estacionária na rede, as ferramentas de leitura de parâmetros de rede foram instaladas no **cliente**, a máquina que recebe o *stream* de vídeo afim de dar início aos testes e recolha de valores pertinentes.

Estes testes, realizados em redes IP em situações de acesso com e sem fios são constituídos por duas situações distintas:

- **Injecção de tráfego na rede**, recorrendo à ferramenta MGEN, que permite a criação de *scripts* de forma a criar padrões de tráfego, gerando situações de congestão e "alívio" da rede. O vídeo é assim afectado negativamente, tal como esperado nas situações de congestão, recuperando o normal funcionamento aquando do fim das mesmas.
- **Movimentação do cliente**, uma vez que os testes estão a ser efectuados em redes sem fios, o deslocamento do cliente provoca flutuações na potência de sinal recebido e obriga mesmo ao "salto" para outro ponto de acesso sem fios, o que obriga os pacotes pertencentes ao *stream* a serem redireccionados e à inevitável perda de pacotes uma vez que o transporte dos dados do *stream* está a ser realizado sobre UDP, inviabilizando as retransmissões.

Para proceder a estas experiências é iniciada a transmissão de um vídeo para o cliente e durante a mesma é injectado tráfego na rede afim de saturar a mesma. Desta forma, recorrendo às aplicações descritas anteriormente é feita a medição em tempo real das condições da rede.

A figura 3.2 demonstra o gráfico resultante das medições do *bitrate*, durante uma experiência em que estava a ser transmitido um vídeo codificado em MPEG-4 a dois *megabits* por segundo. Estas medições são resultantes da criação de um ficheiro, após o tratamento dos dados recolhidos pela aplicação TCPDUMP ? e pré-processados pelo TRPR ?.

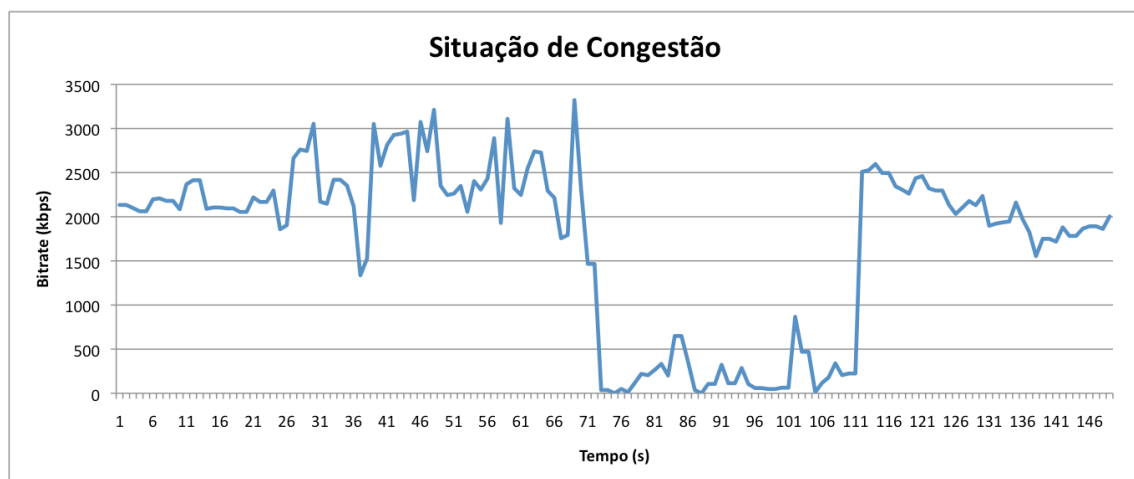


Figura 3.2: Situação de congestão durante a transmissão de um vídeo.

Pode ser observado no gráfico que entre o segundo setenta e cento e dez ocorreu a congestão da rede e que originou na perda de pacotes e consequente quebra na visualização do vídeo.

É precisamente este período que se pretende detectar com antecedência afim de poder accionar o mecanismo de adaptação tentando que o impacto provocado pela congestão seja o menor possível.

Conseguiu-se com estes testes a recolha de elevado número de dados para serem posteriormente "moldados" de forma a habilitar a sua utilização como dados de entrada nos algoritmos de *machine learning*, tanto para a aprendizagem dos mesmos como para a sua avaliação.

Capítulo 4

Teste de Algoritmos de *Machine Learning*

Neste capítulo ficam descritos os procedimentos efectuados afim de testar os diversos algoritmos de *machine learning*.

Para a execução desta tarefa é necessário efectuar o tratamentos dos dados e a formação dos mesmos e só depois foi possível a avaliação dos algoritmos.

4.1 Tratamento de Dados

Como descrito no capítulo anterior, os dados recolhidos nas várias experiências têm de ser modelados afim de se poder utilizar os mesmos para testar vários algoritmos de *machine learning*.

Para a realização do tratamento dos dados foi criada uma aplicação que recolhe os vários ficheiros criados pela aplicação TRPR ?. Uma vez que a mesma para cada parâmetro medido procede à criação de um ficheiro distinto, a aplicação criada lê os dados desses ficheiros e como resultado obtém-se um ficheiro único contendo todos os dados e os cabeçalhos necessários afim de ser dado como entrada no treino e avaliação dos algoritmos de *machine learning*.

De forma a proceder-se a um teste exaustivo dos algoritmos, a aplicação desenvolvida é capaz de, de uma forma dinâmica criar diversos ficheiros de teste com os mesmos valores mas apresentados de formas diferentes. Desta forma, não só se avaliam os algoritmos como também se faz uma optimização pela forma de representação dos dados.

Descreve-se nos pontos seguintes os parâmetros alterados em cada ficheiro:

- **Arredondamento dos valores**, afim de se verificar se a criação de gamas de valores (arredondamento grosseiro) facilitaria na operação dos algoritmos.

