

MARCUS VINÍCIUS MAZEGA FIGUEREDO

**USO DE TRANSFORMADAS
WAVELET E REDES NEURAIS
ARTIFICIAIS NA COMPRESSÃO DO
ELETROCARDIOGRAMA**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

Curitiba
2008

MARCUS VINÍCIUS MAZEGA FIGUEREDO

**USO DE TRANSFORMADAS
WAVELET E REDES NEURAIS
ARTIFICIAIS NA COMPRESSÃO DO
ELETROCARDIOGRAMA**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

Área de Concentração: Ciência da Computação

Orientador: Júlio Cesar Nievola

Curitiba
2008

Figueredo, Marcus Vinícius Mazega
USO DE TRANSFORMADAS WAVELET E REDES NEURAIIS ARTI-
FICIAIS NA COMPRESSÃO DO ELETROCARDIOGRAMA. Curitiba,
2008.

Dissertação - Pontifícia Universidade Católica do Paraná. Programa de
Pós-Graduação em Informática.

1. Compressão de Eletrocardiograma 2. Transformadas Wavelet 3. Redes
Neurais Artificiais I. Pontifícia Universidade Católica do Paraná. Centro de
Ciências Exatas e Tecnologia. Programa de Pós-Graduação em Informática
II - t

Ao Criador, por tudo.
À minha família, pela minha herança.
Aos meus mestres, por seus ensinamentos.

Agradecimentos

Em primeiro lugar, gostaria de agradecer ao orientador Prof. Dr. Júlio Cesar Nievola, pelas preciosas lições de Inteligência Artificial e pelos sábios conselhos durante a execução deste trabalho. Também gostaria de lembrar do Prof. Dr. João da Silva Dias, que orientou meus primeiros trabalhos envolvendo telemedicina e informática em saúde. Em função de todo o apoio fornecido durante minha vida acadêmica, não posso esquecer os nomes dos professores Robert Carlisle Burnett, James Baraniuk, Henri Eberspächer, Lourival Lippmann Jr e Maria Te Vaarwerk. Aproveito para agradecer também aos meus sócios e amigos Alfredo Beckert Neto e Sérgio Renato Rogal Jr, que estiveram sempre presentes em todos os meus projetos. É muito importante agradecer ao Prof. Dr. Paulo Brofman (PUCPR) e ao Prof. Dr. Frank Yanowitz (University of Utah), que participaram da etapa de validação clínica deste trabalho. Sem eles, os resultados do meu trabalho não teriam a mesma significância. Vale ressaltar que este trabalho não seria possível sem o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), que me concedeu uma bolsa de estudos. Ainda quero agradecer aos meus pais Álvaro e Maria José, meus irmãos Thiago e Fernanda e a Liliane Tod, que me “aguentaram” durante toda a minha pesquisa. Finalmente, gostaria de agradecer a todos que direta ou indiretamente colaboraram na execução deste trabalho e, que por acaso, não estejam citados nesta página.

Sumário

Agradecimentos	ii
Lista de Figuras	vi
Lista de Tabelas	viii
Lista de Abreviações	ix
Resumo	x
Abstract	xi
Capítulo 1	
Introdução	1
1.1 Motivações e Desafios	2
1.2 Objetivos	3
1.3 Contribuições Esperadas	4
1.4 Organização do Documento	4
Capítulo 2	
Eletrocardiografia	5
2.1 Eletrofisiologia do Coração	5
2.2 Registro do Eletrocardiograma	6
2.3 Derivações	8
2.4 Eletrocardiograma Normal	8
2.5 Arritmias Cardíacas	12
2.6 Resumo do Capítulo	12
Capítulo 3	
Compressão de ECG: Revisão de Literatura	13
3.1 Abordagens baseadas em Transformadas	14
3.2 Abordagens Bidimensionais	15

3.3	Abordagens baseadas em Similaridades	16
3.4	Outras Abordagens de Compressão	17
3.5	Resumo do Capítulo	18
Capítulo 4		
Wavelets		19
4.1	Transformadas de Fourier	19
4.2	Transformada Wavelet Contínua	23
4.3	Transformada Wavelet Discreta	27
4.4	Resumo do Capítulo	30
Capítulo 5		
Redes Neurais Artificiais		31
5.1	Fundamentos	31
5.2	Algoritmo de Retro-Propagação	32
5.3	Redes Neurais na Compressão de Dados	34
5.3.1	Abordagem utilizando o Algoritmo de Retro-Propagação	34
5.3.2	Abordagem utilizando Aprendizagem Hebbiana	35
5.3.3	Abordagem utilizando Mapas Auto-Organizáveis	35
5.3.4	Abordagem utilizando Redes Preditivas	36
5.4	Resumo do Capítulo	36
Capítulo 6		
Metodologia		37
6.1	Visão Geral	38
6.2	Escolha da Base de Dados	38
6.3	Segmentação	41
6.4	Divisão e Classificação dos Blocos	42
6.5	Normalização dos Coeficientes Wavelet	42
6.6	Treinamento das Redes Neurais	43
6.7	Métricas Utilizadas	45
6.8	Validação da Metodologia	46
6.8.1	Bases de Dados	46
6.8.2	Validação Automática	46
6.8.3	Validação com Especialistas	48
6.9	Resumo do Capítulo	49

Capítulo 7

Resultados e Discussão	50
7.1 Exames de ECG	50
7.2 Medidas de Desempenho	52
7.3 Validação Automática	52
7.4 Validação com Especialistas	55
7.5 Análise da Metodologia	59
7.5.1 Segmentação	59
7.5.2 Classificação	59
7.5.3 Transformada Wavelet	59
7.5.4 Treinamento das Redes Neurais	60
7.6 Resumo do Capítulo	60

Capítulo 8

Conclusão	61
Referências	64

Apêndice A

Código-Fonte	69
A.1 Segmentação	69
A.2 Classificação	71
A.3 Transformada Wavelet	75
A.4 Formatação para o Padrão JavaNNS	78
A.5 Transformada Wavelet Inversa	83
A.6 Reconstrução	85

Lista de Figuras

Figura 2.1	Eletrocardiógrafo construído com amplificadores de instrumentação (UNICAMP, 2007)	7
Figura 2.2	Exemplo de registro do eletrocardiograma (ARTEAGA; GUTIERREZ, 2006)	8
Figura 2.3	As 12 derivações clássicas do ECG: a) I; b) II; c) III; d) aVR; e) aVL; f) aVF; g) V1; h) V2; i) V3; j) V4; k) V5; e, l) V6 (ASCHENBRENNER, 2000)	9
Figura 2.4	Diagrama esquemático exibindo os principais componentes de um ECG normal	10
Figura 2.5	Alguns exemplos de arritmias cardíacas retirada da base de dados utilizada neste trabalho (PHYSIONET, 2007)	11
Figura 4.1	Exemplo de sinal periódico, com componentes de 15, 20, 25, 40 e 70 Hz	20
Figura 4.2	Sinal periódico representado no domínio da frequência	21
Figura 4.3	Exemplo de sinal não-periódico, com componentes de 15, 20, 25, 40 e 70 Hz	22
Figura 4.4	Sinal não-periódico representado no domínio da frequência	23
Figura 4.5	Representação tridimensional relacionando tempo, frequência e amplitude do sinal não-periódico	24
Figura 4.6	Transformada por janelas de Fourier utilizando janelas estreitas	25
Figura 4.7	Transformada por janelas de Fourier utilizando janelas largas	25
Figura 4.8	Diversas famílias de Wavelets: (a) Haar; (b) Symmlet; (c) Daubechies; e, (d) Coiflet	26
Figura 4.9	Algumas fases da computação da TWC para um sinal estacionário	27
Figura 4.10	Transformada Wavelet contínua de um sinal não-estacionário	28

Figura 4.11	Aplicação da TWD em um sinal unidimensional composto por uma senóide de 20Hz e um ruído de alta frequência	28
Figura 4.12	Aplicação da TWD gerando coeficientes de detalhe e aproximação	29
Figura 5.1	Estrutura de uma rede neural utilizada para compressão de dados, treinada com o algoritmo de retro-propagação	35
Figura 5.2	Uso de mapas auto-organizáveis na compressão de dados	36
Figura 6.1	Visão em blocos do projeto	39
Figura 6.2	Fragmento de um arquivo “.DAT” da base de dados.	40
Figura 6.3	Fragmento de um arquivo “.ATR” da base de dados.	40
Figura 6.4	Exemplo de segmentação	41
Figura 6.5	Divisão e classificação do segmento em dois blocos	43
Figura 6.6	Coeficientes Wavelet de um bloco de ECG	44
Figura 6.7	Arquitetura das redes utilizadas	44
Figura 6.8	Escala de qualidade do método automático de validação	48
Figura 7.1	Exame 08730-01, retirado da MIT-BIH ECG Compression Test Database	51
Figura 7.2	Exame 08730-01, após o processo de compressão e expansão	51
Figura 7.3	Exame 08730-02, retirado da MIT-BIH ECG Compression Test Database	51
Figura 7.4	Exame 08730-02, após o processo de compressão e expansão	51
Figura 7.5	Exame 08730-03, retirado da MIT-BIH ECG Compression Test Database	51
Figura 7.6	Exame 08730-03, após o processo de compressão e expansão	51
Figura 7.7	Exame 08730-04, retirado da MIT-BIH ECG Compression Test Database	52
Figura 7.8	Exame 08730-04, após o processo de compressão e expansão	52
Figura 7.9	Tela de entrada do sistema Web de validação	57
Figura 7.10	Tela de análise do ECG do sistema Web de validação	58
Figura 7.11	Tela de saída do sistema Web de validação	58

Lista de Tabelas

Tabela 6.1	Números de neurônios utilizados nos treinamentos	45
Tabela 6.2	Pesos utilizados na validação automática do método proposto . . .	47
Tabela 7.1	Medidas de desempenho para a MIT-BIH Arrhythmia Database . .	53
Tabela 7.2	Medidas de desempenho para a MIT-BIH ECG Compression Test Database	53
Tabela 7.3	Análise em paralelo com outros métodos	54
Tabela 7.4	Resultados da validação automática para a MIT-BIH Arrhythmia Database	55
Tabela 7.5	Resultados da validação automática para a MIT-BIH ECG Com- pression Test Database	55
Tabela 7.6	Resultados da validação com especialistas, em números absolutos .	57
Tabela 7.7	Resultados da validação com especialistas, em números relativos . .	57

Lista de Abreviações

ECG	<i>eletrocardiograma</i>
MLP	<i>multilayer perceptron</i>
DCT	<i>discrete cosine transform</i>
SNR	<i>signal-to-noise ratio</i>
MMP	<i>multidimensional multiscaler parser</i>
TJF	<i>transformada por janelas de Fourier</i>
TWC	<i>transformada Wavelet contínua</i>
TWD	<i>transformada Wavelet discreta</i>
TX	<i>taxa de compressão - “compression rate”</i>
DPRM	<i>diferença percentual da raiz média - “percent root mean squared difference”</i>
EM	<i>erro máximo - “maximum error”</i>
DPE	<i>desvio padrão dos erros - “standard derivation of errors”</i>
DPRM	<i>diferença percentual da raiz média - “percent root mean squared difference”</i>

Resumo

A transmissão em tempo-real do eletrocardiograma (ECG) em situações de urgência pode aumentar consideravelmente as chances de sobrevivência do paciente. No entanto, um dos grandes problemas envolvendo essa aplicação da telemedicina em tempo-real envolve as limitações de banda. Exames de ECG podem gerar um grande número de dados, o que inviabiliza o uso de técnicas de telecardiologia em muitas aplicações do cotidiano. Esse problema instigou muitos autores a buscarem técnicas eficientes de compressão de ECG, que podem ser agrupadas em quatro grandes grupos: abordagens baseadas em transformadas, abordagens bidimensionais, abordagens baseadas em similaridades e outras abordagens. O presente trabalho suscitou a hipótese de que redes neurais artificiais podem ser utilizadas na compressão de ECG em conjunto com as transformadas Wavelet. Nessa abordagem, as transformadas foram responsáveis por ressaltar os padrões do ECG no domínio tempo-frequência. Esses padrões, por sua vez, foram comprimidos por uma rede neural MLP treinada com algoritmo de retro-propagação. Os registros da MIT-BIH Arrhythmia Database foram comprimidos em 28 vezes, enquanto a MIT-BIH ECG Compression Test Database foi comprimida em 20 vezes. O processo obteve bons resultados no quesito de qualidade, que podem ser observados nos baixos valores da diferença percentual da raiz média (0.037% e 0.028%), do desvio padrão dos erros (0.287 e 0.230) e do erro máximo (1.999 e 1.998). Um método automático de validação baseado em Wavelets também foi utilizado para qualificar o método proposto. Pode-se concluir a partir daí que o método proposto neste trabalho apresentou um grau de qualidade variando entre “bom” e “excelente”. Finalmente, especialistas internacionais avaliaram os exames comprimidos, demonstrando que a compressão do ECG não afetou o valor clínico do exame.

Palavras-chave: Compressão de ECG, compressão de dados, transformadas Wavelet, redes neurais artificiais

Abstract

The real-time transmission of the electrocardiogram (ECG) in urgent situations can improve the chances of the patient. However, one of the greatest problems involving this kind of telemedicine application is the leakage of network bandwidth. ECG exams may generate too much data, which makes difficult to apply telecardiology systems in real life. This problem motivated many authors to look for efficient techniques of ECG compression, such as: transform approaches, 2-D approaches, similarity approaches and generic approaches. The present work proposes a new hypothesis: neural networks may be applied together with Wavelet transforms to compress the ECG. In this approach, the Wavelet transform acts as a pre-processor element for a multilayer perceptron neural network, trained with the backpropagation algorithm. The original signal was divided in two parts: the “plain” blocks and the “complex” ones. The plain blocks were compressed with a 40:1 ratio while the “complex” blocks were compressed with a 5:1 ratio. The use of both compressors guaranteed a compression rate of 28:1, approximately. The process obtained good grades in the quality aspect: percent root mean squared difference (0.037% and 0.028%), maximum error (1.999 and 1.998) and standard derivation of errors (0.287 and 0.230). A wavelet-based quality measure was utilized to check the validity of the proposed technique, which returned that the results were inside the “good”-“excellent” range. Furthermore, the results were evaluated by a group of international specialists.

Keywords: ECG compression, data compression, wavelets, neural networks

Capítulo 1

Introdução

Os recentes avanços nas telecomunicações estimularam o desenvolvimento de sistemas de telemedicina, como a telecardiologia. A transmissão em tempo-real do ECG (*electrocardiograma*) em situações de urgência pode aumentar consideravelmente as chances de sobrevivência do paciente. Por exemplo, diversos estudos demonstraram que, realizado dentro de uma ambulância, o ECG de 12 derivações pode indicar antecipadamente a necessidade da aplicação da terapia trombolítica, evitando um ataque cardíaco (SEDGWICK et al., 1993).

No entanto, um dos grandes problemas envolvendo a transmissão de ECG em tempo-real envolve as limitações de banda. Exames de ECG podem gerar um grande número de dados, o que inviabiliza o uso de técnicas de telecardiologia em muitas aplicações do cotidiano. Esse problema instigou muitos autores (GARCÍA, 2006), (CHEN; ZHANG; SHI, 2006), (KU et al., 2006), (KIM; YOO; MOON, 2006), (SOBREIRA, 2006), (CHOU et al., 2006), (HENRIQUES et al., 2006), (CARVALHO; SILVA; FINAMORE, 2002), (FILHO et al., 2006), (BRITO et al., 2005) a buscarem técnicas eficientes de compressão de ECG. De uma forma geral, as técnicas de compressão de ECG podem ser divididas em quatro grandes grupos: abordagens baseadas em transformadas, abordagens bidimensionais, abordagens baseadas em similaridades e outras abordagens.

Nos últimos anos, os métodos baseados em transformadas Wavelet estão sendo amplamente investigados, pois vêm apresentando resultados promissores. Em geral, esses métodos envolvem os seguintes passos na compressão: segmentação do sinal de ECG; aplicação da transformada Wavelet; eliminação dos coeficientes Wavelet inferiores a um limite pré-estabelecido; aplicação de uma técnica consagrada de codificação, como codificação de Huffman ou codificação Golomb. A descompressão, por sua vez, envolve: decodificação dos coeficientes Wavelet e aplicação de uma transformada Wavelet inversa.

O presente trabalho suscita a hipótese de que redes neurais artificiais podem ser utilizadas na compressão de ECG em conjunto com as transformadas Wavelet, substituindo o segundo e terceiro passos, acima listados. Substituindo-se métodos lineares (corte de coeficientes e codificações) por métodos não-lineares (redes neurais), espera-se obter maiores taxas de compressão com menores taxas de erros.

No passado, o uso de redes neurais na compressão de ECG obteve boas taxas de compressão; contudo, as taxas de erro não eram aceitáveis. Isso aconteceu, principalmente, porque as redes foram utilizadas sem um pré-processamento adequado. Neste trabalho, pretende-se utilizar as transformadas Wavelet como elemento pré-processador dos dados. Através de uma transformação para o domínio tempo-frequência, as amostras de sinal serão convertidas em coeficientes Wavelet. Uma rede neural multicamada MLP (*multilayer perceptron*) será treinada, então, para comprimir esses coeficientes. Espera-se que essa abordagem seja mais imune ao ruído do que as tentativas anteriores em função da utilização de Wavelets.

1.1 Motivações e Desafios

As motivações que levaram a escolha do tema deste trabalho podem ser classificadas como sociais, financeiras e científicas. As motivações sociais e financeiras se concentram nas demandas atuais e futuras do setor da saúde pública. Nos últimos anos, a população brasileira vem sofrendo um processo natural de envelhecimento, de forma que a parcela de pessoas com mais de 65 anos cresce cada vez mais. Um número maior de idosos representa, na prática, mais pessoas que necessitam de tratamento clínico constante. É relevante ressaltar ainda o aumento da incidência de diversas doenças crônicas, tais como diabetes, hipertensão, anomalias cardíacas, entre outras. Esse enorme contingente de pacientes crônicos demanda um esforço muito maior do setor da saúde, tanto em recursos humanos quanto financeiros.

Diante da escassez de profissionais de saúde e da necessidade de não estourar o orçamento da saúde, muitos esforços vem sendo dispendidos no desenvolvimento da telemedicina, que pode ser definida como o uso da troca de informações médicas entre dois lugares distintos por meio de comunicação eletrônica a fim de melhorar o estado de saúde do paciente. Muitos serviços remotos podem ser considerados como aplicações de telemedicina, tais como monitoramento de sinais vitais, consultas médicas, diagnóstico de doenças, educação médica e integração de redes de saúde. A grande importância da telemedicina é que ela permite que um profissional de saúde atenda pacientes à distância, o que aumenta virtualmente a cobertura geográfica dos serviços de saúde. A comunicação

entre especialistas médicos, por sua vez, melhora a qualidade do atendimento e reduz erros de diagnóstico, otimizando o uso dos recursos do setor.

Uma das mais importantes aplicações de telemedicina é a telecardiologia, que envolve a transmissão do eletrocardiograma à distância. Uma das maiores restrições às aplicações de telecardiologia no cotidiano é o consumo excessivo dos recursos de telecomunicações e de armazenamento de dados. A criação de um método eficiente de compressão do ECG representa, então, um passo essencial para a popularização dessas aplicações.

A principal motivação científica desse trabalho é unir duas técnicas diferentes na compressão do eletrocardiograma: transformadas Wavelet e redes neurais artificiais. Essa combinação aplicada à compressão representa uma inovação científica, que pretende aliar as altas taxas de compressão obtidas com redes neurais e as baixas taxas de erro obtidas com o uso de transformadas Wavelet. Os trabalhos mais relevantes da área utilizaram, principalmente, transformadas Wavelet em conjunto com técnicas determinísticas de compressão. No geral, essa abordagem vem obtendo uma taxa de compressão de 20 para 1, caracterizando o estado da arte. Dessa forma, um desafio importante desse trabalho é conseguir uma taxa de compressão similar a essa, provando que um método não-determinístico pode substituir os métodos determinísticos nessa área com eficiência melhor ou similar.

Vale ressaltar ainda um importante desafio científico que é implícito à aplicação de redes neurais do tipo MLP na compressão de dados: a existência de uma matriz inversa capaz de expandir os dados comprimidos. Na compressão de dados, a rede MLP pode ser entendida como uma matriz que reduz os dados através de produtórios e somatórios. Para que esses dados possam ser expandidos, é necessário que exista uma matriz inversa capaz de retornar aos dados originais através de produtórios e somatórios. Caso essa matriz inversa não exista, o problema se torna insolúvel, invalidando a hipótese levantada neste trabalho.

1.2 Objetivos

O objetivo principal deste trabalho é verificar, através da metodologia proposta, a validade do uso de redes neurais em conjunto com transformadas Wavelet para compressão de sinais de ECG.

Os objetivos específicos deste trabalho são os seguintes:

- Avaliar se a técnica de compressão de ECG proposta apresenta taxas de compressão similares ou melhores que 20 para 1¹, igualando ou superando o estado da arte.
- Validar a metodologia proposta utilizando técnicas automáticas e técnicas subjetivas (com especialistas humanos) de validação de algoritmos de compressão de ECG².

1.3 Contribuições Esperadas

São esperadas as seguintes contribuições a partir deste trabalho:

- Avanços no estudo da compressão de ECG envolvendo o uso de redes neurais em conjunto com transformadas Wavelet.
- Criação de uma metodologia de compressão de ECG que possa ser, futuramente, adaptada para aplicações práticas de telecardiologia em tempo-real.
- Geração de resultados da metodologia proposta que possam ser utilizados na comparação com outros trabalhos científicos.

1.4 Organização do Documento

Este capítulo introduziu o assunto sobre compressão de ECG. O capítulo 2, por sua vez, apresenta o eletrocardiograma e a eletrocardiografia sob uma ótica técnica. O capítulo 3 traz a revisão da literatura, mostrando os trabalhos mais relevantes na área de compressão de ECG. O capítulo 4 apresenta um estudo sobre transformadas Wavelet, que são utilizadas neste trabalho. O capítulo 5 revisa o uso de redes neurais artificiais em aplicações de compressão de dados. O capítulo 6 apresenta a metodologia utilizada para comprovar a hipótese levantada neste capítulo. No capítulo 7, os resultados obtidos são apresentados e discutidos. Finalmente, o capítulo 9 apresenta as conclusões deste trabalho. Todo o código-fonte desenvolvido neste estudo é disponibilizado em anexo.

¹De forma geral, os trabalhos descritos no estado da arte não obtiveram resultados satisfatórios com taxas de compressão superiores a 20 para 1. Caso a metodologia proposta atinja essa taxa de compressão, ficará provado que a mesma pode apresentar resultados similares ou melhores ao estado da arte.

²No presente trabalho, define-se que o termo “validação” significa a comprovação (utilizando técnicas automáticas ou técnicas subjetivas) de que o método de compressão de ECG não afeta a validade clínica do exame, ou seja, não impede que um especialista dê um diagnóstico médico a partir do mesmo.

Capítulo 2

Eletrocardiografia

De acordo com (AZEVEDO, 1999) a eletrocardiografia é o estudo da atividade elétrica do coração; o registro dessa atividade constitui o eletrocardiograma (ECG); e, o aparelho que o executa chama-se eletrocardiógrafo. Por se tratar de um procedimento não-invasivo, bastante informativo e de baixo custo, o ECG se tornou um dos exames cardiológicos mais utilizados. O Prêmio Nobel de Medicina Bernardo A. Houssay afirmou, em 1946, que “o médico não considerava completo o exame do paciente se não tivesse estudado as manifestações elétricas do coração”.

O presente capítulo apresenta uma visão geral da disciplina da eletrocardiografia, uma vez que este trabalho propõe a compressão do ECG. Compreender o ECG é essencial para compreender a importância médica do mesmo, além de fornecer subsídios para a validação da metodologia proposta neste documento.

2.1 Eletrofisiologia do Coração

Segundo (AZEVEDO, 1999), cerca de 60% do corpo humano é constituído de água, que é distribuída pelos vasos sanguíneos (compartimento intravascular), pelas células (compartimento intracelular) e entre as células (compartimento extracelular). Dissolvidas nessa água, existem muitas substâncias químicas, inclusive íons como o sódio (Na^+), o cloro (Cl^-), o potássio (K^+) e o cálcio (Ca^{2+}). Quando esses íons se movimentam através das membranas das células cardíacas, é produzida corrente elétrica de natureza eletroquímica.

De acordo com (GOLDWASSER, 1997), sarcolema é a membrana que envolve a célula cardíaca, sendo constituída por duas camadas de fosfolipídios e uma de proteínas. Essa membrana apresenta a propriedade da permeabilidade seletiva, permitindo a passagem de íons através de suas camadas segundo alguns fatores. O componente do sarcolema

que permite essa movimentação é denominado de canal iônico, existindo canais próprios para cada tipo de íon. Esse canal funciona como um portão, abrindo-se ou fechando-se:

Quando em repouso, as células musculares cardíacas têm suas cargas elétricas positivas e negativas separadas pelo sarcolema. As positivas estendem-se externamente e as negativas internamente, estando a célula polarizada ou em diástole elétrica. Nesse estado, seu potencial interno negativo equivale a -90mV , enquanto que o externo positivo equivale a $+10\text{mV}$. As células musculares permanecem assim indefinidamente, a menos que recebam algum tipo de estímulo. Quando isso ocorre, o equilíbrio elétrico é alterado e há uma inversão de cargas elétricas, produzindo uma corrente elétrica de natureza eletroquímica denominada despolarização elétrica ou sístole elétrica. Ao fim desse processo, a célula está despolarizada. Tão logo despolarizadas, as células iniciam a sua repolarização, à custa de gradientes eletroquímicos e de trabalho próprio, reorganizando suas cargas elétricas ao longo de suas membranas. Existem dois tipos de células cardíacas: as células contráteis ou de PTR estável, que se despolarizam somente se estimuladas; e, as células marca-passo ou de PTR instável, que são capazes de se despolarizar naturalmente à custa do trabalho de suas mitocôndrias. As células marca-passo são responsáveis por manter o ritmo do processo cardíaco.(AZEVEDO, 1999)

2.2 Registro do Eletrocardiograma

A captação, a amplificação e o registro da atividade elétrica do coração são feitos pelo eletrocardiógrafo. O primeiro eletrocardiógrafo com galvanômetro foi inventado por Willem Einthoven em 1901. Por esse feito, ele recebeu o Prêmio Nobel de Medicina em 1924 (NOBEL, 2007). Atualmente, os eletrocardiógrafos são construídos com amplificadores de instrumentação (UNICAMP, 2007). Um diagrama de blocos de um eletrocardiógrafo desse tipo é mostrado na figura 2.1.

O papel de registro do ECG tem o desenho quadriculado: são pequenos quadrados de 1 mm de cada lado. A abscissa marca o intervalo de tempo, onde cada 1 mm corresponde a 40 ms, considerando-se a velocidade padrão de 25 mm/s (GOLDWASSER, 1997). A diferença de potencial é representada no eixo das ordenadas, onde cada 1 mm equivale a 0,1 mV. Um exemplo de registro do eletrocardiograma é apresentado na figura 2.2.

