

ANDRÉ PAIM LEMOS

**PROPOSTA DE UM ALGORITMO GENÉRICO  
DE DETECÇÃO DE NOVIDADES EM SÉRIES  
TEMPORAIS UTILIZANDO MODELOS DE  
PREVISÃO**

Belo Horizonte  
17 de julho de 2007

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
ESCOLA DE ENGENHARIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**PROPOSTA DE UM ALGORITMO GENÉRICO  
DE DETECÇÃO DE NOVIDADES EM SÉRIES  
TEMPORAIS UTILIZANDO MODELOS DE  
PREVISÃO**

Proposta de dissertação apresentada ao Curso de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Engenharia Elétrica.

ANDRÉ PAIM LEMOS

Belo Horizonte  
17 de julho de 2007

# Resumo

*Novidades* ou *anomalias* em uma série temporal podem ser vistas como valores ou uma sequência de valores inesperados quando comparados a um conjunto de valores considerados *normais*. Um algoritmo de detecção de novidades em séries temporais deve ser capaz de classificar cada ponto da série em normal ou novidade. Existem diversas aplicações para o problema de detecção de novidades em séries temporais, como detecção de falhas em sistemas dinâmicos, detecção de fraudes em sistemas financeiros e monitoramento de sinais fisiológicos. Esse trabalho propõe um algoritmo genérico de detecção de novidades em séries temporais baseado em um modelo de previsão. O algoritmo é genérico pois não define qual modelo a ser utilizado. Duas instanciações do algoritmo são apresentadas, a primeira baseada em um modelo de previsão estatístico e a segunda em um modelo neural. O algoritmo foi utilizado para detectar síndrome de apnéia do sono, através do monitoramento de sinais de eletrocardiogramas e para detectar falhas em um sistema dinâmico através do monitoramento de algumas de suas variáveis amostradas ao longo do tempo. Os resultados obtidos, nos dois problemas, quando comparados aos encontrados na literatura, mostram que o algoritmo proposto é uma alternativa bastante promissora.

# Abstract

*Novelties* or *anomalies* on time series can be seen as unexpected values or a sequence of unexpected values when compared to a dataset considered to be normal. A time series novelty detection algorithm must be capable to classify each point of the time series as normal or novelty. There are many applications to the problem of time series novelty detection including fault detection on dynamic systems, fraud detection on financial systems and physiologic signals monitoring. This work proposes a generic novelty detection algorithm based on a forecasting model. The algorithm is generic because it does not define the forecasting model to be used. Two algorithm instantiations are proposed, the first one, based on a statistical model and the second one based on a neural model. The algorithm is used to screen obstructive sleep apnea through electrocardiogram monitoring and to detect faults on a dynamic system monitoring some of the system variables. The results achieved on both problems are near to the results found on literature.

# Agradecimentos

Agradeço ao meu orientador, professor Walmir de Matos Caminhas, pelo apoio e orientação ao longo desses últimos 2 anos. Agradeço também ao meu co-orientador, professor Carlos Júlio Tierra-Criollo, que tanto me ajudou no desenvolvimento deste trabalho.

Agradeço aos colegas do laboratório de Inteligência Computacional e do CPDEE pela amizade e companherismo ao longo desses últimos dois anos.

Agradeço a minha família, meus pais e irmãs pela paciência, amor e apoio incondicional.

Agradeço a Tati, pessoa especial em minha vida, que está sempre ao meu lado.

Agradeço ao Tiago, criador da charge presente na introdução.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	1
1.2	Definição de Detecção de Novidades . . . . .	3
1.2.1	Detecção de Novidades em Séries Temporais . . . . .	5
1.3	Objetivo . . . . .	6
1.4	Estrutura do Trabalho . . . . .	7
<b>2</b>	<b>Revisão das Técnicas de Detecção de Novidades em Séries Temporais</b>	<b>8</b>
2.1	Técnicas baseadas em classificação . . . . .	9
2.1.1	Classificação de duas classes . . . . .	9
2.1.2	Seleção Negativa . . . . .	10
2.1.3	Classificação de Uma Classe . . . . .	14
2.2	Técnicas baseadas em Modelos de Previsão . . . . .	16
2.2.1	Variação do Erro de Previsão . . . . .	16
2.2.2	Variação da Estrutura do Modelo . . . . .	18
2.3	Detecção de Discórdias . . . . .	19
<b>3</b>	<b>Detecção de Novidades em Séries Temporais Utilizando Modelos de Previsão</b>	<b>22</b>
3.1	Algoritmo Genérico . . . . .	23
3.1.1	Exemplo . . . . .	26
3.2	Detecção de Novidades em Séries Temporais Utilizando Modelos Auto-regressivos e Intervalos de Previsão . . . . .	31
3.2.1	Modelo de Previsão . . . . .	31
3.2.2	Intervalos de Tolerância . . . . .	33
3.2.3	Probabilidade de Ocorrência de uma Surpresa . . . . .	35
3.2.4	Exemplo . . . . .	36
3.2.5	Restrições . . . . .	41
3.3	Detecção de Novidades em Séries Temporais Estacionárias Utilizando Redes Neurais e Intervalos Robustos de Confiança . . . . .	41

3.3.1	Modelo de Previsão . . . . .	43
3.3.2	Intervalo de Tolerância . . . . .	45
3.3.3	Probabilidade de Ocorrência de uma Surpresa . . . . .	47
3.3.4	Exemplo . . . . .	48
<b>4</b>	<b>Experimentos</b>	<b>53</b>
4.1	Análise da Influência dos Parâmetros do Algoritmo . . . . .	54
4.1.1	Nível de Significância $\alpha$ . . . . .	54
4.1.2	Tamanho do Evento $n$ . . . . .	57
4.2	Análise Comparativa . . . . .	61
4.3	Detecção de Apnéia Obstrutiva do Sono . . . . .	63
4.4	Detecção de Falhas em um Atuador . . . . .	68
4.5	Análise dos Resultados . . . . .	77
<b>5</b>	<b>Conclusão</b>	<b>80</b>
5.1	Trabalhos Futuros . . . . .	82
5.1.1	Definição dos Intervalos de Tolerância do Erro de Previsão para Outros Modelos Estatísticos . . . . .	82
5.1.2	Séries Multivariadas . . . . .	83
5.1.3	Modelos Adaptativos . . . . .	83
5.2	Produção Científica . . . . .	84
	<b>Appendices</b>	<b>84</b>
	<b>A Algoritmo com Ordem de Complexidade Linear</b>	<b>85</b>
	<b>Referências Bibliográficas</b>	<b>87</b>

# Lista de Figuras

1.1	Exemplo de um problema detecção de novidades e a diferença entre um sistema de detecção de novidades e um sistema de diagnóstico baseado em uma máquina de aprendizado supervisionado . . . . .	2
1.2	Exemplo de um problema de classificação de uma classe . . . . .	4
2.1	Envelope criado sobre uma janela de pontos para definir a região de normalidade . . . . .	10
2.2	Exemplo de padrões artificiais criados a partir de uma janela de pontos normais utilizando-se um envelope . . . . .	11
2.3	Criação dos receptores do algoritmo imunológico artificial . . . . .	13
2.4	Etapa de detecção do algoritmo imunológico artificial . . . . .	13
2.5	Método híbrido que utiliza seleção negativa e classificação de duas classes .	14
2.6	Série temporal e sua representação simbólica realizada através do método SAX. A série em questão é representada pela sequência cccbedba . . . . .	20
3.1	Valor de $\gamma$ para $q = 0.05$ e $n = 300$ . . . . .	25
3.2	Série temporal utilizada como exemplo do algoritmo de detecção de novidades	27
3.3	Pontos da série utilizados para validação do modelo . . . . .	28
3.4	Pontos da série a serem classificados . . . . .	29
3.5	Ilustração de um intervalo de previsão da saída de um modelo $AR(p)$ . . .	35
3.6	Valor da probabilidade de surpresa, $q$ , para o modelo $AR(p)$ . . . . .	36
3.7	Série temporal gerada a partir de (3.20) . . . . .	37
3.8	Autocorrelação amostral do erro de previsão do conjunto de validação . . .	38
3.9	Gráfico de distribuição normal do erro de previsão . . . . .	38
3.10	Valor observado, previsto e intervalos de previsão para os pontos do intervalo $t = [601, 1200]$ . . . . .	39
3.11	Série temporal gerada a partir de (3.20) no intervalo $t = [1201, 1800]$ contendo uma novidade a ser detectada no intervalo $t = [1500, 1580]$ . . . . .	40
3.12	Classificação final dos pontos do intervalo $t = [1201, 1800]$ . . . . .	40
3.13	Eletrocardiograma apresentando uma contração ventricular prematura (V)	42

3.14	Estrutura de uma rede neural utilizada para previsão de séries temporais . . . . .	43
3.15	Modelo de previsão formado por um comitê de redes baseado na promediação . . . . .	45
3.16	Cálculo das amostras do erro de previsão a partir das redes do modelo . . . . .	47
3.17	Primeiros 200 pontos da série normalizados . . . . .	48
3.18	Série temporal no intervalo [1450, 1650] contendo a novidade a ser detectada no intervalo [1500, 1580] . . . . .	49
3.19	Autocorrelação amostral dos primeiros 1000 pontos da série . . . . .	49
3.20	Valor de BIC em função da topologia da rede . . . . .	50
3.21	Gráfico de correlação amostral do erro de previsão . . . . .	51
3.22	Histograma do erro de previsão e os limiares do intervalo robusto de confi- ança calculado . . . . .	51
3.23	Resultado final da classificação dos pontos do intervalo [1001, 2000] que possui uma novidade no intervalo [1500, 1580] . . . . .	52
4.1	Série temporal gerada a partir de (4.1) no intervalo temporal [5000, 11000].	55
4.2	$P_{FA}$ esperado e medido em função de $\alpha$ . . . . .	56
4.3	$q$ esperado e medido em função de $\alpha$ . . . . .	57
4.4	$P_D$ medido em função de $\alpha$ . . . . .	57
4.5	Ilustração do tempo de recuperação de uma detecção, $t_r$ . . . . .	58
4.6	Exemplo ilustrativo do tempo de detecção, $t_d$ . . . . .	59
4.7	$P_{FA}$ em função de $n$ . . . . .	60
4.8	$P_D$ em função de $n$ . . . . .	60
4.9	Curva ROC da técnica original e da técnica estendida proposta . . . . .	62
4.10	A figura (a) ilustra a série temporal de intervalos RR em um instante de tempo em que não foram detectados eventos de apnéia e a figura (b) ilustra a mesma série durante eventos de apnéia. . . . .	64
4.11	A figura (a) apresenta a foto de um paciente preparado para o exame de polisonografia e a figura (b) um paciente utilizando um aparelho de moni- toramento de ECG portátil . . . . .	64
4.12	A figura (a) apresenta uma série de intervalos RR antes do pré-processamento e a figura (b) a mesma série após o pré-processamento . . . . .	65
4.13	Valor de BIC em função de $p$ para uma série de intervalos RR . . . . .	67
4.14	Diagrama esquemático do atuador . . . . .	68
4.15	Diagrama do simulador do atuador . . . . .	69
4.16	Tipos de tempo de desenvolvimento da falha . . . . .	70
4.17	As figuras ilustram os primeiros 1000 pontos da variável de saída $F$ (a) e $X$ (b) utilizados no treinamento das redes . . . . .	72
4.18	Valor de BIC em função da topologia da rede para a variável $F$ . . . . .	72

4.19	Gráfico de correlação amostral do erro de previsão da variável $F$ . . . . .	73
4.20	Valor de BIC em função da topologia da rede para a variável $X$ . . . . .	74
4.21	Gráfico de correlação amostral do erro de previsão da variável $X$ . . . . .	74
4.22	As figuras ilustram os valores das variáveis monitoradas $F$ (a) e $X$ (b) antes e depois do início da falha f1 abrupta de intensidade baixa ( $t_{from} = 900$ ) .	75
4.23	A figura (a) ilustra os resultados de cada um dos detectores e a figura (b) ilustra o resultado final ( $t_{from} = 900$ ) . . . . .	75
4.24	As figuras ilustram os valores das variáveis monitoradas $F$ (a) e $X$ (b) antes e depois do início da falha f8 abrupta de intensidade alta $t_{from} = 900$ . . .	76
4.25	As figuras ilustram os valores das variáveis monitoradas $F$ (a) e $X$ (b) antes e depois do início da falha f18 abrupta de intensidade alta $t_{from} = 900$ . . .	77
4.26	Valores observados, estimados e o intervalo robusto de confiança do erro para a série de amostragens de $F$ no intervalo temporal [1000, 1200] . . . .	78

# Lista de Tabelas

3.1	Eventos gerados a partir das ocorrências . . . . .	29
3.2	Norma dos eventos . . . . .	30
3.3	Classificação final dos pontos do conjunto de teste . . . . .	30
3.4	Conceitos do algoritmo genérico utilizados no exemplo . . . . .	30
3.5	Valores dos Parâmetros Estimados . . . . .	37
3.6	Intervalos robustos de confiança de uma variável aleatória gaussiana $X \sim N(0, 1)$ para diferentes números de amostras e $\alpha = 0.05$ . . . . .	46
4.1	Valores médios de $P_{FA}$ e $t_r$ para diferentes valores de $n$ . . . . .	61
4.2	Valores médios de $P_D$ e $t_d$ para diferentes valores de $n$ . . . . .	61
4.3	Tipos de Falhas simuladas . . . . .	70
4.4	Cenários de Falhas . . . . .	71
4.5	Índices de Desempenho para os Cenários de Falhas Abruptas . . . . .	76
4.6	Índices de Desempenho para os Cenários de Falhas Incipientes . . . . .	78
4.7	Comparação dos resultados obtidos nos experimentos de detecção de falhas	79

# Capítulo 1

## Introdução

O comportamento normal de um sistema pode ser descrito através de sucessivas observações de suas variáveis ao longo do tempo. O problema de *detecção de novidades ou anomalias em séries temporais* consiste em encontrar intervalos de tempo em que o comportamento do sistema desvia do comportamento definido como padrão.

### 1.1 Motivação

Um motorista está dirigindo seu carro quando escuta um “barulho estranho” proveniente do motor. Ele associa o barulho com algum tipo de falha do veículo.

Para realizar essa associação, o motorista assumiu que o comportamento normal do veículo é descrito por um padrão sonoro peculiar que ele escuta todos os dias ao dirigir. Quando esse padrão foi alterado, ele atribuiu essa novidade ou anomalia a uma falha no automóvel.

Caso o motorista leve seu carro a um mecânico, este, provavelmente, analisando o padrão sonoro anômalo, será capaz de não apenas confirmar a falha suposta pelo motorista, mas também realizar um diagnóstico, mesmo que incerto, do tipo de falha que está ocorrendo. O mecânico seria capaz de distinguir entre diferentes tipos de falha a partir do padrão sonoro, baseando-se em suas experiências passadas.

Este simples exemplo, ilustrado na figura 1.1, descreve um sistema de detecção de novidades em séries temporais, em que o sistema especialista, representado pelo motorista, é capaz de distinguir entre o funcionamento normal e anômalo do sistema através da análise de uma das suas “variáveis”, nesse exemplo, o som emitido pelo motor.

Além disso, esse exemplo ilustra a diferença entre um sistema de detecção de novidades e um sistema de diagnóstico. No primeiro, o sistema é capaz de distinguir entre o funcionamento normal e anômalo de um sistema monitorado baseado apenas no desvio

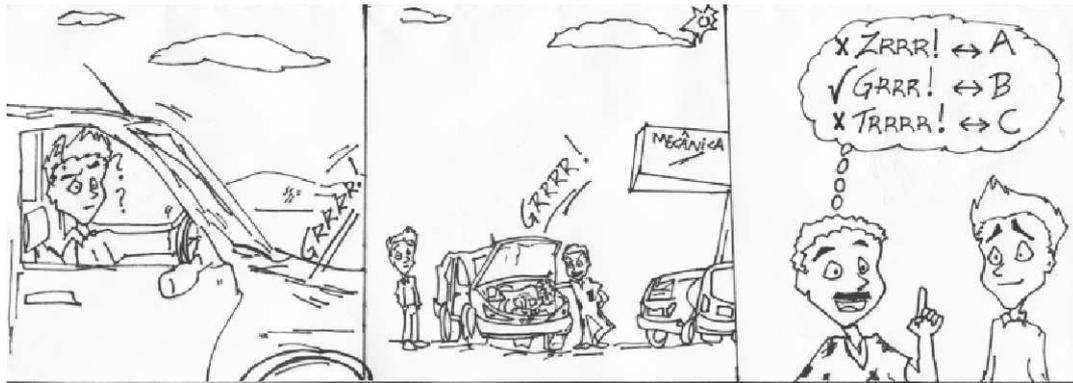


Figura 1.1: Exemplo de um problema de detecção de novidades e a diferença entre um sistema de detecção de novidades e um sistema de diagnóstico baseado em uma máquina de aprendizado supervisionado

do comportamento normal, representado pelo motorista. Já um sistema de diagnóstico, representado pelo mecânico, consegue não só distinguir mais precisamente entre o comportamento normal e anômalo, como também apontar o tipo de falha presente no sistema, baseado em conhecimentos adquiridos sobre as possíveis falhas.

Pode-se questionar a necessidade de desenvolvimento de sistemas de detecção de novidades, uma vez que um sistema de diagnóstico não só é capaz de realizar a detecção de novidades, como também de distinguir entre diferentes situações anômalas.

Porém, em muitos casos práticos, se tem pouco ou nenhum dado do comportamento do sistema em situações de falha, tornando inviável o desenvolvimento de um sistema de diagnóstico baseado em modelos obtidos a partir de dados. Além disso, em muitos casos é inviável, fisicamente ou financeiramente, simular falhas no sistema para adquirir dados do seu funcionamento em tais situações. Dessa maneira, torna-se necessário desenvolver um sistema de detecção de falhas que utilize apenas informações adquiridas durante o funcionamento normal do mesmo.

A detecção de novidades não se limita à detecção de falhas podendo ser aplicada em diversas outras áreas como, por exemplo, na detecção de fraudes. Em alguns sistemas é difícil prever todas as possíveis fraudes que possam ser aplicadas, optando-se, dessa maneira, por construir um modelo de normalidade para o sistema baseado em medições de variáveis ao longo do tempo e, caso as variáveis divirjam dos valores esperados pelo modelo, associa-se a novidade encontrada a uma possível fraude no sistema. Por exemplo, em um sistema de folha de pagamento em que a variável monitorada seria os salários pagos mensais, ou em um sistema de contabilidade em que as variáveis monitoradas seriam contas a pagar ou a receber (Oliveira e Meira (2006)).

Outro exemplo seria no monitoramento de sinais fisiológicos, em que baseando-se

em medições desses sinais, tais como eletrocardiogramas (ECG) e eletroencefalogramas (EEG), para um paciente saudável, constrói-se um modelo de comportamento normal do paciente e, caso esses sinais diverjam de valores esperados pelo modelo, atribui-se a essas anormalidades indicativos de possíveis enfermidades (Keogh et al. (2005)).

## 1.2 Definição de Detecção de Novidades

Do ponto de vista da teoria do aprendizado estatístico, o problema de detecção de novidades consiste em encontrar padrões novos ou desconhecidos que não foram apresentados à máquina de aprendizado durante seu treinamento (Markou e Singh (2003a)).

Um algoritmo de detecção de novidades deve ser capaz de distinguir entre as seguintes hipóteses:

$$H_0 : x_i \text{ é normal}$$

$$H_1 : x_i \text{ é novidade}$$

Ou seja, o algoritmo deve ser capaz de classificar cada padrão de entrada como normal ou novidade.

A probabilidade de detecção  $P_D$  é definida como a probabilidade de o algoritmo classificar o padrão como novidade corretamente. E a probabilidade de falsos alarmes  $P_{FA}$  é a probabilidade do algoritmo ter classificado  $x_i$  como novidade quando na verdade este era um padrão normal (Kay (1993a)). Dessa forma, deseja-se desenvolver uma técnica que maximize a probabilidade de detecção e minimize a probabilidade de falsos alarmes.

Uma revisão completa das técnicas estatísticas e neurais de detecção de novidades pode ser encontrada em Markou e Singh (2003a) e Markou e Singh (2003b). Basicamente, existem duas abordagens para se resolver esse problema:

- Técnicas que utilizam os padrões normais para gerar artificialmente padrões de novidades (Vasconcelos (1995)), ou características que possam identificar padrões de novidades (Dasgupta e Gonzalez (2002)) e a detecção é realizada a partir desse conjunto híbrido de padrões.
- Técnicas baseadas em classificação de uma classe em que apenas padrões normais são utilizados no processo de classificação, realizado através da estimação de limiares de normalidade (Tax (2001)).

O principal problema da primeira abordagem é que dificilmente os padrões de novidade gerados artificialmente representarão sua real distribuição espacial (Vasconcelos

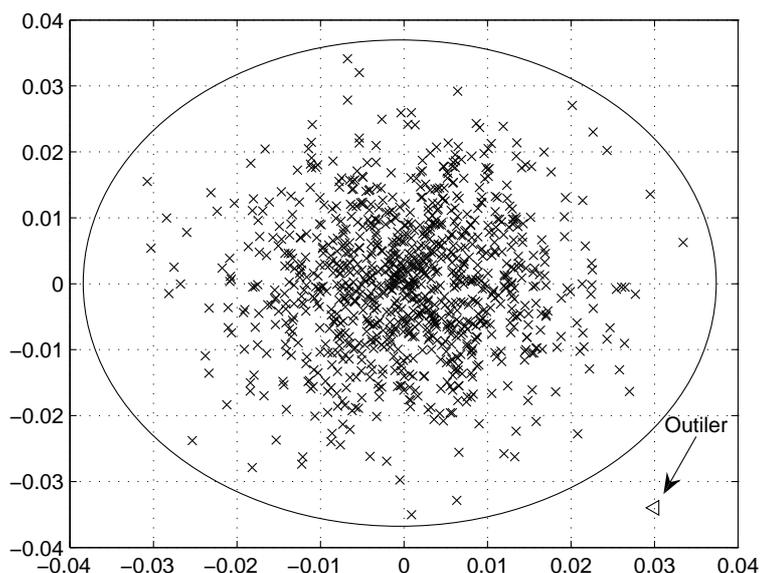


Figura 1.2: Exemplo de um problema de classificação de uma classe

(1995)). Dessa maneira, essa metodologia torna-se bastante dependente da técnica utilizada para geração desses padrões artificiais.