- **Tamanho da janela de valores**, variando o número de valores de instantes anteriores que são fornecidos aos algoritmos, ajudando na previsão.
- **Distância entre valores da janela**, representa o número de segundos de distância entre medições. Com a variação deste parâmetro podemos testar o comportamento dos algoritmos, aliviando a função de recolha de valores, em que no caso de existir uma distância de dois segundos ao invés de um, entre cada valor medido, o que produziria ganhos consideráveis de processamento.
- **Atraso da janela**, representa o atraso dos valores a fornecer ao algoritmo em relação ao valor a prever. Este parâmetro é de extrema importância uma vez que quanto mais cedo se consigam obter uma previsão aceitável, mais cedo se pode tomar medidas de adaptação.

Uma vez que o objectivo da previsão da qualidade da rede é o de tomar decisões de adaptação afim de prevenir situações de falha ou quebra de qualidade na reprodução do vídeo, o **atraso da janela** é um parâmetro a maximizar. Caso a antecedência da previsão não seja superior ao tempo necessário para a adaptação a capacidade de adaptação torna-se ineficaz, uma vez que a comunicação da adaptação para o servidor e a própria adaptação do *stream* são medidas que necessitam de ser executadas atempadamente.

Apresenta-se na tabela 4.1 a definição dos parâmetros utilizados na criação de cada ficheiro de teste. Neste ponto, o arredondamento dos valores não foi feito uma vez que os testes iniciais demonstraram não produzir efeito. Cada ficheiro de teste pretendia simular uma situação de normal utilização da rede, contendo momentos de saturação e congestionamento, criados pela injeção de tráfego ou de perda de sinal de rede simulando a movimentação de um cliente.

4.1.1 Formatação dos Ficheiros de Teste

Veamos um exemplo de um ficheiro formatado de forma a dar entrada nos algoritmos de *machine learning*, apresentando os valores sem arredondamento, com um tamanho de janela igual a dois, uma distância entre valores igual a um e um atraso de dois segundos.

Neste caso é também fornecido ao algoritmo o valor real do *bitrate* no instante que se pretende prever, ou seja, este é um caso de treino de algoritmo recorrendo a dados armazenados das experiências descritas anteriormente.

Seria como saída do algoritmo de *machine learning* a previsão, neste caso do *bitrate* com dois segundos de antecedência.

Tabela 4.1: Diferença de parâmetros entre ficheiros.

	Tamanho da Janela	Distância entre valores	Atraso da Janela
Teste 1	2	1	1
Teste 2	2	1	3
Teste 3	2	2	1
Teste 4	2	2	3
Teste 5	2	3	1
Teste 6	2	3	3
Teste 7	3	1	1
Teste 8	3	1	3
Teste 9	3	2	1
Teste 10	3	2	3
Teste 11	3	3	1
Teste 12	3	3	3
Teste 13	4	1	1
Teste 14	4	1	3
Teste 15	4	2	1
Teste 16	4	2	3
Teste 17	4	3	1
Teste 18	4	3	3

```

@attribute Perdas-4 numeric
@attribute Perdas-3 numeric
@attribute Perdas-2 numeric
@attribute Atraso-4 numeric
@attribute Atraso-3 numeric
@attribute Atraso-2 numeric
@attribute Bitrate-4 numeric
@attribute Bitrate-3 numeric
@attribute Bitrate-2 numeric
@attribute Bitrate numeric

@data
4, 8, 0, 22.158, 22.158, 22.158, 781, 477, 658, 421
8, 0, 4, 22.158, 22.158, 22.158, 477, 658, 784, 1058
0, 4, 4, 22.158, 22.158, 3.983, 658, 784, 910, 951
4, 4, 2, 22.158, 3.983, 3.983, 784, 910, 753, 972

...

```

Para cada conjunto de dados de teste criados foram ainda criados dois ficheiros distintos, um que apresenta os dados em valor absoluto (como podemos observar na caixa de texto anterior) e um segundo que apresenta os mesmos dados mas com valor em diferença (como podemos observar na próxima caixa de texto) ou seja, em vez de apresentar os valores medidos, apresenta para cada parâmetro a diferença entre o valor actual e o anterior.

Podemos observar também na caixa abaixo, que aqui o parâmetro a prever está representado com um "?". Quando os dados são apresentados assim ao algoritmo não podem ser utilizados para treinar os mesmos e perde-se a capacidade de avaliar o resultado da previsão.

Mas será desta forma que na aplicação em tempo real os dados serão fornecidos ao algoritmo.

```

...

@data
0, 4, -8, 0, 0, 0, 0, -304, 181, ?
4, -8, 4, 0, 0, 0, -304, 181, 126, ?
-8, 4, 0, 0, 0, -18.175, 181, 126, 126, ?
4, 0, -2, 0, -18.175, 0, 126, 126, -157, ?

...

```

Tendo estes ficheiros os cabeçalhos necessários à compatibilidade das entradas aceites pelos algoritmos de *machine learning* pode-se então proceder aos testes dos algoritmos em si.

4.2 Avaliação de Algoritmos de Machine Learning

A aplicação Weka ? permite treinar e executar testes a algoritmos. Para a execução de um teste a um algoritmo com uma determinada definição de parâmetros são necessários os seguintes procedimentos:

1. Abrir ficheiro de dados a serem testados;
2. Escolha do algoritmo;
3. Definição dos parâmetros de operação do algoritmo;
4. Execução da aprendizagem e teste do algoritmo (*cross-validation*);
5. Guardar os resultados para posterior avaliação.