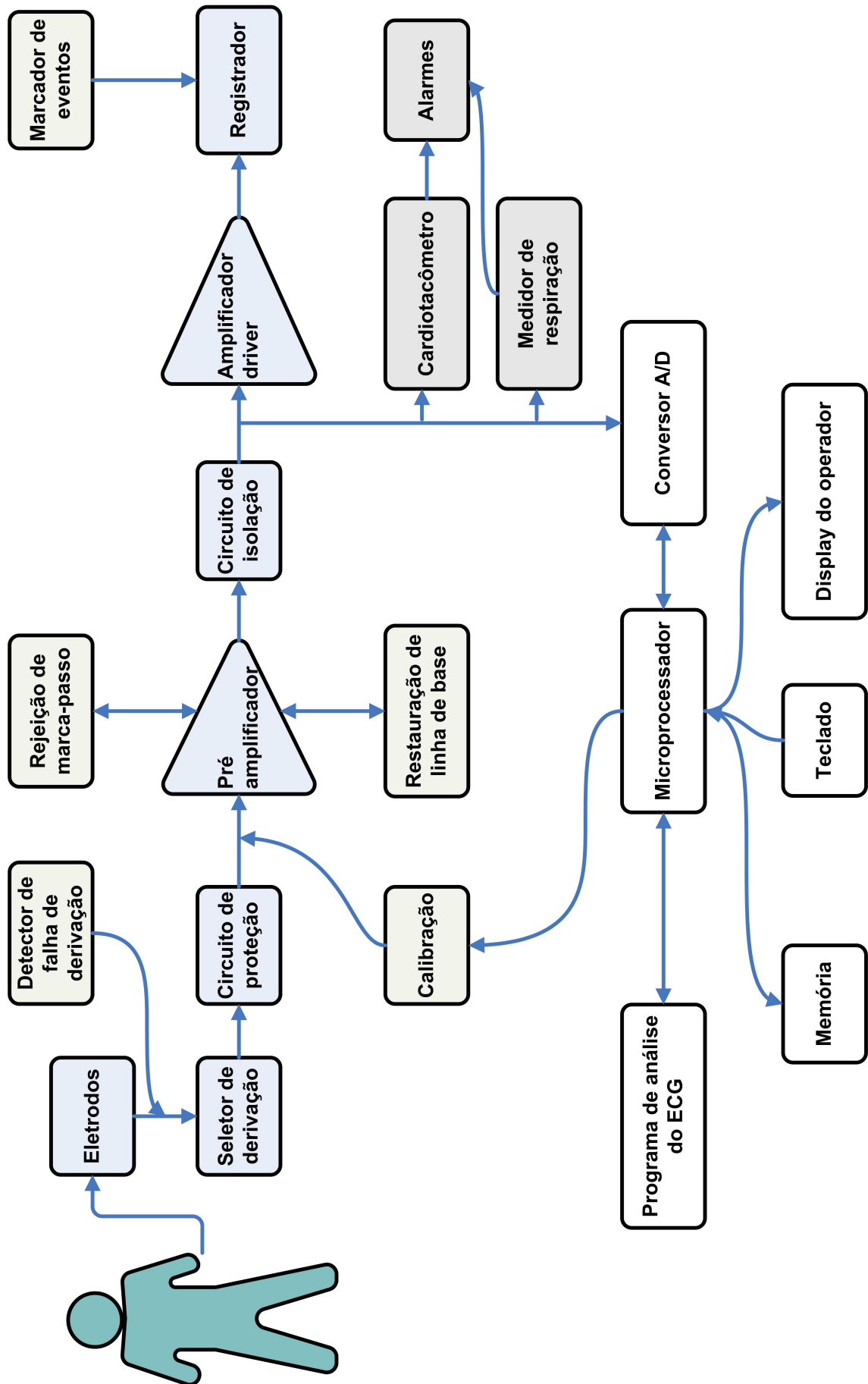


Figura 2.1: Eletrocardiógrafo construído com amplificadores de instrumentação (UNICAMP, 2007)

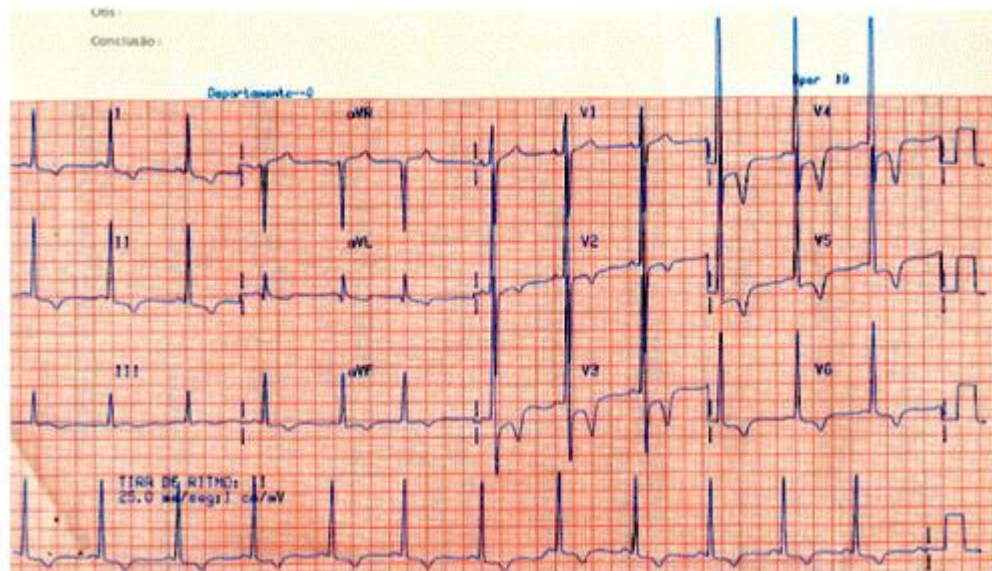


Figura 2.2: Exemplo de registro do eletrocardiograma (ARTEAGA; GUTIERREZ, 2006)

2.3 Derivações

Normalmente, o eletrocardiograma é registrado doze vezes no paciente, usando-se doze combinações diferentes de posições dos eletrodos. Embora a atividade elétrica cardíaca seja sempre a mesma, a captação feita em ângulos diferentes gera sinais com morfologias diferentes. Essas captações são chamadas, no meio clínico, de derivações do ECG. Cada derivação ressalta alguns aspectos da atividade cardíaca. Em conjunto, as doze derivações permitem que um especialista tenha uma perspectiva abrangente do coração. Na prática, existem pontos padronizados para a colocação dos eletrodos. As 12 derivações clássicas são obtidas de diversos sinais captados através de nove eletrodos: dois nos braços, um na perna esquerda e seis sobre o peito. Um eletrodo adicional é colocado na perna direita, servindo como referência para reduzir a interferência externa (UNICAMP, 2007).

A figura 2.3 apresenta visualmente as 12 derivações tipicamente utilizadas nos exames de eletrocardiograma.

2.4 Eletrocardiograma Normal

A figura 2.4 apresenta um ECG normal, identificado os seus principais componentes.

A primeira onda do ciclo cardíaco se denomina P e expressa a despolarização dos átrios. A repolarização dos átrios é expressa na onda Ta, que

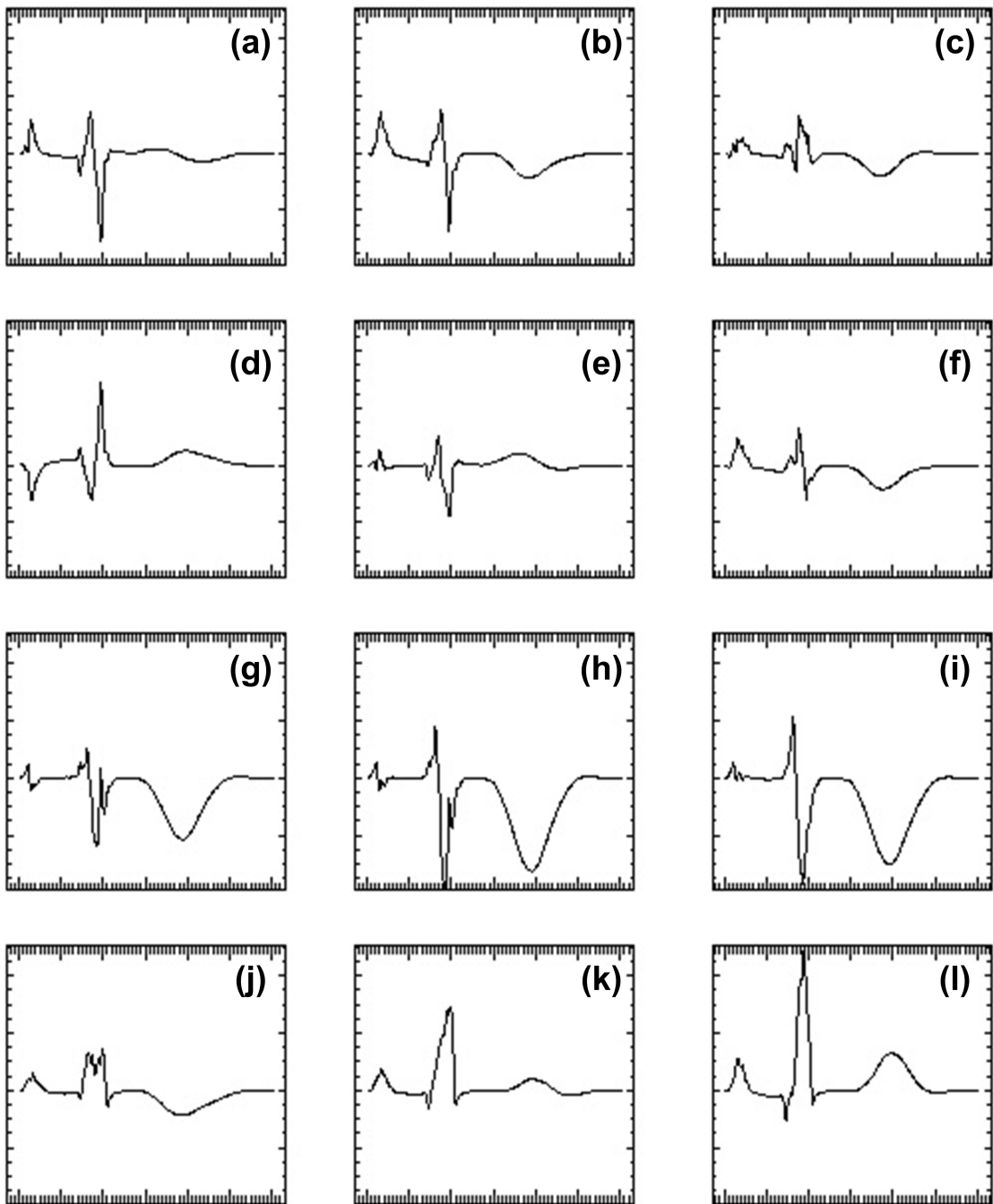


Figura 2.3: As 12 derivações clássicas do ECG: a) I; b) II; c) III; d) aVR; e) aVL; f) aVF; g) V1; h) V2; i) V3; j) V4; k) V5; e, l) V6 (ASCHENBRENNER, 2000)

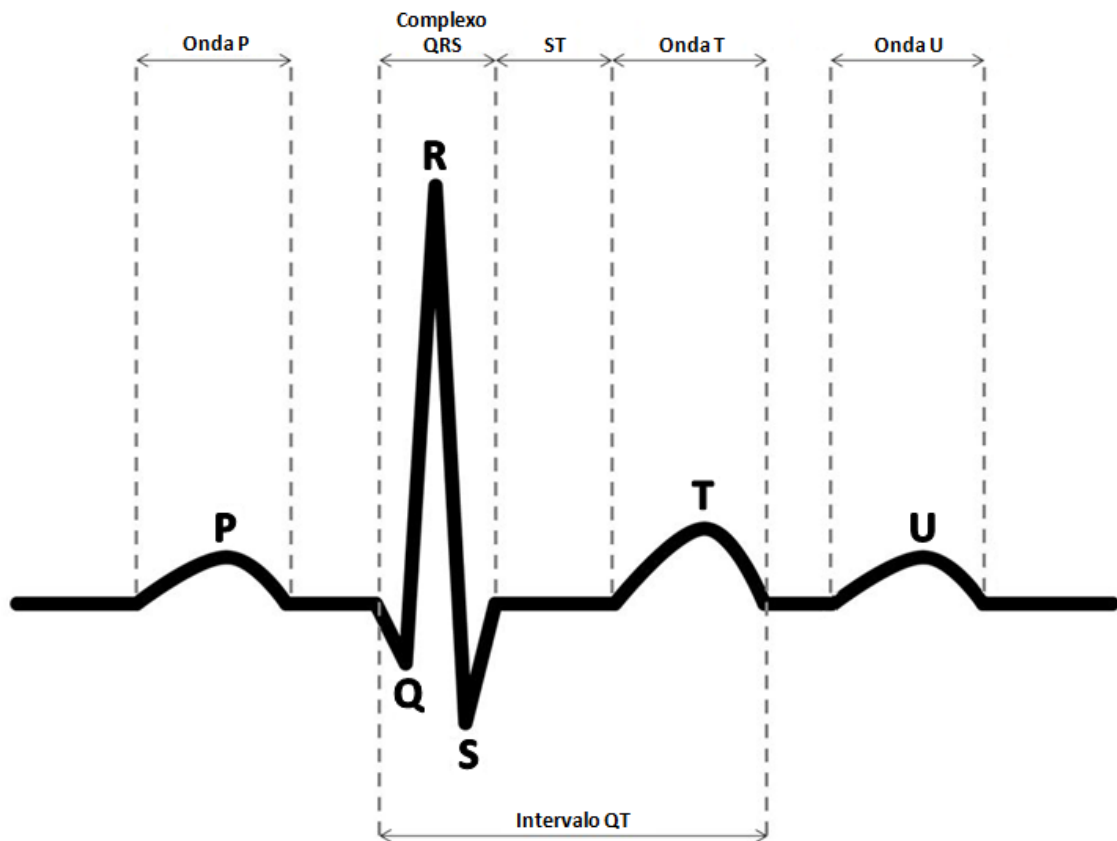


Figura 2.4: Diagrama esquemático exibindo os principais componentes de um ECG normal

geralmente não é visualizada no ECG porque se inscreve simultaneamente à despolarização ventricular, que tem grande potencial e a absorve. O segmento PR começa no final da onda P e termina no início do complexo QRS, correspondendo à passagem do estímulo pela junção atrioventricular, onde sofre um atraso para permitir a contração atrial e o enchimento ventricular. O intervalo PR é a distância entre o início da onda P e o início do complexo QRS, medindo o tempo de condução atrioventricular. O complexo QRS corresponde ao tempo de despolarização dos ventrículos, desde o terço médio do septo até as porções basais dos ventrículos e do septo. Dentro do complexo QRS, a primeira onda negativa que aparece é designada de onda Q, a primeira onda positiva é designada de onda R e a segunda onda negativa é designada de onda S. O ponto J situa-se onde termina o complexo QRS e inicia o segmento ST. O segmento ST começa no final da onda S e termina no início da onda T. O intervalo QT corresponde à sístole elétrica ventricular, iniciando no começo do complexo QRS e termina no final da onda T. O intervalo RR é a distância entre duas ondas R sucessivas. O intervalo PP, por sua vez, é a distância entre duas ondas P sucessivas. Finalmente, a onda U nem sempre aparece no



Figura 2.5: Alguns exemplos de arritmias cardíacas retirada da base de dados utilizada neste trabalho (PHYSIONET, 2007)

registro eletrocardiográfico. Quando aparece, o seu início se confunde com o final da onda T. (AZEVEDO, 1999)

2.5 Arritmias Cardíacas

Diversas perturbações que alteram a frequência e/ou o ritmo dos batimentos cardíacos são chamadas de arritmias cardíacas. Grande parte dessas anomalias são inofensivas. Contudo, algumas delas podem causar a morte do paciente, configurando um caso de emergência médica. Em geral as arritmias podem ser caracterizadas por um ritmo cardíaco rápido (taquicardia), lento (bradicardia) ou irregular (GUYTON; HALL, 2002).

É importante salientar que as arritmias cardíacas apresentam morfologias diferentes do ECG normal, como pode ser visto na figura 2.5. Portanto, é essencial que o método de compressão do ECG seja robusto o suficiente para comprimir essas exceções sem grandes perdas. Esse é um dos maiores desafios neste campo da ciência. Um método de compressão que funcione muito bem para o ECG normal, mas que seja ineficiente para as arritmias cardíacas, não representa uma solução viável clinicamente. Afinal, o grande objetivo da monitorização do ECG é justamente detectar a ocorrência das anomalias.

2.6 Resumo do Capítulo

Compreender a natureza de um sinal é um passo fundamental na busca por novas abordagens e técnicas de compressão. O presente capítulo apresentou a geração, a captura e a análise do sinal de ECG. É importante destacar que o sinal apresenta componentes de baixa frequência de longa duração e componentes de alta frequência de curta duração. Nos próximos capítulos, essa informação será muito útil para justificar o uso de transformadas Wavelet neste trabalho.

Capítulo 3

Compressão de ECG: Revisão de Literatura

Nos últimos anos, diversos autores propuseram diferentes abordagens de compressão de ECG. De forma geral, esses trabalhos científicos podem ser agrupados em quatro grandes grupos:

- Abordagens baseadas em transformadas: valem-se dos mais diversos tipos de transformadas (Fourier, Wavelets, entre outros) para representar o sinal de ECG em outros domínios, que sejam mais adequados à compressão (GARCÍA, 2006), (CHEN; ZHANG; SHI, 2006), (KU et al., 2006), (KIM; YOO; MOON, 2006).
- Abordagens bidimensionais: tratam o sinal de ECG como uma imagem bidimensional e valem-se da interpolação dos pontos para aumentar a eficiência da compressão, utilizando muitas vezes técnicas consagradas na compressão de imagens, como a transformada DCT (*discrete cosine transform*) do padrão JPEG2000 (SOBREIRA, 2006), (CHOU et al., 2006).
- Abordagens baseadas em similaridades: aproveitam-se dos padrões repetitivos existentes nas curvas de ECG para representar o sinal em uma forma comprimida (HENRIQUES et al., 2006), (CARVALHO; SILVA; FINAMORE, 2002), (FILHO et al., 2006).
- Outras abordagens de compressão: compreendem todas as outras técnicas de compressão (redes neurais, algoritmos específicos, entre outros) que não possuem um volume de trabalhos suficientemente grande para comporem grupos específicos (BRITO et al., 2005), (KANNAN; ESWARAN; SRIRAAM, 2002), (ZHAO et al., 1997), (AL-HUKAZI; AL-NASHASH, 1996).

O objetivo deste capítulo é apresentar os principais trabalhos que foram publicados nos últimos anos dentro desses grandes grupos.

3.1 Abordagens baseadas em Transformadas

A telecardiologia é uma das mais maduras aplicações de telemedicina. A princípio, existem duas abordagens envolvendo a transmissão de ECG: a transmissão em tempo-real e a transmissão de exames completos. Historicamente, o envio de exames completos de eletrocardiografia sempre foi mais investigado pela ciência. Isso se devia, principalmente, a limitações de banda de transmissão na época em que a maioria desses estudos começou. Os recentes avanços nas telecomunicações, entretanto, permitiram estudos envolvendo a transmissão em tempo-real do sinal de ECG.

No artigo (GARCÍA, 2006) Alesanco e colegas propõem um método de compressão de ECG baseado em transformadas Wavelet. O sinal é segmentado de forma que cada segmento possua apenas um pico QRS. O segmento é, então, subtraído de um sinal padrão selecionado em um dicionário de padrões. O sinal residual (resultado dessa subtração) e o padrão utilizado são codificados com uma expansão Wavelet Coiflet. A compressão é atingida através da escolha dos coeficientes Wavelet utilizando um limite de corte.

Normalmente, as técnicas de compressão de ECG baseadas em transformadas Wavelet utilizam esquemas baseados em limite de corte (“thresholding”) para melhorar seu desempenho através da eliminação de coeficientes Wavelet próximos a zero. Os coeficientes restantes são, em geral, submetidos à codificação de Huffman, que visa compactar esses dados. A codificação de Huffman exige, por definição, o conhecimento prévio das estatísticas dos símbolos que serão codificados. Esse requisito do método afeta a robustez dessas técnicas em sinais com diferentes características estatísticas.

Diante dessa situação, alguns pesquisadores (CHEN; ZHANG; SHI, 2006) propuseram um esquema de compressão de ECG que utiliza um quantificador escalar de zona morta (“uniform scalar dead zone quantiser”) na codificação dos coeficientes Wavelet. Essa quantização é utilizada após o a aplicação do limite de corte. Os tamanhos das seqüências de coeficientes zeros e não-zeros são codificados com dois tipos de codificação Golomb que não necessitam de um conhecimento a priori dos símbolos. Aplica-se a codificação Golomb (GOLOMB, 1966) para as seqüências de zeros e a codificação Golomb-Rice (GOLOMB, 1966) para os valores não-nulos. Essa abordagem reduz significativamente a distorção oriunda da quantização, além de melhorar a entropia da codificação. O algoritmo se torna bastante robusto por dispensar o conhecimento a priori das estatísticas do sinal. Além disso, a codificação Golomb é bastante eficiente do ponto de vista computacional, sendo recomendada para aplicações em tempo-real.

Um método de compressão de ECG sem corte dos coeficientes Wavelet foi proposto em (KU et al., 2006). Essa abordagem utilizou um tipo especial de transformada

Wavelet denominada 1-DNRDPWT (*reversible round-off nonrecursive one-dimensional (1-D) discrete periodized Wavelet transform*). A transformada 1-DNRDPWT executa uma decomposição em estágios que gera coeficientes menores e resiste a erros propagados de truncamento. Os autores utilizam escalas diferentes de quantização para os coeficientes. Na compressão de dados, o crescimento da taxa de compressão costuma gerar, indesejavelmente, um aumento da taxa de erro médio. O método dos pesquisadores luta contra isso utilizando a tática de permanecer em um zona de operação linear. Dessa forma, eles conseguem atingir boas taxas de erro médio (6.10%) operando com taxas de compressão de até 20 vezes.

O tamanho do atraso é muito importante na compressão de ECG quando o domínio de aplicação envolve sistemas em tempo-real. Em geral, métodos que apresentam altas taxas de compressão apresentam um grande atraso, pois necessitam de grandes porções de amostras de dados para aproveitarem a interpolação característica do sinal de ECG. O uso de janelas de tempo elevadas, todavia, inviabiliza a utilização dessas abordagens em aplicações onde o tempo é um fator crítico, como transmissão de dados em ambulâncias, por exemplo.

No artigo (KIM; YOO; MOON, 2006), os pesquisadores apresentam um método baseado em Wavelets com atraso quase instantâneo. As janelas de tempo do método são ajustadas automaticamente, equivalendo à metade do período do sinal de ECG. O algoritmo proposto separa o sinal em dois tipos de blocos: “complexos” ou “simples”. Os blocos “complexos” agrupam os picos QRS, enquanto os blocos “simples” correspondem às faixas de sinal que não apresentam grande mudança de dados. Todos os blocos são processados por uma transformada Wavelet. A etapa de codificação, entretanto, preserva mais bits dos coeficientes dos blocos “complexos”. Dessa forma, o método consegue comprimir o ECG com taxas razoáveis de compressão em conjunto com baixo atraso

3.2 Abordagens Bidimensionais

Em contraste às abordagens tradicionais que utilizam transformadas unidimensionais, Alexandre e colegas (SOBREIRA, 2006) apresentam uma introdução teórica do uso de transformadas bidimensionais na compressão de ECG. A transformada de cosseno discreto (DCT - “discrete cosine transform”) unidimensional é largamente utilizada em padrões de codificação de áudio, como o ISO/IEC MPEG. A transformada bidimensional, por sua vez, é utilizada no padrão de codificação de imagens JPEG2000. Basicamente, a transformada bidimensional é equivalente à aplicação de uma transformada unidimensional seguida de outra transformada unidimensional. Uma característica interessante da DCT é que ela

conserva a energia do sinal de entrada. Isso implica que a energia do ruído será equivalente nos domínios DCT e do tempo. O mesmo acontece com a razão sinal-ruído SNR (*signal-to-noise ratio*) e com a diferença percentual da raiz média quadrática (*DPRM - percent root-mean-squared differences*). Os autores ressaltam que a transformada Karhunen-Loeve pode oferecer resultados melhores. Todavia, esse tipo de transformada exige uma alta complexidade computacional e, conseqüentemente, uma baixa taxa de transmissão¹. Diante dessa realidade, a DCT bidimensional se mostra como uma opção mais factível.

No artigo (CHOU et al., 2006), Chou e colegas apresentam inovações muito interessantes na compressão de sinais de ECG irregulares. O método descrito utiliza a detecção de QRS para segmentar o sinal. Cada segmento é cortado 130 amostras antes do pico. Os segmentos são então organizados paralelamente em uma matriz 2D. Uma vez que não se usa um tamanho de segmento fixo, torna-se necessário completar os vetores com os valores médios das amostras. Uma das grandes inovações apresentadas é a reordenação dos segmentos através das suas similaridades. Os segmentos ordenados geram uma matriz mais suave. Essa matriz é codificada utilizando-se a codificação JPEG2000.

Os resultados apresentados são bastante promissores e o método se destaca na compressão de sinais extremamente irregulares. Esse ótimo desempenho se deve principalmente à reordenação dos segmentos similares. A abordagem é muito útil para aplicações de telecardiologia que envolvem a transmissão de exames completos. Todavia, não apresenta um bom desempenho em aplicações de tempo-real, pois necessita de janelas de tempo relativamente altas (em torno de 10 minutos, pelo menos).

3.3 Abordagens baseadas em Similaridades

A forte regularidade do sinal de ECG sugere que técnicas de compressão baseadas em similaridades devem se mostrar efetivas. Nesse contexto, é natural dividir a curva em batimentos cardíacos, utilizando, principalmente, o intervalo RR. Isso acontece porque existem diversos algoritmos de detecção do complexo QRS extremamente eficientes. Contudo, essa abordagem apresenta uma séria deficiência, pois só é viável quando aplicada em sinais normais, falhando na presença de casos anormais, como a taquicardia ventricular e a fibrilação ventricular.

Visando uma alternativa a essa inconveniência, Henriques e colegas (HENRIQUES et al., 2006) propuseram um método que busca as similaridades do ECG sem se valer

¹Em aplicações de telecardiologia com compressão do ECG em tempo-real, sempre existe a questão do “atraso”, que é causado pelo tempo necessário para comprimir o sinal. Algoritmos com alta complexidade computacional inviabilizam essa operação em tempo-real, pois gastam mais tempo e geram um “atraso” muito grande.

de algoritmos de segmentação clínica do sinal. De fato, eles utilizaram curvas cúbicas de Bezier para identificar pontos de controle importantes do sinal. Esses pontos simplificam o sinal e permitem que os autores apliquem um algoritmo modificado k-means para agrupar os padrões de batimentos. Os padrões são armazenados em um dicionário de dados, o que evita a repetição de trechos de sinal similares. Essa abordagem acaba por apresentar resultados interessantes, similares aos obtidos com as técnicas baseadas na transformada Wavelet SPIHT.