O problema de classificação de uma classe é um caso especial dos problemas de classificação convencionais. Os métodos tradicionais de classificação realizam a distinção entre duas classes dados exemplos de cada uma das classes. Na classificação de uma classe apenas uma das classes possui exemplos, definida como classe objetivo (*target class*), e todos os outros padrões que não se encaixam na classe objetivo são definidos como *outliers* (Tax (2001)).

A figura 1.2 ilustra o exemplo de classificação de uma classe em que o algoritmo de classificação deve ser capaz de distinguir entre os pontos pertencentes à classe objetivo e os demais pontos.

Existem diversos métodos propostos na literatura para resolver o problema de classificação de uma classe. Em geral, esses métodos constroem um modelo representando a classe objetivo e alguma métrica de distância ou probabilidade, a partir dos dados de treinamento. Em seguida, determina-se um limiar para a métrica utilizada de forma que, para cada novo padrão a ser classificado, caso o seu valor correspondente da métrica utilizada seja maior que o limiar, esse é definido como uma novidade. As técnicas presentes na literatura variam de acordo com o modelo, métrica e metodologia de cálculo do limiar de separação.

Em geral, esses métodos estão divididos em três classes:

- Técnicas baseadas em estimação de densidade de probabilidade, como por exemplo, o modelo Gaussiano e o estimador de Janelas de Parzen (*Parzen Window*) (Parzen (1962)). Nessas técnicas, estima-se a distribuição de probabilidade dos padrões da classe objetivo e utiliza-se um intervalo de confiança como limiar de classificação.
- Técnicas baseadas na construção de limites de separação, como por exemplo, o k-centros (Ypma e Duin (1998)) e classificação de uma classe via máquinas de vetores suportes (Tax (2001)). Nessas técnicas, o modelo construído não se preocupa em estimar a distribuição dos padrões da classe objetivo e sim, apenas estimar os limiares da classe. De forma que o próprio modelo já define os limiares de normalidade.
- Técnicas baseadas em métodos de reconstrução, como por exemplo, o k-médias (Schurmann (1996)) e mapas auto-organizáveis (Haykin (1999)). Essas técnicas capturam características importantes dos padrões da classe objetivo em um espaço de características de dimensão reduzida e realiza a discriminação dos padrões nesse espaço.

Uma completa revisão dessas técnicas pode ser encontrada em Tax (2001).

### 1.2.1 Detecção de Novidades em Séries Temporais

Um algoritmo de detecção de novidades em séries temporais deve ser capaz de distinguir entre as seguintes hipóteses:

$$\begin{aligned}
 H_0 & : x(t) \text{ é normal} \\
 H_1 & : x(t) \text{ é novidade}
 \end{aligned}$$

sendo que  $x(t)$  é o valor observado da série no instante  $t$ .

Dessa maneira, para se resolver o problema de detecção de novidades em séries temporais, é necessário inicialmente extrair características a partir das observações normais da série e utilizar essas características como entradas de um algoritmo de detecção de novidades.

Durante o processo de extração de características, as seguintes particularidades devem ser relevadas:

- a ordem temporal das observações deve ser levada em conta;

- as observações vizinhas são correlacionadas.

Sendo assim, em geral, a extração das características é realizada através de uma janela de pontos ou a partir de resíduos gerados por um modelo de previsão. Isso é feito para se preservar a ordem temporal das medições. Além disso, as características extraídas a partir de diferentes observações da série devem ser descorrelacionadas e identicamente distribuídas para facilitar na estimação de sua distribuição de probabilidade, ou os limites dessa.

Finalmente, em uma grande variedade de problemas, o padrão de novidade não é definido por um único ponto e sim, por uma sequência de pontos. Dessa forma, é necessário que o algoritmo realize o processamento dos pontos da série em conjunto.

### 1.3 Objetivo

A técnica proposta nesse trabalho utiliza como característica o erro de previsão do instante atual. Para isso, um modelo de previsão da série analisada é construído baseado em dados normais. Em seguida, para cada novo ponto a ser analisado, o valor do erro de previsão é calculado como a diferença entre o valor estimado do modelo e o valor observado da série.

Uma vez calculado o valor do erro de previsão, realiza-se um processamento desses valores através de uma janela de pontos e utiliza-se uma técnica de classificação de uma classe baseada na estimação da função de distribuição de probabilidade para se chegar à classificação final de cada padrão.

A técnica proposta nesse trabalho pode ser utilizada com qualquer tipo de modelo de previsão. Dessa forma um algoritmo genérico é proposto de maneira que alguns dos conceitos utilizados no mesmo devam ser definidos conforme o modelo escolhido.

Duas instanciações do algoritmo são apresentadas. Na primeira instanciação, utiliza-se um modelo de previsão estatístico de forma que a função de densidade de probabilidade do erro de previsão é conhecida, sendo necessário apenas estimar seus parâmetros através dos dados.

A segunda instanciação proposta utiliza um modelo de previsão baseado em redes neurais de modo que a forma da função de densidade de probabilidade do erro não é conhecida, sendo então estimada a partir de um conjunto de dados normais.

Esse trabalho resumiu-se a detectar novidades em séries temporais unimodais.

## 1.4 Estrutura do Trabalho

Essa dissertação está organizada em cinco capítulos. Nesse capítulo, inicialmente foi apresentada uma motivação justificando o estudo do problema de detecção de novidades em séries temporais. Em seguida, o problema de detecção de novidades foi definido formalmente, assim como as peculiaridades necessárias para que algoritmos que resolvam esse problema possam ser aplicados em séries temporais.

No capítulo 2 será apresentada uma revisão de algumas das principais técnicas de detecção de novidades em séries temporais encontradas na literatura. Em geral essas técnicas podem ser divididas em técnicas que utilizam um modelo de previsão da série construído a partir de dados normais e técnicas que utilizam diretamente as observações da série.

No capítulo 3, a técnica proposta nesse trabalho será apresentada. Conforme descrito em 1.3, a técnica baseia-se em um algoritmo genérico utilizado para realizar detecção de novidades a partir de métodos de classificação de uma classe utilizando como entrada o erro de previsão. O algoritmo proposto é genérico, pois não define o modelo de previsão a ser utilizado. Dessa maneira, duas instanciações desse algoritmo são apresentadas em seguida. A primeira, baseada em um modelo de previsão estatístico e a segunda baseada em um modelo neural.

Em seguida, no capítulo 4, serão apresentados experimentos com o objetivo de validar o método proposto. Inicialmente será apresentada uma análise da influência dos parâmetros do algoritmo na sua eficiência. A instanciação do algoritmo baseada em redes neurais pode ser vista como uma extensão do método proposto em Oliveira e Meira (2006), com o objetivo de aumentar o desempenho desse método para séries temporais em que uma novidade é caracterizada por uma sequência de pontos. Dessa forma, um experimento será apresentado para comprovar esse ganho de desempenho. Em seguida, o algoritmo proposto será utilizado para detectar síndrome de apnéia obstrutiva do sono a partir da análise de sinais de eletrocardiografia. Por último, o algoritmo será utilizado para detectar falhas em um atuador a partir de dados coletados de um simulador utilizado como *benchmark* de métodos de detecção e diagnóstico de falhas.

Finalmente, no capítulo 5, será apresentada uma análise das principais contribuições geradas nessa dissertação. Além disso, serão apresentados os próximos passos de pesquisa.

## Capítulo 2

# Revisão das Técnicas de Detecção de Novidades em Séries Temporais

Nesse capítulo, algumas das principais técnicas de detecção de novidades em séries temporais encontradas na literatura serão apresentadas. Todas as técnicas descritas contribuíram para o desenvolvimento e análise da técnica proposta nesta dissertação. As técnicas abordadas nesse capítulo podem ser divididas em duas classes distintas:

- técnicas que utilizam diretamente as observações da série para realizar a detecção através de algum algoritmo de classificação;
- técnicas que constroem um modelo de previsão a partir das observações normais e realizam a detecção a partir de características extraídas desse modelo.

Além disso, em Keogh et al. (2005) é proposta uma técnica de detecção de discórdias em séries temporais. Uma discórdia é definida como a subsequência da série que mais se difere do restante das subsequências, dada uma métrica de distância. Dessa maneira, esse algoritmo foi utilizado para detectar novidades em séries temporais, definindo como novidades as  $n$  subsequências com maior distância.

Todas as técnicas descritas nesse capítulo utilizam uma janela de pontos da série para construir um modelo de previsão ou para extrair características utilizadas na classificação.

Dada uma série temporal  $x(t)$ ,  $t = 1, \dots, N$ , define-se um tamanho da janela  $W$  e passo de deslocamento  $K$ . Dessa forma, a primeira janela de dados será formada pelos valores dos pontos no intervalo  $[1, W]$  e a segunda janela pelos pontos  $[1 + K, W + K]$  e assim por diante, formando a matriz:

$$\begin{bmatrix} x(1) & x(2) & \cdots & x(W) \\ x(1+K) & x(2+K) & \cdots & x(W+K) \\ \vdots & \vdots & \ddots & \\ x(N-W+1) & x(N-W+2) & \cdots & x(N) \end{bmatrix}$$

O valor de  $K$  pode variar entre 1 e  $W$ , sendo que, quando  $K = W$ , não há sobreposição de pontos nas janelas, ou seja, cada ponto pertence a apenas uma janela distinta.

## 2.1 Técnicas baseadas em classificação

Nessa seção serão apresentadas algumas técnicas de detecção de novidades em séries temporais que não utilizam modelos de previsão. Ao invés disso, os padrões de entrada de um algoritmo de classificação são gerados a partir dos pontos da série. Essas técnicas podem ser divididas em três grupos distintos:

- técnicas que geram artificialmente padrões de novidades e realizam a detecção através de métodos tradicionais de classificação de duas classes;
- técnicas baseadas em seleção negativa (*negative selection*) em que, a partir de dados normais, são gerados receptores capazes de reconhecer padrões de novidades;
- técnicas que realizam a detecção utilizando apenas padrões normais através de classificação de uma classe.

### 2.1.1 Classificação de duas classes

O método proposto em Oliveira et al. (2003) realiza a detecção de novidades em séries temporais utilizando uma técnica de classificação de duas classes.

Para isso, a partir do conjunto de pontos de treinamento da série tidos como normais, o método utiliza janelas deslizantes para gerar padrões de entrada normais. Dado o tamanho da janela  $W$  e um deslocamento  $K = 1$ ,  $N - W + 1$  janelas são geradas utilizando o algoritmo de janelas deslizantes descrito anteriormente.

Para cada uma das janelas geradas, define-se um *envelope* ao seu redor de forma que os pontos pertencentes às janelas que estejam dentro desse envelope são assumidos como normais e, conseqüentemente, os pontos relativos às janelas que estão fora do envelope são assumidos como novidades. O envelope é construído definindo-se um desvio máximo permitido  $p_1$  positivo ou negativo de cada ponto da janela. A figura 2.1 ilustra esse conceito.

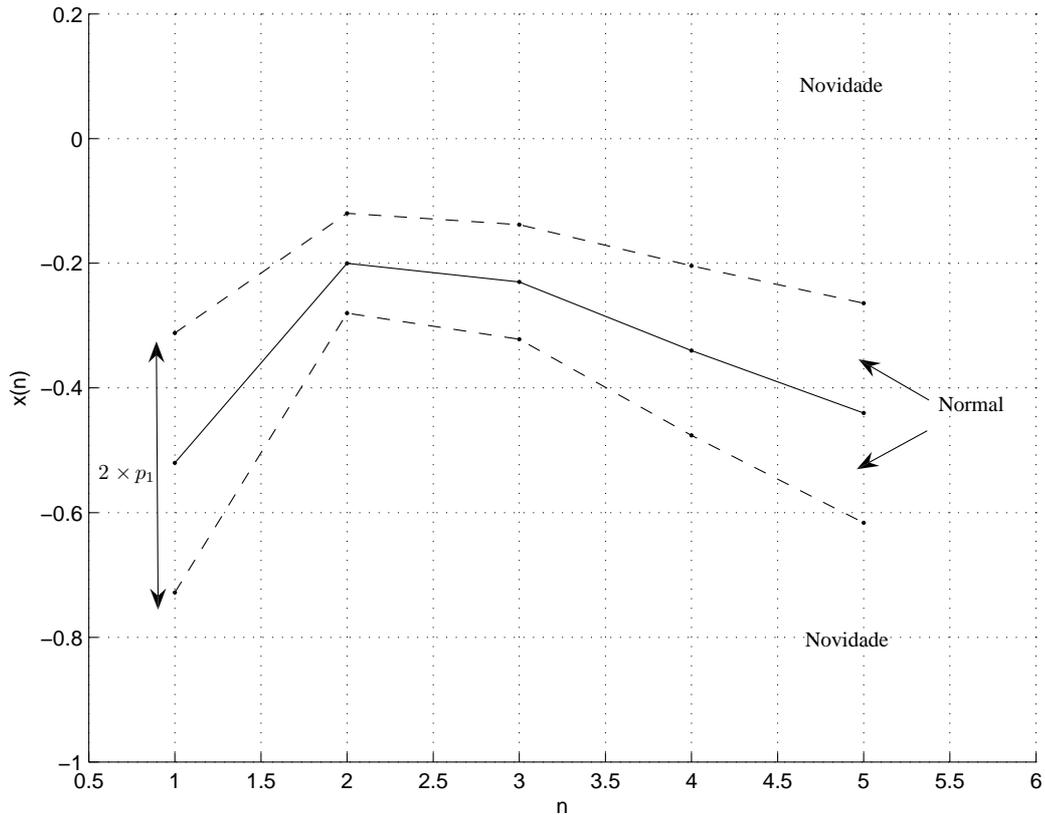


Figura 2.1: Envelope criado sobre uma janela de pontos para definir a região de normalidade

Definidas as regiões de normalidade e novidade, padrões artificiais são gerados. São geradas  $n$  janelas aleatórias, com todos os pontos na região de novidade e  $n - 1$  janelas com todos os pontos na região normal. Dessa maneira cada uma das duas classes terá  $n$  padrões para serem utilizados no treinamento. A figura 2.2 ilustra a criação de 3 padrões definidos como novidades e 2 definidos como normais a partir de uma janela de dados para  $p_1 = 20\%$ .

Finalmente, os padrões artificiais gerados são utilizados para treinar e validar o classificador. Qualquer classificador pode ser utilizado nessa etapa, em Oliveira et al. (2003) utilizou-se redes de base radial (RBF) (Haykin (1999)).

### 2.1.2 Seleção Negativa

Sistemas imunológicos artificiais são algoritmos baseados em princípios de sistemas imunológicos naturais. Existem diversas aplicações desses algoritmos, incluindo detec-

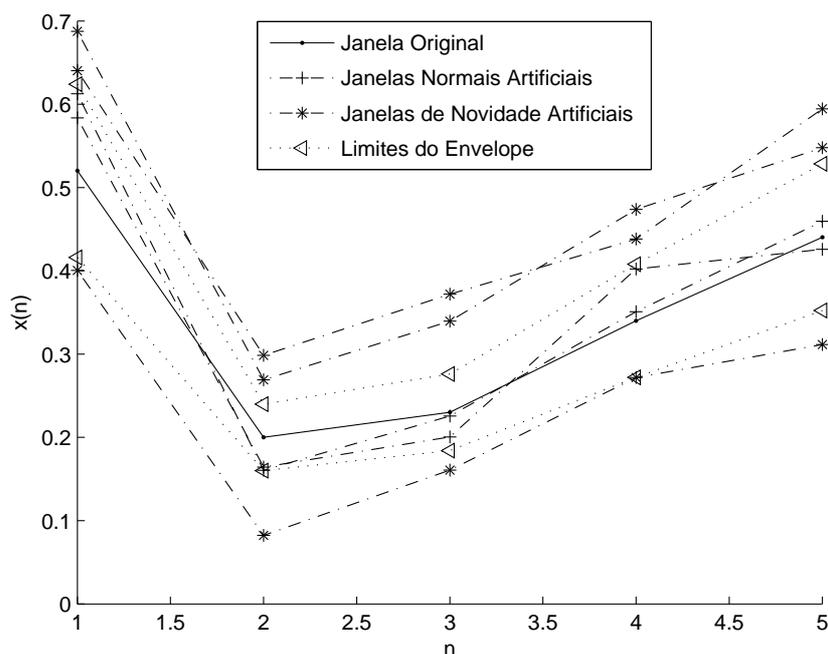


Figura 2.2: Exemplo de padrões artificiais criados a partir de uma janela de pontos normais utilizando-se um envelope

ção de invasão em redes de computadores (Dasgupta e Gonzalez (2002)) e detecção de ponto de mudança em séries temporais (Forrest et al. (1995)).

A técnica proposta em Dasgupta e Forrest (1996) utiliza sistemas imunológicos artificiais para detecção de novidades em séries temporais. Essa técnica baseia-se no princípio de discriminação *self non-self* do sistema imunológico de mamíferos. Esse mecanismo de discriminação é capaz de distinguir entre qualquer célula ou molécula exógena e as células pertencentes ao corpo. Essa discriminação é realizada em parte pelas células T, que possuem receptores na sua superfície capazes de detectar proteínas exógenas. Esses receptores são gerados por um processo pseudo-aleatório de rearranjo genético de forma que alguns desses receptores são capazes de se ligar com as células do corpo (*self*). Esses receptores são regulados pelo Timo. Apenas as células T cujos receptores não são capazes de se ligar com as células do corpo podem deixar o Timo e tornarem-se parte do sistema imunológico do organismo. Esse processo é definido como *seleção negativa*.

O algoritmo proposto em Forrest et al. (1994) baseado no processo de seleção negativa descrito acima é composto pelos seguintes passos:

- define-se *self* como um conjunto de  $S$  strings de tamanho fixo  $l$  representando o comportamento normal do sistema;

- gera-se um conjunto de receptores que não conseguem casar com as strings do conjunto *self*;
- monitora-se o sistema avaliando se os novos valores observados do sistema representados como strings de tamanho  $l$  casam com os receptores gerados. Caso aconteça um casamento, uma novidade é observada.

O conjunto *self* é gerado a partir de um conjunto de pontos relacionados ao funcionamento normal do sistema monitorado. Cada um desses pontos é normalizado e convertido para sua representação binária, dada uma precisão  $m$  e os valores de máximo e mínimo. Em seguida,  $S$  strings são geradas utilizando um algoritmo de janelas deslizantes. Para cada janela gerada, os pontos pertencentes à janela são convertidos para sua representação binária e concatenados, formando uma string binária de tamanho  $l$ . As janelas de pontos são geradas de forma que não haja sobreposição de pontos entre as janelas, ou seja, o valor do deslocamento  $K = W$ , sendo  $W$  o tamanho da janela.

Dado o conjunto *self*, representando o comportamento normal do sistema, geram-se então os receptores. Esses receptores são compostos por strings que não casam com as strings do conjunto *self*. Utiliza-se uma regra de casamento parcial, de forma que duas strings casam, se e somente se, são idênticas em  $r$  posições contínuas, sendo  $r$  um parâmetro do algoritmo. No algoritmo original, descrito em Forrest et al. (1994), os detectores são gerados de maneira aleatória por tentativa e erro, de forma que para cada receptor gerado, caso ele case alguma das  $S$  strings de *self* este é eliminado e um novo receptor é gerado até que  $R$  receptores sejam gerados. O diagrama presente na figura 2.3 ilustra a criação aleatória dos receptores. Um algoritmo mais eficiente é apresentado em D'haeseleer et al. (1996) capaz de gerar os receptores em tempo linear a  $S$ .

Uma vez gerados os receptores, o algoritmo é capaz de detectar novidades em padrões gerados a partir dos novos pontos da série. Dado um conjunto de pontos da série a serem classificados como normais ou novidades, geram-se strings de tamanho  $l$  de maneira análoga à geração do conjunto *self*. Para cada string gerada, caso um dos receptores case com essa string, todos os pontos da janela correspondente são classificados como novidades. O diagrama presente na figura 2.4 ilustra a etapa de detecção.

Em Gonzalez e Dasgupta (2002) é proposta uma representação real dos receptores que são representados como vetores em uma dimensão  $R^W$ , em que  $W$  é o tamanho da janela. Cada receptor gerado possui um raio de detecção e uma novidade é encontrada caso um padrão esteja dentro do seu raio de detecção. Essa representação foi motivada devido a dificuldade de se extrair informações no domínio do problema a partir dos receptores gerados com codificação binária, dificultando a análise das causas das falhas.