Tendo em mente que este é um processo moroso aquando do teste exaustivo de vários algoritmos, foi desenvolvida uma aplicação que realiza todas estas operações de forma dinâmica não necessitando a intervenção humana a cada passo.

A aplicação é capaz de ler todos os ficheiros contendo os dados a serem testados que foram obtidos dos testes realizados numa primeira fase e alimentar todos os algoritmos escolhidos afim de exportar os resultados para avaliação. Para tal, cada algoritmo é executado várias vezes alterando os seus parâmetros intrínsecos. No final de cada teste os resultados são exportados para um ficheiro de texto compatível com programas de folhas de cálculo possibilitando a criação gráficos, facilitando a visualização dos resultados.

4.2.1 Máquinas de Vectores de Suporte (SVM)

Na figura 4.1 podemos observar os valores de erro obtidos do teste realizado ao algoritmo SVM utilizando um dos ficheiros de dados resultantes dos procedimentos descritos em 4.1.1.

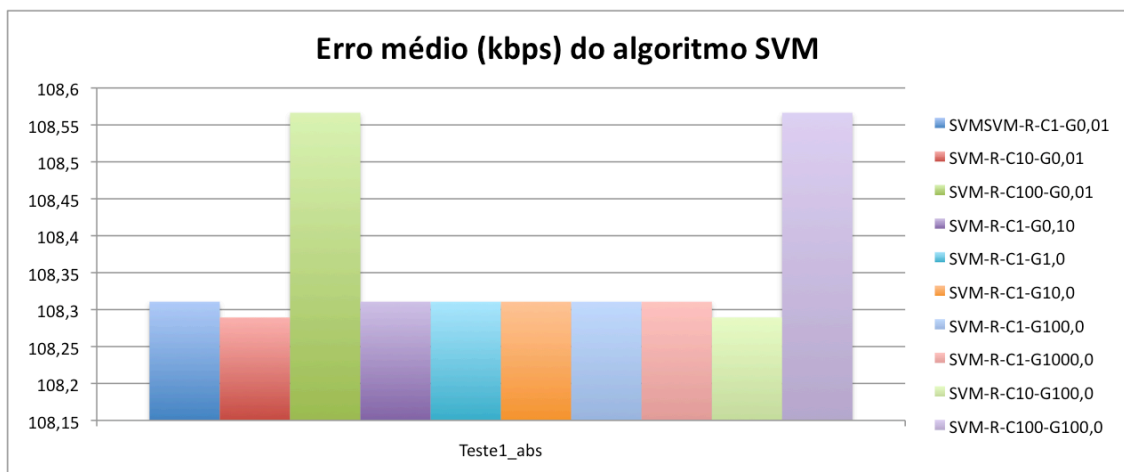


Figura 4.1: Resultados do teste ao algoritmo SVM

Cada uma das barras contidas no gráfico corresponde à utilização de um conjunto de opções fornecidas por este algoritmo nomeadamente:

- **Valor Gamma da kernel RBF** - {0.01; 0.1; 1.0; 10.0; 100.0; 1000.0}
- **Valor da constante de Complexidade (C)** - {1; 10; 100}

4.2.2 Redes Neurais Artificiais (RNA)

Na figura 4.2 podemos observar os valores de erro obtidos do teste realizado ao algoritmo RNA recorrendo ao ficheiro de teste número um.

Podemos observar duas barras com valores semelhantes mas cada uma representa uma configuração diferente, nomeadamente:

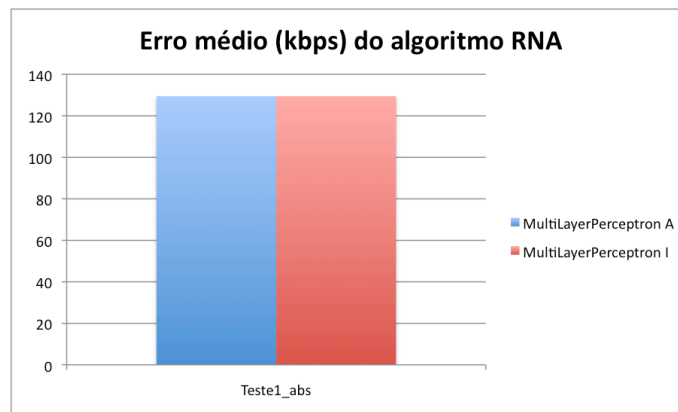


Figura 4.2: Resultados do teste ao algoritmo RNA

- **MultiLayer Perceptron A** - Corresponde a um número de nós da camada escondida (*Hidden Layer*) de metade dos atributos utilizados.
- **MultiLayer Perceptron I** - Corresponde a um número de nós da camada escondida igual ao numero de atributos utilizados.

Observando os resultados obtidos conclui-se que a alteração do número de nós da camada escondida não altera a precisão da previsão.

4.2.3 Regressão Linear

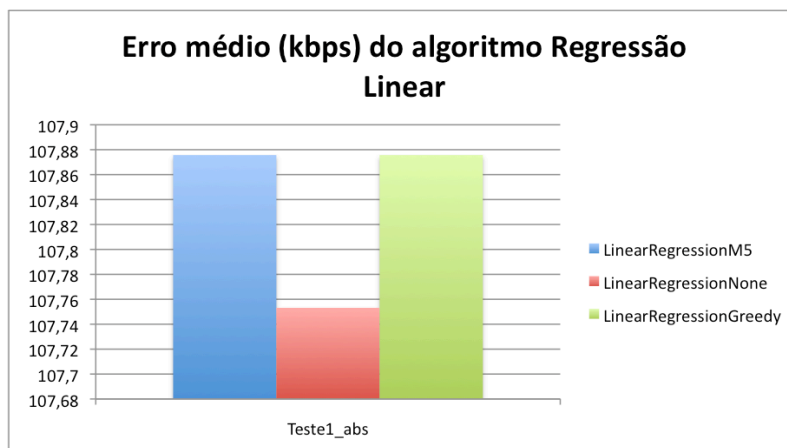


Figura 4.3: Resultados do teste ao algoritmo Regressão Linear

A variação de parâmetros efectuada a este algoritmo consiste no método de geração da fórmula e disponibilizado pela aplicação Weka [?], organizados do método menos selectivo para o mais selectivo são:

- **None** - Método sem critério de escolha de dados.