Recentemente, o MMP (*multidimensional multiscaler parser*) (CARVALHO; SILVA; FINAMORE, 2002) foi utilizado com sucesso para comprimir sinais de ECG. Isso acontece em função da natureza quase periódica desse tipo de sinal, que o qualifica como um candidato natural à aplicação de técnicas baseadas em padrões e similaridades. Todavia, o sinal de ECG apresenta anomalias que são muito relevantes clinicamente. Essas anomalias exigem um tratamento especial para que não afetem o desempenho do MMP.

Em (FILHO et al., 2006), os autores utilizam a interpolação de ECG e a normalização do período para que o intervalo entre os batimentos cardíacos torne-se constante. O sinal é segmentado de forma que cada segmento seja relacionado a um padrão alocado em um dicionário de padrões. Os experimentos realizados na base de dados de arritmias do MIT-BIH revelou que essa abordagem supera o desempenho do método SPIHT (LU; KIM; PEARLMAN, 2000) e é comparável a abordagem baseada em JPEG2000 proposta por (BILGIN; MARCELLIN; ALTBACH, 2003).

3.4 Outras Abordagens de Compressão

No artigo (BRITO et al., 2005), Brito e colegas propõem um método preditivo para comprimir o sinal de ECG. A abordagem utiliza um modelo auto-regressivo para prever o valor do k-ésimo ponto do sinal a partir dos n pontos anteriores. O método foi testado em diferentes tipos de sinais anormais (arritmias, fibrilações, entre outros), apresentando uma baixa taxa de erro. Todavia, não apresentou taxas de compressão semelhantes às melhores descritas na literatura.

Um método de compressão de ECG utilizando redes neurais artificiais é apresentado em (KANNAN; ESWARAN; SRIRAAM, 2002). Os autores segmentaram o sinal em diversos blocos de mesmo tamanho. O treinamento da rede MLP foi realizado com o algoritmo de retro-propagação, onde algumas amostras de cada bloco foram apresentadas à rede. O objetivo do treinamento foi prever as amostras restantes. Dessa forma, só é necessário transmitir algumas porções do sinal. O método foi testado em três registros da MIT-BIH ECG Compression Test Database (PHYSIONET, 2008), apresentando excelentes

taxas de compressão: 46.97 e 155.2. No entanto, os resultados não foram satisfatórios, quando se avaliou a razão sinal-ruído. Embora os resultados desse trabalho não possam ser totalmente considerados, uma vez que não houve uma validação adequada, é importante ressaltar o grande poder de compressão apresentado pelas redes neurais.

No artigo (ZHAO et al., 1997), Zhao e colegas demonstram a utilização de uma rede neural artificial separada em três seções, referenciando as componentes principais do ECG: a onda P, o complexo QRS e a onda T. Essa abordagem apresentou bons desempenhos de compressão; todavia sofreu grande influência da variabilidade das curvas de ECG.

O treinamento Hebbiano (GURNEY, 1997) foi utilizado para desenvolver redes neurais capazes de comprimir o sinal de ECG em (AL-HUKAZI; AL-NASHASH, 1996). Esse método foi capaz de comprimir o sinal em até 30 vezes, apresentado um DPRM inferior a 5%. Não obstante, o método não foi amplamente testado, de forma que os resultados apresentados não são suficientes para determinar sua eficácia clínica.

3.5 Resumo do Capítulo

A literatura científica apresenta muitos trabalhos recentes na área de compressão de ECG. Isso se deve à importância desse tema no domínio da telecardiologia e telemedicina. Dentre as várias abordagens apresentadas, as técnicas baseadas em transformadas Wavelet vêm apresentando os resultados mais promissores. Os próximos capítulos deste trabalho mostrarão como as transformadas Wavelet podem ser associadas a redes neurais na busca por novas abordagens de compressão de ECG.

Capítulo 4

Wavelets

Transformações matemáticas são aplicadas em sinais para se obter informações adicionais que não são claramente visíveis originalmente. Dentre as diversas transformadas existentes, a transformada de Fourier é provavelmente a mais popular, tendo sido descrita ainda no século XIX (MIRADOR, 1993a). Nos últimos anos, entretanto, as transformadas Wavelet vêm ganhando bastante espaço em várias áreas puras ou aplicadas de ciência e engenharia (SWELDENS; SCHRÖDER, 2007). Isso se deve principalmente às informações adicionais que esse tipo de transformada oferece.

O objetivo deste capítulo é apresentar os fundamentos teóricos de Wavelets, que são essenciais neste trabalho. Dessa forma, o capítulo é dividido da seguinte maneira: a primeira seção oferece uma revisão da transformada de Fourier; a seção seguinte apresenta a transformada Wavelet contínua; a terceira seção apresenta a transformada Wavelet discreta, que será efetivamente utilizada neste trabalho; e, finalmente, uma conclusão confrontando o material teórico apresentado e o contexto do problema de compressão de ECG.

4.1 Transformadas de Fourier

Físico e matemático francês, Joseph Fourier publicou a edição final de sua obra mais notável, a *Theorie analytique de la chaleur* (Teoria analítica do calor) em 1822, após investigações conduzidas desde 1807 a propósito da propagação do calor. A sexta seção de seu livro é devotada à solução de “um problema mais geral: desenvolvimento de uma função qualquer em série de senos e co-senos de arcos múltiplos”. Embora esse assunto já tivesse merecido a atenção de Euler, d’Alembert, Daniel Bernoulli e Lagrange, coube a Fourier o mérito de haver criado um novo instrumento matemático capaz de representar funções periódicas descontínuas por meio de funções contínuas (MIRADOR, 1993a).

A análise de Fourier consiste em funções de sobreposição de senos e co-senos, transformando um sinal (função) do domínio do espaço para o domínio da frequência (GONZALEZ; WOODS, 2000). As Transformadas de Fourier (descritas matematicamente na função 4.1, onde ω é a frequência angular, t é a componente do tempo e $i = \pm\sqrt{-1}$) são capazes de representar um sinal temporal no domínio da frequência ou vice-versa. Todavia, essas transformadas não são capazes de representar os domínios do tempo e da frequência ao mesmo tempo. Ou seja, não se pode saber quando efetivamente cada frequência está ocorrendo. Essa característica não representa uma deficiência quando os sinais analisados são periódicos. Isso acontece porque, em um sinal periódico, todas as componentes de frequência ocorrem ao mesmo tempo, durante todo o sinal.

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-i\omega t} dt \quad (4.1)$$

A figura 4.1 apresenta um sinal periódico, enquanto a figura 4.2 apresenta a mesma função no domínio da frequência. A observação desses dois gráficos mostra que o sinal é composto por cinco componentes principais com frequências de 15, 20, 25, 40 e 70Hz que ocorrem todo tempo.

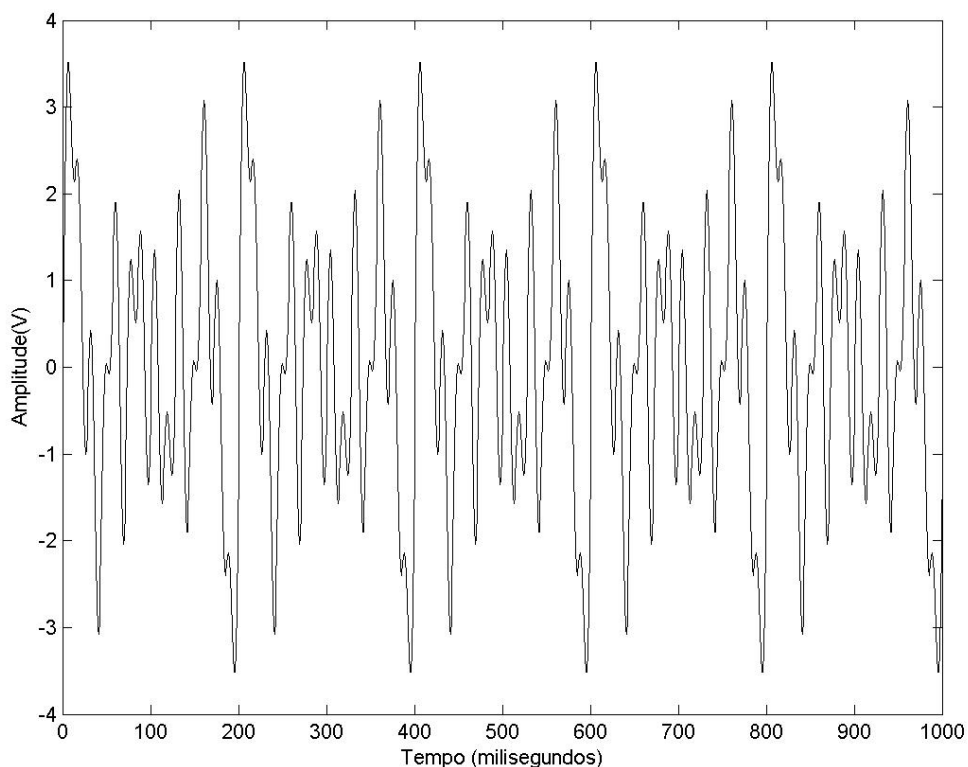


Figura 4.1: Exemplo de sinal periódico, com componentes de 15, 20, 25, 40 e 70 Hz

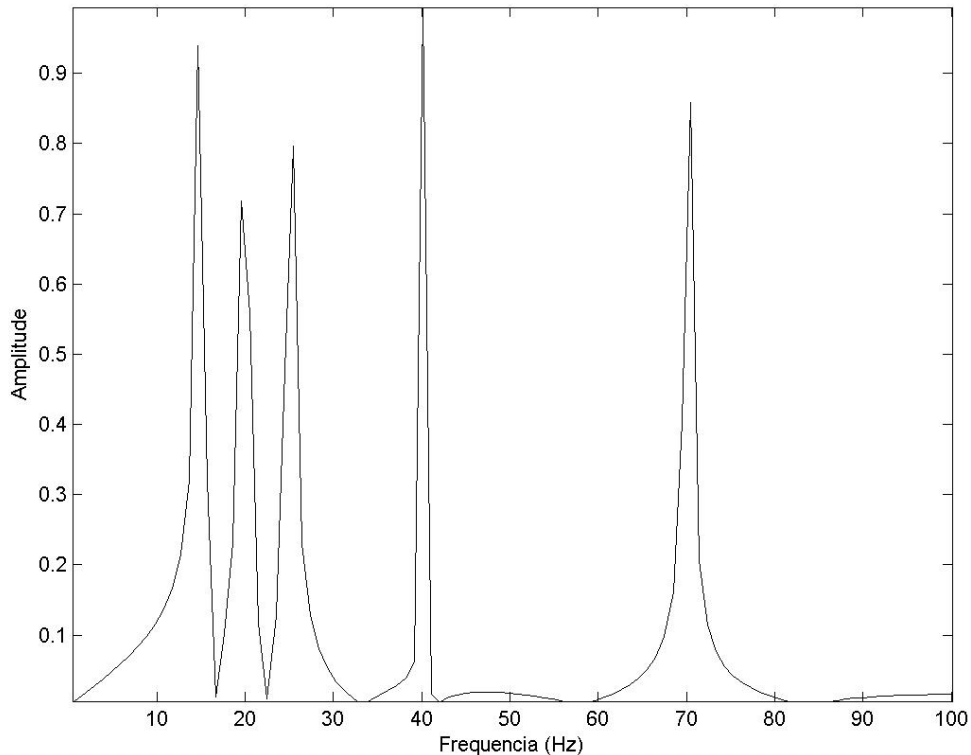


Figura 4.2: Sinal periódico representado no domínio da frequência

No entanto, muitos sinais naturais não apresentam comportamento totalmente periódico. São, na verdade, sinais não-periódicos que possuem diversas componentes de frequência que não ocorrem necessariamente ao mesmo tempo. A figura 4.3 representa um sinal não-periódico com as mesmas cinco componentes de frequência de 15, 20, 25, 40 e 70Hz. O problema da análise de Fourier no tocante a sinais não-periódicos fica evidente na figura 4.4, que apresenta a função no domínio da frequência. Embora as componentes de frequência ocorram em tempos diferentes nessa função, fica impossível precisar as suas ocorrências através da transformada de Fourier. Torna-se necessário, então, buscar outras formas de representação.

A TJF (*transformada por janelas de Fourier*) visa suprir as deficiências apresentadas através da segmentação do sinal analisado. Nessa abordagem, uma janela de tamanho fixo desliza através do tempo, realizando sucessivas transformadas de Fourier. Assim, torna-se possível localizar as componentes de frequência no decorrer do tempo. A figura 4.5 apresenta a representação tridimensional relacionando tempo, frequência e amplitude do sinal não-periódico anteriormente apresentado.

Embora a transformada por janelas de Fourier apresente maior poder de representação, ela ainda é suscetível a um significativo problema. O Princípio da Incerteza

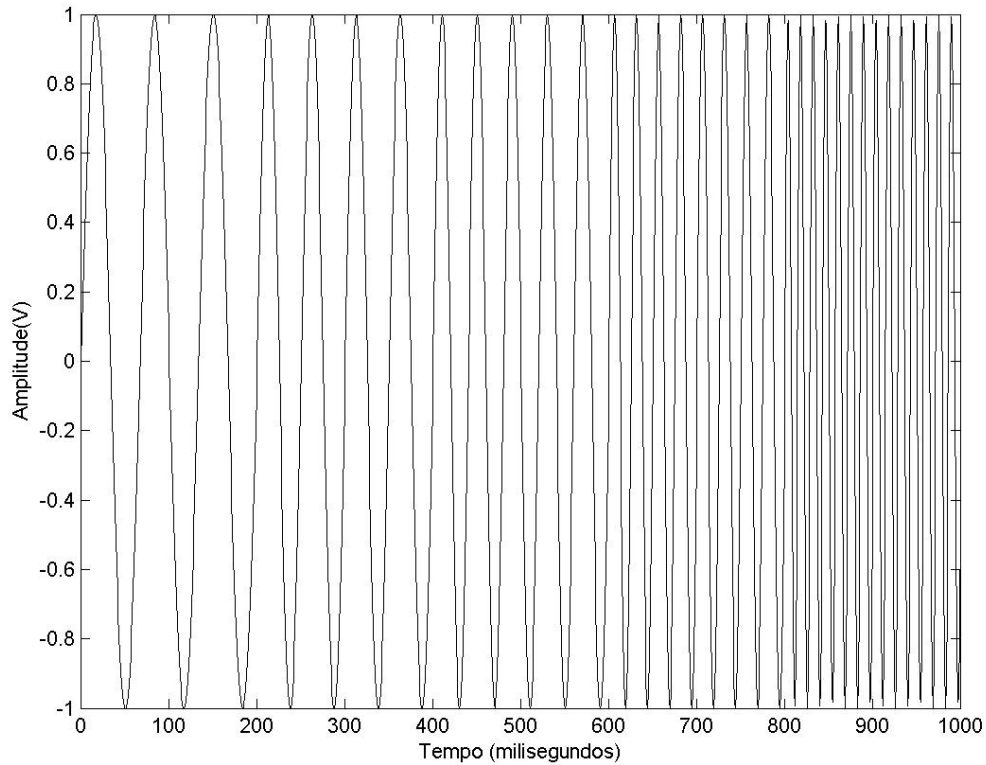


Figura 4.3: Exemplo de sinal não-periódico, com componentes de 15, 20, 25, 40 e 70 Hz

de Heisenberg enuncia que “é impossível especificar e determinar, simultaneamente, a posição e a velocidade de uma partícula, com precisão absoluta” (MIRADOR, 1993b). O mesmo princípio pode ser aplicado ao dilema tempo-frequência dos sinais. Não é factível determinar, com precisão absoluta, as ocorrências das componentes de frequência ao longo do tempo. Em vez disso, torna-se necessário trabalhar com faixas de tempo e faixas de frequência.

O problema da transformada por janelas de Fourier é que ela utiliza janelas de tamanho fixo. Isso ocorre porque esse tamanho influencia diretamente na precisão da análise. Se a janela é estreita, tem-se uma grande precisão no domínio do tempo e uma baixa resolução no domínio da frequência. O inverso ocorre com janelas largas. Os efeitos do tamanho da janela de amostragem são apresentados nas figuras 4.6 (janela estreita) e 4.7 (janela larga).

Os exemplos apresentados ilustram o problema implícito de resolução que envolve a transformada por janelas de Fourier. Em sinais onde as componentes de frequência estão bem definidas, o problema pode ser minimizado. Não obstante, na maioria dos sinais naturais (como é o caso do sinal de ECG) isso não ocorre. Nesses sinais, as altas frequências costumam ocorrer em pontos isolados de tempo, enquanto as baixas

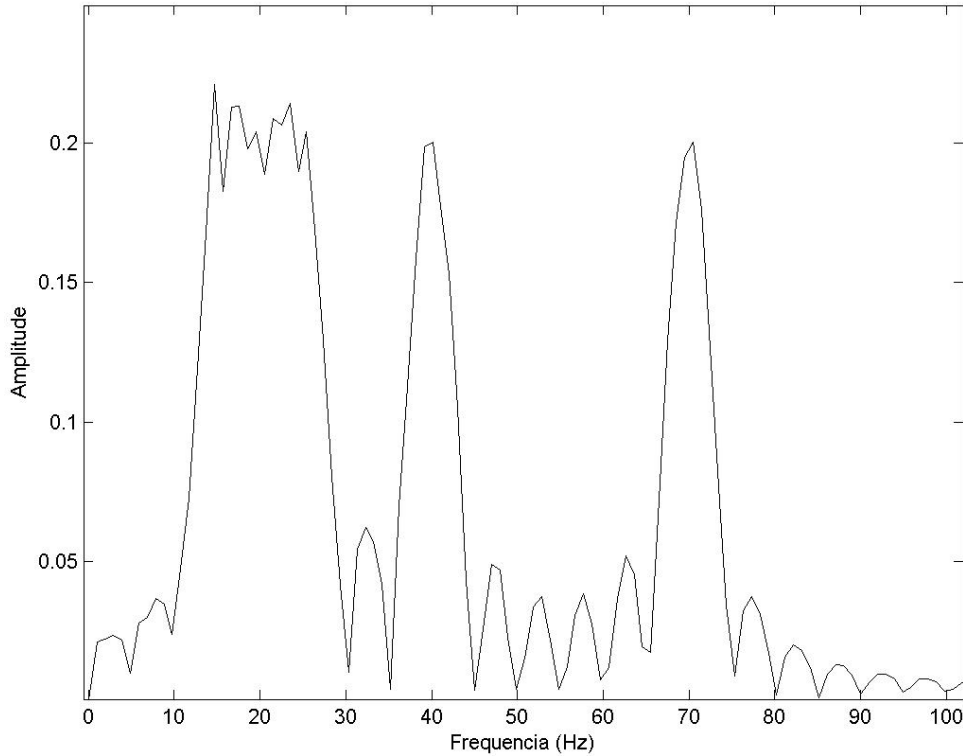


Figura 4.4: Sinal não-periódico representado no domínio da frequência

freqüências ocorrem durante todo o sinal. Essa característica torna necessária a utilização de transformadas que trabalhem com janelas de tamanho variável, como é o caso das transformadas Wavelet que serão apresentadas na seção seguinte.

4.2 Transformada Wavelet Contínua

A TWC (*transformada Wavelet contínua*)(WALKER, 1999) é uma alternativa à transformada por janelas de Fourier, superando o problema da resolução. Como na transformada apresentada na seção anterior, a TWC é uma transformada que é computada separadamente em vários segmentos do sinal. Todavia, a TWC não utiliza um tamanho de janela fixo. De fato, a largura da janela da TWC varia de acordo com o espectro de frequência analisado.

A TWC é definida na forma da equação 4.2, onde ψ é chamada de Wavelet-mãe, τ é a translação, ς é a escala e t é a componente do tempo:

$$TWC_{\varsigma}^{\psi} = \Psi_x^{\psi}(\tau, \varsigma) = \frac{1}{\sqrt{\varsigma}} \int x(t) \psi * \left(\frac{t - \tau}{\varsigma} \right) dt \quad (4.2)$$

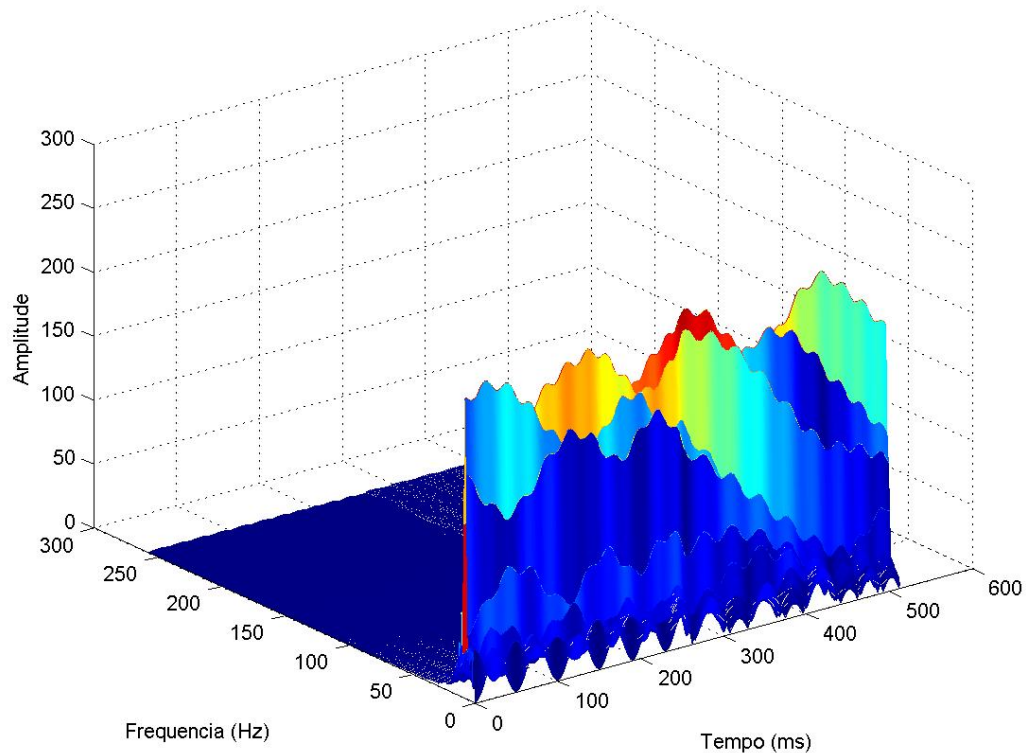


Figura 4.5: Representação tridimensional relacionando tempo, freqüência e amplitude do sinal não-periódico

O sinal transformado é uma função de duas variáveis: τ (translação) e ς (escala). A função ψ é chamada de Wavelet-mãe. O termo Wavelet se refere a uma “pequena onda finita”, que é utilizada como janela de amostragem. A Wavelet-mãe, por sua vez, é a função-protótipo para a geração de várias Wavelets, através de mudanças de τ (translação) e ς (escala).

A translação ocorre na TWC da mesma forma como ocorria na TJF. Assim, a variável translação indica o deslocamento da janela de amostragem ao longo do tempo. Na TWC, entretanto, não existe um parâmetro de freqüência, como na TJF. Em vez disso, tem-se a variável de escala, que é definida como a inversa da freqüência. Altas escalas correspondem a baixas freqüências, representando uma visão global e pouco detalhada do sinal. As baixas escalas, por sua vez, correspondem às componentes de alta freqüência, ou seja, a uma visão local e detalhada do sinal.