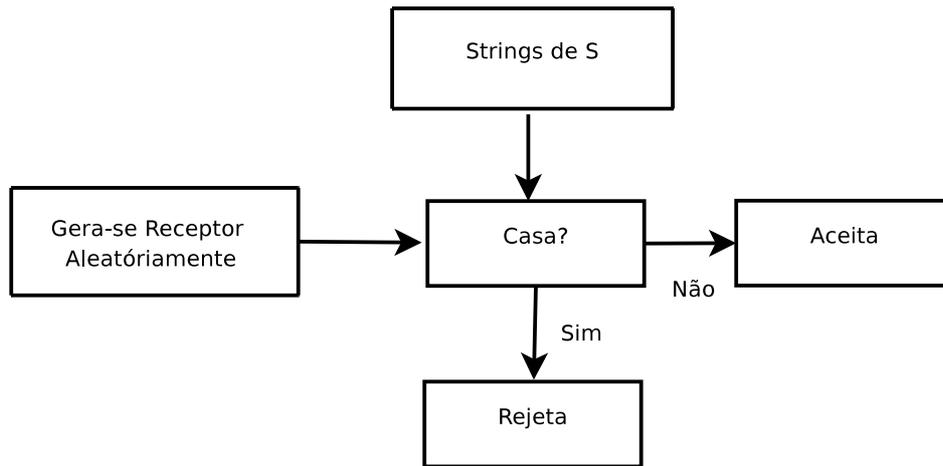


Figura 2.3: Criação dos receptores do algoritmo imunológico artificial

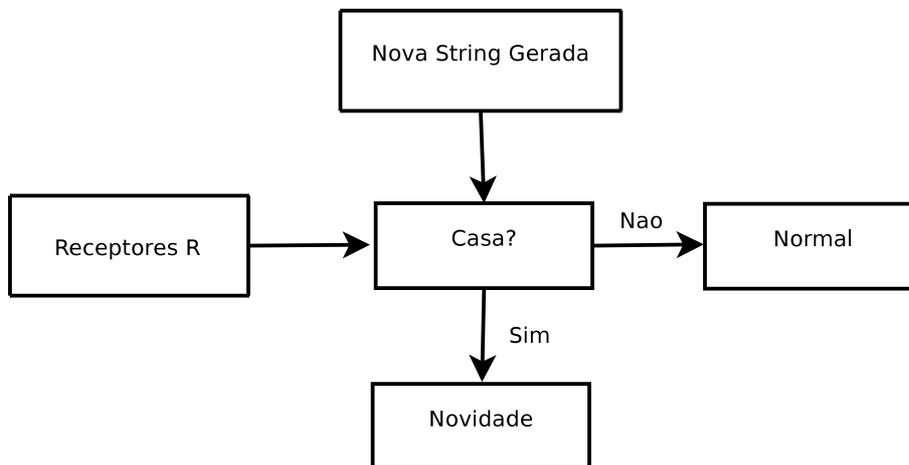


Figura 2.4: Etapa de detecção do algoritmo imunológico artificial

Essa nova representação serviu como motivação no desenvolvimento de um sistema híbrido (Gonzalez e Dasgupta (2002)) que utiliza a seleção negativa em conjunto com um classificador de duas classes para detecção de novidades em séries temporais. Inicialmente são gerados receptores com codificação real a partir de dados normais da série. Em seguida, um classificador de duas classes é treinado para distinguir entre o comportamento normal, representado pelos pontos normais, e novidades, representados pelos receptores gerados. A figura 2.5 ilustra o diagrama do método híbrido.

Conforme descrito anteriormente, o método de seleção negativa e o método de classificação de duas classes possuem um sério problema no sentido de que dificilmente os padrões anômalos ou os receptores gerados conseguirão cobrir toda a área de anormalidade.

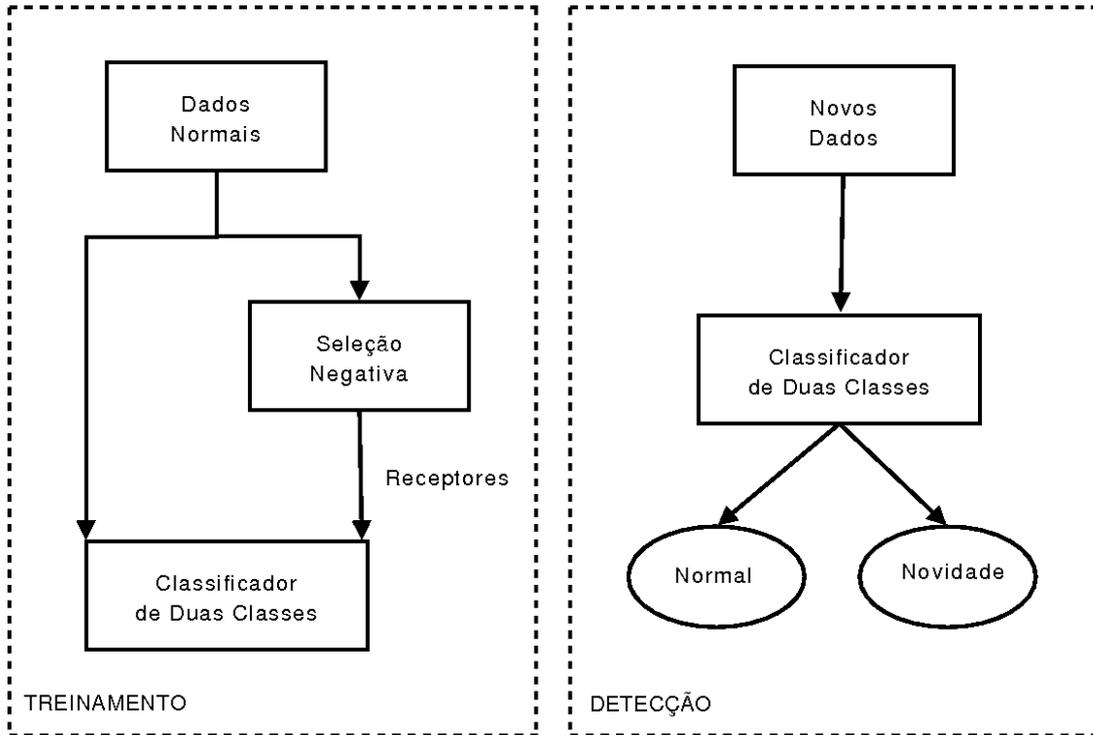


Figura 2.5: Método híbrido que utiliza seleção negativa e classificação de duas classes

O método de seleção negativa também possui um problema de alto custo computacional, pois para cada novo padrão a ser classificado, este deve ser avaliado por todos os receptores criados.

### 2.1.3 Classificação de Uma Classe

A técnica proposta em Ma e Perkins (2003b) utiliza o método de classificação de uma classe utilizando máquinas de vetores suporte para realizar detecção de novidades em séries temporais.

Dado um conjunto de pontos da série tidos como normais  $x(t)$  para  $t = 1, \dots, N$ , gera-se um conjunto de vetores de dimensão  $W$  utilizando um algoritmo de janelas deslizantes com o valor do passo  $K = 1$ :

$$x_W(t) = [x(t - W + 1) \ x(t - W + 2) \ \dots \ x(t)] \quad (2.1)$$

Dessa maneira pode-se converter toda a série temporal em um conjunto de vetores  $T_W(N) = \{x_W(t)\}$  para  $t = W, \dots, N$ . O espaço de dimensão  $W$  em que esses vetores são representados foi denominado *espaço de fase*. Dado esse conjunto de vetores, aplica-se então uma função não-linear  $\Phi(x)$  a cada um destes, de forma a mapeá-los em um espaço de alta dimensão.

A partir desses vetores definidos como normais, constrói-se um hiperplano de separação no espaço de fase definido por:

$$f(x) = W\Phi(x) - b \quad (2.2)$$

Esse hiper-plano é construído de forma que, dado um vetor correspondente a uma sequência de valores da série a serem classificados, caso o valor de  $f(x) < 0$ , esse vetor é definido como um *outlier* e todos os pontos correspondentes são definidos como novidades. O mapeamento realizado por  $\Phi(x)$  é utilizado para que o hiperplano criado tenha maior probabilidade de separar corretamente os vetores normais dos *outliers* (Cover (1965)).

Quando a série temporal é composta basicamente por componentes de baixa frequência, os vetores no espaço de fase estão distribuídos em torno do vetor diagonal  $\mathbf{1} = [1 \ 1 \ \dots \ 1]^T$ . Nesse cenário, o desempenho do detector pode ser gravemente degradado. Para resolver esse problema, em Ma e Perkins (2003b) é proposta uma representação dos vetores em um espaço denominado *espaço de fase projetado*, de forma que os pontos do espaço de fase são projetados para esse espaço através da seguinte transformação:

$$x_W(t)' = \left( I_d - \frac{1}{W} \mathbf{1}\mathbf{1}^T \right) x_W(t) \quad (2.3)$$

sendo  $I_d$  a matriz identidade e  $x_W(t)'$  o vetor no espaço de fase projetado.

O tamanho da dimensão  $W$  utilizado para criar os vetores a serem classificados é decisivo no sucesso do método. Dessa maneira, para tornar o método mais robusto, os pontos são projetados em  $S$  espaços de diferentes dimensões e a classificação final do ponto é definida por:

$$I(t) = \prod_{W \in S} i(W, t) \quad (2.4)$$

sendo  $i(W, t) \in [0, 1]$  a classificação do ponto no instante  $t$  no espaço de fase de dimensão  $W$ , de forma que uma novidade é definida quando  $i(W, t) = 1$ .

Uma técnica semelhante é proposta em Tax (2001). Esse trabalho também utiliza máquinas de vetores suporte para estimar os limiares de normalidade, porém ao invés de se utilizar um hiperplano de separação utiliza-se uma hiperesfera. Esse trabalho lida com o problema de classificação de uma classe e um dos experimentos realizados realiza a detecção de novidades em uma série temporal gerada através de amostras de vibração de uma bomba de água em que as novidades a serem detectadas são associadas a possíveis falhas no sistema. Nesse trabalho, ao invés de utilizar direta-

mente uma janela de pontos como padrão de entrada do classificador de uma classe, utiliza-se características extraídas de cada janela, como os parâmetros de um modelo autoregressivo, o espectro de potência, entre outros.

Vale ressaltar que, apesar das técnicas descritas utilizarem classificação de uma classe através de máquina de vetores suporte, qualquer outra metodologia de classificação de uma classe poderia ser utilizada para classificar os pontos no espaço de fase (projetado).

## 2.2 Técnicas baseadas em Modelos de Previsão

Nessa seção serão apresentadas técnicas de detecção de novidades em séries temporais baseadas em modelos de previsão. Todas estas técnicas necessitam de um conjunto de pontos assumidos como normais para realizar o treinamento e validação do modelo.

Construído o modelo a partir de dados da série tidos como normais, os métodos a serem descritos podem ser divididos em dois grupos distintos:

- métodos baseados na variação do erro de previsão;
- métodos baseados na variação da estrutura do modelo.

### 2.2.1 Variação do Erro de Previsão

O método proposto em Oliveira e Meira (2006) utiliza um modelo de previsão construído a partir de dados da série tidos como normais para prever o valor atual da série. Construído o modelo, calcula-se um limite de tolerância para o erro de previsão do modelo.

Uma vez calculado os limites de tolerância, para cada novo ponto da série a ser avaliado, caso o erro de previsão esteja fora desses limites, o ponto é classificado como uma novidade.

Utilizou-se como modelo de previsão redes neurais artificiais treinadas para realizar previsão do valor atual da série baseado nos valores passados observados. Existem diversas topologias de redes que podem ser utilizadas para realizar essa operação, uma revisão das principais topologias pode ser encontrada em Dorffner (1996).

Em seguida, utilizou-se a técnica de *intervalos robustos de confiança* (Masters (1995)) para calcular os limites de tolerância do erro de previsão.

A técnica de intervalos robustos de confiança calcula numericamente intervalos de confiança da distribuição do erro de previsão, sem que seja necessário definir a forma de sua distribuição de probabilidade, apenas assume-se que esta distribuição é simétrica. Para isso, esta utiliza a *função de distribuição amostral* do erro de previsão,  $S_n(e)$ .

Dado um conjunto de  $n$  valores do erro de previsão ordenados, essa função retorna a fração dessa coleção menor ou igual a um dado valor do erro (Masters (1995)):

$$S_n(e) = \begin{cases} 0, & e < e(1) \\ r/n, & e(r) \leq e \leq e(r+1) \\ 1, & e(n) \leq e \end{cases} \quad (2.5)$$

Caso o conjunto de pontos utilizados para calcular  $S_n(e)$  represente os valores de erros encontrados na prática e o tamanho desse conjunto seja suficientemente grande,  $S_n(e)$  é assumido ser uma boa estimacão da verdadeira função de distribuição cumulativa de probabilidade do erro de previsão  $F(e)$ .

Dessa maneira, para se calcular o intervalo de confiança do erro de previsão dado um nível de significância  $\alpha$ , deve-se descartar  $n \times \alpha/2$  valores de cada extremo do conjunto de valores de  $e(t)$  e os limites do intervalo de confiança serão os valores limites do conjunto resultante. Para um conjunto de amostras de erro relativamente pequeno descarta-se  $n \times \alpha/2 - 1$  (Masters (1995)).

Esse método foi proposto tendo como principal problema a ser tratado a detecção de fraudes em sistemas financeiros. O principal problema das séries utilizadas é que essas séries são formadas por poucos pontos, dessa maneira dificulta-se a estimativa precisa do intervalo. Para solucionar esse problema sugere-se a utilização de um comitê de redes baseadas na promediação (Haykin (1999)) como modelo de previsão e, ao invés de coletar os erros a partir do valor de saída do modelo, utiliza-se a saída de cada uma das redes que compõem o comitê. Dessa maneira, caso o comitê seja composto por  $N$  redes, o conjunto de amostras do erro de previsão será  $N$  vezes maior que o conjunto gerado a partir de uma única rede.

O intervalo robusto de confiança deve ser calculado utilizando-se um conjunto de pontos diferente do utilizado para treinar o modelo. Porém, como séries financeiras em geral possuem poucos pontos, utilizou-se o conjunto de treinamento e validação do modelo de previsão. Segundo Masters (1995), o conjunto de treinamento pode ser utilizado caso tome-se cuidado em evitar *overfitting* (Haykin (1999)) no treinamento.

Dois tipos de experimentos foram utilizados para validar o método proposto. No primeiro experimento utilizaram-se séries temporais financeiras que não possuíam novidades, para mostrar que o método é capaz de construir os intervalos de confiança do erro corretamente. Em seguida utilizaram-se séries temporais fisiológicas contendo novidades detectadas por especialistas.

Conforme descrito anteriormente, esse método foi desenvolvido tendo como principal problema a ser tratado a detecção de fraudes em sistemas financeiros, em que as séries temporais possuem poucos pontos e a novidade a ser encontrada caracteriza-se por

apenas um ponto que não pode ser descrito pelo modelo de previsão.

Em geral, a novidade a ser encontrada não se caracteriza por apenas um ponto, e sim por uma sequência de pontos cuja frequência ou estrutura diferem do padrão assumido como normal (Keogh et al. (2002)). Dessa forma, para se utilizar essa metodologia na detecção de novidades em séries temporais com essa característica, deve-se realizar um processamento do resultado através de uma janela de pontos para que novidades formadas por uma sequência de pontos sejam detectadas corretamente.

O algoritmo proposto no capítulo 3 assemelha-se com essa técnica, no sentido que utiliza a variação do erro de previsão na detecção de novidades, porém essa variação é processada utilizando-se uma janela de pontos desse resultado. Dessa forma, nesse capítulo será apresentada uma extensão da metodologia proposta em Oliveira e Meira (2006), para que esta possa ser utilizada em séries temporais com novidades caracterizadas por uma sequência de pontos.

### 2.2.2 Variação da Estrutura do Modelo

Em Ma e Perkins (2003a), uma metodologia de detecção de novidades em séries temporais é proposta através da análise da variação dos parâmetros do modelo ao longo do tempo. Inicialmente, o modelo é treinado utilizando um conjunto de pontos normais. Em seguida, cada novo ponto a ser classificado é utilizado para atualizar o modelo e uma novidade é detectada quando vários pontos consecutivos geram alterações na sua estrutura.

O modelo de previsão utilizado é baseado em regressão via máquinas de vetores suporte (SVR) (Smola e Schoelkopf (1998)). Dado um conjunto de treinamento  $T = \{(x_i, y_i), i = 1, \dots, N\}$ , sendo  $x_i \in R^D$  e  $y_i \in R$ , constrói-se a seguinte função de regressão linear:

$$f(x) = W\Phi(x) + b \tag{2.6}$$

onde  $\Phi(x)$  é uma função que mapeia os valores de  $x \in R^D$  para um vetor em um espaço  $F$  de alta dimensão e  $W$  e  $b$  são vetores no espaço  $F$ .

Realizada a estimação dos parâmetros desse modelo, apenas um subconjunto dos pontos utilizados no treinamento é necessário para representar os parâmetros. Esse subconjunto de padrões é definido como vetores suporte.

Uma vez treinado o modelo, para cada novo ponto da série observado  $x(t)$ , utiliza-se o par  $T_D(t) = ([x(t - D + 1), \dots, x(t - 1)], x(t))$  para atualizar o modelo de regressão. Caso algum dos pontos utilizados na atualização do modelo passe a fazer parte dos vetores suporte, esse ponto é definido como uma *surpresa*.

Realizada a atualização do modelo de previsão para todos os pontos da série a serem analisados, caso um ponto seja definido como uma surpresa, seu valor de *ocorrência*  $O(t)$  será definido como 1, ou seja,  $O(t) = 1$ . Caso contrário  $O(t) = 0$ .

Os valores de ocorrência são então processados em conjunto. Dado um tamanho de janela  $W$  define-se um *evento* formado por  $W$  valores de ocorrência  $E(t) = [O(t) \ O(t+1) \ \dots \ O(t+W-1)]$  e a *norma de magnitude um do evento* é definida como a soma dos seus valores de ocorrência  $|E(t)| = \sum_{i=1}^{W-1} O(t+i)$ .

Define-se que a função de distribuição de probabilidade da norma dos eventos  $p(|E(t)|)$  é uma distribuição Binomial e, para cada um dos eventos gerados, o evento é definido como *evento de novidade*, caso:

- $|E(t)| > \max(h, E[|E(t)|])$ , ou seja, a norma do evento seja maior que a média da distribuição ou do que o limite inferior,  $h \in N$ ;
- $p(|E(t)|) < 1 - \alpha$ , sendo  $\alpha$  o nível de confiança definido como parâmetro do algoritmo.

Finalmente, caso um evento seja definido como evento de novidade, todos os pontos relativos a esse evento são definidos como novidades.

Essa técnica possui uma vantagem de ser capaz de detectar novidades em séries temporais não estacionárias devido ao modelo temporal adaptativo utilizado. Porém, todos os pontos observados são utilizados para atualizar os parâmetros do modelo, sendo esses pontos classificados como normais ou novidades. Na opinião do presente autor, deveria-se utilizar apenas os pontos classificados como normais para atualizar o modelo, pois utilizando-se também os pontos classificados como novidades, o desempenho preditivo do modelo pode ser deteriorado.

Os conceitos descritos acima são utilizados na técnica proposta no próximo capítulo 3. Dessa maneira, esses conceitos serão descritos detalhadamente nesse capítulo.

## 2.3 Detecção de Discórdias

O trabalho descrito em Keogh et al. (2005) propõe uma técnica de detecção de discórdias em séries temporais que pode ser utilizada como detecção de novidades.

Conforme descrito anteriormente, uma discórdia é definida como a subsequência da série que mais se difere do restante das subsequências dada uma métrica de distância (Keogh et al. (2005)). Dessa maneira, esse algoritmo foi utilizado para detectar novidades em séries temporais definindo como novidades as  $n$  subsequências com maior distância.

Inicialmente, utiliza-se um algoritmo de janelas deslizantes com  $K = 1$  para gerar  $N - W + 1$  janelas de pontos. Em seguida, cada uma das janelas é convertida para uma representação simbólica utilizando um método denominado SAX (*Symbolic Aggregate Approximation*) (Lin et al. (2003)) com o objetivo de reduzir a dimensionalidade dos dados. A figura 2.6 ilustra uma série convertida para essa representação.

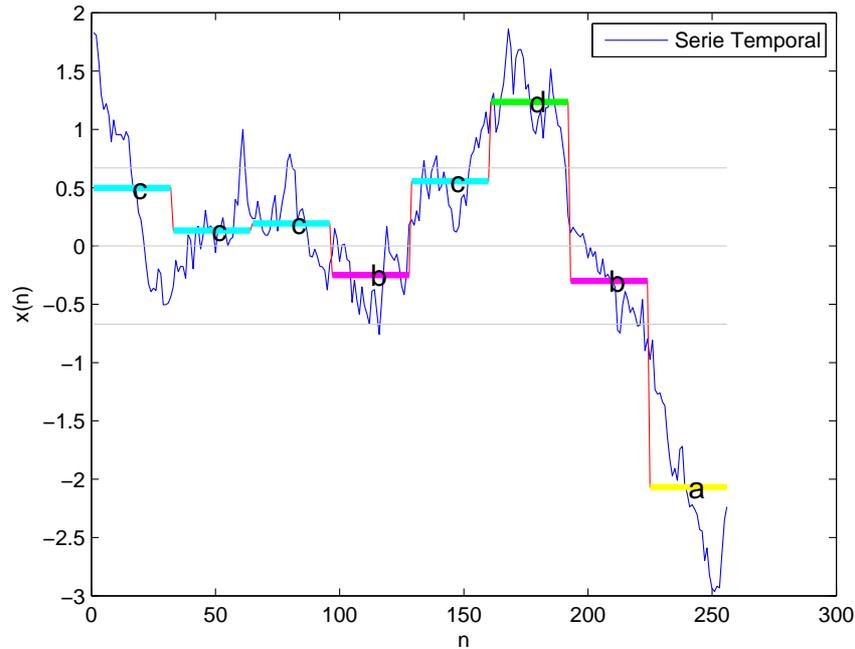


Figura 2.6: Série temporal e sua representação simbólica realizada através do método SAX. A série em questão é representada pela sequência cccbcdba

Cada uma das janelas convertidas para a representação simbólica é definida como uma subsequência. Calcula-se então a distância entre cada subsequência e sua subsequência mais similar ignorando as distâncias calculadas entre subsequências com sobreposição de dados. Dessa forma a subsequência que possuir maior distância é definida como a discórdia. Utilizou-se a distância Euclidiana como métrica de distância:

$$Dist(s1, s2) = \sqrt{\sum_{t=1}^W (s1(t) - s2(t))^2} \quad (2.7)$$

em que  $s1$  e  $s2$  são duas subsequências.

As subsequências são então ordenadas decrescentemente pela distância calculada, de forma que a  $k$ -ésima discórdia é dada pela subsequência com a  $k$ -ésima maior distância.

Um algoritmo simples para a detecção de discórdias seria formado por dois *loops*, o

primeiro percorrendo todas as possíveis subsequências e o segundo medindo a distância entre subsequência atual e as outras subsequências para encontrar a menor distância. Esse algoritmo tem ordem de complexidade  $O(m^2)$  e seria impraticável na detecção de discórdias em séries temporais longas. O algoritmo proposto realiza um ganho no desempenho de 3 a 4 ordens de magnitude armazenando as subsequências extraídas de forma ordenada.