- **M5** - Este método fornece o melhor equilíbrio entre precisão da fórmula e o tempo que a mesma vai requerer para a execução.
- **Greedy** - Este método privilegia a criação de uma fórmula de elevada precisão, mas com o defeito de seleccionar dados "locais" e não globais para a geração da mesma o que se pode traduzir numa menor precisão final.

Na figura 4.3 observamos os resultados da execução deste algoritmo utilizando as opções supra referidas.

Surpreendentemente, o método de escolha mais simples (*None*) e de mais rápida execução demonstrou ter resultados melhores, se bem que ligeiramente. Os métodos M5 e *Greedy* obtiveram resultados semelhantes em todos os testes.

4.2.4 Processos Gaussianos

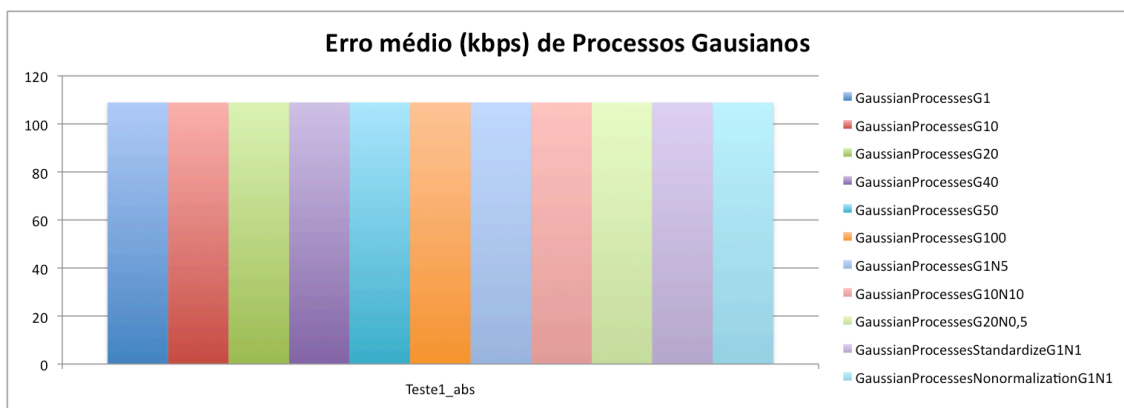


Figura 4.4: Resultados do teste ao algoritmo Processos Gaussianos

A variação efectuada a este algoritmo consiste no método de geração da fórmula, organizados do método menos selectivo para o mais selectivo:

- **G** - Representa o valor do Gamma da *Kernel*.
- **N** - Representa o ruído adicionado à diagonal da Matriz de Covariância, quando o mesmo não é apresentado o seu valor é um.

Na figura 4.4 podemos observar que a variação das opções deste algoritmo não surtiu efeito na melhoria de resultados. Apesar de tudo, as opções mais complexas, ou seja as últimas tenham necessitado de bastante mais tempo para a sua execução.

Variação de Parâmetros: G representa o valor do Gamma da *Kernel*. N representa o ruído adicionado à diagonal da Matriz de Covariância.

4.2.5 M5P

Na figura 4.5 apresenta-se o resultado do erro obtido após testar o algoritmo com o ficheiro de teste número um, tendo para isso sido realizado duas vezes afim de testar dois modos oferecidos por este algoritmo.

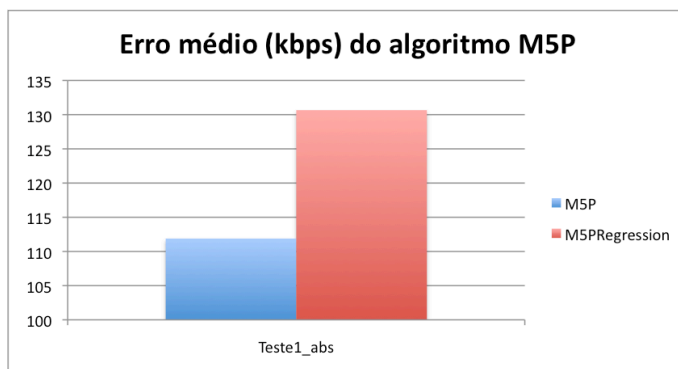


Figura 4.5: Resultados do teste ao algoritmo Processos Gaussianos

Neste algoritmo foi efectuada a comparação entre a utilização da árvore/regra modelo e a árvore/regra de regressão.

4.2.6 Locally Weighted Learning (LWL)

A variação de parâmetros efectuada neste algoritmo consiste na escolha da função de determinação dos pesos dos atributos, a escolha destas funções é possível mais uma vez pela utilização do Weka ?. Sendo utilizadas as funções **Lineares, Epnechnikov, Tricube, Inverse, Gaussian e Constant**.

Podemos observar as variações produzidas pela escolha de diferentes funções no erro médio após execução do teste número (figura 4.7).