Esse comportamento da escala na TWC é bastante conveniente para a análise de sinais naturais. Como já se comentou anteriormente, em sinais naturais, as baixas freqüências tendem a ocorrer durante todo o sinal, enquanto as altas freqüências ocorrem

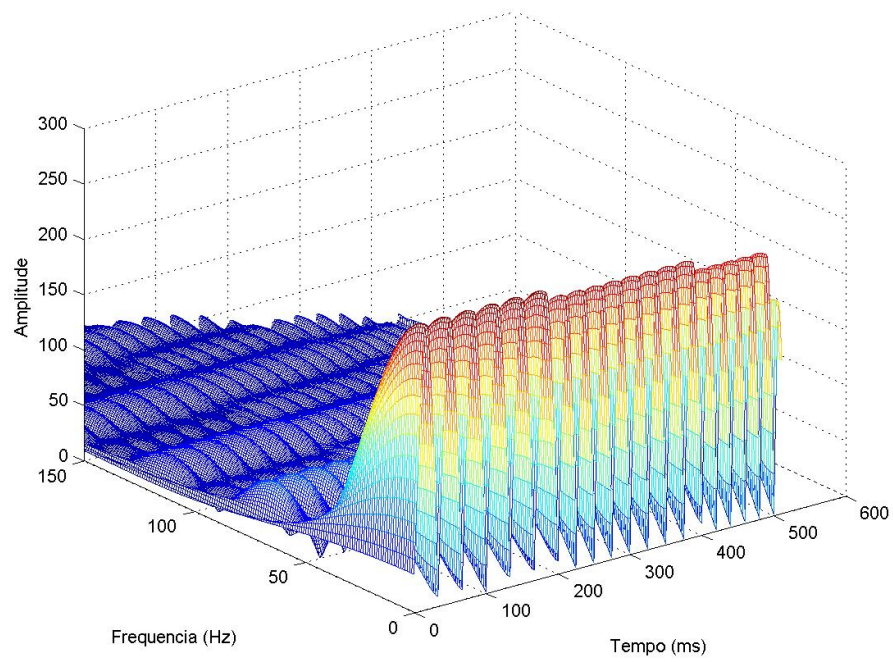


Figura 4.6: Transformada por janelas de Fourier utilizando janelas estreitas

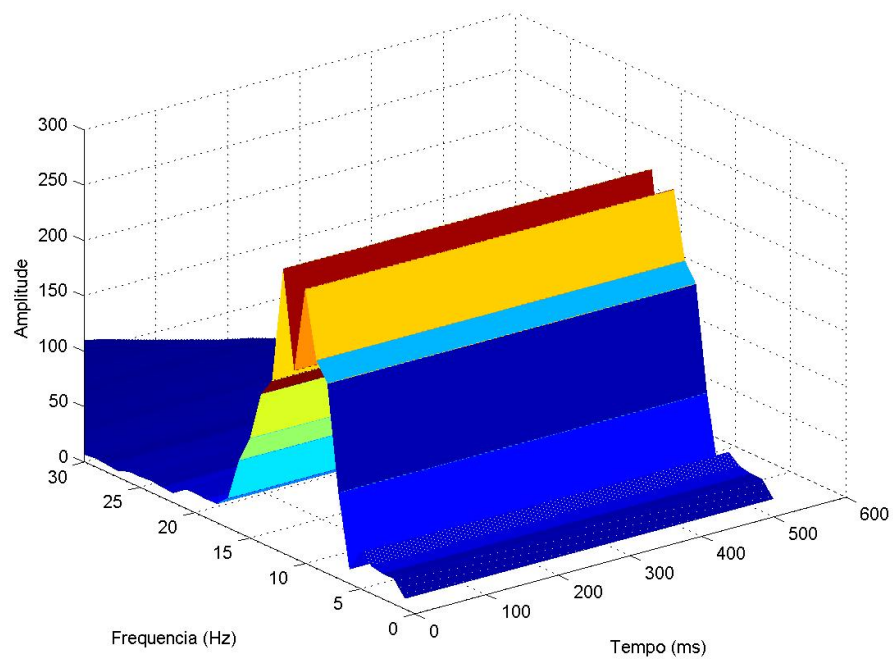


Figura 4.7: Transformada por janelas de Fourier utilizando janelas largas

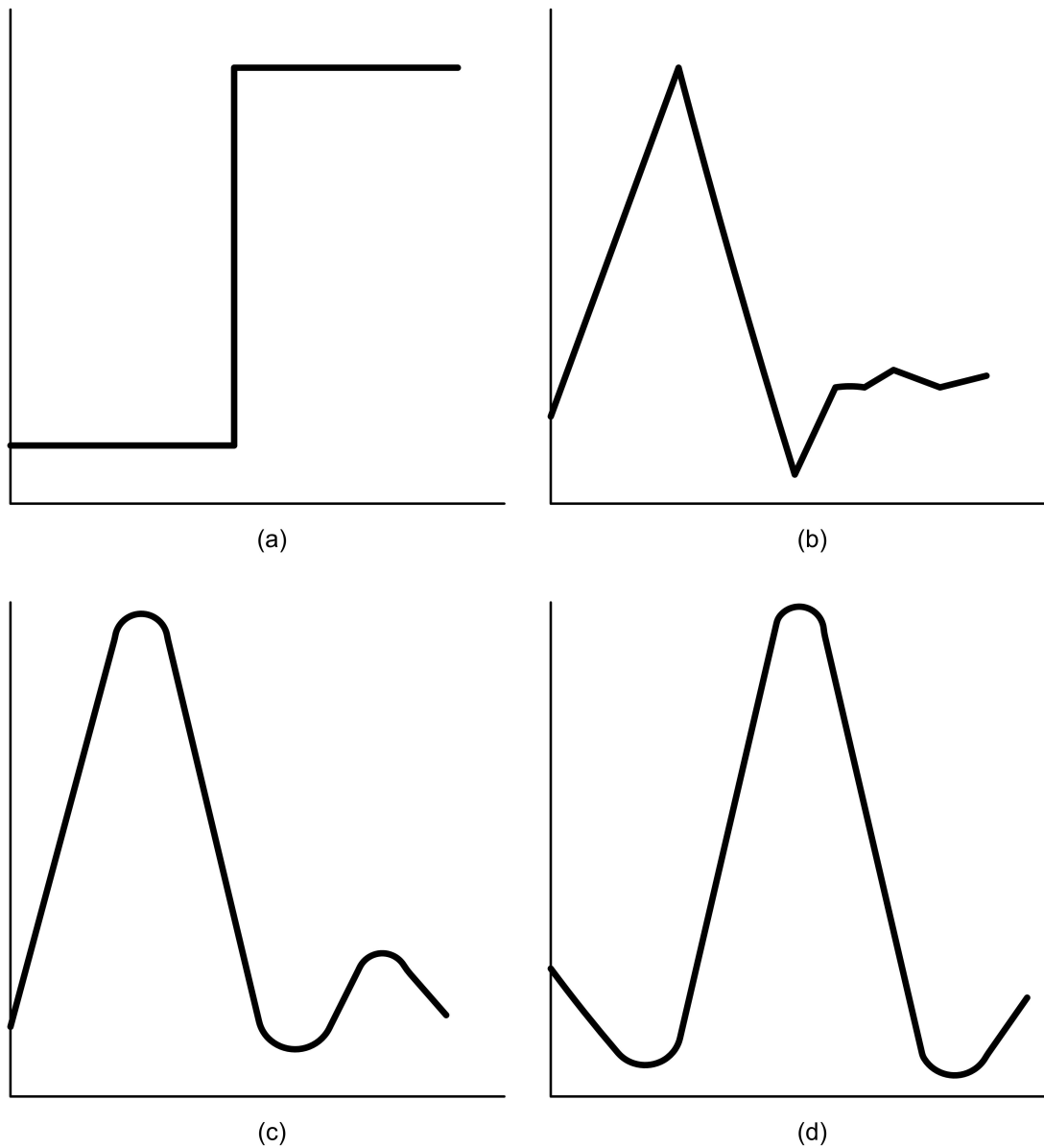


Figura 4.8: Diversas famílias de Wavelets: (a) Haar; (b) Symmlet; (c) Daubechies; e, (d) Coiflet

em picos isolados. Dessa forma, a variação da escala tende a representar cada componente de frequência adequadamente.

A função Wavelet-mãe serve como protótipo para a geração de janelas com diferentes escalas, que serão transladadas pelo sinal para a computação da TWC. Existem diversos tipos de Wavelet-mãe que podem ser utilizados em uma TWC. Cada tipo se revela mais adequado a um determinado sinal. A figura 4.8 apresenta algumas Wavelets bastante populares: Haar, Daubechies, Coiflet e Symmlet.

Dado o sinal não-estacionário mostrado anteriormente na figura 4.3, pode-se verificar na figura 4.9 como ocorre a computação da TWC. Uma vez escolhida a função Wavelet-mãe, são geradas janelas de diferentes escalas, cobrindo todo o espectro da

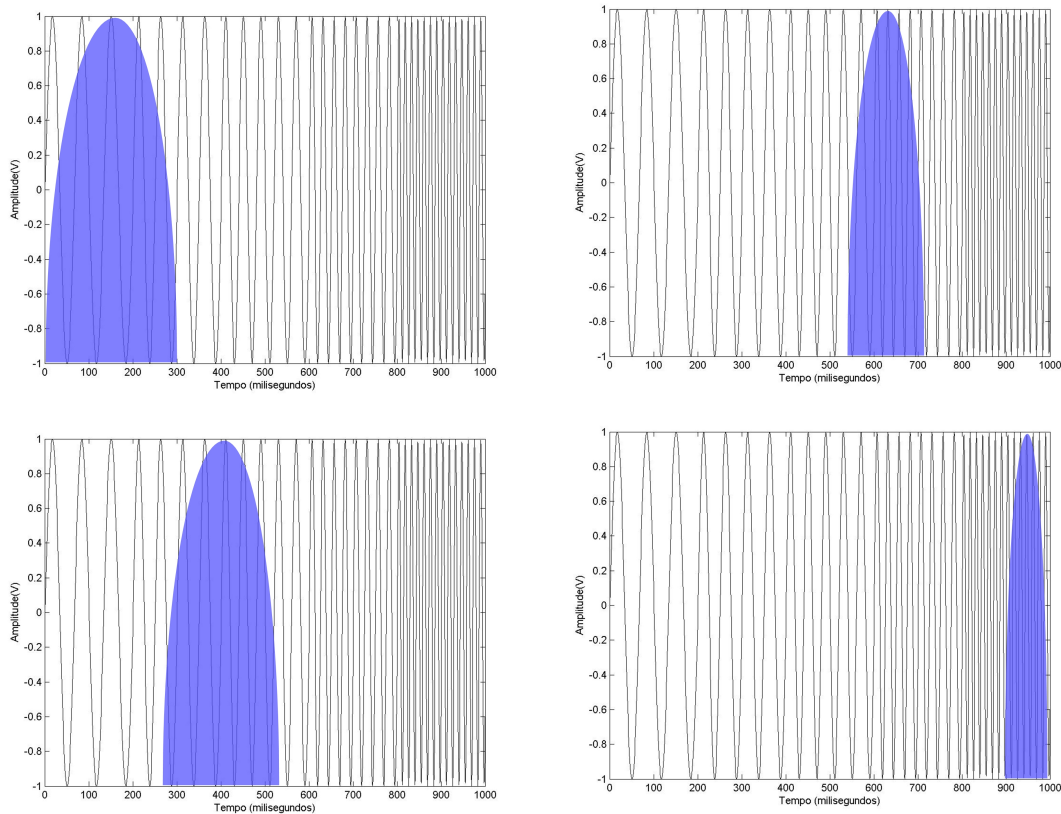


Figura 4.9: Algumas fases da computação da TWC para um sinal estacionário

freqüência do sinal. Cada janela é transladada ao longo do tempo por todo o sinal. Em cada translação, a função Wavelet é convolucionada com o sinal e integrada ao longo do tempo. Assim, é possível se obter a amplitude de energia para cada janela de freqüência em um determinado intervalo de tempo. Essa informação pode ser representada tridimensionalmente em um gráfico que relaciona translação, escala e amplitude, como é mostrado na figura 4.10.

4.3 Transformada Wavelet Discreta

O cálculo da TWC é realizado discretamente nos computadores, gerando uma alta redundância de informação, além de criar uma demanda alta de recursos computacionais de tempo e memória. Por isso, a TWD (*transformada Wavelet discreta*) (WALKER, 1999) se revela tão interessante, já que ela fornece informação suficiente para analisar e reconstruir o sinal analisado, em um menor tempo computacional.

A TWD é implementada por uma seqüência de filtragens digitais sobre o sinal original, de acordo com um algoritmo conhecido como codificação sub-banda, onde um par de filtros digitais, definidos como filtros em quadratura, é representado pelas funções

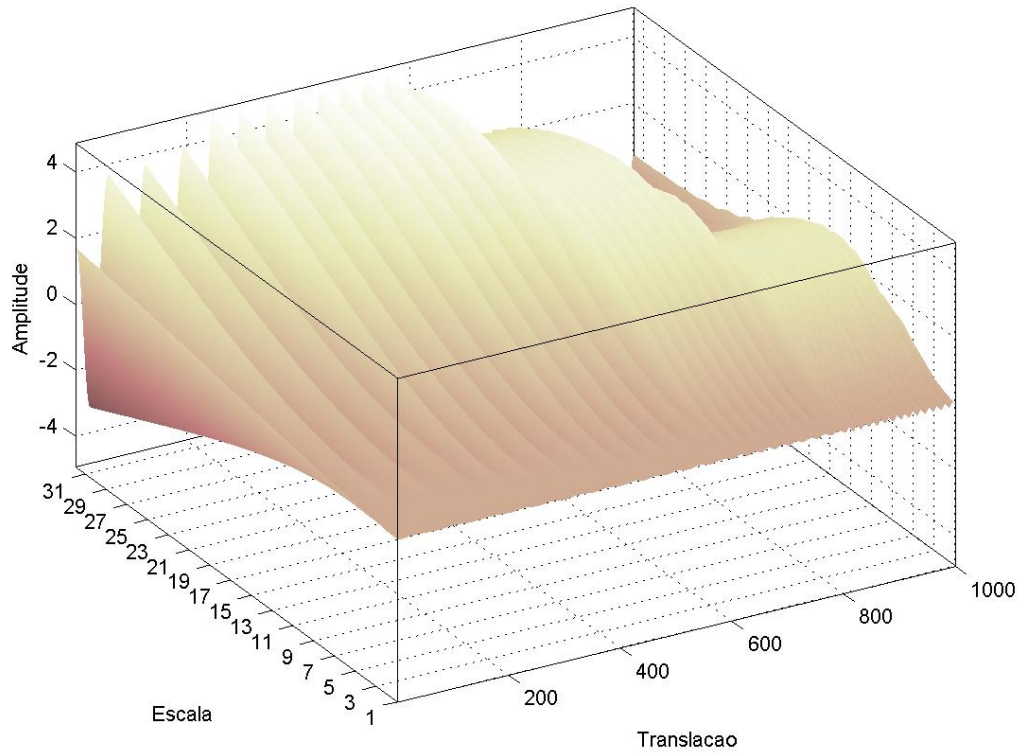


Figura 4.10: Transformada Wavelet contínua de um sinal não-estacionário

$h(n)$ e $g(n)$, que são, respectivamente, as funções resposta ao impulso de um filtro passa-baixa e passa-alta, cada um com metade da banda do sinal original (CASTELANO, 2006).

Na figura 4.11, pode-se observar a aplicação da TWD em um sinal unidimensional composto por uma senóide de 20Hz e um ruído de alta frequência.

As filtragens podem ser executadas recursivamente, como é mostrado na figura 4.12. Um sinal original $x[n]$ com 512 amostras e uma frequência de π rad/s¹ é processado

¹Embora na maior parte deste trabalho, a unidade Hz seja utilizada para expressar as frequências, é muito importante salientar que, para sinais discretizados, a unidade adequada de frequência é o rad/s.

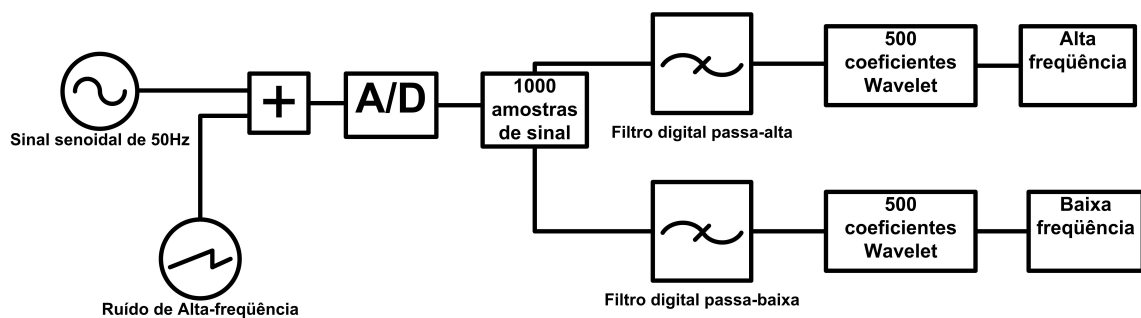


Figura 4.11: Aplicação da TWD em um sinal unidimensional composto por uma senóide de 20Hz e um ruído de alta frequência

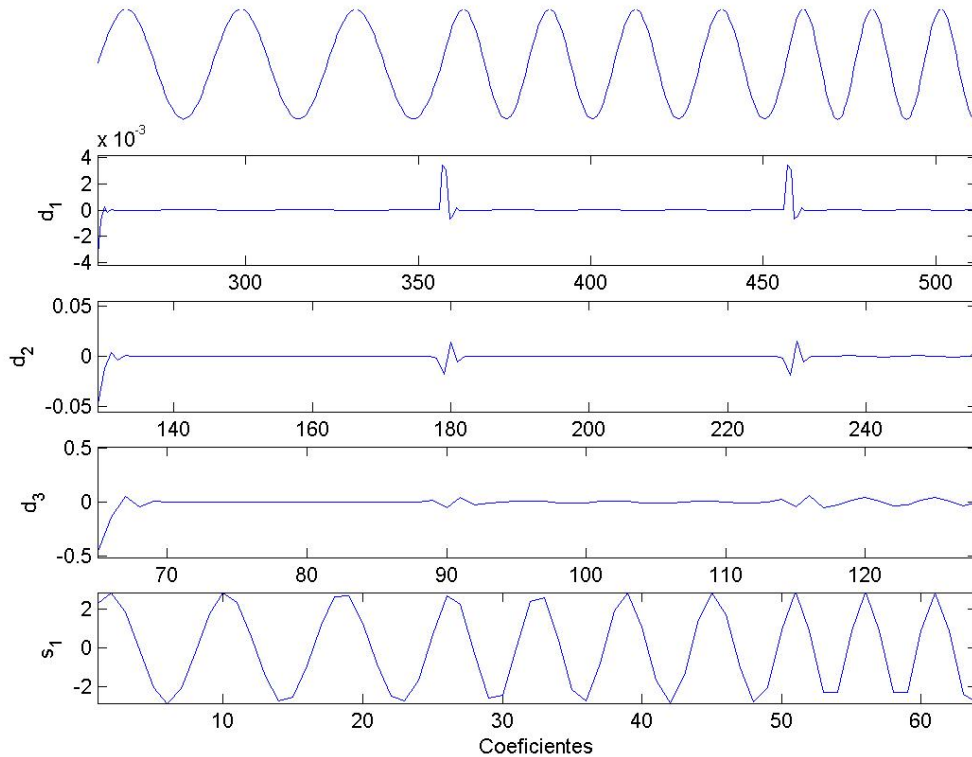


Figura 4.12: Aplicação da TWD gerando coeficientes de detalhe e aproximação

por um filtro passa-alta e um filtro passa-baixa. O primeiro filtro gera um sinal que possui frequências entre $\pi/2$ rad/s e π rad/s, sendo representado por 256 amostras². Essas amostras irão constituir os primeiros coeficientes de detalhe da TWD. O filtro passa-baixa, por sua vez, gera um sinal com também 256 amostras, contendo as frequências entre 0 rad/s e π rad/s. Esses são os coeficientes de aproximação da TWD. Deve-se notar que, no total, são mantidas as 512 amostras do sinal original. Os coeficientes de aproximação podem ser filtrados mais uma vez, gerando novos coeficientes de detalhe e aproximação. Essa operação pode ser repetida várias³ vezes, criando várias camadas de análise, com diferentes níveis de resolução.

²O Critério de Nyquist diz que um sinal analógico pode ser reproduzido sem distorções (erros) a partir de um conjunto de amostras, desde que a frequência de amostragem seja superior ao dobro da maior frequência existente no sinal contínuo original [WIKIPEDIA: Nyquist]. Como a faixa de frequências do sinal foi reduzida pela metade, torna-se possível reduzir pela metade o número de amostras.

³Em um sinal amostrado de tamanho $2n$, as operações de filtragem podem ser repetidas n vezes.

4.4 Resumo do Capítulo

Este capítulo apresentou os fundamentos conceituais das transformadas Wavelet, partindo de uma introdução sobre transformadas de Fourier. As noções apresentadas serão significativamente importantes nas próximas seções, uma vez que as transformadas Wavelet serão utilizadas para representar o sinal de ECG nos domínios do tempo e da frequência. Os sinais transformados serão utilizados na compressão.

Capítulo 5

Redes Neurais Artificiais

As redes neurais artificiais têm sua inspiração biológica originada dos neurônios típicos, formados basicamente por dendritos, o corpo celular e os axônios, que podem ter várias combinações com as mais variadas topologias. A idéia básica é sumarizada na abordagem reducionista: reproduzindo-se com suficiente detalhe a suposta “máquina” biológica responsável pelo comportamento inteligente (o cérebro), um comportamento inteligente emergirá do sistema (ALMEIDA, 2006).

5.1 Fundamentos

Os neurônios naturais são generalizados em modelos matemáticos baseados nas seguintes suposições (FAUSSET, 1994): o processamento da informação ocorre em unidades simples chamadas neurônios; sinais são transmitidos entre os neurônios através de conexões; cada conexão possui um peso associado que é multiplicado pelo sinal transmitido; e, cada neurônio aplica uma função de ativação para determinar o sinal de saída. Assim, é possível dizer que as redes neurais artificiais consistem em diversas unidades de processamento conectadas entre si através de conexões nas mais diversas topologias.

As redes neurais artificiais possuem boas aplicações onde: as regras de resolução do problema são desconhecidas ou difíceis de formalizar; dispõe-se de um grande conjunto de exemplos e suas soluções; necessita-se de grande rapidez na resolução do problema; ou, não existem soluções tecnológicas atuais. Os domínios de aplicação se espalham pelo reconhecimento de formas, tratamento de sinal, visão, fala, previsão, modelagem, auxílio à decisão e robótica (GURNEY, 1997).

Uma das etapas da utilização de redes neurais é o treinamento, onde a rede “aprende” a partir de exemplos do domínio de aplicação. Essa aprendizagem pode

ser supervisionada (onde os exemplos já possuem soluções) ou não-supervisionada. A aprendizagem supervisionada é normalmente utilizada em problemas de classificação, enquanto a aprendizagem não-supervisionada é geralmente utilizada em problemas de agrupamento.

5.2 Algoritmo de Retro-Propagação

Neste trabalho, utilizar-se-á o método de retro-propagação (“backpropagation”), que é um método de aprendizagem supervisionada. A arquitetura de uma rede neural MLP com algoritmo de retro-propagação padrão consiste de uma camada de entrada, uma ou mais camadas escondidas e uma camada de saída. Os neurônios de saída e escondidos possuem uma “polarização”, chamada “bias”. Os neurônios de cada camada se conectam a todos os neurônios das camadas adjacentes. Cada conexão possui um peso atribuído. O treinamento é composto por duas fases. Na primeira fase, os exemplos são apresentados à rede, que chega a um resultado qualquer. Na segunda fase, esse resultado é comparado ao resultado verdadeiro, calculando-se um erro. O erro é então propagado pela rede, começando-se pela camada de saída, camadas escondidas e camada de entrada. Durante essa retro-propagação do erro, os pesos das conexões são corrigidos. Essas duas fases se repetem até a rede obter uma taxa de erro aceitável.

O algoritmo de retro-propagação (LNCC, 2007)(HEATON, 2005) é apresentado abaixo:

1. Inicialização: Inicie os pesos sinápticos e os bias aleatoriamente, com valores no intervalo $[-1;1]$;
2. Apresentação dos Exemplos de Treinamento:
 - a) Treinamento “on-line”: Para cada exemplo do conjunto de treinamento, efetue os passos 3 e 4.
 - b) Treinamento “em lote”: Para cada “época” do conjunto de treinamento, efetue os passos 3 e 4.
3. Computação para Frente (Propagação): Depois de apresentado o exemplo do conjunto de treinamento $t = x(n), d(n)$, sendo $x(n)$ a entrada apresentada à rede e $d(n)$ a saída desejada, calcule o valor de ativação v_j e a saída para cada unidade da rede, da seguinte forma:

- a) $v_j = \sum_{i=1}^m w_{ji}x_i + b$, para o cálculo do valor de ativação;

b) $f(v) = \frac{1}{1+e^{-av}}$, para o cálculo da saída y da unidade k , utilizando a função sigmóide ou uma outra função, se necessário.

c) Utilize a saída das unidades de uma camada como entradas para a seguinte, até a última camada. A saída das unidades da última camada será a resposta da rede.

4. Calcule o Sinal de Erro: Fazendo a saída $y_j = O_j(n)$, será $y_j = O_j(n)$ a resposta da rede, calcule o sinal através do seguinte cálculo:

a) $e_j(n) = d_j(n) - O_j(n)$, onde $d_j(n)$ é a saída desejada com resposta para cada unidade na interação n .

b) Este sinal de erro será utilizado para computar os valores dos erros das camadas anteriores e fazer as correções necessárias nos pesos sinápticos.

5. Computação para Trás (Retro-propagação): Calcule os erros locais, δ , para cada unidade, desde a camada de saída até a de entrada. O gradiente local é definido por:

a) $\delta_j(n) = e_j(n)O_j(n)(1 - O_j(n))$ para a unidade da camada de saída;

b) $\delta_j(n) = O_j(n)(1 - O_j(n)) \sum \delta_k w_{jk}$ para as unidades das demais camadas.

Onde:

i) $O_j(1 - O_j)$ é a função de ativação diferenciada em função do argumento, i.e., valor de ativação;

ii) δ_k é o erro das unidades da camada anterior conectadas a unidade j ;

iii) w_{jk} são os pesos das conexões com a camada anterior.

c) Após o cálculo dos erros de cada unidade, calcule o ajuste dos pesos de cada conexão segundo a regra delta generalizada e atualize os pesos:

i) $\Delta w_{kj}(n+1) = \alpha w_{kj}(n) + \eta \delta_j y_j$ para o cálculo dos ajustes dos pesos.

d) Faça:

i) $w(n+1) = w(n) + \Delta w_{kj}(n)$, onde: α é a constante de momentum, quando $\alpha = 0$, esta função funciona como a regra delta comum; η é a taxa de aprendizado; δ_j é o erro da unidade; y_j é a saída produzida pela unidade j .

6. Interação: Refaça os itens 3, 4 e 5 referentes à propagação, cálculo do erro e retro-propagação, apresentando outros estímulos de entrada, até que sejam satisfeitas as condições de treinamento; as quais podem ser:

- a) O erro da rede está baixo, sendo pouco alterado durante o treinamento;
- b) O número máximo de ciclos de treinamento foi alcançado.

5.3 Redes Neurais na Compressão de Dados

Redes neurais artificiais são aplicadas em vários problemas, inclusive na compressão de dados (VERMA; BLUMESTEIN; KULKARNI, 1997). De fato, esse tipo de técnica apresenta um desempenho bastante satisfatório nesse tipo de aplicação, pois as redes são capazes de processar padrões de entrada produzindo padrões mais simples, com menos componentes (JIANG, 1996). A informação comprimida preserva grande parte das informações originais, embora se trate de um método com perdas. Existem diversas abordagens para o uso de redes neurais na compressão de dados. De forma geral, essas abordagens podem ser divididas em:

- Utilizando o algoritmo de retro-propagação;
- Utilizando aprendizagem hebbiana;
- Utilizando mapas auto-organizáveis;
- Utilizando redes preditivas.

As próximas subseções apresentam essas diferentes abordagens.

5.3.1 Abordagem utilizando o Algoritmo de Retro-Propagação

Dentre as várias abordagens descritas, o algoritmo de retro-propagação apresentou alguns dos melhores sucessos envolvendo a aplicação de redes neurais na compressão de dados (ASH, 1989) (KENUE, 1992). A figura 5.1 apresenta a estrutura de uma rede neural utilizada para compressão de dados. Ela é formada por uma camada de entrada, uma camada escondida e uma camada de saída. Os dados a serem comprimidos são apresentados à camada de entrada da rede. O algoritmo de retro-propagação é utilizado para treinar a rede, de forma que a saída seja igual à entrada. Os neurônios da camada escondida representam os dados comprimidos. Isso significa que a mesma rede faz a compressão e a expansão dos dados. A taxa de compressão é obtida através da razão neurônios de entrada por neurônios da camada escondida.