Esse algoritmo foi utilizado na detecção de novidades em séries temporais fisiológicas e em sistemas dinâmicos. Em todos os experimentos realizados sabia-se a priori o número de novidades a serem encontradas. Dessa forma, caso a série possua  $k$  novidades, essas seriam relacionadas às  $k$  primeiras discórdias.

Porém na prática não se sabe o número de novidades que o algoritmo deve encontrar, dessa maneira, esse algoritmo carece de uma metodologia de determinação do limiar de distância, de forma que, subsequências que ultrapassem o limiar sejam classificadas como novidades.

## Capítulo 3

# Detecção de Novidades em Séries Temporais Utilizando Modelos de Previsão

Nesse capítulo serão apresentadas duas técnicas de detecção de novidades em séries temporais baseadas em modelos de previsão.

As duas técnicas propostas baseiam-se em um algoritmo genérico de detecção de novidades em séries temporais baseados em um modelo de previsão. Inicialmente, um modelo de previsão é treinado e validado a partir de um conjunto de pontos da série assumidos normais. Em seguida, para cada novo ponto da série a ser classificado como normal ou novidade, calcula-se o erro de previsão associado a esse ponto e, caso esse erro ultrapasse um determinado limiar, o ponto é definido como uma surpresa. Finalmente aplica-se um processamento utilizando uma janela de valores para se chegar a classificação final de cada ponto.

A primeira técnica é baseada em análise estatística de séries temporais e utiliza um modelo de previsão autorregressivo e intervalos de previsão calculados analiticamente como limiares do erro de previsão.

A segunda técnica utiliza redes neurais como modelo de previsão e a técnica de *intervalos robustos de confiança* (Masters (1995)) como limites do erro de previsão.

Conforme dito anteriormente, alguns dos conceitos utilizados no desenvolvimento desse algoritmo foram baseados no *framework* definido em Ma e Perkins (2003a). Porém foram realizadas algumas alterações nesses conceitos para serem utilizados no algoritmo proposto.

Esse capítulo está dividido da seguinte maneira: inicialmente, na seção 3.1, o algoritmo genérico que serve como base das duas técnicas propostas será apresentado. Em seguida, a técnica de detecção de novidades proposta baseada em modelos de previsão

autorregressivos será apresentada na seção 3.2. Finalmente, a técnica baseada em redes neurais será descrita na seção 3.3

### 3.1 Algoritmo Genérico

Nessa seção será apresentado um algoritmo genérico de detecção de novidades em séries temporais baseado em um modelo de previsão.

Esse algoritmo é denominado genérico, pois não define o modelo de previsão a ser utilizado. Dessa maneira, pode-se utilizar a modelagem mais conveniente para a série. São definidos alguns conceitos que o algoritmo utiliza na sua execução e alguns desses conceitos representam partes do algoritmo que variam de acordo com a escolha do modelo de previsão. Assim, esses conceitos devem ser definidos após a escolha do modelo atendendo a alguns pré-requisitos.

Inicialmente constrói-se um *modelo de previsão*  $M_x(t)$  responsável por prever o valor atual da série. Esse modelo pode ser um modelo físico, construído a partir do conhecimento prévio do sistema descrito pela série, ou um modelo construído apenas a partir dos dados da série.

O modelo pode utilizar um conjunto de dados de treinamento formado por pontos da série tidos como normais para estimar seus parâmetros e um conjunto de dados de validação, também formado por pontos normais, para validá-los.

A partir do modelo, calcula-se o erro de previsão  $e(t)$  relativo ao conjunto de pontos a serem classificados como normais ou novidades. Dado o valor observado da série  $x(t)$  e o valor previsto do modelo para o instante atual  $\hat{x}(t)$ , tem-se que:

$$e(t) = \hat{x}(t) - x(t) \quad (3.1)$$

Deve-se então definir um *intervalo de tolerância*,  $[\varepsilon_1, \varepsilon_2]$ , para os valores do erro de previsão a partir de um nível de significância  $\alpha \in [0, 1]$ , definido como parâmetro do algoritmo. Definido esse intervalo, para todos os pontos a serem classificados, calcula-se sua *ocorrência*:

$$O(t) = \begin{cases} 0, & \text{para } e(t) \in [\varepsilon_1, \varepsilon_2] \\ 1, & \text{caso contrário} \end{cases} \quad (3.2)$$

Uma *surpresa* é observada se  $O(t) = 1$ , ou seja, uma surpresa ocorre quando o valor de  $e(t)$  está fora do intervalo de tolerância.

Um *evento*  $E_n(t)$  é definido como uma sequência de valores de ocorrência:

$$E_n(t) = [O(t) \ O(t+1) \ \dots \ O(t+n-1)] \quad (3.3)$$

sendo  $n$  a duração do evento, definido como um parâmetro do algoritmo.

A partir dos valores de ocorrência e dado o tamanho do evento  $n$ ,  $m - n + 1$  eventos são gerados utilizando-se uma janela deslizante, sendo  $m$  o número total de pontos da série a serem classificados. Dessa forma, o primeiro evento contém os valores de ocorrência relativos aos pontos no intervalo  $[1, n]$ , o segundo contém os valores relativos ao intervalo  $[2, n + 1]$ , assim por diante, formando o seguinte sistema de equações:

$$\begin{aligned} E_n(1) &= [O(1) & O(2) & \cdots & O(n-1)] \\ E_n(2) &= [O(2) & O(3) & \cdots & O(n)] \\ &\vdots \\ E_n(m-n+1) &= [O(m-n+1) & O(m-n+2) & \cdots & O(m)] \end{aligned}$$

Para cada um dos eventos gerados, calcula-se a *norma de magnitude 1 do evento*  $|E_n(t)|$  definida como:

$$|E_n(t)| = \sum_{i=0}^{i=n-1} O(t+i) \quad (3.4)$$

Essa grandeza mede o número de surpresas que ocorreram no evento  $E_n(t)$ .

A função de probabilidade de  $|E_n(t)|$  é representada como  $p(|E_n(t)|)$  para  $|E_n(t)| = 0 \cdots n$ . Para se definir a forma de  $p(|E_n(t)|)$ , assume-se que  $E_n(t)$  é uma sequência de variáveis aleatórias independentes de Bernoulli com a mesma probabilidade de sucesso  $q$ . A probabilidade de sucesso corresponde a probabilidade de que o erro de previsão esteja fora dos intervalos de tolerância, ou seja, a probabilidade de ocorrência de uma surpresa. Dessa maneira a função de densidade de probabilidade discreta de  $|E_n(t)|$  pode ser descrita como uma distribuição Binomial (Papoulis (1991)):

$$p(|E_n(t)|) = \begin{cases} \binom{n}{|E_n(t)|} q^{|E_n(t)|} (1-q)^{n-|E_n(t)|} & \text{para } |E_n(t)| = 0 \cdots n \\ 0 & \text{caso contrário} \end{cases} \quad (3.5)$$

Dado o nível de significância  $\alpha$ , um evento  $E_n(t)$  é definido como um *evento de novidade* se o valor de  $|E_n(t)|$  for maior que o limite do intervalo de confiança unilateral construído a partir de  $p(|E_n(t)|)$ .

Para se calcular o intervalo de confiança unilateral de  $p(|E_n(t)|)$  deve-se encontrar o menor valor inteiro  $\gamma$ , de forma que a função de probabilidade acumulada de  $|E_n(t)|$  avaliada em  $\gamma$  seja igual ou exceda  $1 - \alpha$ . O algoritmo 1 ilustra o pseudocódigo utilizado

para calcular o valor de  $\gamma$  e a figura 3.1 ilustra o valor de  $\gamma$  para  $q = 0.05$  e  $n = 300$  e  $\alpha = 0.05$ .

loa 1: Pseudocódigo utilizado no cálculo do valor de  $\gamma$  dada a probabilidade de ocorrência de uma surpresa, o tamanho do evento e o nível de significância

```

1: function GAMMA( $q, n, \alpha$ )
2:    $cdf \leftarrow 0$ 
3:   for  $k \leftarrow 0, n$  do
4:      $cdf \leftarrow cdf + \binom{n}{k} q^k (1 - q)^{n-k}$ 
5:     if  $cdf \geq 1 - \alpha$  then
6:       break
7:     end if
8:   end for
9:   return  $k$ 
10: end function

```

Dadas as ocorrências e o limiar do intervalo de confiança  $p(|E_n(t)|)$ , os eventos são classificados como normais ou eventos de novidades. Caso o evento seja classificado como evento de novidade, o último ponto pertencente ao evento é classificado como novidade, ou seja, caso o evento  $E_n(t_i)$  seja classificado como um evento de novidade, o ponto  $x(t_i + n - 1)$  é definido como novidade.

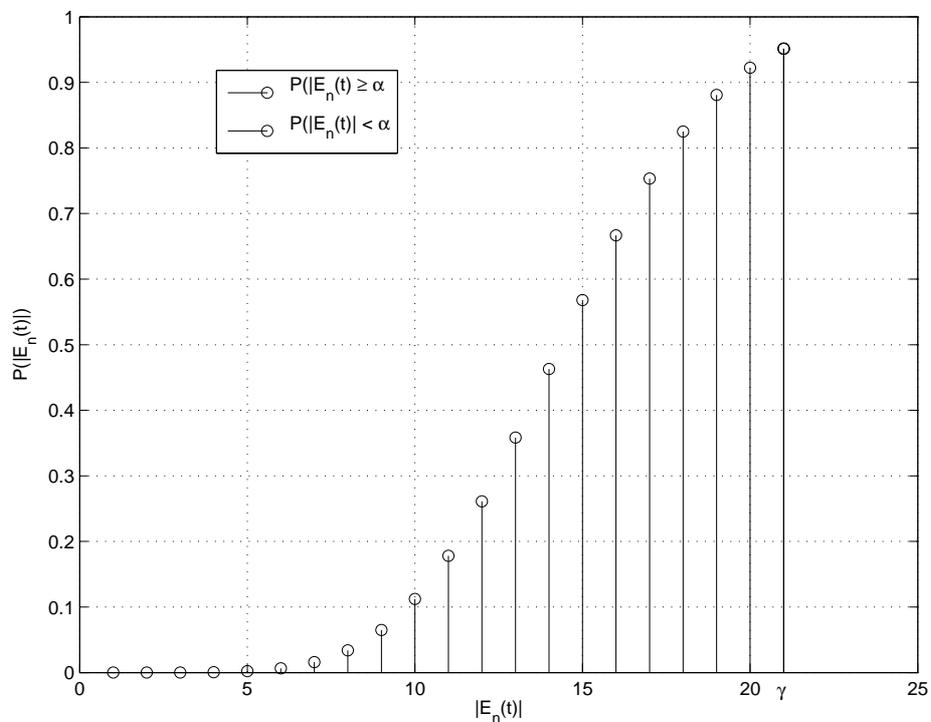


Figura 3.1: Valor de  $\gamma$  para  $q = 0.05$  e  $n = 300$

Para se utilizar o algoritmo proposto os seguintes conceitos devem ser implementados:

- $M_x(t)$ , o modelo responsável por descrever o conhecimento adquirido da série.
- $[\varepsilon_1, \varepsilon_2]$ , o intervalo de tolerância para valores de  $e(t)$ .
- $q$ , a probabilidade de ocorrência de uma surpresa.

Além disso, o algoritmo proposto possui os seguintes parâmetros:

- $\alpha$ , o nível de significância utilizado para calcular o intervalo de tolerância de  $e(t)$  e o intervalo unilateral de confiança de  $p(|E_n(t)|)$ .
- $n$ , o tamanho do evento, ou seja, o tamanho da janela utilizada para processar as ocorrências relativas a cada ponto da série.

O algoritmo 2 apresenta o pseudocódigo da etapa de classificação do algoritmo proposto.

Em seguida um exemplo será apresentado para ilustrar o funcionamento do algoritmo.

### 3.1.1 Exemplo

Dada a série temporal ilustrada na figura 3.2 formada por 80 pontos, sabe-se que os primeiros 60 pontos representam seu comportamento normal e deseja-se classificar seus últimos 20 pontos como normais ou novidades.

Inicialmente deve-se construir um modelo de previsão  $M_x(t)$  para a série. Utilizou-se como modelo de previsão um polinômio de primeira ordem cujos coeficientes foram estimados através de uma regressão linear realizada utilizando os primeiros 30 pontos da série:

$$M_x(t) = \hat{x}(t) = 1.203t + 0.4788 \quad (3.6)$$

Em seguida, definiu-se que o intervalo de tolerância do erro de previsão será:

$$[\varepsilon_1, \varepsilon_2] = \hat{x}(t) \pm 0.05\hat{x}(t) \quad (3.7)$$

Ou seja, caso a diferença entre o valor previsto do modelo e o valor observado para um dado instante  $t$  ultrapasse 5% do valor previsto,  $O(t) = 1$ , isto é, uma surpresa é observada no instante  $t$ .

loa 2: Pseudocódigo da etapa de classificação do algoritmo genérico de detecção de novidades proposto

```

1: function NOVELTY_DETECTOR( $x, M_x(t), [\varepsilon_1, \varepsilon_2], \gamma, n, \alpha$ )
2:    $m = \text{length}(x)$ 
3:   for  $t \leftarrow 1, m$  do
4:      $e(t) \leftarrow M_x(t) - x(t)$ 
5:     if  $e(t) \in [\varepsilon_1, \varepsilon_2]$  then
6:        $O(t) \leftarrow 1$ 
7:     else
8:        $O(t) \leftarrow 0$ 
9:     end if
10:  end for
11:  for  $t \leftarrow 1, m - n + 1$  do
12:     $|E_n(t)| \leftarrow 0$ 
13:    for  $i \leftarrow t, t + n - 1$  do
14:       $|E_n(t)| \leftarrow |E_n(t)| + O(i)$ 
15:    end for
16:    if  $|E_n(t)| > \gamma$  then
17:       $\text{class}(t + n - 1) \leftarrow 1$ 
18:    else
19:       $\text{class}(t + n - 1) \leftarrow 0$ 
20:    end if
21:  end for
22:  return  $\text{class}$ 
23: end function

```

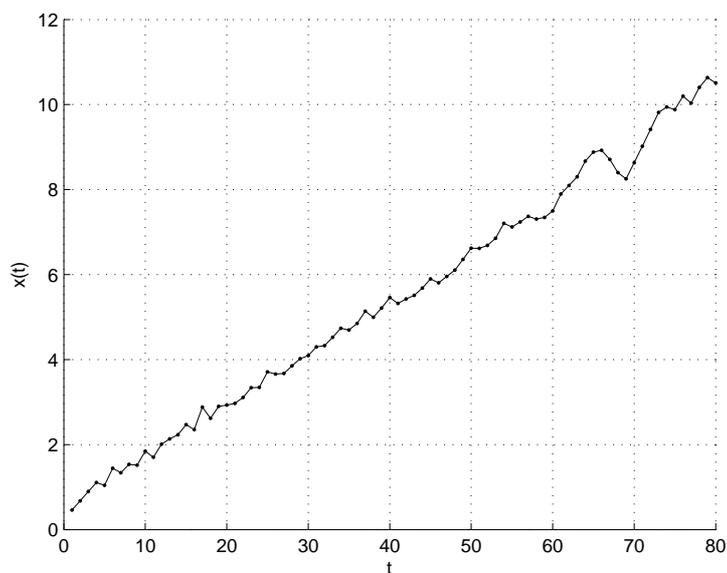


Figura 3.2: Série temporal utilizada como exemplo do algoritmo de detecção de novidades

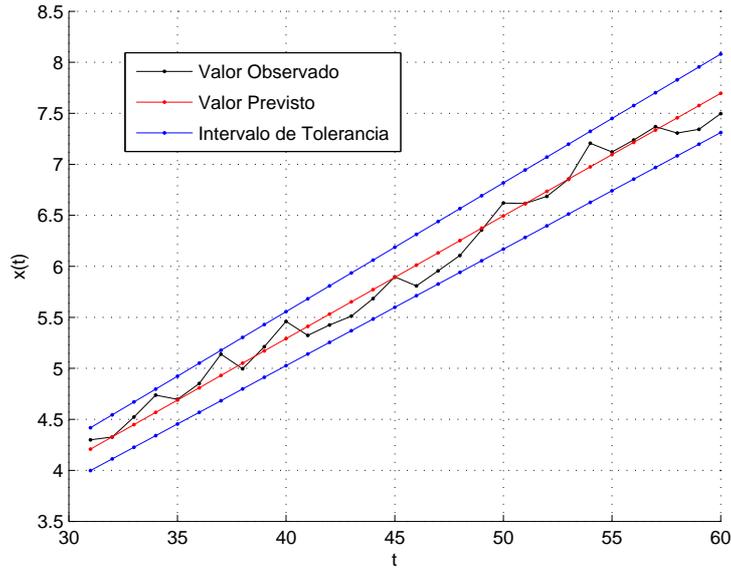


Figura 3.3: Pontos da série utilizados para validação do modelo

A figura 3.3 ilustra os valores da série, o valor previsto e os intervalos de tolerância no intervalo temporal  $t = [31, 60]$ . Analisando essa figura, percebe-se que a diferença entre o valor observado e o previsto da série nesse intervalo temporal não ultrapassa os limites do intervalo de tolerância. Sabe-se que os pontos desse intervalo são definidos como normais, sendo assim, valida-se o modelo proposto.

Finalmente, os pontos do intervalo  $[61, 80]$  serão utilizados como entrada do algoritmo de classificação. A figura 3.4 ilustra o valor observado e previsto dos pontos nesse intervalo assim como os limites do intervalo de tolerância. Percebe-se que alguns dos valores observados estão fora do intervalo de tolerância definido, sendo assim, gerou-se o seguinte vetor de ocorrências  $[O(1), \dots, O(20)]$ :

$$\left[ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \right]$$

Assumindo-se um tamanho de evento  $n = 5$ , foram gerados 16 eventos ( $m - n + 1 = 20 - 5 + 1 = 16$ ). A tabela 3.1 representa os eventos gerados:

A norma de cada um dos eventos é representada pela tabela 3.2.

Assume-se hipoteticamente que a probabilidade de ocorrência de uma surpresa é de 10%,  $q = 0.1$ . Dessa maneira, o limite do intervalo de confiança unilateral de 5% ( $\alpha = 0.05$ ) calculado a partir de  $p(|E_n(t)|)$  é 2. Sendo assim os eventos que possuem a norma maior que 2 são definidos como eventos de novidades, ou seja, os eventos:

$$[E_5(2) \ E_5(3) \ E_5(4) \ E_5(5) \ E_5(12) \ E_5(13) \ E_5(14) \ E_5(15) \ E_5(16)]$$

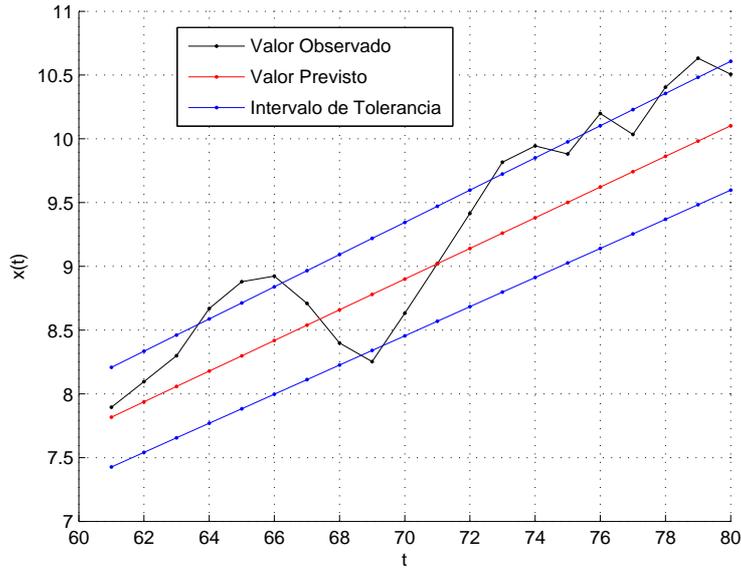


Figura 3.4: Pontos da série a serem classificados

Tabela 3.1: Eventos gerados a partir das ocorrências

$E_n(t)$	Ocorrências correspondentes	Valores
$E_5(1)$	$[O(1) O(2) O(3) O(4) O(5)]$	$[0 0 0 1 1]$
$E_5(2)$	$[O(2) O(3) O(4) O(5) O(6)]$	$[0 0 1 1 1]$
$E_5(3)$	$[O(3) O(4) O(5) O(6) O(7)]$	$[0 1 1 1 0]$
$E_5(4)$	$[O(4) O(5) O(6) O(7) O(8)]$	$[1 1 1 0 0]$
$E_5(5)$	$[O(5) O(6) O(7) O(8) O(9)]$	$[1 1 0 0 1]$
$E_5(6)$	$[O(6) O(7) O(8) O(9) O(10)]$	$[1 0 0 1 0]$
$E_5(7)$	$[O(7) O(8) O(8) O(10) O(11)]$	$[0 0 1 0 0]$
$E_5(8)$	$[O(8) O(9) O(9) O(11) O(12)]$	$[0 1 0 0 0]$
$E_5(9)$	$[O(9) O(10) O(10) O(12) O(13)]$	$[1 0 0 0 1]$
$E_5(10)$	$[O(10) O(11) O(11) O(13) O(14)]$	$[0 0 0 1 1]$
$E_5(11)$	$[O(11) O(12) O(12) O(14) O(15)]$	$[0 0 1 1 0]$
$E_5(12)$	$[O(12) O(13) O(13) O(15) O(16)]$	$[0 1 1 0 1]$
$E_5(13)$	$[O(13) O(14) O(14) O(16) O(17)]$	$[0 1 1 0 1]$
$E_5(14)$	$[O(14) O(15) O(15) O(17) O(18)]$	$[0 1 1 0 1]$
$E_5(15)$	$[O(15) O(16) O(16) O(18) O(19)]$	$[0 1 1 0 1]$
$E_5(16t)$	$[O(16) O(17) O(18) O(19) O(20)]$	$[1 1 0 1 0]$

Finalmente, o último ponto pertencente a cada um dos eventos classificados como eventos de novidade é classificado como novidade. A tabela 3.3 ilustra a classificação final de cada ponto.