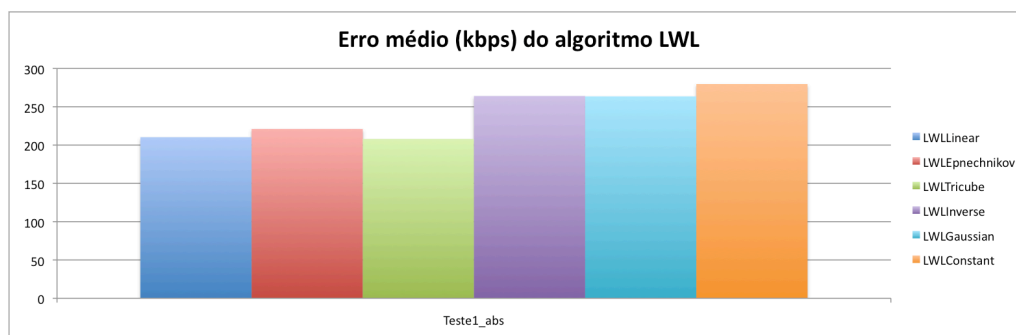


Figura 4.6: Resultados do teste ao algoritmo Processos Gaussianos

4.3 Análise

Como resultado deste processo foi obtida uma primeira avaliação sobre os algoritmos de *machine learning* e formas de apresentação dos valores aos mesmos.

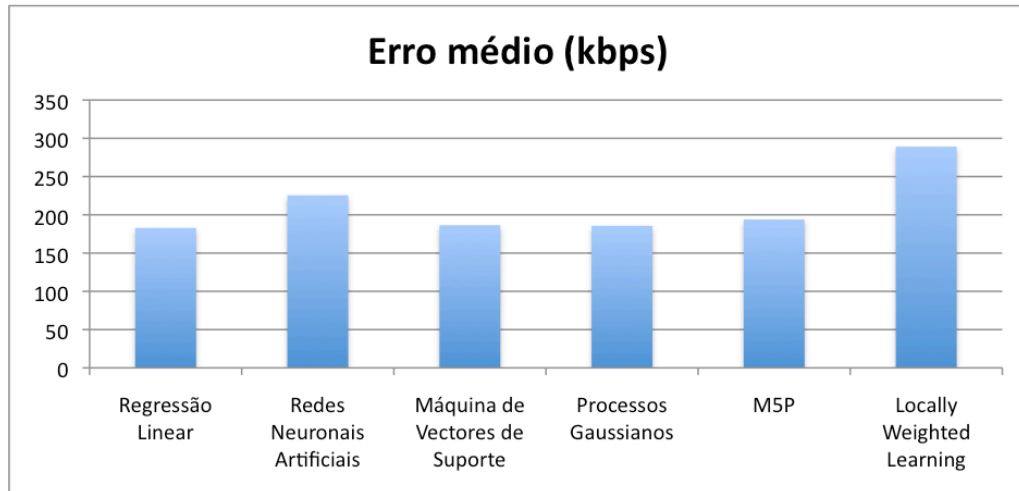


Figura 4.7: Comparação dos resultados médios dos algoritmos.

Consegue-se assim eliminar alguns dos algoritmos, ficando apenas com aqueles que obtiveram os melhores resultados de previsão para utilização na aplicação que opera em tempo real, sendo estes o **SVM** e a **Regressão Linear** tendo os **Processos Gaussianos** ficado de fora devido ao tempo excessivo requerido para o treino. E sendo que Redes Neurais e LWL obtiveram os piores resultados de previsão.

Consideram-se satisfatórios os resultados obtidos nestes testes, uma vez que um erro de cem *kbit* numa transmissão a dois mil *kbit* representa um erro de cinco por cento.

Capítulo 5

Previsão em Tempo-Real

Tendo a realização do estudo dos algoritmos e da forma de representação dos dados de entrada aos mesmos atingido os objectivos desejados e possíveis para testes realizados com base em dados históricos foi necessário iniciar o desenvolvimento de uma aplicação que permitisse a execução das mesmas funções que as aplicações desenvolvidas anteriormente, ou seja, a captura de dados, o treino e a obtenção da previsão, mas em tempo real.

5.1 Aquisição de Parâmetros em Tempo Real

Desta forma a nova aplicação terá de ser capaz de efectuar o treino do algoritmo numa primeira instância, baseando-se em dados históricos, mas findo esse procedimento, o algoritmo seria alimentado de uma forma contínua, obtendo assim as previsões em tempo real.

Para tal foi necessário o estudo de como obter parâmetros de rede em tempo real que fizesse a passagem desses valores à aplicação a executar o algoritmo de *machine learning*.

Em etapas anteriores o projecto estava a ser utilizado o Wireshark ? para a visualização de tais parâmetros em tempo real, mas as tentativas de utilizar esses dados à medida que são apresentados mostrou-se infrutífera.

Como a aplicação que se utilizou para a criação dos ficheiros contendo dados de rede, o TRPR ?, que faz os cálculos de forma a determinar esses parâmetros através dos dados de saída do TCPDUMP ? apenas permitia obter um parâmetro de cada vez, ou seja, para cada parâmetro de rede que se pretendesse medir seria necessário um processo a correr essas duas aplicações, tornou-se evidente que tal não seria uma opção, nomeadamente pela capacidade de processamento que iria exigir do dispositivo terminal.

Após a exploração do programa TCPDUMP ? e dos dados de saída que este apresenta, foi encontrada uma solução que passava pela realização dos cálculos necessários à obtenção dos parâmetros no próprio programa que corria o tratamento de dados e o algoritmo de *machine learning*.

Dado que o TCPDUMP ? não é mais que uma ferramenta de *sniffing* de rede, a passagem do cabeçalho dos pacotes obtidos na rede é suficiente para o cálculo de alguns dos parâmetros nomeadamente:

1. **Bitrate**, através da acumulação do tamanho dos pacotes e medido segundo a segundo, obtendo-se assim um valor em *kbps* é ainda calculada a percentagem de tráfego correspondente ao *stream* por comparação do restante tráfego na rede.
2. **Pacotes por segundo**, pela simples contagem do número de pacotes.
3. **Jitter**, uma vez que é apresentado o *time-stamp* à chegada dos pacotes, consegue-se obter a variação do atraso entre cada dois pacotes consecutivos pela simples subtracção dos tempos de chegada.