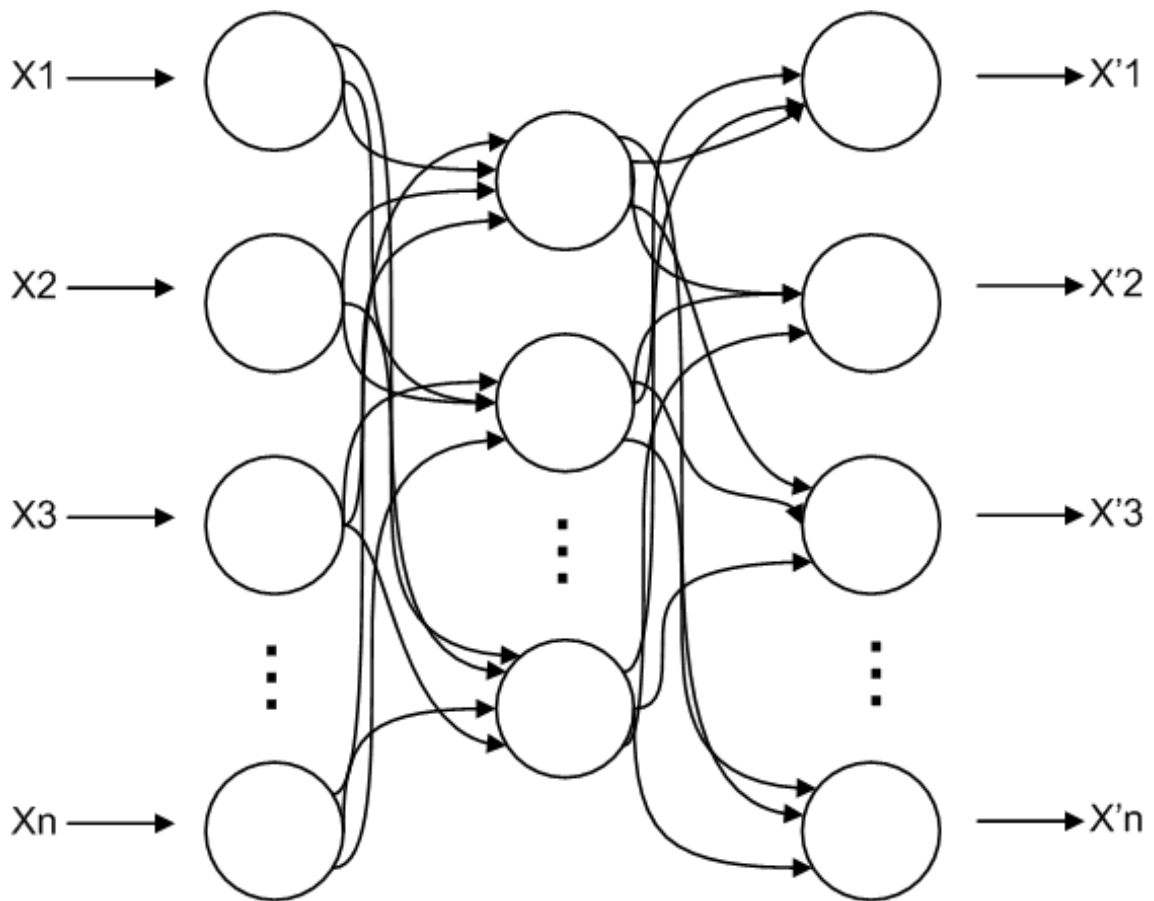


Figura 5.1: Estrutura de uma rede neural utilizada para compressão de dados, treinada com o algoritmo de retro-propagação

5.3.2 Abordagem utilizando Aprendizagem Hebbiana

É possível também utilizar a aprendizagem hebbiana para treinar redes neurais que, a partir de uma entrada de dados de alta dimensionalidade, projetem esses dados em um espaço com dimensionalidade menor, preservando o máximo das características originais. Esse tipo de extração de características também é chamado de Análise da Componente Principal, sendo muito interessante pois tenta preservar a variância dos dados originais. Nesse tipo de aplicação, a compressão é obtida pela redução da dimensionalidade do espaço de dados. Esse tipo de abordagem já foi utilizado na compressão do ECG (AL-HUKAZI; AL-NASHASH, 1996).

5.3.3 Abordagem utilizando Mapas Auto-Organizáveis

Mapas auto-organizáveis são um meio eficiente de se obter a compressão de dados. Isso é possível porque esse tipo de rede neural é capaz de mapear uma entrada em uma saída com dimensionalidade menor, preservando a mesma topologia. A figura 5.2

apresenta essa abordagem. Esta abordagem foi muito utilizada em problemas envolvendo compressão de imagens com redes neurais (BAIRAKTARIS, 1991) (VERLEYSEN, 2003).

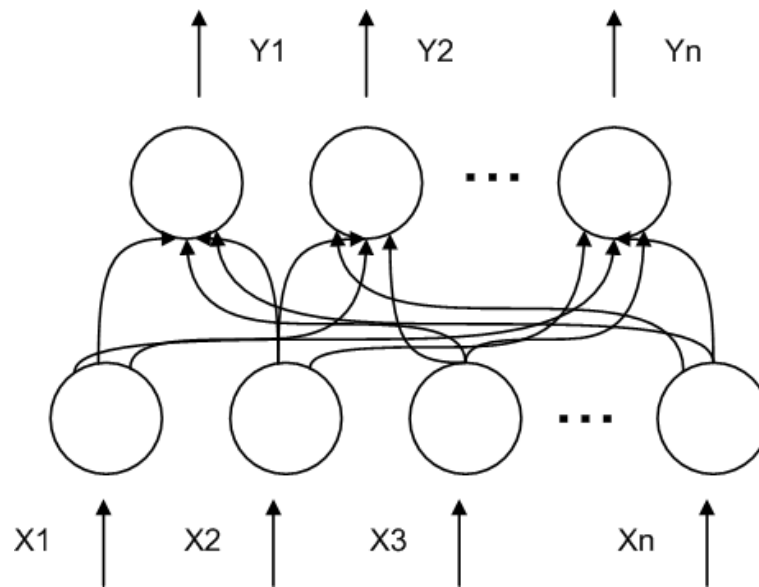


Figura 5.2: Uso de mapas auto-organizáveis na compressão de dados

5.3.4 Abordagem utilizando Redes Preditivas

Redes neurais preditivas podem ser utilizadas na compressão de dados. Nesse tipo de aplicação, a rede neural é treinada para prever as próximas amostras de um sinal a partir das amostras anteriores. A compressão é obtida pela supressão dos dados que serão previstos pela rede neural. Vários tipos de redes neurais temporais podem ser aplicadas, entre elas as redes de Jordan e as redes de Elman. A abordagem preditiva já foi utilizada na compressão do ECG (KANNAN; ESWARAN; SRIRAAM, 2002).

5.4 Resumo do Capítulo

Este capítulo apresentou alguns fundamentos das redes neurais artificiais e o algoritmo de retro-propagação, que utiliza aprendizagem supervisionada. No próximo capítulo, será demonstrado como uma rede MLP será utilizada na compressão do ECG, em conjunto com as transformadas Wavelet.

Capítulo 6

Metodologia

A partir da observação do sinal de ECG, é natural afirmar que se trata de um sinal não-estacionário composto por baixas e altas frequências. As componentes de alta frequência ocorrem em intervalos curtos de tempo, enquanto as componentes de baixa frequência tendem a ocorrer durante todo o sinal. A ocorrência dessas componentes se dá em padrões claramente visíveis, inclusive no domínio do tempo. Todos esses padrões, por sua vez, formam um conjunto finito de elementos.

A fundamentação teórica demonstrou que a transformada Wavelet discreta é capaz de representar altas escalas com uma visão global e baixas escalas com uma visão local e detalhada. Ou seja, as componentes de alta frequência são analisadas em intervalos curtos de tempo, enquanto as componentes de baixa frequência são analisadas globalmente.

Acerca da compressão de dados com redes neurais, a revisão revelou que ela pode apresentar boas taxas de compressão. Além disso, possui um bom desempenho computacional, pois é implementada como uma série de produtórios e somatórios. Finalmente, diversos trabalhos científicos demonstraram que uma transformada Wavelet discreta pode representar com satisfatório grau de fidelidade um sinal de ECG. Pode-se dizer, inclusive, que os padrões do ECG ficam mais evidentes no domínio do tempo e da frequência.

Diante de todas as evidências apresentadas, é possível inferir que uma rede neural pode, provavelmente, comprimir um sinal de ECG utilizando a transformada Wavelet como uma etapa de pré-processamento dos dados. Para provar a hipótese acima levantada, propôs-se a metodologia descrita nas seções deste capítulo.

6.1 Visão Geral

O projeto foi desenvolvido com a linguagem de programação Java. Uma visão em blocos do projeto pode ser visualizada na figura 6.1. Cada bloco do esquema representa uma classe Java. O código-fonte das classes é disponibilizado no apêndice A.

O processo se inicia com a divisão do ECG em segmentos, onde cada segmento inclui apenas um batimento cardíaco. Os segmentos são iniciados 250 ms antes do pico de cada complexo QRS. Em seguida, cada segmento é dividido ao meio, gerando dois blocos de mesmo tamanho. Os blocos são classificados como “simples” ou “complexos”, de acordo com o valor do seu desvio-padrão.

As amostras de ECG são então submetidas a uma transformada Wavelet do tipo Coiflet com 6 níveis. Os coeficientes Wavelet são processados, de forma que restam apenas os 40 maiores coeficientes, em valores absolutos. Esses coeficientes são ordenados de forma decrescente.

Os coeficientes selecionados são apresentados à etapa de compressão da rede neural MLP treinada. Existem duas redes treinadas, uma para cada tipo de bloco. Os blocos “complexos” são comprimidos a uma taxa de 5:1, enquanto os blocos “simples” sofrem uma redução de 40:1. Neste momento, tem-se o ECG comprimido, que pode ser transmitido em uma aplicação de telemedicina ou persistido em banco de dados.

A reconstrução do sinal é iniciada com a etapa de expansão das redes neurais. Em seguida, os coeficientes são processados e submetidos a uma transformada Wavelet inversa. Os blocos de ECG são reorganizados, retornando-se ao sinal de ECG, com tamanho normal. As métricas são calculadas neste momento, através da comparação entre a entrada e a saída do processo.

6.2 Escolha da Base de Dados

A MIT-BIH Arrhythmia Database (PHYSIONET, 2007) foi a base de dados utilizada nos experimentos descritos neste trabalho. Optou-se por ela por se tratar de uma base de dados “padrão-ouro”, utilizada largamente nos trabalhos científicos que envolvem compressão ou interpretação de ECG. Dessa forma, os resultados aqui obtidos puderam ser comparados diretamente com o que está relatado na literatura internacional.

Essa base foi criada a partir das pesquisas em arritmias conduzidas pelo Boston’s Beth Israel Hospital (BIH, 2007) e pelo MIT (MIT, 2007) entre os anos de 1975 e 1979, tendo sido publicada em 1980. No total, quarenta e sete pacientes foram monitorados e avaliados pelo BIH Arrhythmia Laboratory, gerando 48 horas e meia de registros de ECG.

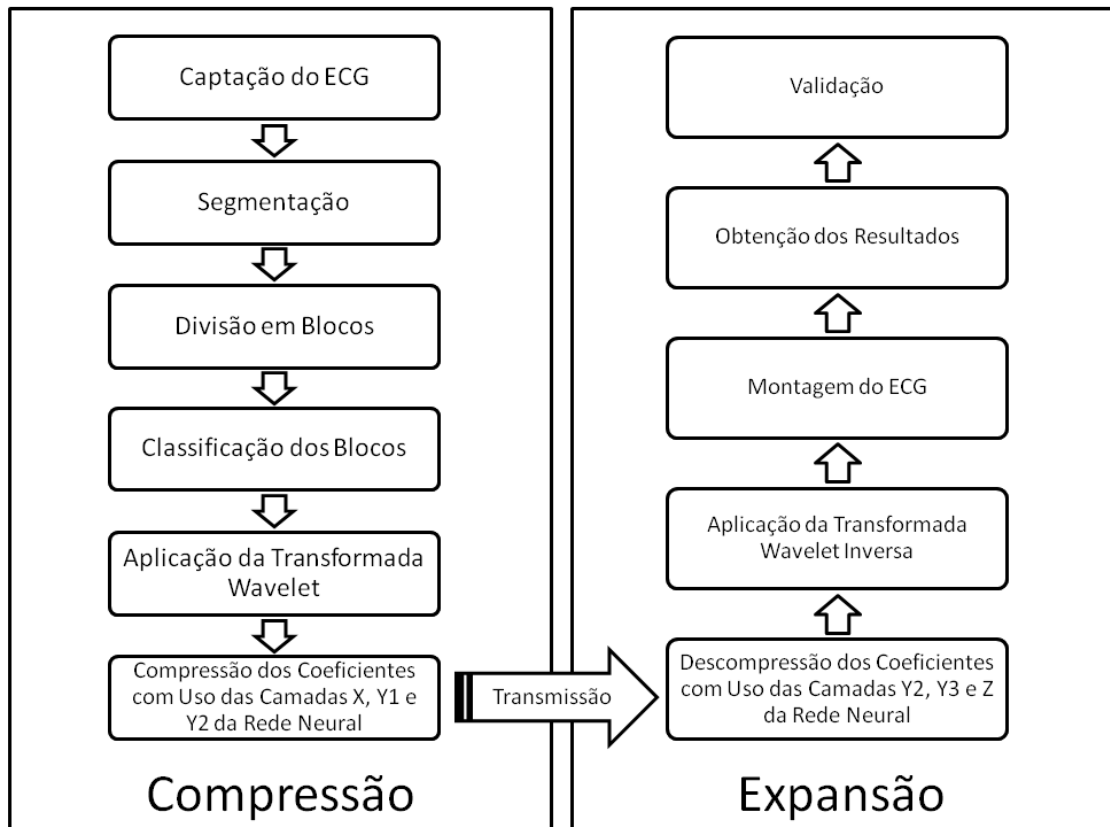


Figura 6.1: Visão em blocos do projeto

Vinte e três registros foram coletados de um conjunto de 4000 exames de 24 horas, sendo 60% de pacientes internos e 40% de pacientes externos. Os 25 registros restantes foram selecionados para representar arritmias importantes, porém pouco frequentes.

Os registros foram digitalizados com uma taxa de amostragem de 360 amostras por segundo, utilizando 11 bits para mapear 10 mV. Dois ou mais cardiologistas avaliaram independentemente cada batimento cardíaco dos exames, informando a ocorrência de uma arritmia ou não. As eventuais discordâncias foram resolvidas para gerar uma base de dados de ECG com um total de 110.000 batimentos cardíacos, onde cada batimento foi classificado como normal ou anômalo. Em caso de anormalidade, está indicado o tipo de anomalia cardíaca.

Cada registro da base de dados é formado por dois arquivos distintos. O primeiro arquivo possui a terminação “.DAT” e contém todas as amostras de ECG do registro, como pode ser visto na figura 6.2. Já o segundo arquivo possui a terminação “.ATR” e contém as ocorrências de cada batimento cardíaco, como é visto na figura 6.3.

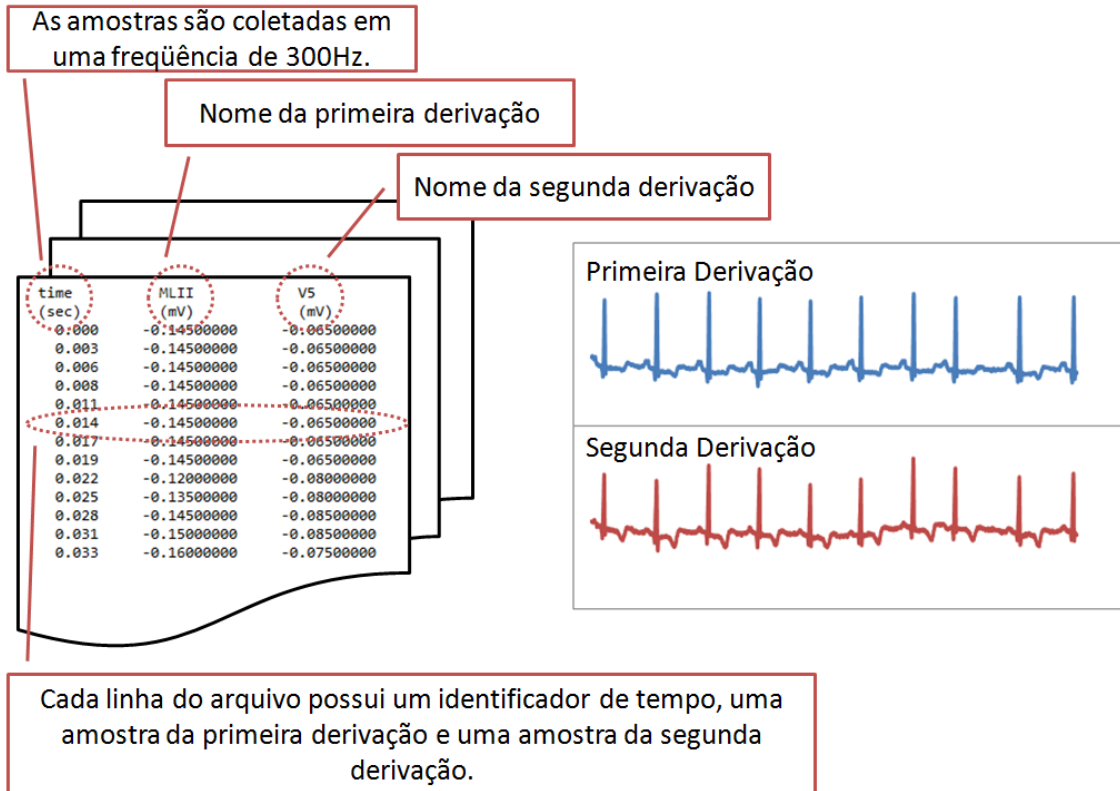


Figura 6.2: Fragmento de um arquivo “.DAT” da base de dados.

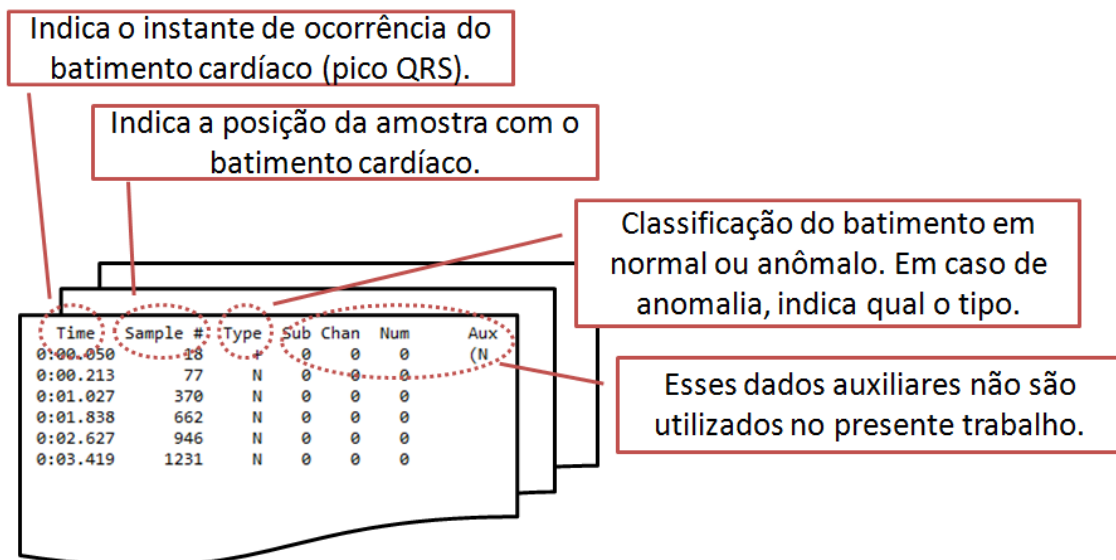


Figura 6.3: Fragmento de um arquivo “.ATR” da base de dados.

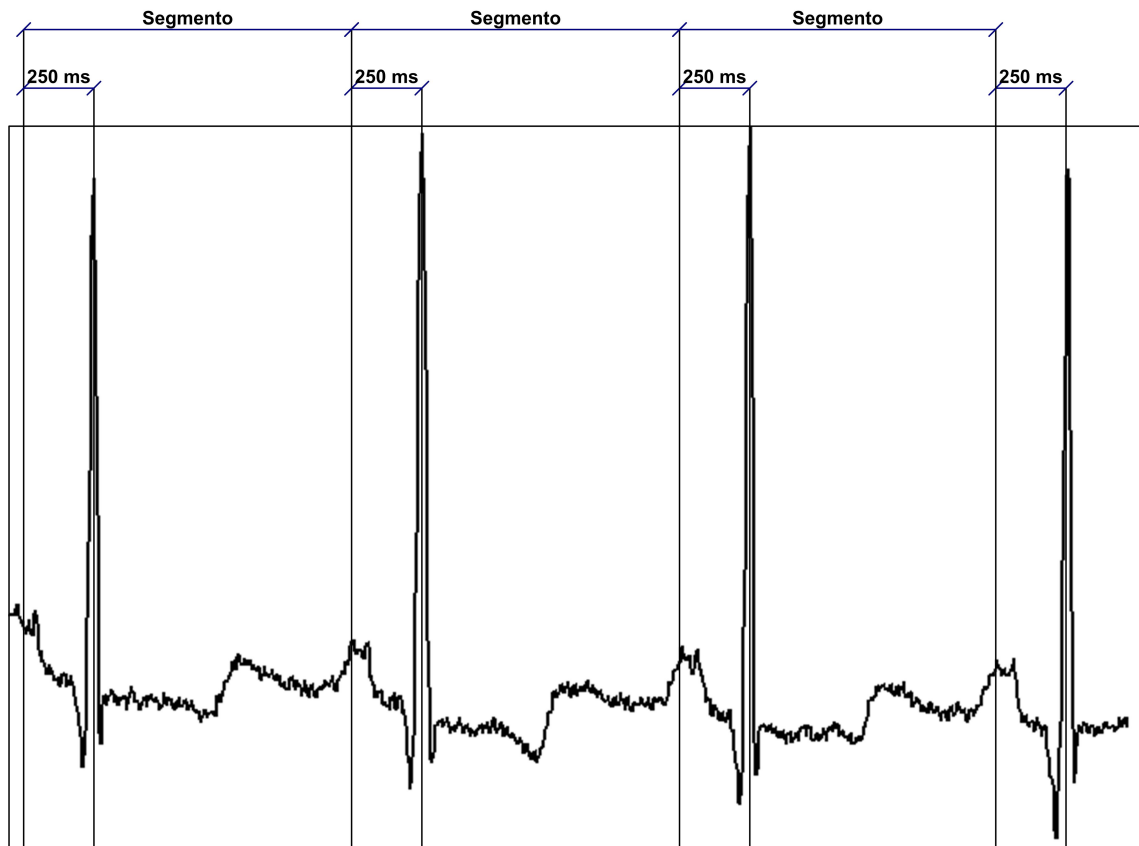


Figura 6.4: Exemplo de segmentação

6.3 Segmentação

Todos os sinais da base de dados foram segmentados durante a fase de pré-processamento. Cada segmento é iniciado 250 ms antes da detecção do pico QRS, seguindo a técnica de segmentação proposta por Alesanco e colegas (GARCÍA, 2006). O tamanho máximo do segmento é igual a 110 amostras, sendo aplicado em situações onde não seja possível detectar o pico QRS. A figura 6.4 apresenta um exemplo da fase de segmentação.

A base de dados selecionada para este trabalho já possuía as anotações indicando a ocorrência dos picos QRS, o que agilizou a etapa de pré-processamento. É importante relatar que cada derivação de ECG é segmentada independentemente, como também é recomendado por (GARCÍA, 2006).

Optou-se por essa técnica de segmentação porque ela gera, satisfatoriamente, segmentos de ECG contendo apenas 1 (um) ciclo cardíaco. Assim, a aplicação posterior de uma transformada Wavelet nos segmentos tende a gerar grupos de coeficientes que evidenciem os padrões do sinal. Esperou-se que esse comportamento reforçasse a capacidade compressora das redes neurais aplicadas em passos posteriores.

6.4 Divisão e Classificação dos Blocos

Os segmentos foram divididos em dois blocos de mesmo tamanho, como é mostrado na figura 6.5. Cada bloco foi classificado como “complexo” ou “simples”. O objetivo dessa classificação foi criar um parâmetro para definir o grau de compressão. Blocos complexos, em tese, possuem mais informação e, portanto, devem sofrer uma menor compressão. Os blocos simples, por sua vez, podem sofrer uma maior compressão, pois contém menos informações.

O desvio-padrão de cada bloco foi calculado de acordo com a equação 6.1, onde x_i representa a i -ésima amostra do sinal, N representa o total de amostras e \bar{X} representa o valor médio das amostras. Se o valor do desvio-padrão fosse maior que um valor pré-definido de corte, então o bloco era dito “complexo”. Caso contrário, ele era classificado como um bloco “simples”. Essa técnica de classificação de blocos foi descrita inicialmente por (KIM; YOO; MOON, 2006). O limite de corte utilizado neste trabalho foi determinado de forma empírica, tendo sido estabelecido o valor de 0.1.

$$Desvio - padrão = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{X})^2}{N}} \quad (6.1)$$

6.5 Normalização dos Coeficientes Wavelet

Cada bloco de dados foi codificado na forma de uma expansão Wavelet Coiflet com 6 níveis, como é mostrado na figura 6.6. A opção pelas Coiflet se deu porque estudos anteriores (GARCÍA, 2006) demonstraram que esse tipo de Wavelet possui o melhor poder de representação do sinal de ECG. Os coeficientes Wavelet foram, então, pré-processados da seguinte maneira:

1. O sinal de cada coeficiente foi removido com a aplicação de uma função “módulo”.
2. Os coeficientes foram ordenados em ordem decrescente.
3. Os 40 primeiros coeficientes¹ foram selecionados.
4. Os metadados (sinal e posição original) de cada coeficiente foram armazenados em um arquivo auxiliar.

¹Após vários experimentos realizados neste trabalho, determinou-se que esta era a quantidade de coeficientes necessária para reconstruir o sinal de forma adequada.

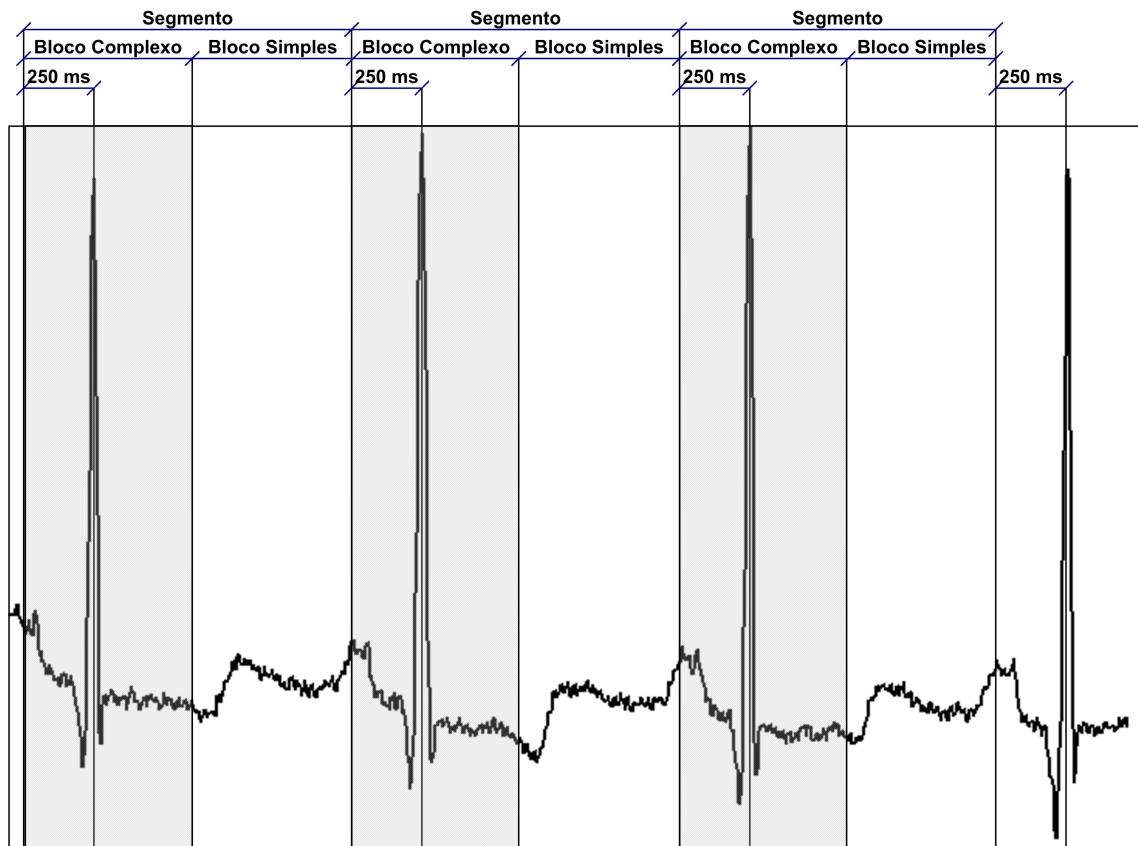


Figura 6.5: Divisão e classificação do segmento em dois blocos

6.6 Treinamento das Redes Neurais

Duas redes neurais MLP (*multilayer perceptron*) de 5 (cinco) camadas foram treinadas para comprimir e expandir os coeficientes Wavelet do sinal de ECG. A primeira rede foi treinada apenas com coeficientes “complexos”, enquanto a segunda só trabalhou com coeficientes “simples”. Utilizou-se o algoritmo de aprendizagem de retropropagação (“backpropagation”). O objetivo do treinamento foi repetir na saída os valores apresentados na entrada. A opção por esse algoritmo deve-se à sua ampla utilização na literatura científica em tarefas de compressão de dados, o que o qualifica como uma opção inicial adequada para a investigação da validade da combinação de Wavelets e redes neurais na compressão do ECG. A arquitetura utilizada nas redes pode ser vista na figura 6.7.