A tabela 3.4 sumariza os conceitos do algoritmo utilizados nesse exemplo.

Tabela 3.2: Norma dos eventos

$E_5(t)$	$ E_5(t) $	$E_5(t)$	$ E_5(t) $
$E_5(1)$	2	$E_5(9)$	2
$E_5(2)$	3	$E_5(10)$	2
$E_5(3)$	3	$E_5(11)$	2
$E_5(4)$	3	$E_5(12)$	3
$E_5(5)$	3	$E_5(13)$	3
$E_5(6)$	2	$E_5(14)$	3
$E_5(7)$	1	$E_5(15)$	3
$E_5(8)$	1	$E_5(16)$	3

Tabela 3.3: Classificação final dos pontos do conjunto de teste

$x_t$	classificação	$x_t$	classificação
$x(1)$	normal	$x(11)$	normal
$x(2)$	normal	$x(12)$	normal
$x(3)$	normal	$x(13)$	normal
$x(4)$	normal	$x(14)$	normal
$x(5)$	normal	$x(15)$	normal
$x(6)$	novidade	$x(16)$	novidade
$x(7)$	novidade	$x(17)$	novidade
$x(8)$	novidade	$x(18)$	novidade
$x(9)$	novidade	$x(19)$	novidade
$x(10)$	normal	$x(20)$	novidade

Tabela 3.4: Conceitos do algoritmo genérico utilizados no exemplo

Conceito	Instanciação
Modelo de Previsão ( $M_x(t)$ )	Polinômio de primeiro grau
Intervalo de Tolerância ( $[\epsilon_1, \epsilon_2]$ )	5% de variação do valor previsto
Probabilidade de surpresa	$q = 10\%$ ( valor escolhido hipoteticamente )

Além disso, os seguintes parâmetros foram utilizados:

- Nível de significância:  $\alpha = 0.05$ .
- Tamanho do evento:  $n = 5$ .

## 3.2 Detecção de Novidades em Séries Temporais Utilizando Modelos Autoregressivos e Intervalos de Previsão

Nessa seção será apresentada uma técnica de detecção de novidades em séries temporais estacionárias utilizando um modelo de previsão autoregressivo e intervalo de previsão como o intervalo de tolerância. Essa técnica baseia-se no algoritmo genérico proposto na seção anterior.

### 3.2.1 Modelo de Previsão

O modelo de previsão  $M_x(t)$  responsável por descrever o conhecimento adquirido da série é representado por um modelo autorregressivo de ordem  $p$   $AR(p)$ .

Dado um processo estocástico estacionário  $X_t$ , um modelo autoregressivo linear de ordem  $p$ ,  $AR(p)$ , é o modelo em que o valor atual do processo é expresso como uma combinação linear dos  $p$  valores anteriores  $[X_{t-1}, X_{t-2}, \dots, X_{t-p}]$  acrescido de um ruído branco gaussiano  $a_t$ . Assume-se que o processo de ruído branco  $a_t$  é gaussiano, independente e identicamente distribuído (i.i.d), com média zero e variância  $\sigma_a^2$ . Dessa maneira o modelo autoregressivo pode ser escrito como:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + a_t \quad (3.8)$$

dado que  $\phi = \phi_1, \phi_2, \dots, \phi_p$  e  $\sigma_a^2$  são parâmetros do modelo.

Dada uma série temporal, que possa ser descrita como um processo estocástico, para se construir um modelo de previsão para essa série utilizando a equação (3.8) deve-se inicialmente estimar os valores dos parâmetros  $(\phi, \sigma_a^2)$ , para diversos valores de  $p$  e em seguida avaliar qual é o valor mais plausível para  $p$  a partir de algum critério estatístico. Definido os valores estimados de  $(\phi, \sigma_a^2)$  e  $p$  pode-se então utilizar o modelo construído para descrever o comportamento da série.

Existem diversas técnicas para se construir o estimador dos parâmetros do modelo. O estimador baseado na teoria de inferência estatística clássica é definido como *estimador de máxima verossimilhança* (Kay (1993b)). Para construir esse estimador, inicialmente deve-se determinar a *função de verossimilhança* dos parâmetros  $L(X|\theta)$  a partir de dados da série, onde  $X = (X_1, X_2, \dots, X_m)$  são os dados da série temporal observados e  $\theta = (\phi, \sigma_a^2)$ . De acordo com Box e Jenkins (1990), pode ser demonstrado que a função de verossimilhança aproximada para um modelo autoregressivo de ordem

$p$ , dadas as  $p$  primeiras observações  $X = (X_1, X_2, \dots, X_p)$ , para  $p \ll m$  é dada por:

$$L(X|\theta) \propto \tau^{\frac{N-p}{2}} \exp -\frac{\tau}{2} \sum_{t=p+1}^N (X_t - \phi_1 X_{t-1} - \phi_2 X_{t-2} - \dots - \phi_p X_{t-p})^2 \quad (3.9)$$

Determinada a função de verossimilhança dos parâmetros (3.9), deve-se encontrar o valor estimado dos parâmetros  $\theta$  de forma a maximizar a probabilidade dos dados  $X$  serem gerados a partir de  $L(X|\theta)$ . O estimador de máxima verossimilhança é dado por:

$$\begin{aligned} \hat{\phi} &= (X'X)^{-1}(X'Z) \\ \hat{\sigma}_a^2 &= \frac{1}{N-p} (Z - X\hat{\phi})(Z - X\hat{\phi})^{-1} \end{aligned} \quad (3.10)$$

$$\text{onde } Z = \begin{bmatrix} X_{p+1} \\ X_{p+2} \\ \vdots \\ X_N \end{bmatrix} \text{ e } X = \begin{bmatrix} X_p & \cdots & X_1 \\ X_{p-1} & \cdots & X_2 \\ \vdots & \vdots & \vdots \\ X_{N-1} & \cdots & X_{N-p} \end{bmatrix}$$

A partir do estimador definido em (3.10) e dos dados da série, deve-se então estimar a ordem do modelo  $p$  a ser utilizado.

Para isso estimam-se os valores dos parâmetros para diversos valores de  $p$  e utiliza-se um *critério de informação* para escolher a ordem do modelo mais adequada. Esse critério leva em conta não apenas a qualidade do ajuste do modelo, mas também penaliza a inclusão de parâmetros extras ao modelo. Dessa forma, escolhe-se a ordem do modelo cujo critério de informação calculado seja mínimo.

O *Critério Bayesiano de Informação de Schwarz (BIC)* (Schwarz (1978)) é comumente utilizado para selecionar a ordem do modelo  $AR(p)$ :

$$BIC = \log(\hat{\sigma}_a^2) + \frac{(p+1) \log(m)}{m} \quad (3.11)$$

onde  $p$  é a ordem do modelo autoregressivo e  $m$  é o número de pontos da série.

Estimados os parâmetros do modelo e sua ordem, deve-se então verificar a adequação do modelo antes de utilizá-lo para realizar previsões.

Essa verificação é realizada através da análise de resíduos do modelo. Para o modelo  $AR(p)$ , o resíduo é definido como o erro de previsão no instante  $t$ .

Caso o modelo seja adequado para os dados, espera-se que os resíduos sejam des-correlacionados ao longo do tempo. Além disso, espera-se que os resíduos sejam provenientes de uma distribuição de probabilidade fixa. Como o ruído do processo  $AR(p)$  definido em (3.8) é gaussiano, espera-se que o resíduo também possua distribuição gaussiana, pois o resíduo é uma combinação linear da variável aleatória gaussiana  $\hat{X}_t$ .

Uma maneira intuitiva de verificar se o resíduo segue o padrão descrito acima seria através da análise do seu gráfico de correlação amostral e de seu gráfico de distribuição normal.

Através do gráfico de correlação amostral, pode-se verificar se há correlação entre seus valores. Espera-se que os coeficientes de correlação calculados sejam provenientes de uma distribuição gaussiana com média zero e  $\sigma = \text{sqr}(m)$ . O gráfico em questão ilustra os valores dos coeficientes e um intervalo de confiança desses valores. Dessa forma, espera-se que os valores dos coeficientes estejam dentro dos intervalos.

Para verificar a gaussianidade, utiliza-se o gráfico de distribuição normal. Esse gráfico, ilustra uma linearização da distribuição normal esperada e dos dados amostrados. Dessa forma, espera-se que os dados estejam o mais próximo possível da reta da distribuição esperada.

Realizada a análise dos gráficos, caso o modelo não seja validado, escolhe-se a ordem do modelo para o valor de  $p$  relativo ao segundo menor valor de BIC e assim por diante até que este seja validado.

Uma vez que o modelo foi estimado e validado, pode-se então utilizá-lo para realizar a previsão do valor atual da série dados os últimos  $p$  valores observados:

$$\hat{X}_t = E[X_t | X_{t-1}, \dots, X_{t-p}] = \phi_1 X_{t-1} + \dots + \phi_p X_{t-p} \quad (3.12)$$

sendo  $E[\ ]$  o valor esperado.

### 3.2.2 Intervalos de Tolerância

Os intervalos de tolerância para os valores de  $e(t)$  são construídos a partir dos intervalos de previsão do valor previsto do modelo  $AR(p)$ . Para calcular esse intervalo utiliza-se o valor do parâmetro nível de significância  $\alpha$  do algoritmo.

A função de densidade de probabilidade condicional da saída do modelo  $\hat{X}_t$  dadas as últimas  $p$  observações é dada por:

$$p(X_t | X_{t-1}, \dots, X_{t-p}) = \frac{1}{\sqrt{2\pi\hat{\sigma}_a^2}} \exp - \frac{1}{2\hat{\sigma}_a^2} (\hat{X}_t - (\phi_1 X_{t-1} + \dots + \phi_p X_{t-p}))^2 \quad (3.13)$$

ou seja, assumido que o ruído presente no sinal é gaussiano,  $p(X_t|X_{t-1}, \dots, X_{t-p})$  é também gaussiana com média igual ao valor previsto pelo modelo e variância igual à variância do ruído.

Para calcular os limiares do intervalo de previsão  $[\gamma_1, \gamma_2]$  dado o valor de  $\alpha$  deve-se encontrar os valores que limitam a área de  $(1 - \alpha) * 100\%$  da probabilidade de  $p(X_t|X_{t-1}, \dots, X_{t-p})$ :

$$\begin{aligned} Pr(\gamma_1 < \hat{X}_t < \gamma_2) &= 1 - \alpha \\ Pr(\hat{X}_t < \gamma_1) + Pr(\hat{X}_t > \gamma_2) &= \alpha; \end{aligned}$$

Como  $p(X_t|X_{t-1}, \dots, X_{t-p})$  é simétrica em torno da média, os limiares  $[\gamma_1, \gamma_2]$  também são simétricos em torno da média, e  $Pr(\hat{X}_t < \gamma_1) = Pr(\hat{X}_t > \gamma_2)$ , dessa maneira tem-se que:

$$2Pr(\hat{X}_t > \gamma_2) = \alpha$$

Sabe-se que:

$$Pr(\hat{X}_t > \gamma) = \int_{\gamma}^{+\infty} p(X_t|X_{t-1}, \dots, X_{t-p}) d\hat{X}_t = Q\left(\frac{\gamma - \mu_{\hat{X}_t}}{\hat{\sigma}_a}\right)$$

sendo  $Q(x)$  a *função de distribuição cumulativa complementar* de uma variável aleatória gaussiana (Kay (1993a)), tem-se que:

$$\gamma_2 = \mu_{\hat{X}_t} + Q^{-1}\left(\frac{\alpha}{2}\right) \hat{\sigma}_a \quad (3.14)$$

E, devido a simetria da distribuição:

$$\gamma_1 = \mu_{\hat{X}_t} - Q^{-1}\left(\frac{\alpha}{2}\right) \hat{\sigma}_a \quad (3.15)$$

O valor de  $\mu_{\hat{X}_t}$  é dado por:

$$\mu_{\hat{X}_t} = E[X_t|X_{t-1}, \dots, X_{t-p}] = \hat{X}_t \quad (3.16)$$

Ou seja, o intervalo de previsão é:

$$\hat{X}_t \pm Q^{-1}\left(\frac{\alpha}{2}\right) \hat{\sigma}_a \quad (3.17)$$

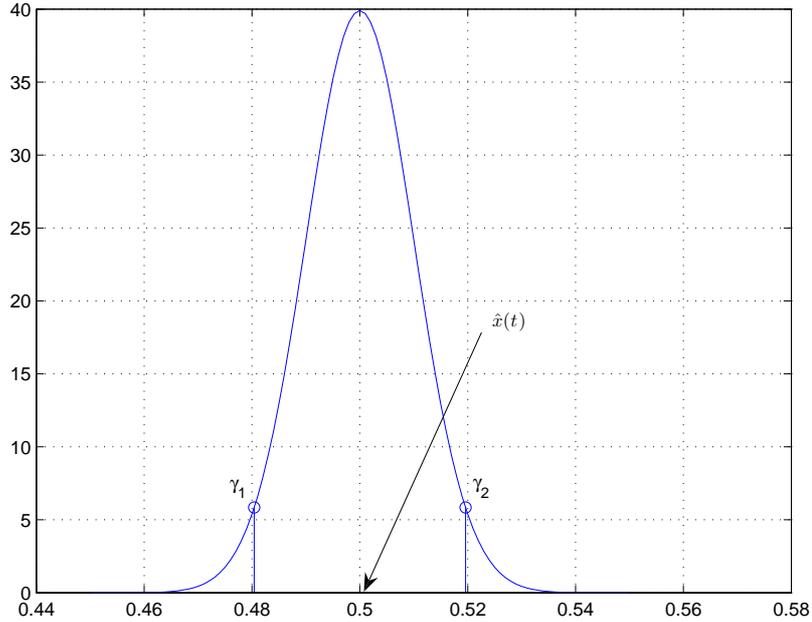


Figura 3.5: Ilustração de um intervalo de previsão da saída de um modelo  $AR(p)$

A figura 3.5 ilustra o intervalo de previsão para  $\hat{X}_t = 0.5$ ,  $\sigma_a = 0.01$  e  $\alpha = 0.05$ . Dessa forma, o intervalo de tolerância do erro de previsão é dado por:

$$\begin{aligned}\varepsilon_1 &= -Q^{-1}\left(\frac{\alpha}{2}\right)\hat{\sigma}_a \\ \varepsilon_2 &= Q^{-1}\left(\frac{\alpha}{2}\right)\hat{\sigma}_a\end{aligned}\tag{3.18}$$

Ou seja, caso o erro de previsão esteja fora do intervalo de tolerância, uma surpresa é definida no instante  $t$ .

### 3.2.3 Probabilidade de Ocorrência de uma Surpresa

O valor da probabilidade de ocorrência de uma surpresa é dado pelo complemento da probabilidade de um dado ponto estar dentro do intervalo de previsão:

$$q = \alpha\tag{3.19}$$

Ou seja, a probabilidade de ocorrência de uma surpresa é igual ao nível de significância utilizado para construir o intervalo de previsão do modelo. A figura 3.6 ilustra essa probabilidade para um modelo  $AR(p)$  com  $\sigma_a = 0.01$  e  $\alpha = 0.05$ .

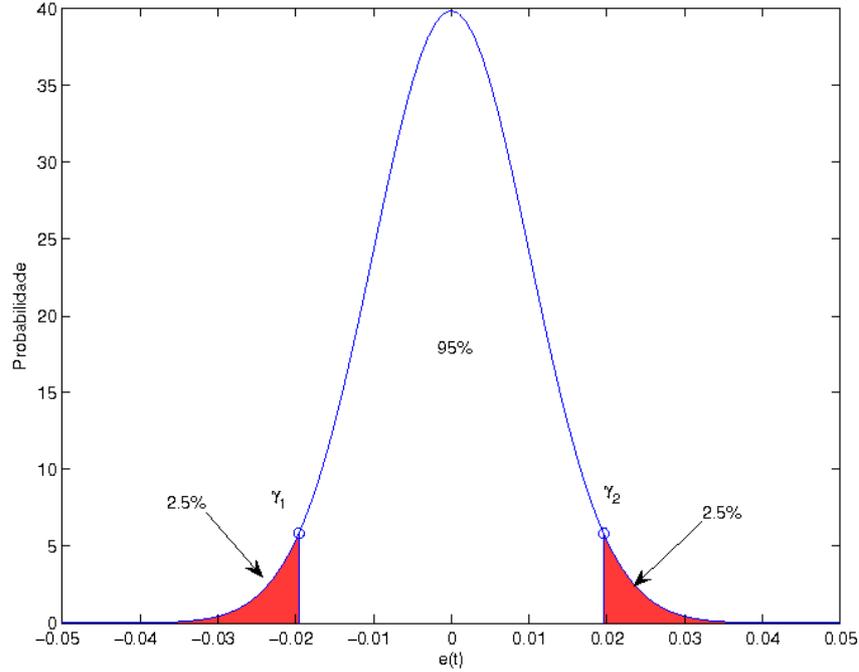


Figura 3.6: Valor da probabilidade de surpresa,  $q$ , para o modelo  $AR(p)$

### 3.2.4 Exemplo

A técnica proposta será utilizada para detectar novidades em uma série temporal gerada artificialmente. Os dados da série foram gerados a partir do seguinte processo estocástico:

$$Z_t = \sin\left(\frac{40\pi t}{N}\right) + a_t + e_t \quad (3.20)$$

sendo  $t = 1, 2, \dots, N$ ,  $N = 1800$  e  $a_t \sim N(0, 0.1)$ , isto é, um ruído branco aditivo gaussiano de média 0 e desvio padrão  $\sigma = 0.1$ .  $e_t$  é uma novidade a ser detectada:

$$e_t = \begin{cases} n_t, & t \in [1500, 1580] \\ 0, & \text{caso contrário} \end{cases} \quad (3.21)$$

onde  $n_t \sim N(0, 0.2)$ .

Foram utilizados os primeiros 600 pontos da série para estimar o modelo de previsão. A figura 3.7 ilustra a série nesse intervalo.

O critério de informação BIC definiu um modelo  $AR(23)$ . A tabela 3.5 ilustra os

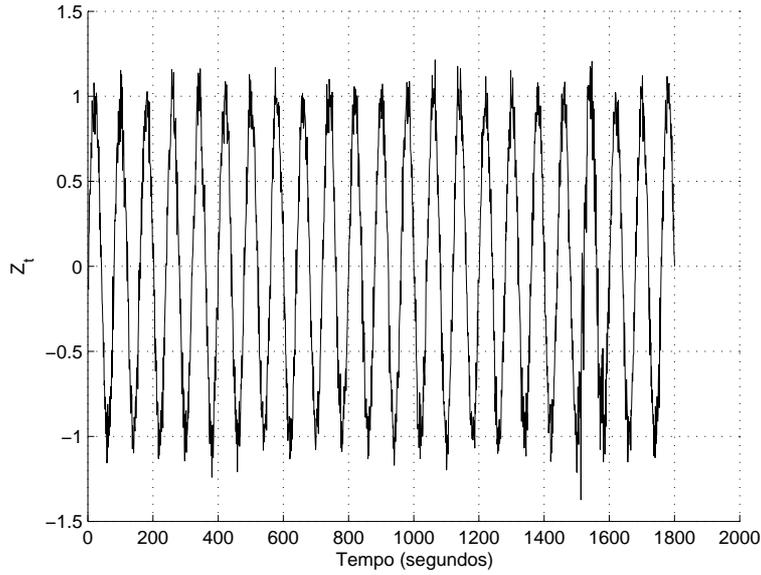


Figura 3.7: Série temporal gerada a partir de (3.20)

Tabela 3.5: Valores dos Parâmetros Estimados

Parâmetro	Valor	Parâmetro	Valor
$\hat{\phi}_1$	0.1426	$\hat{\phi}_2$	0.0783
$\hat{\phi}_3$	0.1317	$\hat{\phi}_4$	0.0478
$\hat{\phi}_5$	0.1010	$\hat{\phi}_6$	0.1165
$\hat{\phi}_7$	0.1103	$\hat{\phi}_8$	0.1194
$\hat{\phi}_9$	-0.004	$\hat{\phi}_{10}$	0.0033
$\hat{\phi}_{11}$	0.0241	$\hat{\phi}_{12}$	0.0561
$\hat{\phi}_{13}$	-0.0245	$\hat{\phi}_{14}$	-0.0347
$\hat{\phi}_{15}$	0.0294	$\hat{\phi}_{16}$	-0.0273
$\hat{\phi}_{17}$	-0.0187	$\hat{\phi}_{18}$	-0.1294
$\hat{\phi}_{19}$	-0.1011	$\hat{\phi}_{20}$	0.0323
$\hat{\phi}_{21}$	-0.0941	$\hat{\phi}_{22}$	-0.0280
$\hat{\phi}_{23}$	-0.1361	$\hat{\sigma}_a$	0.1056

parâmetros estimados do modelo.

Em seguida os pontos do intervalo  $t = [601, 1200]$  são utilizados para validar o modelo. Conforme descrito na seção 3.2 a validação é realizada analisando os gráficos de autocorrelação amostral (figura 3.8) e distribuição normal (figura 3.9).

Analisando o gráfico de autocorrelação amostral (3.8) percebe-se que o erro é descorrelacionado e analisando o gráfico 3.9 verifica-se que o erro segue uma distribuição normal. Dessa maneira, o modelo é validado.