Embora estes três valores sejam de máxima importância, não são suficientes e ficariam aquém do proposto.

Para a monitorização do **atraso**, visto que na realidade é pouco prático garantir o sincronismo de relógios em computadores com precisão suficiente, nem tal seria prático, optou-se pela medição do **Round-Trip-Time** (RTT), recorrendo ao comando *Ping*, que consiste no envio de pacotes para o destinatário, que por sua vez responde prontamente, enviando um pacote de volta fazendo assim a medição do RTT.

Recorrendo à utilização do VLC ? em modo de linha de comandos, consegue-se obter em tempo real as falhas detectadas pelo mesmo, ou seja, **erros** de decodificação do *stream* enquanto ao mesmo tempo se está a visualizar o vídeo. Com isto conseguimos observar todos os erros que estão a ocorrer na visualização do vídeo, nomeadamente provenientes de perdas ou erros dos pacotes.

Este último parâmetro provou ser o mais importante para os algoritmos de *machine learning* uma vez que é o parâmetro que tem uma ligação directa e aparentemente linear com a qualidade do vídeo a ser visualizado.

Todos estes dados são tratados por uma função, encarregue de, a todos os segundos (unidade mínima comum de precisão de entre os vários parâmetros), formatar os dados recolhidos nesse instante anexando-os a valores em histórico afim de os passar ao algoritmo de *machine learning* em utilização.

5.2 Mecanismo de Previsão

Com os dados recolhidos a cada segundo, os mesmos são utilizados para obter uma previsão recorrendo ao do algoritmo de *machine learning*.

Este processo é simples, uma vez que os cálculos da previsão são efectuados pelo algoritmo em si. Apenas é necessário a passagem dos valores obtidos em tempo real ao algoritmo e a consequente obtenção da resposta que é dada consoante o treino do mesmo.

5.3 Mecanismo de Adaptação

Após os valores em obtidos em tempo real serem passados ao algoritmo de *machine learning*, este produz uma resposta, ou seja, o valor que pretendemos prever, sendo que o parâmetro que provou ser mais eficaz na relação com a qualidade do vídeo recebido foi o de erros apresentados pelo descodificador.

Na figura abaixo 5.1 podemos observar uma imagem retirada de um vídeo a ser recebido por *streaming* na qual ocorreram erros devido à perda de pacotes na rede. É precisamente este o tipo de situações a evitar.

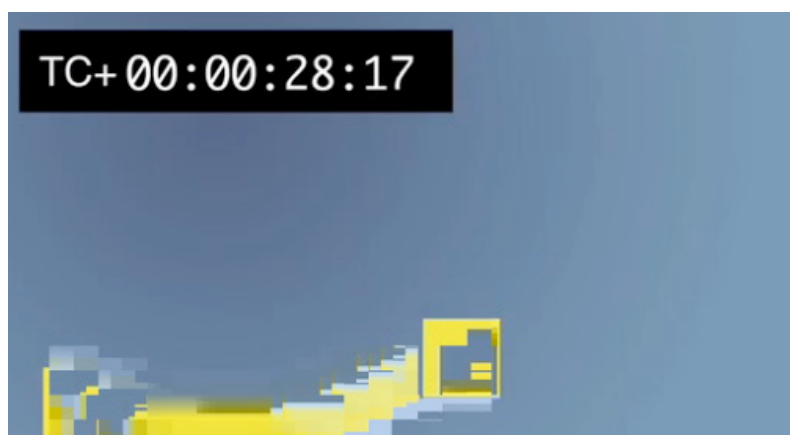


Figura 5.1: Visualização dos erros de descodificação.

A tomada de decisão sobre a adaptação a efectuar não é objectivo desta tese. Até porque o assunto é tema de muitas teses e projectos de dimensão considerável, o que inviabilizaria a tentativa de o efectuar aqui, embora sejam apresentados documentos durante a revisão bibliográfica e o mesmo tenha sido objecto de estudo numa primeira fase.

Tendo esta explicação em mente, foi implementado o mecanismo de tomada de decisão, embora não considerando muitas das recomendações encontradas na bibliografia estudada, mas com o intuito de se observar uma diminuição na capacidade de rede necessária à transmissão do vídeo, tentando assim observar uma melhoria na qualidade observada pelo utilizador, nomeadamente pela diminuição das dimensões do vídeo que fazem requerer um menor *bitrate* na transmissão do mesmo, "aliviando" situações de congestão.

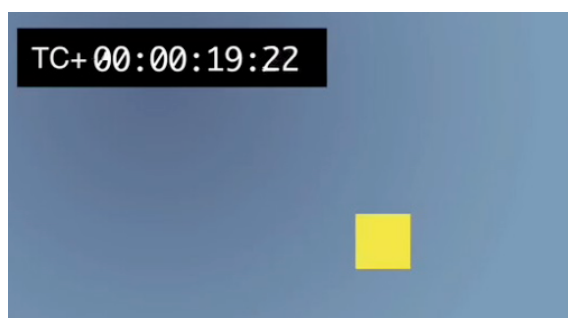


Figura 5.2: Imagem retirada de um vídeo recebido por *streaming*.

Podemos observar na figura 5.2, apresentada aqui com dimensões propositalmente reduzidas, por comparação com a figura 5.1, afim de mostrar que uma adaptação espacial que passe pela redução do tamanho da imagem é geralmente melhor suportada pelo utilizador que uma imagem maior contendo erros.