A fase de pré-processamento gerou, aproximadamente, 220.000 blocos de ECG, entre “simples” e “complexos”. Cerca de 70% desses blocos foram utilizados na fase de treinamento. O restante foi utilizado durante a fase de testes.

Os blocos “simples” foram comprimidos com um fator de compressão superior aos blocos “complexos”. De fato, o fator de compressão foi determinado pelo número de neurônios utilizado em cada rede. Isso pode ser visto na tabela 6.1.

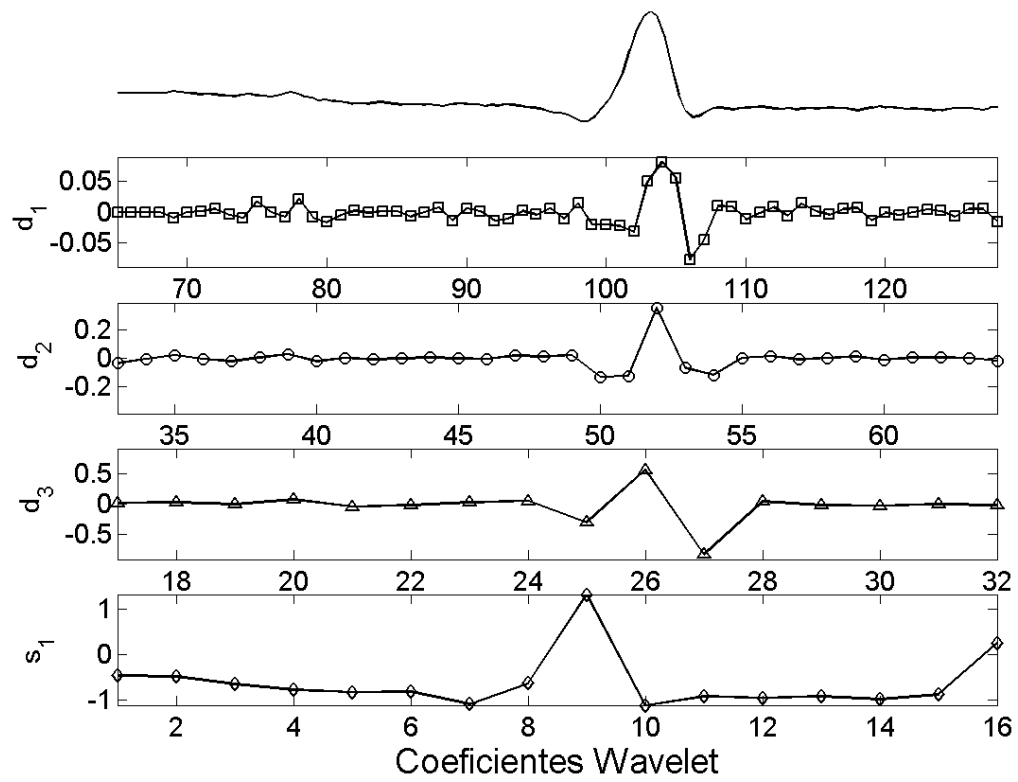


Figura 6.6: Coeficientes Wavelet de um bloco de ECG

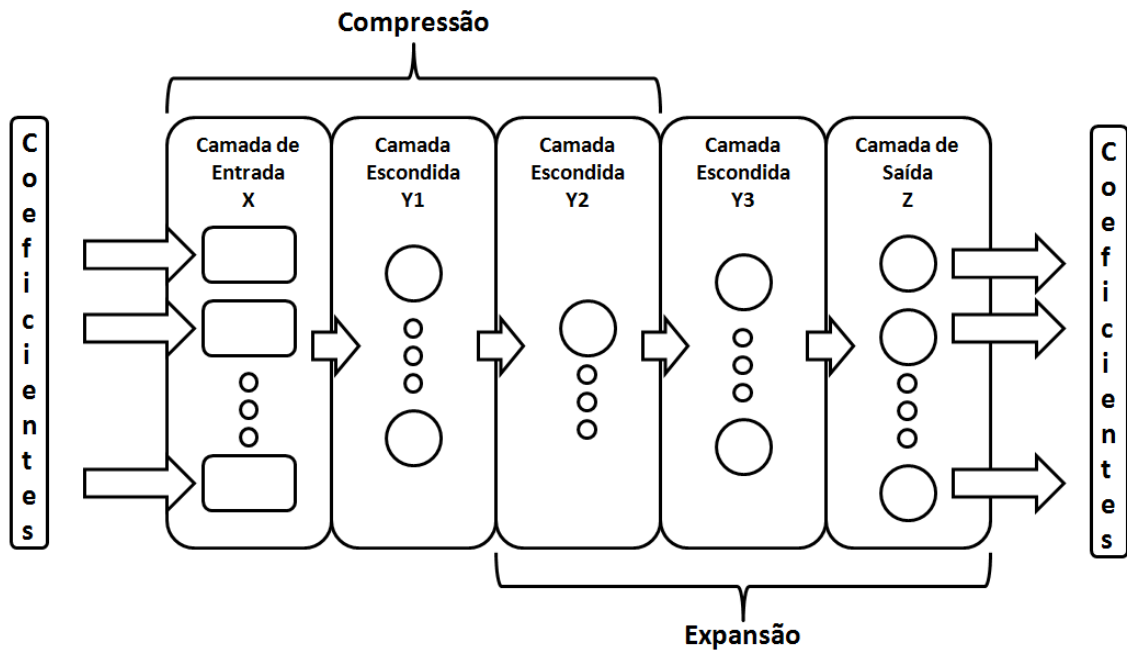


Figura 6.7: Arquitetura das redes utilizadas

Tabela 6.1: Números de neurônios utilizados nos treinamentos

Rede	Taxa de Compressão	X	Y1	Y2	Y3	Z
Blocos “simples”	40:1	40	39	1	39	40
Blocos “complexos”	5:1	40	26	8	26	40

6.7 Métricas Utilizadas

Quatro métricas foram utilizadas neste trabalho para mensurar o desempenho e a validade da metodologia proposta: TX (*taxa de compressão* - “*compression rate*”); DPRM (*diferença percentual da raiz média* - “*percent root mean squared difference*”); EM (*erro máximo* - “*maximum error*”); e, DPE (*desvio padrão dos erros* - “*standard derivation of errors*”).

A taxa de compressão é calculada através da divisão do número total de amostras do sinal original de ECG e o número total de amostras do sinal comprimido, conforme é descrito na fórmula 6.2, onde a representa a quantidade de amostras originais e b representa a quantidade de amostras comprimidas. A taxa de compressão é uma das métricas mais importantes na avaliação de uma metodologia de compressão de ECG, pois ela indica, ignorando-se os erros e as perdas, a eficiência desse método.

$$TX = a \div b \quad (6.2)$$

Existem várias formas de se calcular a diferença percentual da raiz média. Neste trabalho, optou-se pela forma “1024-offset”, que é descrita na fórmula 6.3, onde $y(n)$ representa a n ésima amostra original do sinal, $\hat{y}(n)$ representa a n ésima amostra processada do sinal e N representa o total de amostras do sinal. Essa escolha se deve ao fato dessa fórmula ser amplamente utilizada na literatura, facilitando a comparação com outros trabalhos. Essa métrica normaliza o erro médio quadrático do método. Trata-se de uma métrica global que não indica, necessariamente, o desempenho do algoritmo testado. Por exemplo, se um algoritmo de compressão corta apenas os complexos QRS do sinal, ele ainda pode apresentar um bom valor de DPRM, apesar de ser um método ruim. Mesmo não sendo uma métrica definitiva, é importante utilizá-la, pois os trabalhos comparados utilizam-na.

$$DPRM = \sqrt{\frac{\sum_{n=1}^N [y(n) - \hat{y}(n)]^2}{\sum_{n=1}^N [y(n) - 1024]^2}} \times 100\% \quad (6.3)$$

Os erros são definidos como a diferença entre o valor da amostra original e o valor da amostra reconstruída, conforme descrito na fórmula 6.4. O erro máximo é o maior valor absoluto entre todos esses erros, como é mostrado na fórmula 6.5.

$$erro(n) = y(n) - \hat{y}(n) \quad (6.4)$$

$$EM = \max_{n=1}^N |erro(n)| \quad (6.5)$$

O desvio padrão dos erros é dado pela fórmula 6.6, onde \overline{erro} é a média dos erros.

$$DPE = \sqrt{\frac{\sum_{n=1}^N [erro(n) - \overline{erro}]^2}{N - 1}} \quad (6.6)$$

6.8 Validação da Metodologia

6.8.1 Bases de Dados

Durante a validação da metodologia, foram utilizadas duas bases de dados: (1) a MIT-BIH Arrhythmia Database (PHYSIONET, 2007) que já foi apresentada neste capítulo; e, (2) a MIT-BIH ECG Compression Test Database (PHYSIONET, 2008). A MIT-BIH ECG Compression Test Database contém 68 (sessenta e oito) registros de ECG de curta duração (cada um durando 20,48 segundos) selecionados especialmente por oferecer desafios aos métodos de compressão de ECG (MOODY; MARK; GOLDBERGER, 1988).

6.8.2 Validação Automática

A validação automática da metodologia seguiu o método proposto por (AL-FAHOUM, 2006). O método de validação utiliza uma transformada Wavelet para decompor o sinal de ECG em vários níveis. Cada um desses níveis representa uma ou mais componentes do ECG (onda P, complexo QRS, onda U, segmento ST, entre outros). As diferenças entre o sinal original e processado são calculadas separadamente para cada um dos níveis. Esses valores são utilizados no cálculo de uma diferença média ponderada, onde cada nível tem um determinado grau de importância. A grande inovação desse método é criar uma forma matemática de se mensurar quão boa uma técnica é, permitindo ainda que a nota calculada seja traduzida em valores mais compreensíveis: “excelente”, “muito boa”, “boa”, “regular” e “ruim”. Isso é possível porque os autores criaram uma correlação entre os valores obtidos com os métodos e as avaliações subjetivas de especialistas humanos.

Tabela 6.2: Pesos utilizados na validação automática do método proposto

Nível de Análise Wavelet	Peso
A1	6/27
D5	9/27
D4	7/27
D3	3/27
D2	1/27
D1	1/27

Essa correlação é representada na forma de uma escala de possibilidades, apresentada na figura 6.8. Por exemplo, um valor de 17.23 está na escala de 15.45 a 25.18. Logo, isso significa que a qualidade do método de compressão tem 7% de chances de ser classificada como “muito boa” e 93% de chances de ser classificada como “boa” por um especialista humano.

No presente trabalho, os seguintes passos foram realizados:

1. Os registros de ECG da base de dados foram comprimidos de acordo com a metodologia apresentada no presente trabalho.
2. Os dados comprimidos foram novamente reconstruídos.
3. Os registros de ECG foram decompostos em seis níveis (D1, D2, D3, D4, D5 e A1) por uma transformada Wavelet, gerando coeficientes Wavelet.
4. Os registros reconstruídos de ECG também foram decompostos em seis níveis (D1, D2, D3, D4, D5 e A1) pela mesma transformada Wavelet, gerando outros coeficientes Wavelet.
5. Para cada um dos seis níveis de decomposição, calculou-se a diferença percentual da raiz média (DPRM) entre os coeficientes dos registros originais e os coeficientes dos registros reconstruídos.
6. Calculou-se a média ponderada dos valores de DPRM, utilizando os pesos da tabela 6.2.
7. O valor desse DPRM médio foi convertido em um grau de qualidade através da escala que é mostrada na figura 6.8.

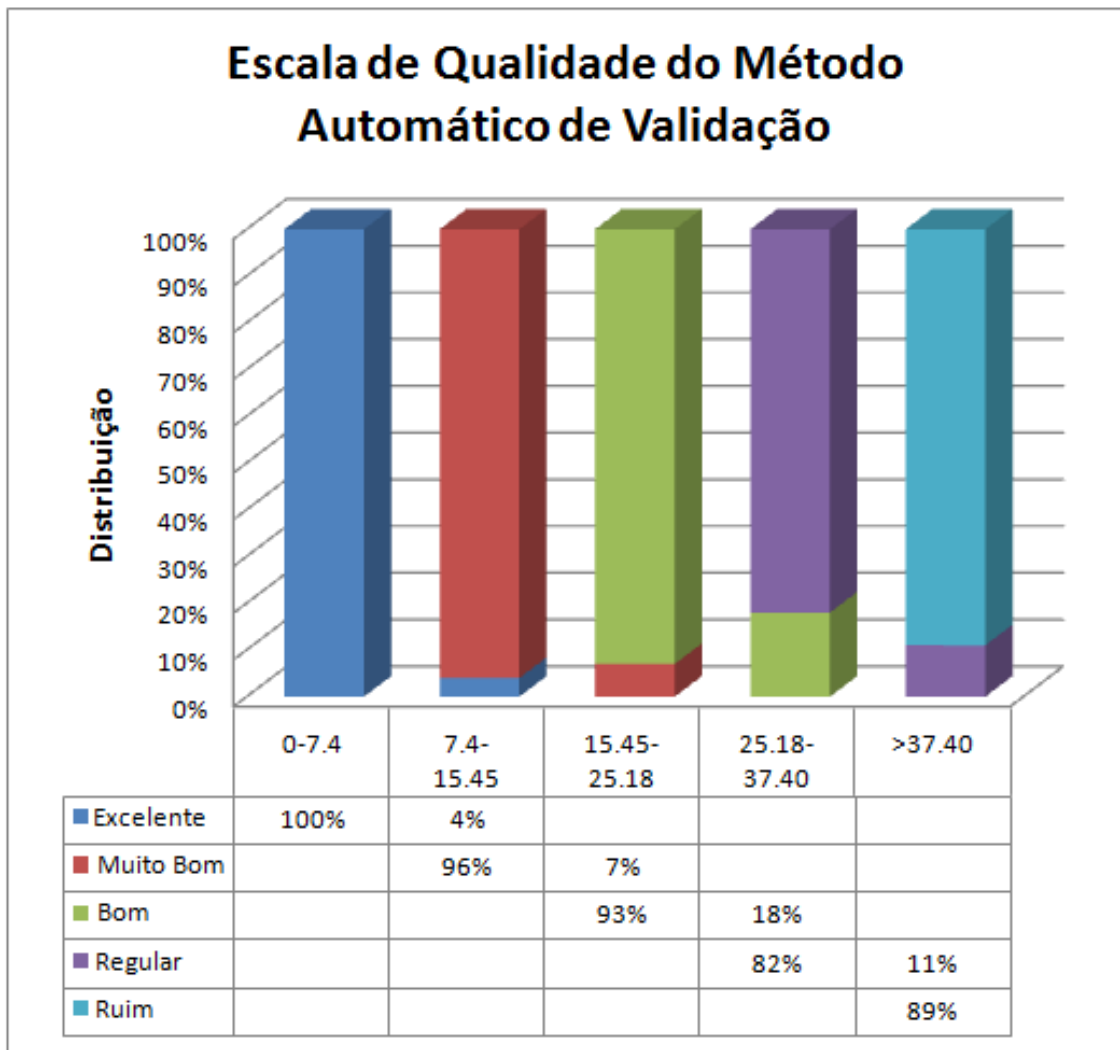


Figura 6.8: Escala de qualidade do método automático de validação

6.8.3 Validação com Especialistas

Os 68 (sessenta e oito) registros da MIT-BIH ECG Compression Test Database foram submetido ao processo de compressão e expansão descrito neste trabalho. Os registros originais e processados foram misturados e ordenados aleatoriamente. Os registros foram mostrados a especialistas através de um sistema Web, em um teste às escuras. Os especialistas foram informados que, em cada conjunto, não existiam exames repetidos. Essa orientação visava garantir que eles não fossem influenciados durante a análise. Cada registro foi analisado individualmente. Os resultados foram compilados no trabalho. Essa técnica de validação foi proposta por (MOODY; MARK; GOLDBERGER, 1988).

6.9 Resumo do Capítulo

Este capítulo apresentou a metodologia utilizada para comprovar ou rejeitar a hipótese proposta neste trabalho. Os resultados obtidos são descritos no próximo capítulo.

Capítulo 7

Resultados e Discussão

Os experimentos descritos no capítulo anterior foram realizados, tendo seus resultados descritos neste capítulo. A seção 1 apresenta vários exemplos de compressão e expansão do sistema utilizando a metodologia proposta. A seção 2 apresenta as medidas obtidas com as métricas de desempenho selecionadas, comparando o método proposto com outras abordagens do estado da arte. Por sua vez, a seção 3 apresenta os resultados obtidos com o sistema de validação automático. Por sua vez, a seção 4 apresenta o sistema Web desenvolvido para a validação com especialistas, além dos resultados obtidos com ele. Finalmente, a seção 5 apresenta uma análise mais detalhada da metodologia utilizada, comentando alguns pontos interessantes.

7.1 Exames de ECG

Para demonstrar a qualidade do método proposto, diversos exemplos de exames originais e processados são apresentados nas figuras 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7 e 7.8. Os exames mostrados nessas figuras fazem parte da MIT-BIH ECG Compression Test Database (PHYSIONET, 2008), que agrega 68 (sessenta e oito) registros de ECG que oferecem grandes desafios aos métodos de compressão de dados. É relevante observar que não existem grandes diferenças morfológicas entre o sinal original e o sinal processado pelo método de compressão proposto neste trabalho. Em alguns exames, é possível detectar um pequeno ruído na base do sinal. Contudo, quando ocorre, esse leve ruído não chega a descaracterizar o sinal. É muito importante ainda destacar a robustez apresentada pelo método de compressão. Ela pode ser verificada na capacidade do método de comprimir tanto sinais normais de ECG quanto sinais com anomalias e arritmias cardíacas.

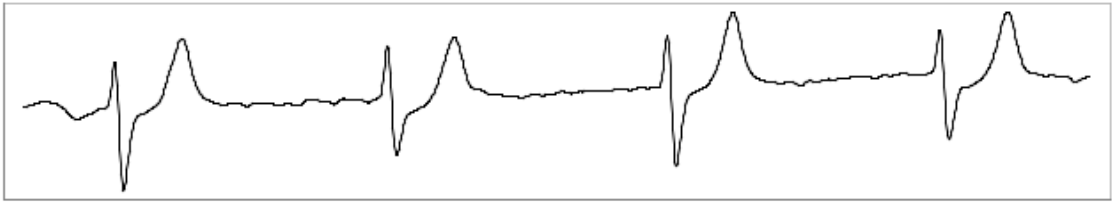


Figura 7.1: Exame 08730-01, retirado da MIT-BIH ECG Compression Test Database

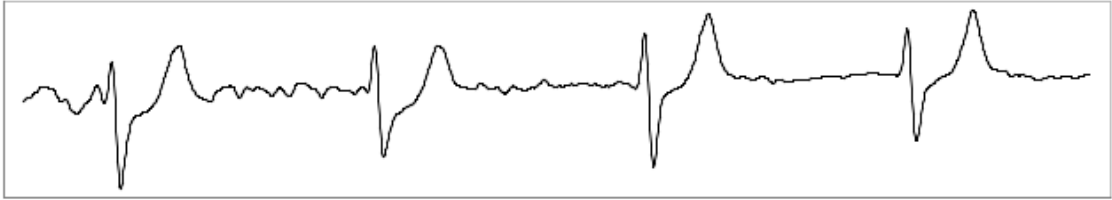


Figura 7.2: Exame 08730-01, após o processo de compressão e expansão

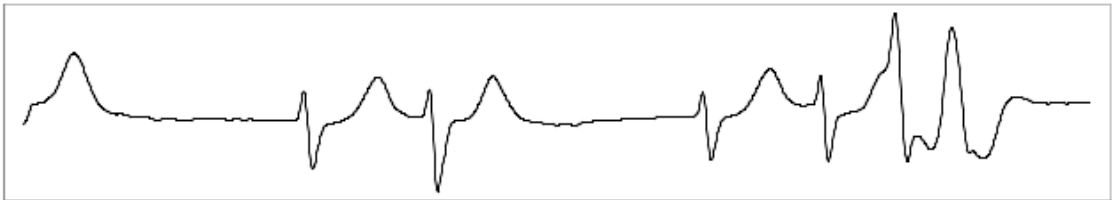


Figura 7.3: Exame 08730-02, retirado da MIT-BIH ECG Compression Test Database

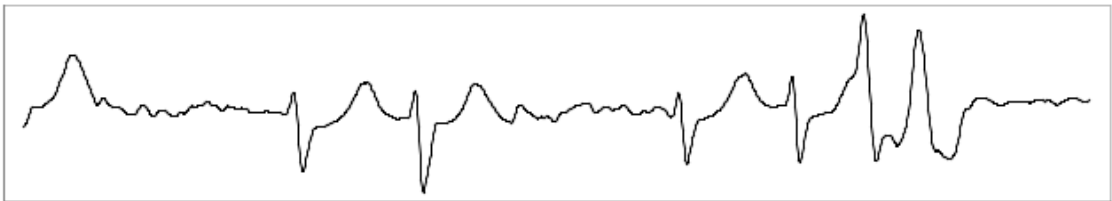


Figura 7.4: Exame 08730-02, após o processo de compressão e expansão



Figura 7.5: Exame 08730-03, retirado da MIT-BIH ECG Compression Test Database

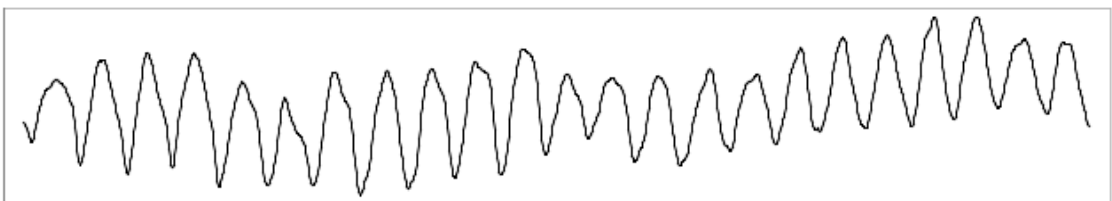


Figura 7.6: Exame 08730-03, após o processo de compressão e expansão

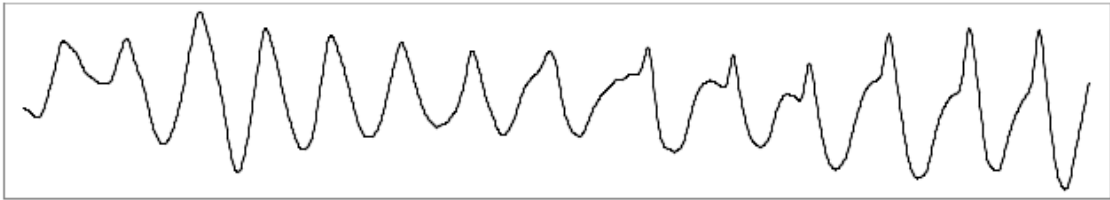


Figura 7.7: Exame 08730-04, retirado da MIT-BIH ECG Compression Test Database



Figura 7.8: Exame 08730-04, após o processo de compressão e expansão

7.2 Medidas de Desempenho

As medidas de desempenho obtidas com a metodologia são listadas nas tabelas 7.1 e 7.2. A primeira tabela apresenta os resultados para a MIT-BIH Arrhythmia Database enquanto a segunda referencia a MIT-BIH ECG Compression Test Database. Os dados obtidos por esse trabalho são confrontados com o estado da arte na tabela 7.3.

A metodologia adotada apresentou bons resultados, similares ao do estado da arte. Os blocos “simples” foram comprimidos a uma taxa de 40:1, enquanto os blocos “complexos” foram comprimidos a uma taxa de compressão de 5:1. O uso de ambos compressores levaram a uma taxa de compressão de 28:1 para a MIT-BIH Arrhythmia Database, aproximadamente. A MIT-BIH ECG Compression Test Database, por sua vez, foi comprimida em 20 vezes. O processo obteve bons resultados no quesito de qualidade, que podem ser observados nos baixos valores da diferença percentual da raiz média, do desvio padrão dos erros e do erro máximo.