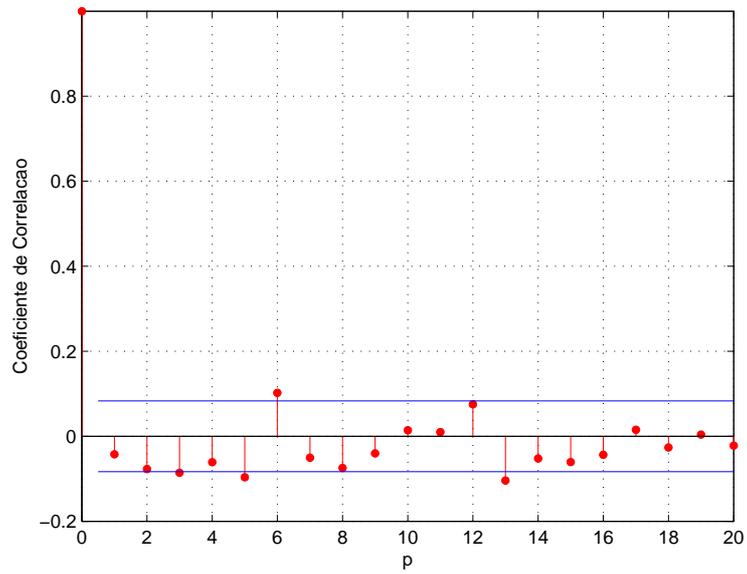


Figura 3.8: Autocorrelação amostral do erro de previsão do conjunto de validação

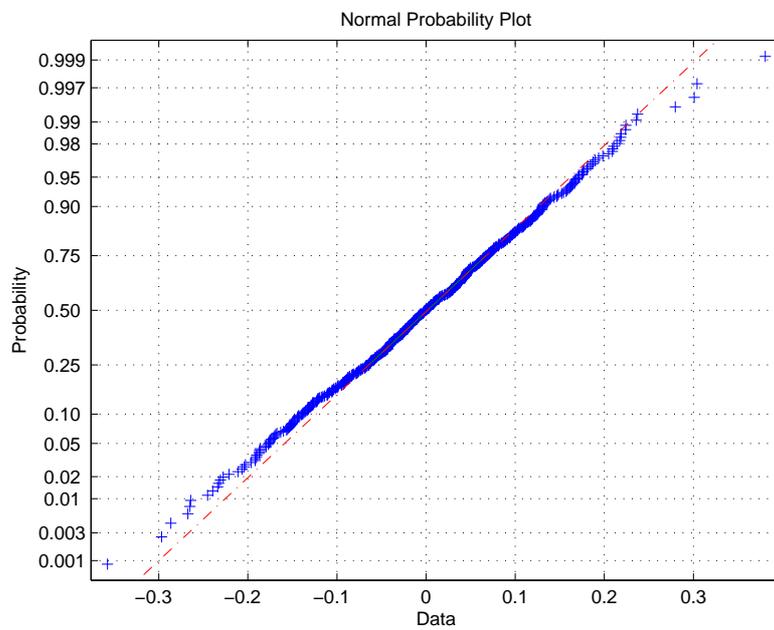


Figura 3.9: Gráfico de distribuição normal do erro de previsão

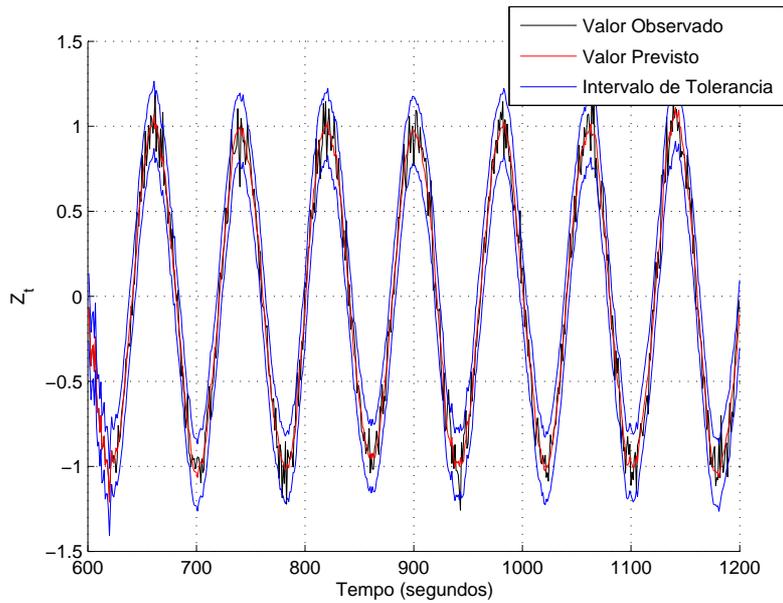


Figura 3.10: Valor observado, previsto e intervalos de previsão para os pontos do intervalo  $t = [601, 1200]$

Outra forma de validar o modelo seria utilizando o intervalo de previsão do mesmo, descrito pela equação (3.17). Espera-se que, para o conjunto de pontos de validação, aproximadamente  $\alpha * 100\%$  dos valores observados estejam fora do intervalo de previsão. A figura 3.10 ilustra o valor observado da série, o valor previsto e os intervalos de previsão para os pontos da série utilizados na validação do modelo ( $t = [601, 1200]$ ). 6.3% dos valores observados estão fora do intervalo de previsão. Esse valor pode ser assumido como próximo ao valor desejado e espera-se que, caso o tamanho do conjunto de validação seja suficientemente grande, esse valor convirja para  $\alpha * 100\%$ .

Finalmente, os pontos do intervalo  $t = [1201, 1800]$  são utilizados como entrada do algoritmo de detecção. Espera-se que o algoritmo consiga detectar a novidade presente no intervalo  $t = [1500, 1580]$ . A figura 3.11 ilustra a série nesse intervalo.

Os parâmetros do algoritmo foram ajustados para  $\alpha = 0.01$ ,  $n = 45$ . A figura 3.12 ilustra os valores de ocorrência e a classificação final dos pontos do intervalo  $t = [1201, 1800]$ . Valores iguais a 0 correspondem a pontos classificados como normais e diferentes de 0, como novidades. Já os valores de ocorrência foram multiplicados por 0.8 para facilitar sua visualização. Essa representação gráfica do resultado da classificação será utilizada em todos os experimentos realizados. Analisando essa figura, percebe-se que o método foi capaz de detectar a novidade conforme desejado.

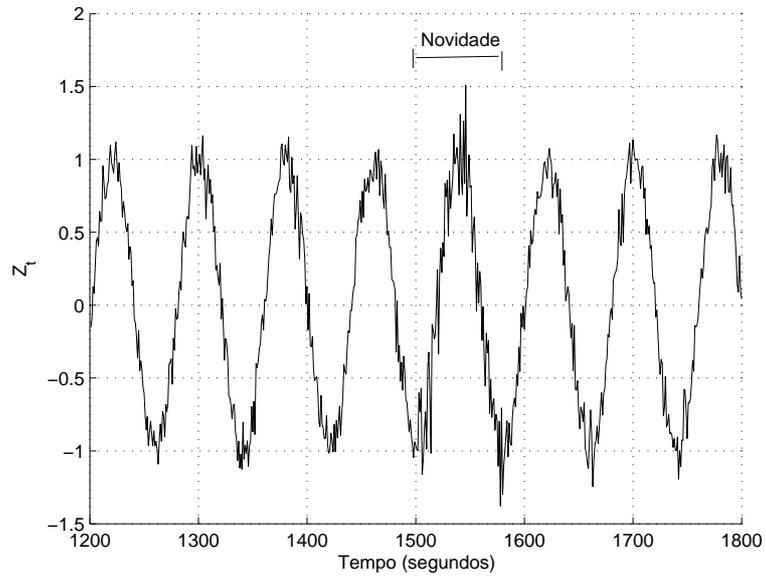


Figura 3.11: Série temporal gerada a partir de (3.20) no intervalo  $t = [1201, 1800]$  contendo uma novidade a ser detectada no intervalo  $t = [1500, 1580]$

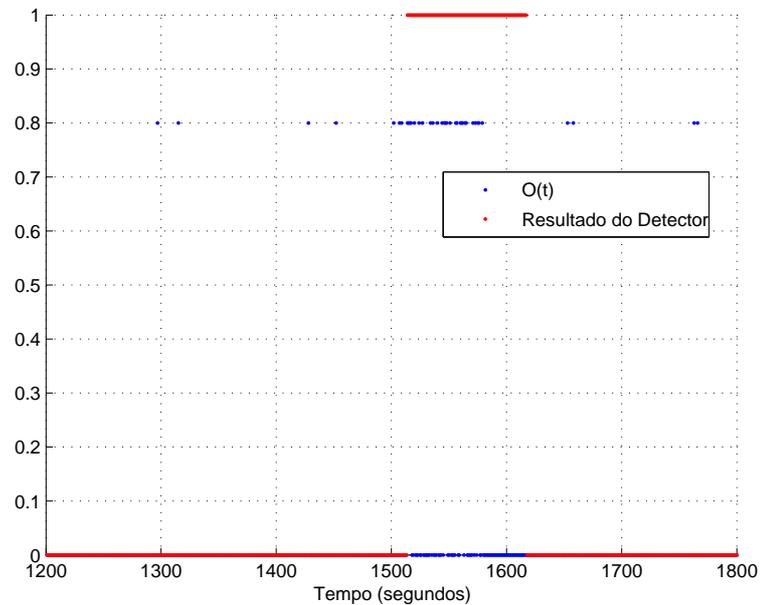


Figura 3.12: Classificação final dos pontos do intervalo  $t = [1201, 1800]$

### 3.2.5 Restrições

Definidos os conceitos e operações, o algoritmo pode então ser utilizado para realizar a detecção de novidades em uma série temporal. Porém, para que a série temporal possa ser modelada a partir de um modelo  $AR(p)$  conforme definido na seção anterior, alguns pré-requisitos devem ser assumidos. Em geral, a série deve (Ehlers (1996)):

- ser fracamente ou estritamente estacionária;
- possuir uma função de autocorrelação com decaimento exponencial ou oscilatório;
- possuir uma função de autocorrelação parcial amostral com valores nulos após o instante  $p$  (Box e Jenkins (1990));
- possuir ruído descorrelacionado e gaussiano.

## 3.3 Detecção de Novidades em Séries Temporais Estacionárias Utilizando Redes Neurais e Intervalos Robustos de Confiança

Nessa seção será apresentada uma metodologia de detecção de novidades em séries temporais que pode ser utilizada quando a série temporal não atenda aos pré-requisitos descritos em 3.2.5 de um modelo  $AR(p)$ . Além disso, a metodologia apresentada pode ser aplicada em séries sem nenhum pressuposto sobre a distribuição do ruído. O único requisito necessário é que a série seja estacionária.

A metodologia proposta nessa seção baseia-se em conceitos definidos em Oliveira e Meira (2006). Esse trabalho descreve uma técnica de detecção de novidades em séries temporais utilizando previsão da série através de redes neurais e limites de tolerância do erro construídos a partir da técnica de intervalos robustos de confiança.

A técnica descrita nesse trabalho foi desenvolvida tendo como problema a ser tratado a detecção de fraudes em sistemas financeiros onde as séries temporais analisadas possuem poucos pontos. Tendo em vista esse problema, durante a etapa de classificação, um único ponto que não pode ser descrito pelo modelo, ou seja, seu erro de previsão é maior que o intervalo de confiança construído, é definido como uma novidade.

Essa abordagem condiz com o tipo de série temporal tratado em que uma novidade é caracterizada por um ponto. Porém a técnica descrita não é apropriada para séries temporais em que uma novidade é caracterizada por uma sequência de pontos que não podem ser descritos pelo modelo de previsão.

Considere, por exemplo, a análise de um eletrocardiograma em que se deseja encontrar arritmias cardíacas causadas por batimentos ectópicos cuja presença é usada no diagnóstico de doenças cardíacas.

A figura 3.13 ilustra um eletrocardiograma que apresenta uma contração ventricular prematura. O instante de ocorrência da anomalia foi anotado por um cardiologista e é representado na figura pela anotação "V". Pode-se notar que a anomalia cardíaca caracteriza-se como um batimento que difere dos outros batimentos anotados como normais ".". A figura 3.13 foi retirada de Goldberger et al. (2000).

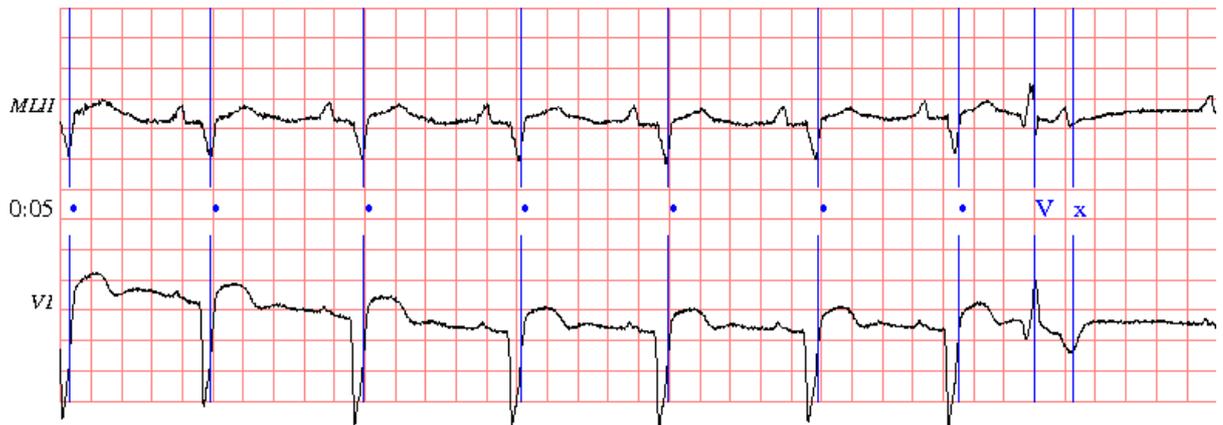


Figura 3.13: Eletrocardiograma apresentando uma contração ventricular prematura (V)

Dessa maneira, o problema de detecção de novidades aplicado a esse exemplo consiste em encontrar batimentos cardíacos que diferem dos batimentos anotados como normais, considerando que um batimento é formado por uma sequência de pontos da série. Deve-se ressaltar que o problema apresentado caracteriza-se pela detecção de arritmias e não por seu diagnóstico. Embora já existam diversos métodos presentes na literatura que realizam o diagnóstico de arritmias, tais como Dingfei Ge e Krishnan (2002), Ubeyli (2007) e Zhao e Zhang (2005), todas essas técnicas necessitam de dados de exemplos dos diferentes tipos de arritmia a serem classificados durante seu treinamento. Deseja-se desenvolver uma técnica que seja capaz de detectar batimentos anômalos utilizando no treinamento apenas resultados de eletrocardiogramas contendo batimentos classificados como normais.

Um exemplo de aplicação para essa técnica seria no monitoramento remoto de sinais vitais de pacientes. Esses sinais são monitorados constantemente e, caso haja um desvio do comportamento padrão, parte desses sinais são enviados a uma central de forma que

um especialista possa realizar um diagnóstico das anomalias encontradas. Um sistema como esse foi proposto em Virone et al. (2006).

Motivado por exemplos como esse, em que uma novidade é caracterizada por uma sequência de pontos, optou-se por desenvolver uma técnica baseada nos conceitos propostos em Oliveira e Meira (2006), porém utilizando-se o algoritmo genérico. A técnica a ser descrita pode ser vista como uma extensão da técnica proposta em Oliveira e Meira (2006) para séries temporais em que uma novidade é caracterizada como uma sequência de pontos que fogem do comportamento padrão. No capítulo 4 serão realizados experimentos para comprovar o ganho de desempenho dessa extensão.

### 3.3.1 Modelo de Previsão

O modelo de previsão utilizado para realizar a previsão do valor atual da série é composto por redes neurais estáticas em que a dinâmica do sistema é representada por valores temporais atrasados no tempo como entradas da rede. A figura 3.14 ilustra esse tipo de rede.

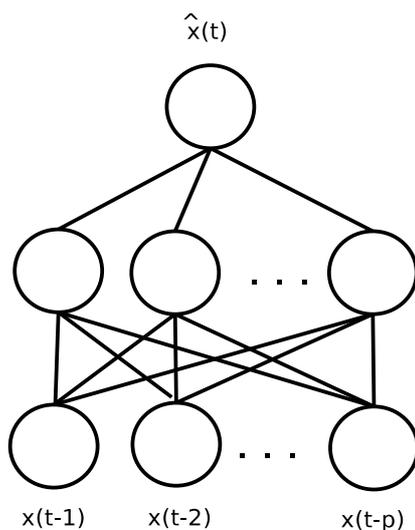


Figura 3.14: Estrutura de uma rede neural utilizada para previsão de séries temporais

Em todos os experimentos realizados utilizaram-se redes de *perceptrons* de múltiplas camadas totalmente conectadas com funções de ativação sigmoidais (Haykin (1999)) nas camadas escondidas e na camada de saída.

Como as redes possuem funções de ativações sigmoidais, os valores da série devem ser normalizados para o intervalo  $[0, 1]$  (Zhang et al. (1998)). A normalização dos

pontos é realizada da seguinte maneira:

$$x(t) = \frac{x(t) - x_{min}}{x_{max} - x_{min}} \quad (3.22)$$

Em relação ao número de camadas escondidas, utiliza-se redes com apenas uma camada, já que somente uma camada é suficiente para aproximar qualquer função não-linear contínua em um domínio compacto com qualquer precisão desejada (Hornik et al. (1989)).

Finalmente, para selecionar o número de entradas e neurônios na camada escondida, utiliza-se o seguinte critério: treinam-se várias redes com valores de entradas variando de 1 a  $p$  e número de neurônios na camada escondida variando de 1 a  $l$  e, para cada combinação possível, calcula-se um critério de informação. Dessa maneira, a topologia escolhida corresponde ao menor valor do critério. O Critério Bayesiano de Informação de Schawar'z (BIC) para a rede MLP é dado por (Cottrell et al. (1995)):

$$BIC = \log \left( \frac{\sum_{i=1}^{m-p+1} e(i)^2}{m-p+1} \right) + \frac{pr \log(m)}{m} \quad (3.23)$$

sendo  $pr$  o número de pesos da rede MLP e  $m$  o número de pontos do conjunto de treinamento. O número de pesos da rede é calculado da seguinte maneira:

$$pr = pl + 2l + 1 \quad (3.24)$$

em que  $p$  é o atraso temporal utilizado como entrada das redes e  $l$  o número de neurônios na camada escondida.

Como o resultado final do treinamento das redes depende dos valores iniciais dos pesos, treinam-se 5 redes com a mesma topologia e utiliza-se o valor médio de BIC.

Além disso, conforme proposto em Oliveira e Meira (2006), utiliza-se um comitê de redes baseado na promediação (Haykin (1999)) como modelo de previsão. Todas as redes do comitê possuem a mesma configuração, porém são treinadas a partir de condições iniciais diferentes. A figura 3.15 ilustra o modelo final de previsão  $M_x(t)$  proposto.

A justificativa de se optar por um comitê de redes é dada por:

- a variância do modelo utilizando promediação é menor que a variância de um modelo com apenas uma rede (Haykin (1999));
- utilizando-se um comitê, pode-se aumentar o número de amostras do erro de previsão do conjunto de validação. Essas amostras são utilizadas para estimar os

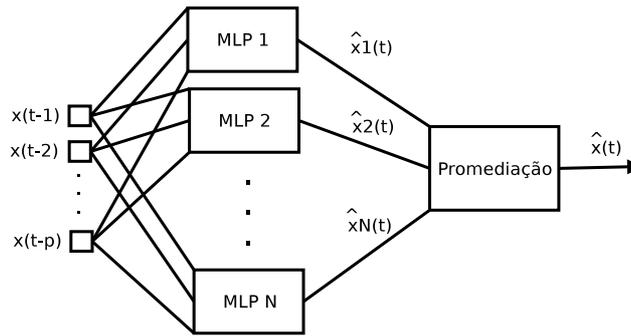


Figura 3.15: Modelo de previsão formado por um comitê de redes baseado na promedição

intervalos de tolerância do erro de previsão. A metodologia de estimação desses intervalos será explicada na próxima seção.

Definida a topologia da rede e o tamanho do comitê, deve-se validar o modelo de previsão. Essa validação é realizada através da análise de resíduos do modelo, definido como a diferença entre a saída do modelo e o valor observado para cada instante  $t$ .

Caso o modelo seja adequado para os dados, espera-se que os resíduos sejam descorrelacionados ao longo do tempo. Uma maneira intuitiva de verificar se o resíduo segue o padrão descrito acima seria através da análise de seu gráfico de correlação amostral.

De maneira análoga à metodologia descrita para o modelo de previsão autoregressivo, caso o modelo não seja validado, escolhe-se a topologia com o segundo menor valor associado de BIC e assim por diante, até que o modelo seja validado.

Além disso, como o resultado final do treinamento das redes depende dos valores iniciais dos pesos, 5 comitês com a mesma topologia são treinados e escolhe-se o comitê com o menor erro quadrático médio para os pontos do conjunto de validação.

Uma completa revisão da utilização de redes neurais de atrasos de tempo em previsão de séries temporais pode ser encontrada em Zhang et al. (1998).

### 3.3.2 Intervalo de Tolerância

Conforme descrito anteriormente, o intervalo de tolerância do erro de previsão é calculado através da técnica de intervalos robustos de confiança (Masters (1995)).

A técnica de intervalos robustos de confiança calcula numericamente intervalos de confiança da distribuição do erro de previsão, sem que seja necessário definir a forma de sua distribuição de probabilidade, apenas assume-se que esta distribuição é simétrica. Para isso, esta utiliza a *função de distribuição amostral* do erro de previsão,  $S_n(e)$ .

Dado um conjunto de  $n$  valores do erro de previsão ordenados, essa função retorna a fração dessa coleção menor ou igual a um dado valor do erro (Masters (1995)):

$$S_n(e) = \begin{cases} 0, & e < e(1) \\ r/n, & e(r) \leq e \leq e(r+1) \\ 1, & e(n) \leq e \end{cases} \quad (3.25)$$

Caso o conjunto de pontos utilizados para calcular  $S_n(e)$  represente os valores de erros encontrados na prática e o tamanho desse conjunto seja suficientemente grande,  $S_n(e)$  é assumido ser uma boa estimacão da verdadeira função de distribuição de probabilidade do erro de previsão  $F(e)$ .

Dessa maneira, para se calcular o intervalo de confiança do erro de previsão dado um nível de significância  $\alpha$ , deve-se descartar  $n \times \alpha/2$  valores de cada extremo do conjunto de valores de  $e(t)$  e os limites do intervalo de confiança serão os valores limites do conjunto resultante. Para um conjunto de amostras de erro relativamente pequeno descarta-se  $n \times \alpha/2 - 1$ .