5.4 *Stream* de Vídeo

Após a tomada de decisão explicada na secção anterior, para ocorrer a adaptação do conteúdo de vídeo a ser recebido, o programa envia uma mensagem ao servidor que está a emitir o *stream*, que por sua vez actua sobre a linha de comandos do VLC ?, alterando assim os parâmetros da transcodificação do vídeo.

Esta adaptação provoca uma quebra no *stream*, uma vez que o servidor necessita de enviar os novos parâmetros de vídeo ao cliente afim do mesmo o descodificar correctamente, embora essa quebra seja de curta duração (inferior a meio segundo, dependendo das condições da rede).

Existem já ferramentas desenvolvidas por grupos de trabalho na área mas as mesmas não se encontram disponíveis para utilização pública, como no exemplo dado na subsecção 2.1.2.3.

5.5 Sumário de Funcionalidades e Resultados

De acordo com o descrito nas secções anteriores fica assim completa mas aberta a melhoramentos uma aplicação capaz de iniciar a recepção de um *stream* de vídeo. Com capacidade de ler e calcular em tempo real parâmetros de rede e erros de descodificador de vídeo, tanto para armazenamento como para utilização em tempo real. É ainda capaz de utilizar esses parâmetros no treino e na utilização de algoritmos de *machine learning* afim de obter previsão sobre factores que influenciam a qualidade com que o vídeo está a ser recepcionado.

Os resultados obtidos na tentativa de previsão não foram os ideais uma vez que existem previsões erradas e mesmo as correctas não são obtidas com a antecedência necessária à tomada de uma decisão de adaptação, mas mantém-se a convicção de que com a introdução de novos parâmetros a capacidade preditiva pode realmente ocorrer.

Na figura 5.3 vemos o resultado de uma experiência recorrendo ao algoritmo de regressão linear em que podemos observar ter uma prestação muito pior que os resultados apresentados pelo algoritmo SVM.

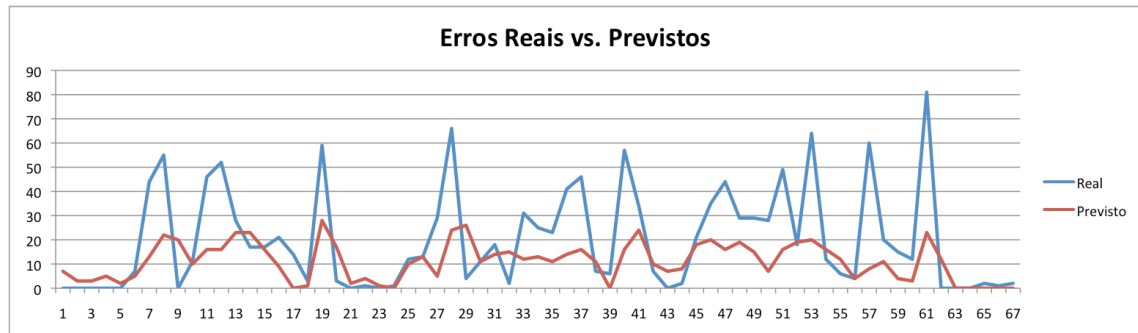


Figura 5.3: Comparação de erros reais de descodificação vs. erros previstos pelo algoritmo Regressão Linear.

Na figura 5.4 podemos observar que a previsão, apesar de ainda não ser feita com a desejada antecedência, ocorre.

Se considerarmos o instante "32", podemos observar um pico nos erros detectados e previstos. Esta previsão foi feita sem conhecimento dos dados nesse instante pelo que podemos afirmar existir alguma capacidade de previsão, mas instantânea, ou seja, prevê-se no momento o que vai ocorrer nesse mesmo instante. A situação ideal seria o gráfico da previsão a ocorrer com dois segundos de antecedência.

A diferença de amplitudes observada deve-se ao facto de nos dados de treino do algoritmo terem ocorrido menos erros que nesta situação pelo que algum trabalho ainda terá de ser feito afim de melhorar a qualidade dos resultados.

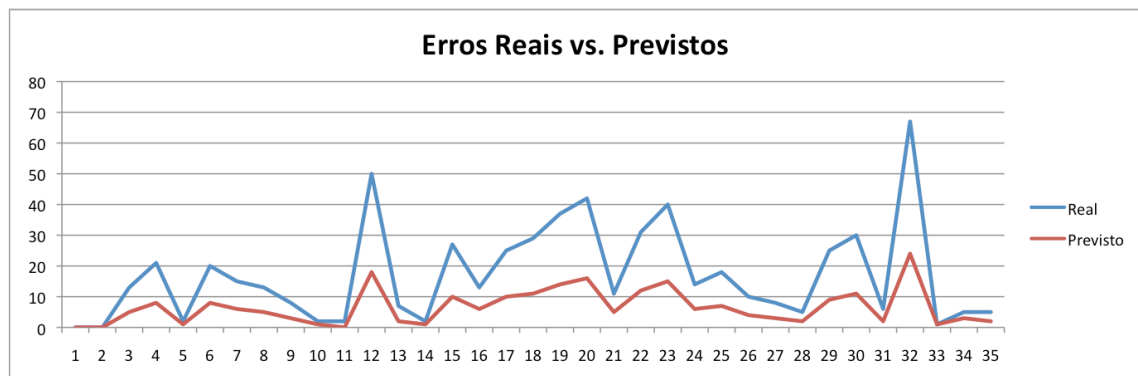


Figura 5.4: Comparação de erros reais de descodificação vs. erros previstos pelo algoritmo SVM.