7.3 Validação Automática

Os sinais originais e processados foram decompostos em seis níveis com o uso de uma transformada Wavelet, seguindo a metodologia proposta em (AL-FAHOUM, 2006). Os níveis foram comparados separadamente, com o cálculo da DPRM (*diferença percentual da raiz média - “percent root mean squared difference”*). Os níveis foram calculados com uma média ponderada. Os resultados para a MIT-BIH Arrhythmia Database são mostrados na tabela 7.4, enquanto os da MIT-BIH ECG Compression Test Database

Tabela 7.1: Medidas de desempenho para a MIT-BIH Arrhythmia Database

Métrica	Objeto	Métrica
TX	Blocos “simples”	40:1
	Blocos “complexos”	5:1
	Ambos	28:1
DPRM	Blocos “simples”	0.025%
	Blocos “complexos”	0.043%
	Ambos	0.037%
EM	Blocos “simples”	1.999
	Blocos “complexos”	1.999
	Ambos	1.999
DPE	Blocos “simples”	0.181
	Blocos “complexos”	0.307
	Ambos	0.287

Tabela 7.2: Medidas de desempenho para a MIT-BIH ECG Compression Test Database

Métrica	Objeto	Métrica
TX	Blocos “simples”	40:1
	Blocos “complexos”	5:1
	Ambos	20:1
DPRM	Blocos “simples”	0.013%
	Blocos “complexos”	0.041%
	Ambos	0.028%
EM	Blocos “simples”	1.982
	Blocos “complexos”	1.998
	Ambos	1.998
DPE	Blocos “simples”	0.105
	Blocos “complexos”	0.301
	Ambos	0.230

Tabela 7.3: Análise em paralelo com outros métodos

Autor	Registros da Base ¹	TX	DPRM
Método Proposto	Todos ²	28:1	1.85
Método Proposto	Todos ³	20:1	1.85
(CHEN; ZHANG; SHI, 2006)	104, 107, 111, 112, 115, 116, 117, 118, 119, 201, 207, 208, 209, 212, 213, 214, 228, 231, 232	20:1	6.13
		16:1	4.67
		12:1	3.46
		10:1	2.93
		8:1	2.39
(HENRIQUES et al., 2006)	Primeiros 400s de todos os registros	33.4:1	7.6
		30.7:1	7.1
		25.7:1	7.2
		22.6:1	7.8
(FILHO et al., 2006)	100, 101, 102, 103, 107, 109, 111, 115, 117, 118 e 199	20:1	3.41
		16:1	2.95
		10:1	2.14
		8:1	1.83
		4:1	1.02
(LU; KIM; PEARLMAN, 2000)	104, 107, 111, 112, 115, 116, 117, 118, 119, 201, 207, 208, 209, 212, 213, 214, 228, 231, 232	20:1	7.52
		16:1	5.46
		12:1	3.82
		10:1	2.11
		8:1	2.5
(LEE; BUCKLEY, 1999)	100	24:1	8.10
(KU et al., 2006)	Vários registros da base	20:1	6.10
(KIM; YOO; MOON, 2006)	200	24.77:1	7.32
		5.63:1	7.33
(CHOU et al., 2006)	100	24:1	4.06
(HENRIQUES et al., 2006)	Primeiros 5s de todos os registros	13.7:1	6.6
		14.4:1	4.6
		9.3:1	5.7
		8.6:1	3.9
		8.5:1	4.0
		40.2:1	7.9

¹ MIT-BIH Arrhythmia Database.² Conjunto de testes, formado por 30% dos dados de cada registro da base de dados.³ MIT-BIH ECG Compression Test Database.

Tabela 7.4: Resultados da validação automática para a MIT-BIH Arrhythmia Database

Objeto	A5	D5	D4	D3	D2	D1	Total
Blocos “simples”	72.15	48.32	13.45	17.26	17.71	13.75	19.39
Blocos “complexos”	50.78	25.45	9.37	12.97	15.63	12.82	15.29
Ambos	60.11	35.43	11.15	14.84	16.54	13.23	17.08

Tabela 7.5: Resultados da validação automática para a MIT-BIH ECG Compression Test Database

Objeto	A5	D5	D4	D3	D2	D1	Total
Blocos “simples”	68.57	36.93	14.25	12.97	13.73	21.75	18.26
Blocos “complexos”	58.76	21.12	7.39	9.10	8.89	13.28	12.05
Ambos	64.05	29.65	11.09	11.18	11.49	17.85	15.40

estão na tabela 7.5. Utilizando-se a escala apresentada no capítulo anterior (6.8), pode-se concluir que o método proposto neste trabalho apresentou um grau de qualidade variando entre “bom” e “muito bom” para a MIT-BIH Arrhythmia Database, enquanto a MIT-BIH ECG Compression Test Database foi comprimida com um grau de qualidade variando entre “muito bom” e “excelente”.

7.4 Validação com Especialistas

Toda técnica de compressão de ECG deve, necessariamente, ser avaliada por um ou mais especialistas em eletrocardiografia. Essa avaliação é muito importante, pois nunca se deve esquecer que o exame de ECG é utilizado para diagnóstico médico. Dessa forma, espera-se que um médico possa diagnosticar utilizando um ECG comprimido, caso contrário a técnica de compressão não terá validade.

Todavia, não se pode simplesmente apresentar o mesmo exame duas vezes (original e comprimido) a um médico. Esse tipo de abordagem não tem validade científica, pois induz o especialista, uma vez que ele já sabe de antemão que os dois exames deveriam ser iguais. Da mesma forma, tentar esconder essa informação do especialista é inútil, pois eles rotineiramente acabam por perceber as semelhanças entre os sinais.

Pode-se propor, então, uma estratégia onde o mesmo exame é apresentado a dois especialistas diferentes, sendo que um deles recebe o exame original e o outro recebe o exame comprimido. A princípio, poderia se esperar que ambos os diagnósticos fossem semelhantes. Contudo, essa premissa não é válida, pois o diagnóstico sobre o eletrocardiograma é uma tarefa em muitos casos subjetiva, de forma que cada especialista pode dar uma resposta diferente sobre o mesmo exame.

Diante de tal panorama, optou-se pela estratégia proposta em (MOODY; MARK; GOLDBERGER, 1988), onde os exames são apresentados a especialistas diferentes. Nenhum especialista sabe se está avaliando um exame original ou processado. No final, o importante é avaliar se os especialistas conseguem ou não utilizar os exames como base para diagnóstico. Da mesma forma, é importante analisar se os eventuais ruídos originados pela técnica são percebidos e se podem constituir em um inconveniente durante o diagnóstico.

Para aplicar essa estratégia, um sistema Web de validação foi desenvolvido com o uso das seguintes tecnologias:

- Linguagem de programação Java;
- Servidor de aplicações JBoss;
- Banco de dados HypersonicSQL;
- *Framework* Icefaces;
- *Framework* JBoss Seam.

Utilizou-se a base MIT-BIH ECG Compression Test Database (PHYSIONET, 2008), pois ela contém um conjunto de 68 exames de ECG que representam desafios para a maioria das técnicas de compressão. Esses exames foram comprimidos com a técnica proposta e então apresentados a um grupo de especialistas, utilizando-se um sistema Web. Exames originais também foram incluídos no sistema, para garantir que nenhum especialista soubesse se estava avaliando um exame comum ou processado.

Dois médicos especialistas em eletrocardiografia acessaram o sistema através da Internet, analisando uma série de 18 exames de ECG. Todos os exames foram retirados da MIT-BIH ECG Compression Test Database. Não foi comunicado a nenhum médico se o exame analisado era original ou processado. Os resultados da análise são apresentados nas tabelas 7.6 e 7.7. As figuras 7.9, 7.10 e 7.11 apresentam as telas do sistema de validação.

Os diagnósticos produzidos pelos especialistas permitem concluir que a compressão não afetou de forma significativa os exames, do ponto de vista clínico. Todos os exames comprimidos puderam ser utilizados para diagnóstico, o que indica claramente que o método não prejudica a validade clínica do exame. Em alguns exames, houve a reclamação da presença de ruído (proveniente da compressão), porém isso também não afetou o diagnóstico.

Tabela 7.6: Resultados da validação com especialistas, em números absolutos

Resultado	Original	Processado	Total
Conseguiu diagnosticar	7	5	12
Conseguiu diagnosticar, mas reclamou de ruído	1	5	6
Não conseguiu diagnosticar	0	0	0
Total	8	10	18

Tabela 7.7: Resultados da validação com especialistas, em números relativos

Resultado	Original	Processado
Conseguiu diagnosticar	87.5%	50.0%
Conseguiu diagnosticar, mas reclamou de ruído	12.5%	50.0%
Não conseguiu diagnosticar	0%	0%

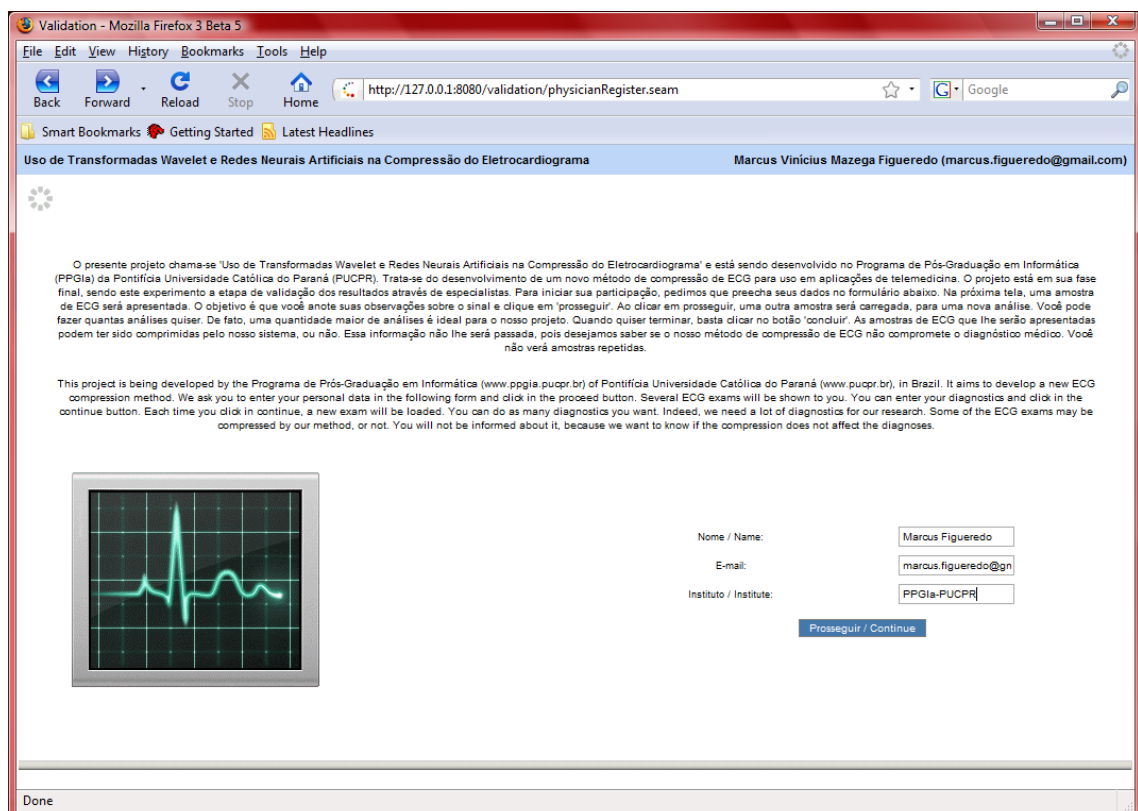


Figura 7.9: Tela de entrada do sistema Web de validação

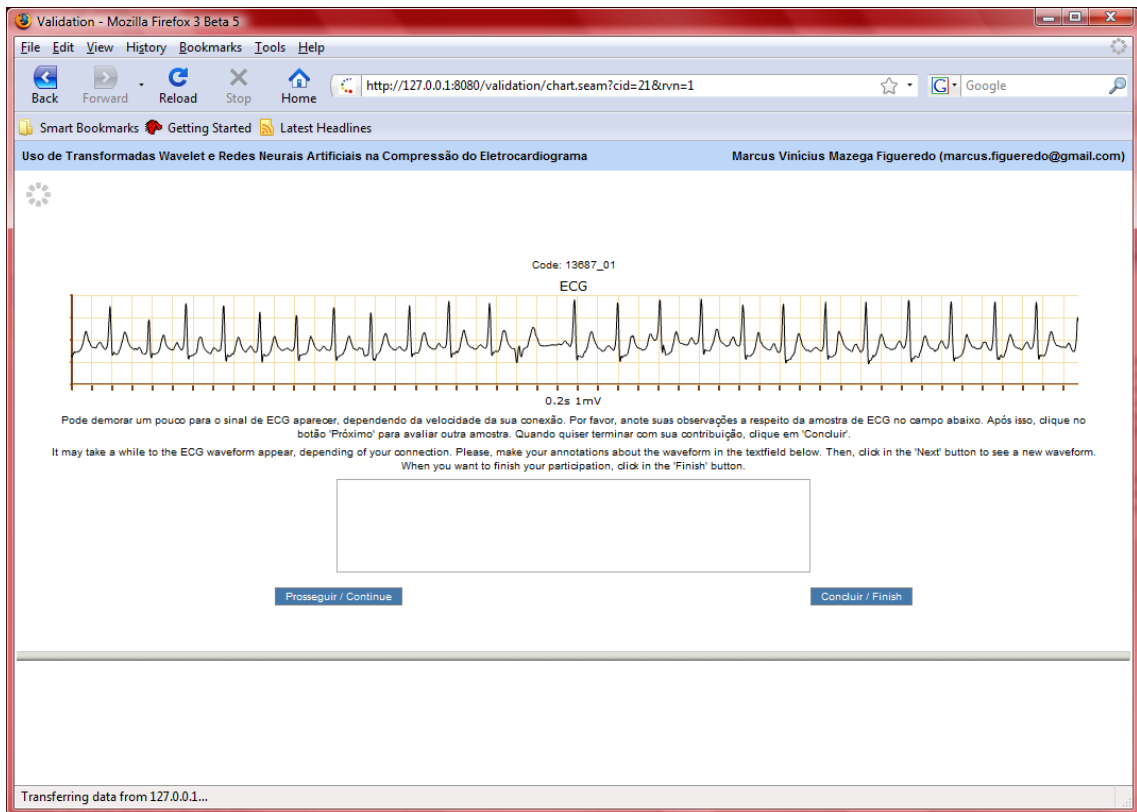


Figura 7.10: Tela de análise do ECG do sistema Web de validação

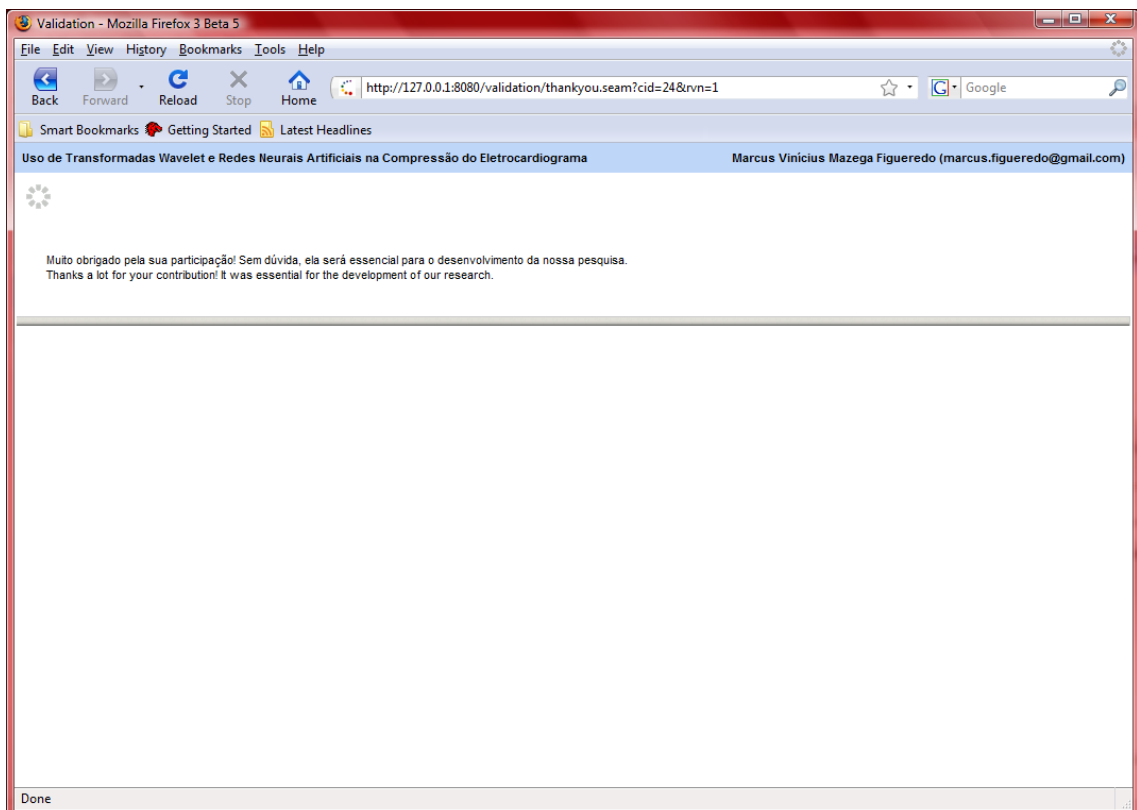


Figura 7.11: Tela de saída do sistema Web de validação

7.5 Análise da Metodologia

7.5.1 Segmentação

A etapa de segmentação é necessária na maioria dos métodos de compressão de ECG. Neste trabalho, optou-se pela estratégia de segmentação proposta em (GARCÍA, 2006). O maior impacto dessa escolha foi a garantia de um único batimento cardíaco em cada segmento. Isso reforçou os padrões do sinal, o que foi bastante positivo durante a fase de treinamento das redes neurais artificiais.

Durante a fase de validação com especialistas, houve a necessidade de se aplicar o método proposto na base de dados MIT-BIH ECG Compression Test Database. Uma vez que essa base não era anotada, não havia como aplicar a técnica de segmentação sem a execução de um detector de complexos QRS. O uso do detector, entretanto, também não apresentou bons resultados, pois a base de dados apresentava vários exames de ECG anômalos. Diante dessa situação, optou-se pelo uso de uma estratégia de segmentação diferente para a MIT-BIH ECG Compression Test Database. Usou-se, então, a técnica de segmentação de janela fixa, criando segmentos de tamanhos iguais. Os resultados demonstraram que, uma vez que as redes neurais já estejam treinada, a técnica de segmentação já não afeta o desempenho do processo. Isso é muito interessante, pois evidencia a robustez do método proposto.

7.5.2 Classificação

A classificação dos blocos de ECG em “simples” e “complexos” foi inicialmente proposta em (KIM; YOO; MOON, 2006). Essa técnica teve um grande impacto no desempenho da metodologia proposta, pois com ela foi possível aumentar a taxa de compressão sem comprometer significativamente a qualidade. Os blocos foram submetidos a diferentes taxas de compressão, de acordo com a relevância da informação contida dentro de si. Os blocos “simples” foram comprimidos em 40 vezes, enquanto os blocos “complexos” receberam uma compressão menos agressiva.

7.5.3 Transformada Wavelet

O uso da transformada Wavelet do tipo Coiflet mais uma vez provou sua utilidade na representação do ECG. Testes realizados mostraram que a transformada do tipo Daubechies também podem ser utilizada, com eficiência similar.

7.5.4 Treinamento das Redes Neurais

Muitas abordagens foram testadas para criar uma MLP capaz de comprimir coeficientes Wavelet. Inicialmente, tentou-se utilizar redes MLP com 3 camadas. Todavia, os resultados não foram positivos. Isso aconteceu em razão da grande disparidade entre a quantidade de neurônios da camada de entrada e a quantidade de neurônios da camada escondida. De fato, as primeiras redes testadas não possuíam uma quantidade suficiente de neurônios e conexões para estabelecer um processo eficiente de compressão e expansão.

O uso de redes MLP com 5 camadas, entretanto, mostrou-se bastante robusto para a compressão e expansão dos coeficientes Wavelet. As 2 camadas adicionais tornaram todo o processo mais estável.

Também é importante salientar a importância de 2 passos do pré-processamento: a remoção dos sinais dos coeficientes e a ordenação decrescente dos mesmos. De fato, no começo da pesquisa, os coeficientes não eram normalizados dessa maneira. Os sinais negativos prejudicavam a performance do treinamento, afetando a eficiência do algoritmo de aprendizagem de retropropagação. A normalização corrigiu isso, reforçando os padrões do ECG e melhorando a qualidade do treinamento.

7.6 Resumo do Capítulo

O presente capítulo apresentou todos os resultados que foram obtidos com a execução da metodologia proposta. A análise desses resultados demonstrou que a metodologia proposta atendeu de forma bastante satisfatória os objetivos e metas traçados. Verificou-se também que o método é bastante robusto e que, provavelmente, pode ser utilizado em outros campos de aplicação. Essa e outras considerações são feitas no próximo capítulo.

Capítulo 8

Conclusão

O presente trabalho levantou a seguinte hipótese: “uma rede neural pode, provavelmente, comprimir um sinal de ECG utilizando uma transformada Wavelet como uma etapa de pré-processamento dos dados”. Essa hipótese foi comprovada através de uma metodologia que tentou extrair as principais vantagens das transformadas Wavelet e das redes neurais MLP. De fato, pode-se afirmar que as Wavelets influenciaram positivamente o algoritmo de aprendizagem das redes, uma vez que o domínio tempo-freqüência é mais imune ao ruído, além de tornar mais evidentes os padrões do ECG.

Os resultados obtidos foram comparáveis aos melhores que estão descritos na literatura científica. O objetivo do presente trabalho era atingir uma taxa de compressão de 20:1, que serviria como justificativa para a importância da técnica. No entanto, a aplicação da técnica atingiu a taxa de 28:1, que superou significativamente as expectativas iniciais. Isso demonstrou que resultados ainda mais positivos podem vir de estudos futuros envolvendo redes neurais artificiais e transformadas Wavelet.

Para garantir que o método proposto tinha validade clínica, foram utilizadas duas abordagens diferentes de validação. Em ambas, os resultados foram bastante conclusivos, indicando que a metodologia não compromete o diagnóstico médico. Da mesma forma, as medidas de desempenho obtidas permitem concluir que o método obteve resultados iguais ou superiores aos melhores do estado da arte.

As várias contribuições deste trabalho podem ser resumidas da seguinte forma:

- Provou-se que redes neurais artificiais podem ser utilizadas com sucesso na compressão do eletrocardiograma, desde que os dados sejam pré-processados por uma técnica capaz de “blindar” os efeitos do ruído. Nesse caso, uma transformada Wavelet foi utilizada com elemento pré-processador.

- Chegou-se a uma abordagem capaz de igualar e/ou superar os resultados do estado da arte. Os resultados obtidos indicam que essa abordagem pode render frutos ainda melhores no futuro.
- Os bons resultados obtidos aumentam a viabilidade de aplicações de telecardiologia, uma vez que diminuem os custos com a transmissão e o armazenamento dos eletrocardiogramas. Isso se torna muito relevante, uma vez que a telemedicina é uma das soluções para os problemas atuais da saúde pública, que precisa prestar um atendimento médico de qualidade a uma parcela cada vez maior de pacientes crônicos e/ou idosos.
- A metodologia foi validada com estratégias objetivas e subjetivas, aumentando a credibilidade dos resultados obtidos, que poderão ser utilizados como referência para trabalhos futuros. Isso é muito importante porque a maioria dos trabalhos descritos da literatura não aplica técnicas de validação clínica, embora isso seja muito importante.

De uma forma simples, o caminho da ciência pode ser visto como a eterna busca da Verdade. A cada passo dado, tenta-se chegar um pouco mais próximo da Verdade. Porém, cada nova descoberta sempre leva a novas questões que também precisam ser respondidas, gerando um ciclo que nunca se conclui. O presente trabalho comprovou a hipótese proposta de que redes neurais artificiais e transformadas Wavelet podem ser utilizadas em conjunto na compressão do eletrocardiograma. Uma vez que isso foi demonstrado, surgem novas questões a serem respondidas:

- Quais outras transformadas Wavelet podem ser utilizadas?
- Quais outros tipos de transformações matemáticas podem ser aplicadas com êxito?
- Qual é o impacto da escolha da transformada na qualidade da compressão?
- Quais outros tipos de redes neurais podem ser utilizados?
- Qual é a taxa máxima de compressão que pode ser atingida seguindo a estratégia proposta?
- A mesma metodologia pode ser aplicada em quais outros campos de aplicação?

Os próximos passos da pesquisa serão dados de forma a descobrir qual é o limite máximo de compressão que pode ser obtido com essa abordagem. Além disso, é muito relevante verificar se a mesma técnica pode ser aplicada em campos como a

eletroencefalografia e a eletromiografia, que também envolvem sinais biológicos. Dessa forma, este capítulo passa a ser, além da conclusão do presente trabalho, o início de novas pesquisas.

Referências

- AL-FAHOUM, A. S. Quality assessment of ecg compression techniques using a wavelet-based diagnostic measure. *IEEE Transactions on Information Technology in Biomedicine*, v. 10, n. 1, p. 182–191, 2006.
- AL-HUKAZI, E.; AL-NASHASH, H. Ecg data compression using hebbian neural networks. *Journal of Medical Engineering & Technology*, p. 211–218, 1996.
- ALMEIDA, M. *Introdução ao Estudo das Redes Neurais Artificiais*. 2006. Disponível em: <<http://twiki.im.ufba.br/pub/MAT054/ToDoMaterial/RNA.PDF>>. Acesso em: 22 maio 2006.
- ARTEAGA, E.; GUTIERREZ, P. S. Choque em mulher de 78 anos portadora de cardiomiopatia hipertrófica e fibrilação atrial persistente. *Arquivos Brasileiros de Cardiologia*, v. 87, n. 3, 2006. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0066-782X2006001600021&tlng=en&lng=en&nrm=iso>. Acesso em: 15 abril 2007.
- ASCHENBRENNER, T. The hybrid method: Simulation of the electrical activity of the human heart. *International Journal of Bioelectromagnetism*, v. 2, n. 2, 2000. Disponível em: <http://www.rgi.tut.fi/ijbem/volume2/number2/aschenbrenner/paper_ijbem.htm>. Acesso em: 15 abril 2007.
- ASH, T. Dynamic node creation in backpropagation networks. *Connection Science*, v. 1, p. 365–375, 1989.
- AZEVEDO, D. F. *Iniciação à Eletrocardiografia*. Porto Alegre: Artmed, 1999.
- BAIRAKTARIS, S. T. V. P. M. P. D. Image data compression using a self-organizing neural network with adaptive thresholds. In: IEEE INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, 3., 1991, Seattle, Washington, USA.

Proceedings of 1991 IEEE International Joint Conference on Neural Networks. Seattle, Washington, USA: IEEE Press, 1991. p. 646–651.

BIH. *Beth Israel Hospital Inc.* 2007. Disponível em: <<http://www.bih.harvard.edu>>. Acesso em: 21 abril 2007.

BILGIN, A.; MARCELLIN, M. W.; ALTBACH, M. I. Compression of electrocardiogram signals using jpeg2000. *IEEE Transactions on Consumer Electronics*, v. 49, n. 4, p. 833–840, 2003.

BRITO, M. et al. A predictive adaptive approach to generic ecg data compression. In: IEEE, 1., 2005, Faro, Portugal. *IEEE International Symposium on Intelligent Signal Processing*. Faro, Portugal: IEEE Press, 2005. p. 32–37.

CARVALHO, M. de; SILVA, A. da; FINAMORE, W. Multidimensional signal compression using multiscale recurrent patterns. *Signal Processing: Image and Video Coding beyond Standards*, n. 2, p. 3201–3204, 2002.

CASTELANO, C. R. *Estudo Comparativo da Transforma Wavelet no Reconhecimento de Padrões da Íris Humana, Dissertação (Mestrado em Engenharia Elétrica)*. São Carlos, São Paulo: Departamento de Engenharia Elétrica, Universidade de São Paulo, 2006. 136 p.

CHEN, J.; ZHANG, Y.; SHI, X. Ecg compression based on wavelet transform and golomb coding. *Electronic Letters*, v. 42, n. 6, p. 322–324, 2006.