Por exemplo, dado um conjunto de 10 amostras do erro de previsão formado pelos seguintes valores  $\vec{e} = \{-0.4, -0.3, -0.1, 0.0, 0.0, 0.0, 0.1, 0.2, 0.4\}$ , deseja-se construir um intervalo de confiança com um nível de significância de 0.4. Dessa maneira  $n \times \alpha/2 - 1 = 10 - 0.4/2 - 1 = 1$  valores devem ser descartados de cada extremo do conjunto e o limite do intervalo de confiança é  $\{-0.3, 0.2\}$ , ou seja, caso o valor observado da série  $x(t)$  esteja fora do intervalo  $\{\hat{x}(t) - 0.3, \hat{x}(t) + 0.2\}$ , o ponto é definido como uma novidade.

A tabela 3.6 ilustra os limiares do intervalo robusto de confiança, com um nível de significância  $\alpha = 0.05$ , calculados a partir de amostras de uma variável aleatória gaussiana com média zero e variância um. Os valores dos limiares do intervalo de confiança calculados teoricamente são  $[-1.960, 1.960]$ . Percebe-se que à medida que o número de amostras cresce, o valor dos limiares do intervalo robusto de confiança tende assintoticamente para os valores teóricos.

Tabela 3.6: Intervalos robustos de confiança de uma variável aleatória gaussiana  $X \sim N(0, 1)$  para diferentes números de amostras e  $\alpha = 0.05$

Número de Amostras	$\gamma_1$	$\gamma_2$
100	-1.506	2.135
10000	-1.954	1.964
1000000	-1.957	1.961
10000000	-1.960	1.960

Para se calcular o intervalo robusto de confiança do erro de previsão da série, ini-

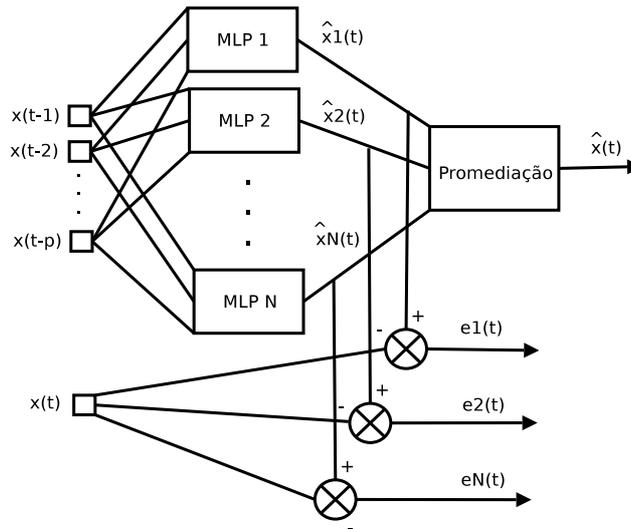


Figura 3.16: Cálculo das amostras do erro de previsão a partir das redes do modelo

cialmente, é necessário gerar as amostras do erro. Para isso, utilizam-se os pontos do conjunto de validação do modelo e, ao invés de gerar as amostras através da diferença entre o valor observado e saída do comitê, calcula-se o erro como a diferença entre o valor esperado e a saída de cada uma das redes que compõem o comitê conforme proposto em Oliveira e Meira (2006). O erro é gerado dessa maneira para aumentar o número de amostras, fazendo com que os limiares do intervalo robusto de confiança calculado tendam para os limiares reais. A figura 3.16 ilustra o cálculo dos valores do erro de previsão.

### 3.3.3 Probabilidade de Ocorrência de uma Surpresa

O valor da probabilidade de ocorrência de uma surpresa é dado pelo complemento da probabilidade de uma amostra do erro de previsão estar fora do intervalo robusto de confiança:

$$q = \alpha \tag{3.26}$$

Ou seja, a probabilidade de ocorrência de uma surpresa é igual ao nível de significância utilizado para construir o intervalo robusto de confiança do modelo.

### 3.3.4 Exemplo

Nesta seção, a técnica descrita será utilizada para detectar novidades em uma série temporal. A série em questão foi gerada a partir do seguinte processo estocástico:

$$Z_t = 0.2 - 3Z_{t-1} + 3\sqrt{Z_{t-1}} + a_t + e_t \quad (3.27)$$

sendo  $t = 1, 2, \dots, m$ ,  $m = 2000$  e  $a_t \sim N(0, 0.05)$ , isto é, um ruído branco aditivo gaussiano de média 0 e desvio padrão  $\sigma = 0.05$  e  $Z_1 = 0.3$ .  $n_t$  é uma novidade a ser detectada:

$$n_t = \begin{cases} g_t, & t \in [1500, 1580] \\ 0, & \text{caso contrário} \end{cases} \quad (3.28)$$

onde  $g_t \sim N(0, 0.1)$ .

A figura 3.17 ilustra os primeiros 200 pontos da série normalizados utilizados no treinamento e a figura 3.18 ilustra a série no intervalo temporal  $[1450, 1650]$ , sendo que a novidade a ser detectada encontra-se no intervalo  $[1500, 1580]$ .

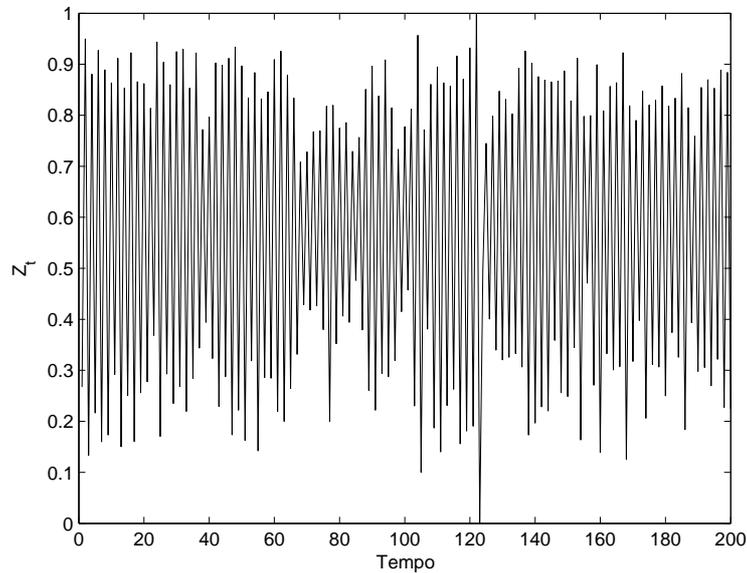


Figura 3.17: Primeiros 200 pontos da série normalizados

Analisando o gráfico de autocorrelação dos primeiros 1000 pontos da série, ilustrado pela figura 3.19, percebe-se que esta não pode ser modelada utilizando-se um modelo

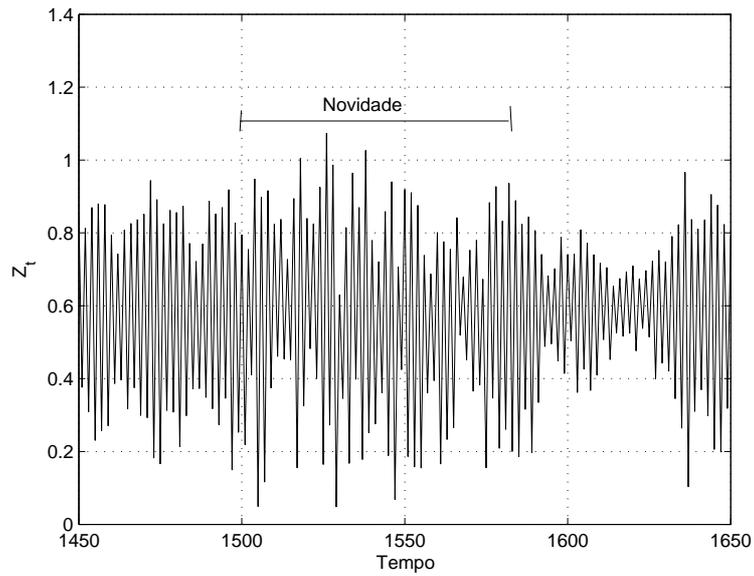


Figura 3.18: Série temporal no intervalo [1450, 1650] contendo a novidade a ser detectada no intervalo [1500, 1580]

$AR(p)$ . Isso pode ser justificado pelo fato do modelo  $AR(p)$  ser linear e o processo que gerou a série ter dinâmica não linear.

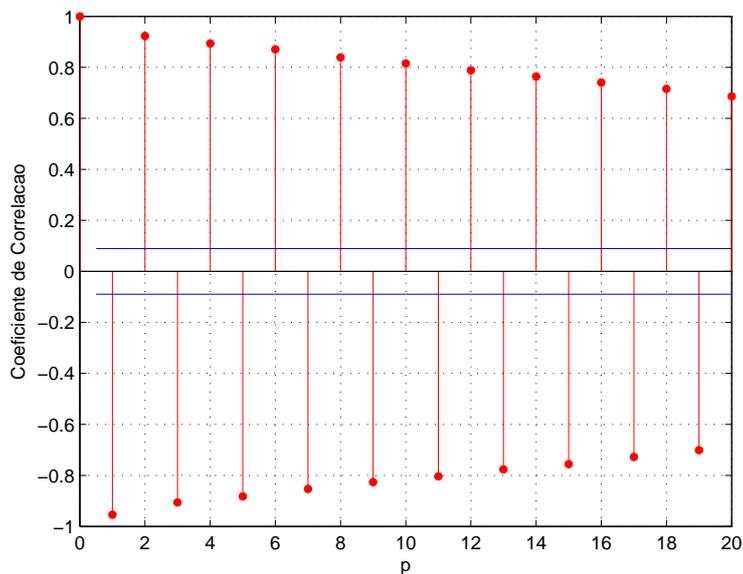


Figura 3.19: Autocorrelação amostral dos primeiros 1000 pontos da série

Os primeiros 500 pontos da série foram utilizados para escolher a topologia da rede

e treinar o modelo de previsão e os pontos do intervalo [501, 1000] foram utilizados para validá-lo. Analisando a figura 3.20 percebe-se que o menor valor de BIC em função do atraso temporal  $p$  e do número de neurônios na camada escondida  $l$  está relacionado a  $p = 1$  e  $l = 3$ . Dessa forma, o modelo previsão escolhido consiste em um comitê composto por 3 redes MLP com 1 entrada ( $x(t - 1)$ ) e 3 neurônios na camada escondida.

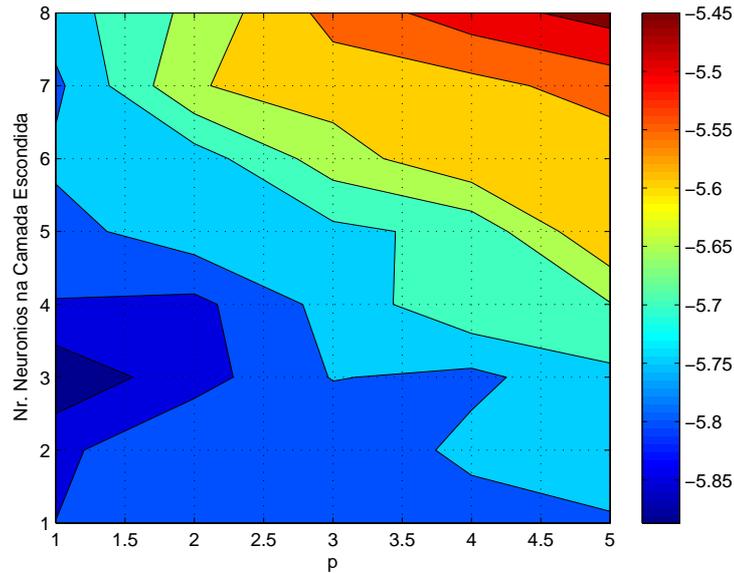


Figura 3.20: Valor de BIC em função da topologia da rede

Cada uma das redes do comitê foi treinada durante 100 épocas utilizando o algoritmo de treinamento supervisionado *Levenberg-Marquardt* (Marquardt (1963)) com um erro objetivo de 0.000001. A figura 3.21 ilustra o gráfico de correlação amostral do erro de previsão calculado a partir do conjunto de validação. Analisando esse gráfico, percebe-se que o erro é descorrelacionado e, portanto, a topologia é validada. Foram treinados 5 comitês com a mesma topologia e o comitê selecionado como modelo de previsão apresentou erro quadrático médio do conjunto de validação igual a 0.0024.

O histograma do erro de previsão e os limiares do intervalo robusto de confiança são ilustrados na figura 3.22. Os limiares do intervalo foram definidos como:  $\{x(\hat{t}) - 0.0953, x(\hat{t}) + 0.0947\}$ .

Finalmente, os pontos do intervalo [1001, 2000] foram utilizados como entradas do método de detecção. A novidade a ser detectada encontra-se no intervalo [1500, 1580]. Os parâmetros do algoritmo foram ajustados para  $n = 50$  e  $\alpha = 0.05$ . A figura 3.23 ilustra a classificação final dos pontos. Analisando essa figura, percebe-se que o algoritmo foi capaz de detectar a novidade corretamente.

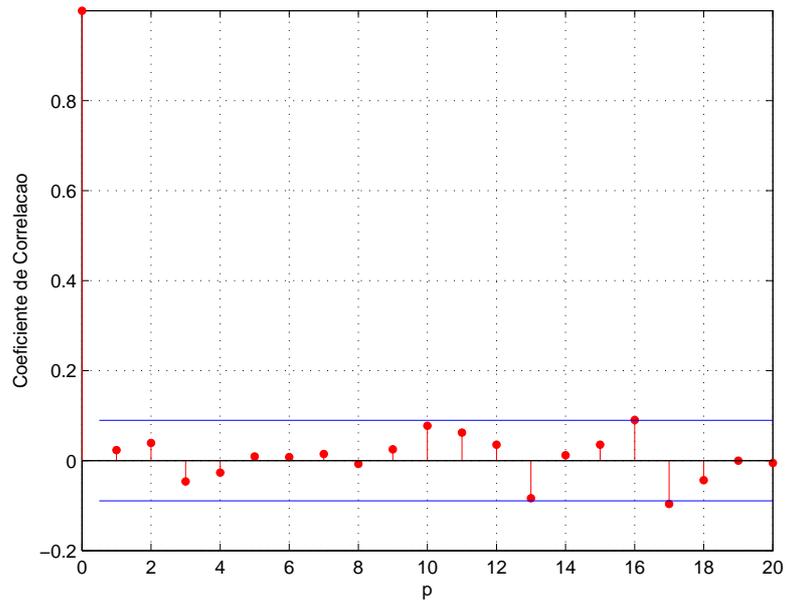


Figura 3.21: Gráfico de correlação amostral do erro de previsão

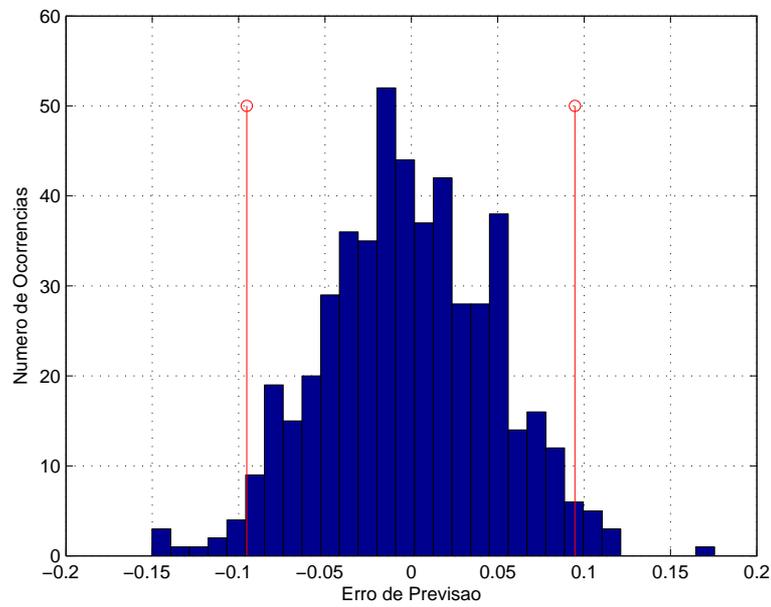


Figura 3.22: Histograma do erro de previsão e os limiares do intervalo robusto de confiança calculado

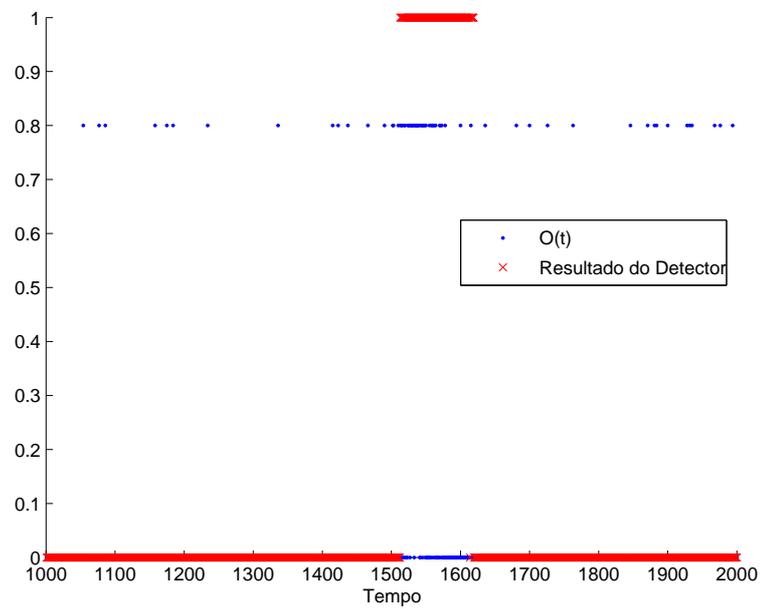


Figura 3.23: Resultado final da classificação dos pontos do intervalo  $[1001, 2000]$  que possui uma novidade no intervalo  $[1500, 1580]$

# Capítulo 4

## Experimentos

Nesse capítulo, as instanciações do algoritmo genérico propostas no capítulo 3 serão utilizadas na detecção de novidades em séries temporais com o intuito de testar sua eficácia e eficiência.

Inicialmente, uma série temporal artificial será utilizada para avaliar a influência dos parâmetros do algoritmo na sua eficiência.

Conforme descrito anteriormente, a instanciação do algoritmo utilizando redes neurais e intervalos robustos de confiança apresentada na seção 3.3 é uma extensão do método proposto em Oliveira e Meira (2006), para tratar de novidades que se caracterizam como uma sequência de pontos cuja frequência ou estrutura difere do conjunto de pontos assumidos como normais. Dessa maneira, um experimento será apresentado comparando o algoritmo original e a extensão proposta, com objetivo de comprovar o aumento de desempenho sugerido anteriormente.

Em seguida, o algoritmo proposto será utilizado para detectar síndrome de apnéia obstrutiva do sono em pacientes através do monitoramento de eletrocardiogramas.

Finalmente, o algoritmo será utilizado para detectar falhas em um atuador através do monitoramento de algumas variáveis do sistema. Os dados utilizados são provenientes de um simulador que serve como *benchmark* de técnicas de detecção e diagnóstico de falhas.

Sumarizando, os seguintes experimentos serão apresentados:

1. Análise dos parâmetros do algoritmo.
2. Comparação do método proposto em Oliveira e Meira (2006) e sua extensão proposta na seção 3.3.
3. Detecção de apnéia do sono através do monitoramento de eletrocardiogramas.
4. Detecção de falhas em um atuador.

Para todos os experimentos realizados nesse capítulo, com exceção do segundo, optou-se por utilizar a instanciação do algoritmo baseada em modelos autoregressivos, nos casos em que a série satisfaça os pré-requisitos necessários. Caso esses pré-requisitos não sejam satisfeitos, utiliza-se a instanciação baseada em redes neurais e intervalos robustos de confiança.

## 4.1 Análise da Influência dos Parâmetros do Algoritmo

Todos os experimentos dessa seção foram realizados com séries temporais geradas a partir do seguinte processo estocástico:

$$Z_t = 0.9Z_{t-1} - 0.4Z_{t-2} + a_t + e_t \quad (4.1)$$

sendo  $t = 1, 2, \dots, m$ ,  $m = 100000$  e  $a_t \sim N(0, 0.1)$ , isto é, um ruído branco aditivo gaussiano de média 0 e desvio padrão  $\sigma = 0.1$  e  $Z_1 = 0.3$  e  $Z_2 = 0.4$ . E  $e_t$  é uma novidade a ser detectada:

$$e_t = \begin{cases} n_t, & t \in [8000, 8500] \\ 0, & \text{caso contrário} \end{cases} \quad (4.2)$$

onde  $n_t \sim N(0, 0.11)$ .

A figura 4.1 ilustra a série temporal em questão no intervalo temporal [5000, 11000] contendo a novidade a ser detectada.

### 4.1.1 Nível de Significância $\alpha$

A princípio, foi realizado um experimento para avaliar a influência do nível de significância no desempenho do algoritmo.

Espera-se que:

- a probabilidade de falsos alarmes seja similar ao nível de significância, acrescido de uma constante que depende do tamanho do evento e o número de novidades presentes na série;
- a probabilidade de ocorrência de uma surpresa seja igual ao nível de significância, ou seja,  $q = \alpha$ ;

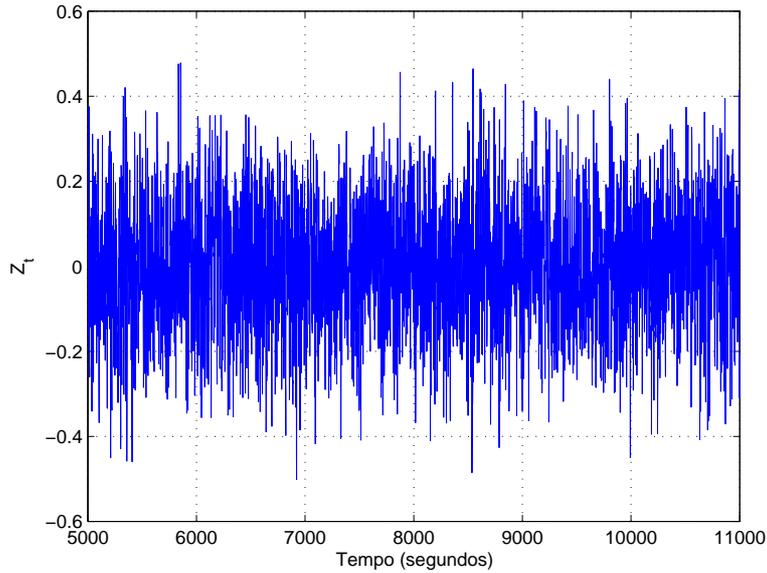


Figura 4.1: Série temporal gerada a partir de (4.1) no intervalo temporal [5000, 11000].