Capítulo 6

Conclusões e Trabalho Futuro

6.1 Satisfação dos Objectivos

A realização desta tese, deixa como grande mais-valia as aplicações desenvolvidas e que consumiram uma grande parte do tempo disponível para a realização da mesma.

Nomeadamente a aplicação de teste de algoritmos de *machine learning* que revelou ser de extrema importância ao eliminar do leque de algoritmos disponíveis alguns dos mesmos. Com esta aplicação a comparação de performance entre algoritmos torna-se uma tarefa rápida e simples fornecendo todos os dados necessários à avaliação dos mesmos.

A aplicação de funcionamento em tempo real ficou também concluída oferecendo as seguintes três funcionalidades fundamentais:

1. Treino do algoritmo de *machine learning*.
2. Levantamento e cálculo de parâmetros, tais como **bitrate** e **percentagem de bitrate** pertencente ao *stream* vídeo, a **latência** e o **jitter** da ligação, o **número de pacotes por segundo** e a **percentagem dos mesmos** correspondentes ao *stream* e os **erros por segundo do decodificador** de vídeo.
3. Obtenção da previsão dada pelo algoritmo de *machine learning*.

Contudo, apesar da realização de inúmeros testes, adicionados parâmetros e testados novamente sendo sempre também testadas as diferentes formas de apresentação dos mesmos dados de formas diferentes, a capacidade de prever situações em que seja necessária a adaptação do conteúdo vídeo não se revelou eficiente.

De facto, embora tendo existido situações como as apresentadas na figura 6.1 em que se pode observar que nos momentos em que os erros de decodificação aumentavam, também o algoritmo de *machine learning* SVM previu tais situações embora com uma amplitude desadequada.

Tal acontece uma vez que o treino do algoritmo é realizado com um ficheiro de dados de histórico e durante este teste as condições da rede a serem obtidas em tempo real estavam a ser bem reconhecidas pelo algoritmo em utilização conseguindo desta forma uma detecção da deterioração ou melhoramento da qualidade do vídeo a ser apresentado.

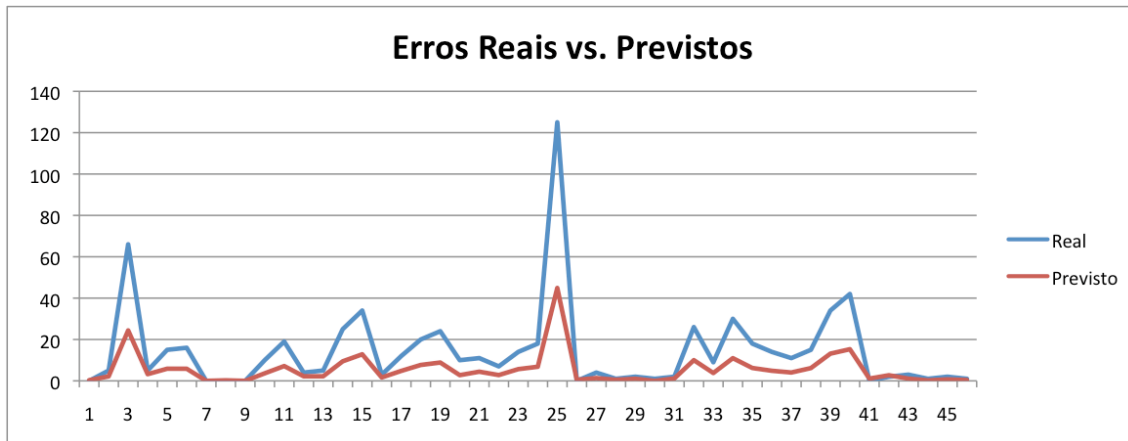


Figura 6.1: Comparação de erros reais de descodificação vs. erros previstos pelo algoritmo SVM.

Apesar do gráfico aqui apresentado ser encorajador não se pode considerar que a aplicação desenvolvida no decorrer desta tese possua capacidade de previsão. Em muitos testes os parâmetros utilizados não são suficientes para considerar de forma definitiva que uma decisão de adaptação deve ser tomada. Querendo com isto dizer que em determinadas situações, os mesmos valores de entrada (ou seja, medidos) conduzem a situações distintas, tanto aumentando o erro como a diminuição do mesmo.

Pode-se desta forma afirmar que o algoritmo de *machine learning* aqui utilizado, o SVM, produz boas respostas mas é necessário considerar a introdução de mais parâmetros afim de resolver as ambiguidades ainda existentes.

6.2 Trabalho Futuro

Tendo em mente o tempo de trabalho que foi necessário e utilizado no desenvolvimento e teste das aplicações nomeadamente na obtenção de parâmetros mínimos para a realização de testes iniciais, fica agora o caminho aberto para a aquisição de mais parâmetros que possam fornecer indicações sobre o comportamento da rede num futuro próximo.

Para tal será necessário continuar o trabalho recorrendo à ferramenta já desenvolvida sendo apenas necessário modificar ou criar funções de modo a que se possam adquirir novos parâmetros, relacionados com a rede ou mesmo o contexto, uma vez que toda a estrutura da aplicação não necessita de grandes modificações para tal.

Alguns parâmetros de interesse que podem vir a ser implementados na aplicação e que poderão fornecer mais informações sobre como a rede se irá comportar são a localização do utilizador e a potência de sinal obtida da placa sem fios.

A localização do utilizador poderá ser muito útil caso se consigam mapear zonas e associar as mesmas a medições da qualidade da rede efectuadas. Tendo esta funcionalidade seríamos capazes de detectar com alguma antecedência a deslocação do utilizador para uma zona conhecida por oferecer baixa qualidade de serviço.

Quanto à utilização da potência de sinal, combinada com a posição do utilizador pode ser utilizada para criar o referido mapeamento.