CHOU, H.-H. et al. An effective and efficient compression algorithm for ecg signals with irregular periods. *IEEE Transactions on Biomedical Engineering*, v. 53, n. 6, p. 1198–1205, 2006.

FAUSSET, L. *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, 1994. 461 p.

FILHO, E. B. et al. Ecg compression using multiscale recurrent patterns with period normalization. In: IEEE, 2006, Kos, Grécia. *Proceedings of the IEEE International Symposium on Circuits and Systems*. Kos, Grécia: IEEE Press, 2006. p. 4.

GARCÍA Álvaro A. S. O. R. S. I. J. Enhanced real-time ecg coder for packetized telecardiology applications. *IEEE Transactions on Information Technology in Biomedicine*, v. 10, n. 2, p. 229–236, 2006.

GOLDWASSER, G. P. *O Eletrocardiograma Orientado para o Clínico Geral*. Rio de Janeiro: Revinter, 1997.

- GOLOMB, S. W. Run-length encodings. *IEEE Transactions on Information Theory*, v. 12, n. 3, p. 399–401, 1966.
- GONZALEZ, R.; WOODS, R. *Processamento de Imagens Digital*. São Paulo: Edgar Blucher, 2000.
- GURNEY, K. *An Introduction to Neural Networks*. New York, NY, USA: Cambridge University Press, 1997. 234 p.
- GUYTON, A. C.; HALL, J. E. *Tratado de Fisiologia Médica*. Rio de Janeiro: Guanabara Koogan, 2002. 973 p.
- HEATON, J. *Introduction to Neural Networks with Java*. Chesterfield, MO, USA: Heaton Research, Inc., 2005. 380 p.
- HENRIQUES, J. et al. Searching for similarities in nearly periodic signals with application to ecg data compression. In: ICPR 2006, 2006, Hong Kong. *18th International Conference on Pattern Recognition*. Hong Kong, 2006. p. 942–945.
- JIANG, J. A neural network design for image compression and indexing. In: IASTED, 1996, Hawaii, USA. *Proceedings of IASTED International Conference: Artificial Intelligence, Expert Systems & Neural Networks*. Hawaii, USA, 1996. p. 296–299.
- KANNAN, R.; ESWARAN, C.; SRIRAM, N. Neural networks methods for ecg data compression. In: ICONIP 02, 2002. *Proceedings of the 9th International Conference on Neural Information Processing*. Cingapura, 2002. p. 2317–2319.
- KENUE, S. Modified back-propagation neural network with applications to image compression. *Applications of Artificial Neural Networks*, 1992.
- KIM, B.; YOO, S. K.; MOON, H. Wavelet-based low-delay ecg compression algorithm for continuous ecg transmission. *IEEE Transactions on Information Technology in Biomedicine*, v. 10, n. 1, p. 77–83, 2006.
- KU, C.-T. et al. A novel ecg data compression method based on nonrecursive discrete periodized wavelet transform. *IEEE Transactions on Biomedical Engineering*, v. 53, n. 12, p. 2577–2583, 2006.
- LEE, H.; BUCKLEY, K. M. Data compression using cut and align beats approach and 2-d transforms. *IEEE Transactions on Biomedical Engineering*, v. 46, n. 5, p. 556–564, 1999.

LNCC. *Algoritmo Backpropagation*. 2007. Disponível em: <<http://www.lncc.br/~labinfo/tutorialRN/>>. Acesso em: 15 maio 2007.

LU, Z.; KIM, D. Y.; PEARLMAN, W. Wavelet compression of ecg signals by the set partitioning in hierarchical trees (spihit) algorithm. *IEEE Transactions on Biomedical Engineering*, v. 47, n. 7, 2000.

MIRADOR. Fourier. In: *Enciclopédia Mirador*. São Paulo: Encyclopaedia Britannica do Brasil Publicações Ltda, 1993. v. 9.

MIRADOR. Heisenberg. In: *Enciclopédia Mirador*. São Paulo: Encyclopaedia Britannica do Brasil Publicações Ltda, 1993. v. 11.

MIT. *Massachusetts Institute of Technology*. 2007. Disponível em: <<http://www.mit.edu>>. Acesso em: 21 abril 2007.

MOODY, G. B.; MARK, R. G.; GOLDBERGER, A. L. Evaluation of the “trim” ecg data compressor. In: COMPUTERS IN CARDIOLOGY, 1988. *Proceedings of Computers in Cardiology 1988*. Washington, DC, USA: IEEE Computer Society Press, 1988. p. 167–170.

NOBEL. *Willem Einthoven: Biography*. 2007. Disponível em: <http://nobelprize.org/nobel_prizes/medicine/laureates/1924/einthoven-bio.html>. Acesso em: 29 mar. 2007.

PHYSIONET. *MIT-BIH Arrhythmia Database*. 2007. Disponível em: <<http://www.physionet.org/physiobank/database/mitdb>>. Acesso em: 20 abril 2007.

PHYSIONET. *The MIT-BIH ECG Compression Test Database*. 2008. Disponível em: <<http://www.physionet.org/physiobank/database/mitdb>>. Acesso em: 21 janeiro 2008.

SEDGEWICK, M. et al. Performance of an established system of first responder out-of-hospital defibrillation: the results of the second year of the heartstart soctland project in the utslein style. *Resuscitation*, v. 6, p. 75–78, 1993.

SOBREIRA, E. A. A. P. M. On the use of 2-d coding techniques for ecg signals. *IEEE Transactions on Information Technology in Biomedicine*, v. 10, n. 4, p. 809–811, 2006.

SWELDENS, W.; SCHRÖDER, P. *Building Your Own Wavelets at Home*. 2007. Disponível em: <<http://cm.bell-labs.com/who/wim/papers/athome/athome.pdf>>. Acesso em: 10 fevereiro 2007.

UNICAMP, D. de Engenharia Biomédica da. *Eletrocardiografia*. 2007. Disponível em: <<http://www.fee.unicamp.br/deb/ecg1.pdf>>. Acesso em: 29 março 2007.

VERLEYSEN, C. A. J. L. M. Image compression using self-organizing maps. *Special issue: Digital signal processing and control*, v. 43, n. 11, p. 1529–1543, 2003.

VERMA, B.; BLUMESTEIN, M.; KULKARNI, S. A neural network based technique for data compression. *Journal of Intelligent Systems*, v. 9, 1997.

WALKER, J. S. *A Primer on Wavelets and Their Scientific Applications*. Florida, USA: Chapman & Hall, 1999. 168 p.

ZHAO, Y. et al. Applying incompletely connected feedforward neural network to ambulatory ecg data compression. *Electronic Letters*, v. 33, n. 6, p. 220–221, 1997.

Apêndice A

Código-Fonte

O projeto foi desenvolvido com a linguagem de programação Java. O código fonte é disponibilizado a seguir.

A.1 Segmentação

Optou-se pela técnica de segmentação proposta por Alesanco e colegas (GARCÍA, 2006), onde cada segmento do ECG é iniciado 250 ms antes do pico do complexo QRS. Uma vez que a base de dados MIT-BIH Arrhythmia Database é anotada, o algoritmo se aproveita das marcações de batimentos cardíacos, que são incluídas em arquivos acessórios.

```
package br.pucpr.ppgia.figueredo;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.PrintWriter;
import java.util.StringTokenizer;

public class SegmentationProcessor extends Processor {

    @Override
    public void process() throws Exception {
        // acessa os canais de entrada
        InputStream inAtr = getInput(FileTypes.atr.toString());
        BufferedReader readerAtr = new BufferedReader(new InputStreamReader(inAtr));
```

```
InputStream inDat = getInput(FileTypes.dat.toString());
BufferedReader readerDat = new BufferedReader(new InputStreamReader(inDat));

// acessa os canais de saída
OutputStream out = getOutput(FileTypes.seg.toString());
PrintWriter writer = new PrintWriter(out);

// ignora as linhas de cabeçalho
readerAtr.readLine();readerAtr.readLine();
readerDat.readLine();readerDat.readLine();

int beginSample = -1;
int endSample = -1;
int cutSample = -1;
String type;

// processa os dados
while(readerAtr.ready()) {
    // le uma linha de dados
    String lineAtr = readerAtr.readLine();
    StringTokenizer tokAtr = new StringTokenizer(lineAtr);
    // ignora o timestamp
    tokAtr.nextToken();
    // descobre a posição da última amostra
    endSample = Integer.parseInt(tokAtr.nextToken());
    // descobre o tipo de batimento (normal, arritmia, fibrilação etc)
    type = tokAtr.nextToken();

    if(beginSample<0) {
        beginSample = endSample;
        continue;
    }

    // corta 250 ms antes do pico
    // o período é de 3 ms
    // 250/3 =~ 83
```

```

cutSample = endSample-beginSample-83;

writer.println(cutSample + " " + type + " ");
for(int i=0; i<cutSample; i++) {
// le a amostra
String lineDat = readerDat.readLine();
if(lineDat == null)
continue;
StringTokenizer tokDat = new StringTokenizer(lineDat);
// ignora o timestamp
tokDat.nextToken();
// salva as amostras do segmento
writer.print(tokDat.nextToken() + " ");
writer.flush();
}
writer.println();
writer.flush();

beginSample = endSample;
}
}
}

```

A.2 Classificação

Cada segmento foi dividido em dois blocos, que foram classificados como “complexos” ou “simples”. O critério para a classificação foi o valor do desvio-padrão de cada bloco. Blocos com valores elevados foram classificados como “complexos”, enquanto os outros foram denominados “simples”. Essa técnica foi apresentada em (KIM; YOO; MOON, 2006).

```

package br.pucpr.ppgia.figueredo;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;

```

```
import java.io.OutputStream;
import java.io.PrintWriter;
import java.util.StringTokenizer;

import org.apache.commons.math.stat.descriptive.DescriptiveStatistics;

public class BlocksClassificationProcessor extends Processor {

    // limite de corte
    private static final double THRESHOLD = 0.1d;

    @Override
    public void process() throws Exception {
        // acessa os canais de entrada
        InputStream in = getInput(FileTypes.seg.toString());
        BufferedReader reader = new BufferedReader(new InputStreamReader(in));

        // acessa os canais de saída
        OutputStream outSbl = getOutput(FileTypes.sbl.toString());
        PrintWriter writerSbl = new PrintWriter(outSbl);
        OutputStream outCbl = getOutput(FileTypes.cbl.toString());
        PrintWriter writerCbl = new PrintWriter(outCbl);
        OutputStream outClas = getOutput(FileTypes.clas.toString());
        PrintWriter writerClas = null;
        if(outClas != null) {
            writerClas = new PrintWriter(outClas);
        }

        // processa os dados
        while (reader.ready()) {
            // lê o cabeçalho
            String header = reader.readLine();
            if (header == null)
                return;
            if ("".equals(header.trim()))
                return;
        }
    }
}
```

```

// separa o cabeçalho e descobre quantas amostras o segmento possui
// assim, pode-se encontrar a metade do segmento
StringTokenizer tokHeader = new StringTokenizer(header);
int size = Integer.parseInt(tokHeader.nextToken().trim());
int middle = size / 2;
// descobre o tipo de segmento (normal, arritmia etc)
String classification = tokHeader.nextToken();

    // acessa as amostras do segmento
String data = reader.readLine();
StringTokenizer tokData = new StringTokenizer(data);

if (size < 0)
continue;

// transforma todos os dados em doubles
double[] segment = new double[size];
for (int i = 0; i < size; i++) {
if(!tokData.hasMoreTokens()) {
break;
}
segment[i] = Double.parseDouble(tokData.nextToken());
}

// calcula o desvio-padrão da primeira metade
DescriptiveStatistics stats = DescriptiveStatistics.newInstance();
for (int i = 0; i < middle; i++) {
stats.addValue(segment[i]);
}
double dp = stats.getStandardDeviation();

// joga os blocos na saída, classificando como “complexo” ou simples
if(dp < THRESHOLD) {
writerSbl.println(middle + " " + classification + " " + dp);
for (int i = 0; i < middle; i++) {

```



```

writerSbl.print(segment[i]);
writerSbl.print(" ");
}
writerSbl.println();
writerSbl.flush();

if(writerClas != null) {
writerClas.println("s");
writerClas.flush();
}
} else {
writerCbl.println(middle + " " + classification + " " + dp);
for (int i = 0; i < middle; i++) {
writerCbl.print(segment[i]);
writerCbl.print(" ");
}
writerCbl.println();
writerCbl.flush();

if(writerClas != null) {
writerClas.println("c");
writerClas.flush();
}
}

// calcula o desvio-padrão da segunda metade
stats = DescriptiveStatistics.newInstance();
for (int i = middle; i < size; i++) {
stats.addValue(segment[i]);
}
dp = stats.getStandardDeviation();

// joga os dados no output
if(dp < THRESHOLD) {
writerSbl.println((size - middle) + " " + classification + " " + dp);
for (int i = middle; i < size; i++) {

```



```

import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.PrintWriter;
import java.util.StringTokenizer;

public class WaveletsProcessor extends Processor {

    @Override
    public void process() throws Exception {
        // acessa os canais de entrada
        InputStream inSbl = getInput(FileTypes.sbl.toString());
        BufferedReader readerSbl = new BufferedReader(new InputStreamReader(inSbl));
        InputStream inCbl = getInput(FileTypes.cbl.toString());
        BufferedReader readerCbl = new BufferedReader(new InputStreamReader(inCbl));

        // acessa os canais de saída
        OutputStream outSwc = getOutput(FileTypes.swc.toString());
        PrintWriter writerSwc = new PrintWriter(outSwc);
        OutputStream outCwc = getOutput(FileTypes.cwc.toString());
        PrintWriter writerCwc = new PrintWriter(outCwc);

        // utiliza uma transformada Wavelet Coiflet com 6 níveis
        Wavelet wavelet = new Wavelet(Wavelet.c6);

        // processa os blocos ‘‘simples’’
        while(readerSbl.ready()) {
            try {
                // lê as amostras e gera um array de doubles
                String header = readerSbl.readLine();
                StringTokenizer tokHeader = new StringTokenizer(header);
                int length = Integer.parseInt(tokHeader.nextToken().trim());
                String lineSamples = readerSbl.readLine();
                StringTokenizer tokSamples = new StringTokenizer(lineSamples);
                double[] samples = new double[length];
                for(int i=0; i<length; i++) {

```

```

samples[i] = Double.parseDouble(tokSamples.nextToken().trim());
}
// executa a transformada Wavelet
Wavelet timeDomain = new Wavelet(samples);
timeDomain.normalize();
Wavelet timefrequencyDomain = Wavelet.dwt(timeDomain, wavelet, 2);
timefrequencyDomain.normalize();
// grava o resultado na saída
writerSwc.print(timefrequencyDomain.length() + " ");
writerSwc.println(header);
for(int i=0; i<timefrequencyDomain.length(); i++) {
writerSwc.print(timefrequencyDomain.valueAt(i) + " ");
}
writerSwc.println();
writerSwc.flush();
} catch(Exception e1) {
e1.printStackTrace();
}
}

// processa os blocos ‘complexos’
while(readerCbl.ready()) {
try {
// lê as amostras e gera um array de doubles
String header = readerCbl.readLine();
StringTokenizer tokHeader = new StringTokenizer(header);
int length = Integer.parseInt(tokHeader.nextToken().trim());
String lineSamples = readerCbl.readLine();
StringTokenizer tokSamples = new StringTokenizer(lineSamples);
double[] samples = new double[length];
for(int i=0; i<length; i++) {
samples[i] = Double.parseDouble(tokSamples.nextToken().trim());
}
// executa a transformada Wavelet
Wavelet timeDomain = new Wavelet(samples);
timeDomain.normalize();

```

```

Wavelet timefrequencyDomain = Wavelet.dwt(timeDomain, wavelet, 2);
timefrequencyDomain.normalize();
// grava o resultado na saída
writerCwc.print(timefrequencyDomain.length() + " ");
writerCwc.println(header);
for(int i=0; i<timefrequencyDomain.length(); i++) {
writerCwc.print(timefrequencyDomain.valueAt(i) + " ");
}
writerCwc.println();
writerCwc.flush();
} catch(Exception e1) {
e1.printStackTrace();
}
}
}
}
}

```

A.4 Formatação para o Padrão JavaNNS

As redes neurais artificiais utilizadas neste projeto foram implementadas no ambiente JavaNNS. Para permitir isso, implementou-se uma classe Java com a função de converter os coeficientes Wavelet para o padrão suportado pelo JavaNNS. Durante essa etapa de formatação, o mesmo algoritmo seleciona os 40 maiores coeficientes em termos absolutos. Esses coeficientes são apresentados em ordem decrescente, sem os respectivos sinais. O sinal e a posição de cada coeficiente são armazenados na forma de metadados, em um arquivo auxiliar.

```

package br.pucpr.ppgia.figueredo;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.Collections;

```

```

import java.util.List;
import java.util.StringTokenizer;

public class SsnsPreProcessor extends Processor {

    // número de neurônios das camadas de entrada e saída
    private static final int NEURONS_COUNT = 40;

    @Override
    public void process() throws Exception {
        // acessa os canais de entrada
        InputStream inSwc = getInput(FileTypes.swc.toString());
        BufferedReader readerSwc = new BufferedReader(new InputStreamReader(inSwc));
        InputStream inCwc = getInput(FileTypes.cwc.toString());
        BufferedReader readerCwc = new BufferedReader(new InputStreamReader(inCwc));

        // acessa os canais de saída
        OutputStream outSpat = getOutput(FileTypes.spat.toString());
        PrintWriter writerSpat = new PrintWriter(outSpat);
        OutputStream outCpat = getOutput(FileTypes.cpat.toString());
        PrintWriter writerCpat = new PrintWriter(outCpat);
        OutputStream outSme = getOutput(FileTypes.sme.toString());
        PrintWriter writerSme = new PrintWriter(outSme);
        OutputStream outCme = getOutput(FileTypes.cme.toString());
        PrintWriter writerCme = new PrintWriter(outCme);

        // processa os blocos ‘‘simples’’
        int counter = 1;
        while(readerSwc.ready()) {
            try {
                // descobre o número de coeficientes
                String lineHeader = readerSwc.readLine();
                StringTokenizer tokHeader = new StringTokenizer(lineHeader);
                int length = Integer.parseInt(tokHeader.nextToken().trim());
                // transforma os coeficientes em um array de doubles
                String lineCoefs = readerSwc.readLine();
            }
        }
    }
}

```

```

StringTokenizer tokCoefs = new StringTokenizer(lineCoefs);
double[] coefs = new double[length];
if(length<NEURONS_COUNT) {
    continue;
}
for(int i=0; i<length; i++) {
    coefs[i] = Double.parseDouble(tokCoefs.nextToken().trim());
}
// ordena os coeficientes
List<SortableCoefficient> sortedCoefs = new ArrayList<SortableCoefficient>();
for(int i=0; i<length; i++) {
    sortedCoefs.add(new SortableCoefficient(i, coefs[i]));
}
Collections.sort(sortedCoefs, new ValueCoefficientsComparator());
// seleciona os coeficientes mais importantes
writerSpat.println("# Input pattern " + counter + ":");
for(int i=0; i<NEURONS_COUNT; i++) {
    writerSpat.print(Math.abs(sortedCoefs.get(i).getValue()) + " ");
}
writerSpat.println();
writerSpat.println("# Output pattern " + counter + ":");
for(int i=0; i<NEURONS_COUNT; i++) {
    writerSpat.print(Math.abs(sortedCoefs.get(i).getValue()) + " ");
}
writerSpat.println();
// processa os metadados
writerSme.println(length + " " + NEURONS_COUNT);
for(int i=0; i<NEURONS_COUNT; i++) {
    writerSme.print(sortedCoefs.get(i).getPosition() + " ");
}
writerSme.println();
for(int i=0; i<NEURONS_COUNT; i++) {
    if(sortedCoefs.get(i).getValue()>=0) {
        writerSme.print("1 ");
    } else {
        writerSme.print("-1 ");
    }
}

```

```

}
}
writerSme.println();
// incrementa o contador
counter++;
} catch(Exception e1) {
e1.printStackTrace();
}
}

// processa os blocos ‘‘complexos’’
counter = 1;
while(readerCwc.ready()) {
try {
// descobre o número de coeficientes
String lineHeader = readerCwc.readLine();
StringTokenizer tokHeader = new StringTokenizer(lineHeader);
int length = Integer.parseInt(tokHeader.nextToken().trim());
// transforma os coeficientes em um array de doubles
String lineCoefs = readerCwc.readLine();
StringTokenizer tokCoefs = new StringTokenizer(lineCoefs);
if(length<NEURONS_COUNT) {
continue;
}
double[] coefs = new double[length];
for(int i=0; i<length; i++) {
coefs[i] = Double.parseDouble(tokCoefs.nextToken().trim());
}
// ordena os coeficientes
List<SortableCoefficient> sortedCoefs = new ArrayList<SortableCoefficient>();
for(int i=0; i<length; i++) {
sortedCoefs.add(new SortableCoefficient(i, coefs[i]));
}
Collections.sort(sortedCoefs, new ValueCoefficientsComparator());
// seleciona os coeficientes mais importantes
writerCpat.println("# Input pattern " + counter + ":");

```



```

for(int i=0; i<NEURONS_COUNT; i++) {
writerCpat.print(Math.abs(sortedCoefs.get(i).getValue()) + " ");
}
writerCpat.println();
writerCpat.println("# Output pattern " + counter + ":");
for(int i=0; i<NEURONS_COUNT; i++) {
writerCpat.print(Math.abs(sortedCoefs.get(i).getValue()) + " ");
}
writerCpat.println();
// processa os metadados
writerCme.println(length + " " + NEURONS_COUNT);
for(int i=0; i<NEURONS_COUNT; i++) {
writerCme.print(sortedCoefs.get(i).getPosition() + " ");
}
writerCme.println();
for(int i=0; i<NEURONS_COUNT; i++) {
if(sortedCoefs.get(i).getValue()>=0) {
writerCme.print("1 ");
} else {
writerCme.print("-1 ");
}
}
writerCme.println();
// incrementa o contador
counter++;
} catch(Exception e1) {
e1.printStackTrace();
}
}
}
}
}

```

A.5 Transformada Wavelet Inversa

A saída das redes neurais é processada com uma transformada Wavelet inversa, para que se obtenham os blocos de ECG. Durante essa etapa, os metadados (sinal e posição) são recuperados para garantir que o sinal possa ser reconstruído com sucesso.

```
package br.pucpr.ppgia.figueredo;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.PrintWriter;
import java.util.StringTokenizer;

public class ExpansionProcessor extends Processor {

    @Override
    public void process() throws Exception {
        // acessa os canais de entrada
        InputStream inCres = getInput(FileTypes.cres.toString());
        BufferedReader readerCres = new BufferedReader(new InputStreamReader(inCres));
        InputStream inCme = getInput(FileTypes.cme.toString());
        BufferedReader readerCme = new BufferedReader(new InputStreamReader(inCme));

        // acessa os canais de saída
        OutputStream outCexp = getOutput(FileTypes.cexp.toString());
        PrintWriter writerCexp = new PrintWriter(outCexp);

        // utiliza a Wavelet Coiflet com 6 níveis
        Wavelet wavelet = new Wavelet(Wavelet.c6);

        // descarta os cabeçalhos do arquivo JavaNNS
        for(int i=0; i<8; i++) {
            readerCres.readLine();
        }
    }
}
```

```

// processa os dados
int counter = 0;
while(readerCme.ready()) {
counter++;
// processa os metadados
String lineCme = readerCme.readLine();
StringTokenizer tokCme = new StringTokenizer(lineCme);
int length = Integer.parseInt(tokCme.nextToken().trim());
int positionsCount = Integer.parseInt(tokCme.nextToken().trim());
String linePositions = readerCme.readLine();
StringTokenizer tokPositions = new StringTokenizer(linePositions);
String lineSignals = readerCme.readLine();
StringTokenizer tokSignals = new StringTokenizer(lineSignals);
// transforma os coeficientes em um array de doubles
double[] coefs = new double[length];
// descarta os comentários do arquivo
readerCres.readLine();
for(int i=0; i<4; i++) {
String lineRes = readerCres.readLine();
StringTokenizer tokRes = new StringTokenizer(lineRes);
while(tokRes.hasMoreTokens()) {
coefs[Integer.parseInt(tokPositions.nextToken().trim())] = Double.parseDouble(tokRe
}
}
readerCres.readLine();
// aplica a transformada Wavelet inversa (IDWT)
Wavelet timeFrequencyDomain = new Wavelet(coefs);
timeFrequencyDomain.normalize();
Wavelet timeDomain = Wavelet.idwt(timeFrequencyDomain, wavelet, 2);
timeDomain.normalize();
// salva os dados na saída
writerCexp.println(length);
for(int i=0; i<length; i++) {
writerCexp.print(coefs[i] + " ");
}
writerCexp.println();

```

```

for(int i=0; i<timeDomain.length(); i++) {
writerCexp.print(timeDomain.valueAt(i) + " ");
}
    writerCexp.println();
writerCexp.flush();
}
}
}

```

A.6 Reconstrução

A etapa de reconstrução envolve a união dos blocos “simples” e “complexos” na forma do sinal de ECG.

```

package br.pucpr.ppgia.figueredo;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.PrintWriter;

public class ReconstructionProcessor extends Processor {

    @Override
    public void process() throws Exception {
        // acessa os canais de entrada
        InputStream inClas = getInput(FileTypes.clas.toString());
        BufferedReader readerClas = new BufferedReader(new InputStreamReader(inClas));
        InputStream inCexp = getInput(FileTypes.cexp.toString());
        BufferedReader readerCexp = new BufferedReader(new InputStreamReader(inCexp));
        InputStream inSexp = getInput(FileTypes.sexp.toString());
        BufferedReader readerSexp = new BufferedReader(new InputStreamReader(inSexp));
        InputStream inCbl = getInput(FileTypes.cbl.toString());
    }
}

```

```

BufferedReader readerCbl = new BufferedReader(new InputStreamReader(inCbl));
InputStream inSbl = getInput(FileTypes.sbl.toString());
BufferedReader readerSbl = new BufferedReader(new InputStreamReader(inSbl));

// acessa os canais de saída
OutputStream outProc = getOutput(FileTypes.proc.toString());
PrintWriter writerProc = new PrintWriter(outProc);
OutputStream outOri = getOutput(FileTypes.ori.toString());
PrintWriter writerOri = new PrintWriter(outOri);

// reconstrói os dados
while(readerClas.ready()) {
String classification = readerClas.readLine().trim().toLowerCase();
if("s".equals(classification)) {
writerProc.println(readerSexp.readLine());
writerProc.println(readerSexp.readLine());
writerProc.println(readerSexp.readLine());
writerProc.flush();
writerOri.println(readerSbl.readLine());
writerOri.println(readerSbl.readLine());
writerOri.flush();
} else if("c".equals(classification)) {
writerProc.println(readerCexp.readLine());
writerProc.println(readerCexp.readLine());
writerProc.println(readerCexp.readLine());
writerProc.flush();
writerOri.println(readerCbl.readLine());
writerOri.println(readerCbl.readLine());
writerOri.flush();
}
}
writerOri.flush();
writerOri.close();
writerProc.flush();
writerProc.flush();
}}

```