- à medida que o nível de significância aumente, a probabilidade de detecção também aumente.

Em uma série temporal sem novidades, caso  $p(|E_n(t)|)$  fosse uma distribuição contínua, o valor de  $P_{FA}$  esperado seria igual a  $\alpha$ . Porém, como essa distribuição é discreta, existe uma discrepância entre esses valores, pois o intervalo de confiança de  $p(|E_n(t)|)$  não pode ser calculado com a precisão desejada. Por exemplo, o limiar do intervalo de confiança de  $p(|E_n(t)|)$  para  $n = 90$ ,  $q = 0.05$  e  $\alpha = 0.05$  é  $\gamma = 8$ . Dessa forma, os eventos que apresentem normas maiores que 8 são classificados como eventos de novidades e eventos com normas menores ou iguais a 8 são classificados como normais. Como a distribuição cumulativa da norma para valores iguais ou menores que 8 é 96.38%, é esperado que a probabilidade de falsos alarmes seja 3.62% e não 5%. Dessa forma, o valor esperado de  $P_{FA}$  seria igual a:

$$P_{FA} = 1 - P_{|E_n|}(\gamma) \quad (4.3)$$

sendo  $P_{|E_n|}(|E_n|)$  a função cumulativa da norma dos eventos e  $\gamma$  o limiar do intervalo de confiança de  $p(|E_n(t)|)$  calculado a partir dos parâmetros do algoritmo.

Ao contrário da probabilidade de falsos alarmes, não é possível calcular analiticamente o valor esperado da probabilidade de detecção, pois seria necessário conhecer  $P(H_1; H_1)$ , ou seja, a função de densidade de probabilidade das novidades.

Sendo assim, foi realizado um experimento para comprovar as suposições sobre a influência de  $\alpha$  nas probabilidades de falsos alarmes, detecção e ocorrência de uma surpresa.

Foram geradas 180 séries temporais a partir do processo estocástico descrito em (4.1). Para todas as séries, utilizou-se um modelo  $AR(2)$ , conforme os resultados obtidos através dos valores de BIC. Os primeiros 1000 pontos foram utilizados para treinar o modelo e os próximos 1000 para validá-lo. O valor do tamanho do evento foi ajustado para  $n = 50$  e as probabilidades de falsos alarmes, ocorrência de surpresas e detecção foram calculadas para valores de  $\alpha$ , variando de 0.01 a 0.99, a partir dos pontos do intervalo [2051, 100000].

As figuras 4.2, 4.3 e 4.4 ilustram, respectivamente, os valores de  $P_{FA}$  esperados e medidos experimentalmente,  $q$  esperados e medidos experimentalmente e  $P_D$  medidos em função dos valores de  $\alpha$ . Como  $n \ll m$ , não se levou em conta a contribuição do valor constante, dependente de  $n$ , no cálculo da  $P_{FA}$  esperada.

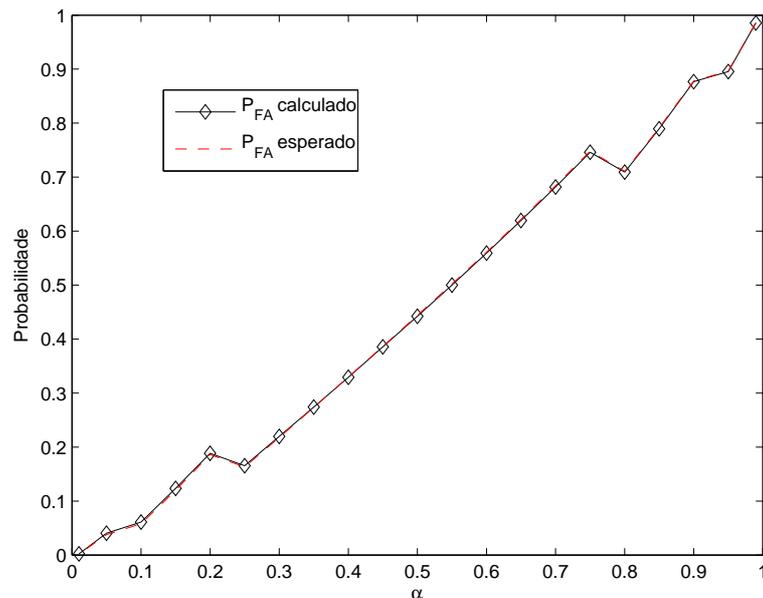


Figura 4.2:  $P_{FA}$  esperado e medido em função de  $\alpha$

Analisando esses dados, percebe-se que:

- a  $P_{FA}$  medida está próxima ao valor esperado;
- o valor de  $q$  é, experimentalmente, próximo ao valor de  $\alpha$ ;
- o  $P_D$  cresce de forma aproximadamente linear com o valor de  $\alpha$ .

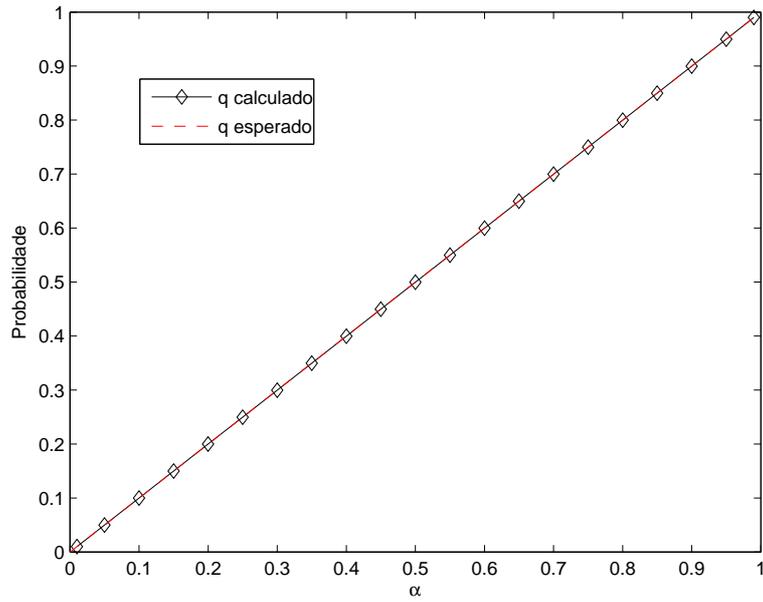


Figura 4.3:  $q$  esperado e medido em função de  $\alpha$

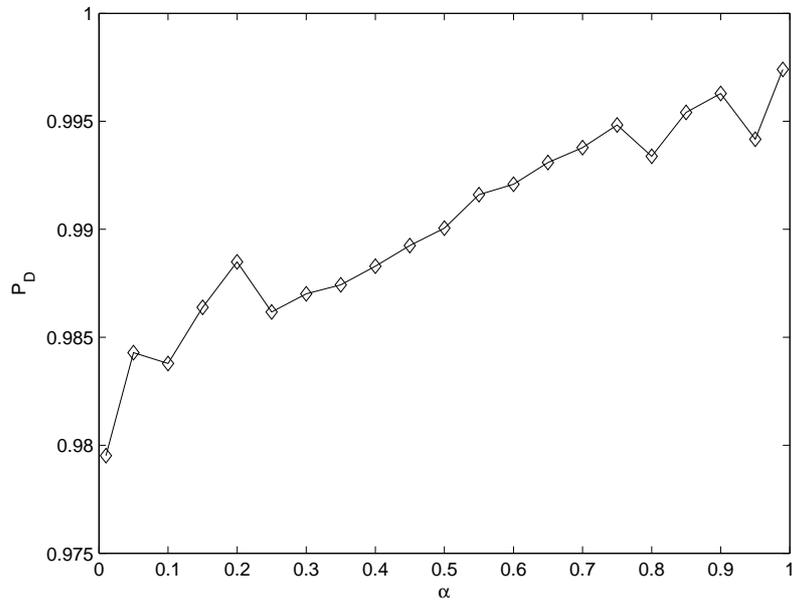


Figura 4.4:  $P_D$  medido em função de  $\alpha$

### 4.1.2 Tamanho do Evento $n$

Em seguida, serão apresentados experimentos realizados para analisar a influência do tamanho do evento nas probabilidades de detecção e falsos alarmes.

Conforme descrito anteriormente, a probabilidade de falsos alarmes esperada é proporcional ao nível de significância, acrescida de uma constante proporcional ao tamanho do evento e o número de novidades presentes. Esse acréscimo deve-se a um atraso temporal do detector no final da detecção de uma novidade. Esse atraso é denominado *tempo de recuperação de uma detecção*,  $t_r$  (Bartys et al. (2006)). A figura 4.5 ilustra esse fenômeno. Assim, espera-se que a  $P_{FA}$  seja igual a calculada pela equação (4.3) acrescida de um valor constante proporcional ao tamanho do evento e o número de novidades presentes.

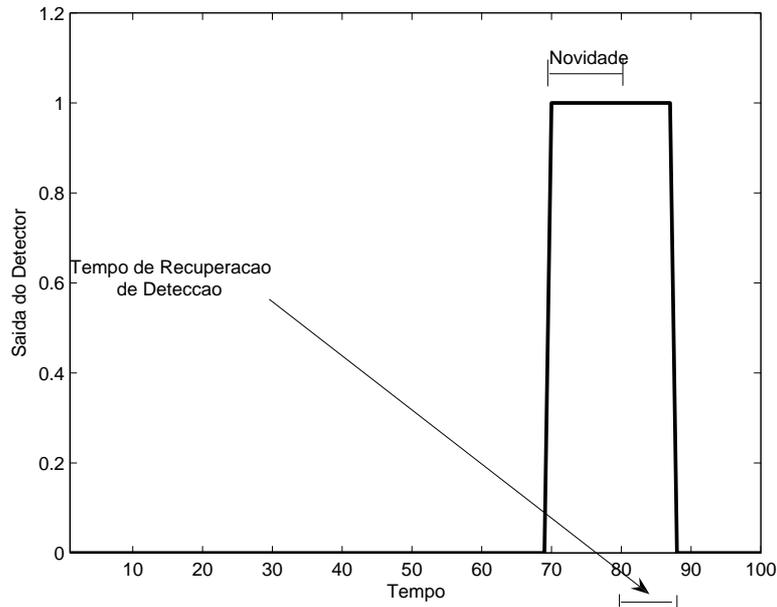


Figura 4.5: Ilustração do tempo de recuperação de uma detecção,  $t_r$

Em relação à probabilidade de detecção, espera-se que, à medida que o tamanho do evento aumente, a probabilidade de detecção aumente até um certo valor  $n_{otimo}$  e, a partir desse valor, comece a diminuir. Ou seja, espera-se que exista um valor ótimo do tamanho do evento que maximize a probabilidade de detecção.

O aumento da probabilidade de detecção, proporcional ao aumento do tamanho do evento, até um valor  $n_{otimo}$  ocorre pois, para valores menores que  $n_{otimo}$ , durante o período temporal da novidade, a saída do detector oscila entre 0 e 1. Já a diminuição de  $P_D$  para valores maiores que  $n_{otimo}$  ocorre pois, com o aumento do evento, o tempo necessário para que o detector comece a detectar uma novidade aumenta, já que o valor de  $\gamma$  depende do tamanho do evento.

Espera-se que o valor de  $n_{otimo}$  possua o menor *tempo de detecção*  $t_d$  (Bartys et al. (2006)). O tempo de detecção é definido como o tamanho do intervalo temporal entre

o início da novidade  $t_{from}$  e o início do último intervalo temporal em que a saída do detector é igual a 1, no intervalo da novidade. A figura 4.6 ilustra esse fenômeno.

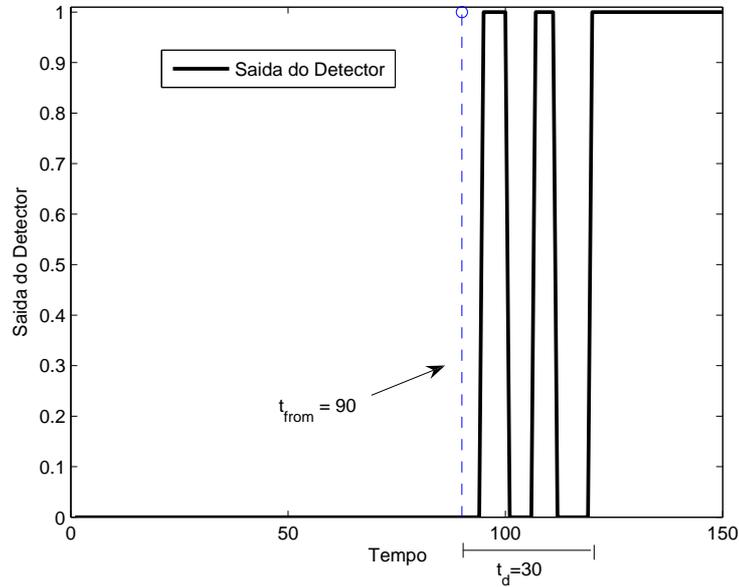


Figura 4.6: Exemplo ilustrativo do tempo de detecção,  $t_d$

Dessa maneira, foram realizados experimentos para comprovar as suposições descritas. Foram geradas 1000 séries temporais a partir do processo estocástico descrito em (4.1) no intervalo temporal  $[1, 10000]$ . Para todas as séries, utilizou-se um modelo  $AR(2)$ , conforme os resultados obtidos através dos valores de BIC, e os primeiros 1000 pontos foram utilizados para treinar o modelo e os próximos 1000 para validá-lo. O valor de  $\alpha$  foi ajustado para 0.05 e, para cada série gerada, os valores de  $P_{FA}$  e  $P_D$  foram calculados para  $n$  variando de 1 a 1500, a partir dos pontos do intervalo  $[2501, 10000]$ .

A figura 4.7 ilustra a  $P_{FA}$  medido e esperado, calculado a partir da equação (4.3), em função do tamanho do evento e a figura 4.8 ilustra a variação de  $P_D$ .

Analisando a figura 4.7, pode-se notar que a  $P_{FA}$  aumenta com o tamanho do evento. Esse aumento é justificado pelo tempo de recuperação de uma detecção  $t_r$ . Porém, o primeiro valor de  $P_{FA}$ , para  $n = 1$ , é maior que o três próximos valores. Isso ocorre porque, para o primeiro valor,  $P_{FA}$  esperado é igual a  $q = \alpha$ , pois o evento é composto por apenas um ponto. Já para os outros valores de  $n$ ,  $P_{FA}$  esperado é ligeiramente menor que  $\alpha$ , devido ao fenômeno explicado pela equação (4.3), acrescida de uma constante dependente de  $t_r$ . Dessa forma, para  $n = [5, 25, 50, 100]$ , o valor esperado de  $P_{FA}$  ainda é menor que  $\alpha$ , pois o valor de  $t_r$  ainda é pequeno. Já para os próximos valores,  $t_r$  é suficientemente grande para que  $P_{FA}$  seja maior de  $\alpha$ . A tabela

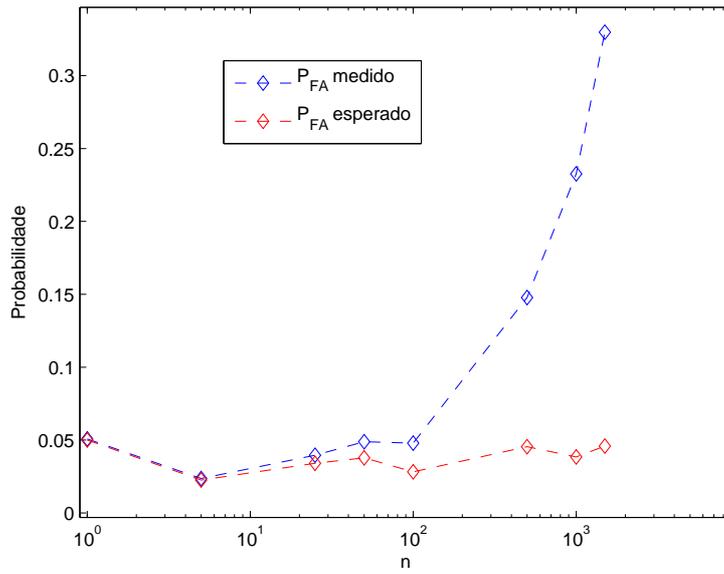


Figura 4.7:  $P_{FA}$  em função de  $n$

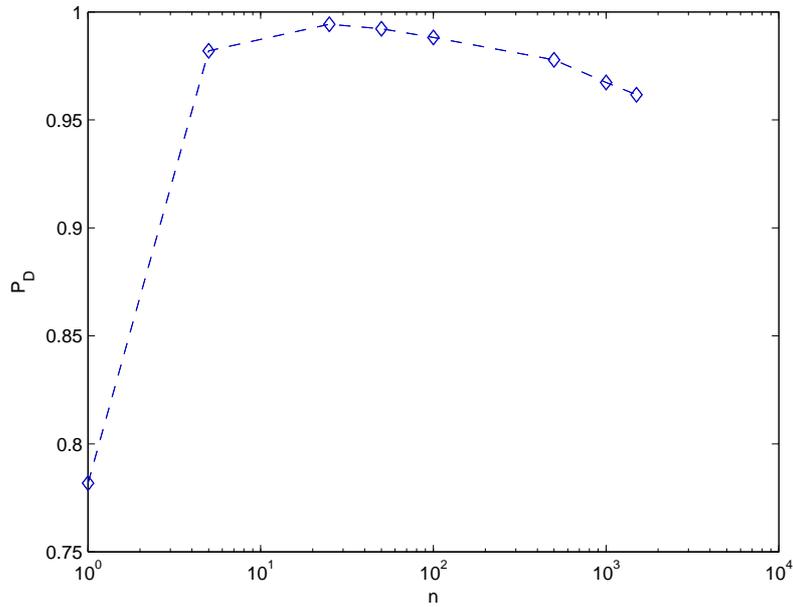


Figura 4.8:  $P_D$  em função de  $n$

4.1 ilustra valores médios de  $P_{FA}$  e  $t_r$  para diferentes valores de  $n$ . Analisando essa tabela, percebe-se a influência do aumento de  $n$  em  $P_{FA}$  e  $t_r$ .

Analisando a figura 4.8 percebe-se que  $P_D$  aumenta até  $n = 25$ , depois começa a diminuir. A tabela 4.2 ilustra valores médios de  $P_D$  e  $t_d$  em função de  $n$ . Para os

Tabela 4.1: Valores médios de  $P_{FA}$  e  $t_r$  para diferentes valores de  $n$

$n$	$P_{FA}$	$t_r$
1	5.07%	2
5	2.39%	5
25	3.93%	24
50	4.88%	37
100	4.79%	81
500	14.78%	475
1000	23.26%	973
1500	32.98%	1440

valores dessa tabela,  $n_{otimo} = 25$ . Para valores inferiores a 25, o resultado do detector no período de ocorrência de uma novidade varia entre 0 e 1. Esse fenômeno é relacionado a altos valores de  $t_d$ . Já para valores maiores que 100, o resultado do detector permanece constante em 1 durante o período de ocorrência de uma novidade, porém  $t_d$  cresce com o aumento de  $n$ .

Tabela 4.2: Valores médios de  $P_D$  e  $t_d$  para diferentes valores de  $n$

$n$	$P_D$	$t_d$
1	78.36%	497
5	98.20%	387
25	99.44%	3
50	99.22%	4
100	98.82%	6
500	97.78%	11
1000	96.74%	16
1500	96.17%	19

## 4.2 Análise Comparativa

Conforme descrito anteriormente, o método proposto em Oliveira e Meira (2006) foi desenvolvido tendo como principal problema a ser tratado a detecção de fraudes em sistemas financeiros, em que as séries temporais possuem poucos pontos e a novidade a ser encontrada caracteriza-se por apenas um ponto que não pode ser descrito pelo modelo de previsão. Porém, em muitos casos práticos, a novidade a ser detectada é caracterizada por uma sequência de pontos cuja frequência ou estrutura diferem do padrão assumido como normal (Keogh et al. (2002)). Dessa forma, para se utilizar essa metodologia na detecção de novidades em séries temporais com essa característica,

deve-se realizar um processamento do resultado através de uma janela de pontos para que novidades formadas por uma sequência de pontos sejam detectadas corretamente.

Fazendo uma analogia entre o algoritmo genérico proposto e o método descrito em Oliveira e Meira (2006), esse método define que um ponto é classificado como novidade quando uma surpresa ocorre, ou seja, um ponto é definido como uma novidade caso o erro de previsão esteja fora dos intervalos de tolerância. Dessa maneira, a instanciação do algoritmo genérico proposta na seção 3.3 é uma extensão desse método para que este possa ser utilizado em séries temporais em que uma novidade é caracterizada por uma sequência de pontos.

A seguir será apresentado um experimento com o intuito de comprovar a eficiência ganha utilizando-se a extensão proposta.

Foram geradas 100 séries temporais a partir do processo estocástico descrito pela equação (3.27). Utilizou-se a mesma topologia de rede, número de pontos de treinamento e validação e tamanho do evento descritos no exemplo da seção 3.3.4. A partir dos pontos a serem classificados, as probabilidades de detecção e falsos alarmes das duas metodologias foram medidas para valores de  $\alpha$  variando de 0.01 a .99. Em seguida, as curvas ROC (*Receiver Operating Characteristic*) (Kay (1993a)) dos dois detectores foram geradas. A figura 4.9 ilustra essas curvas. Analisando essa figura, percebe-se o aumento significativo no desempenho do detector utilizando a técnica estendida proposta nessa dissertação, já que a  $P_D$  é maior para qualquer valor de  $P_{FA}$ .

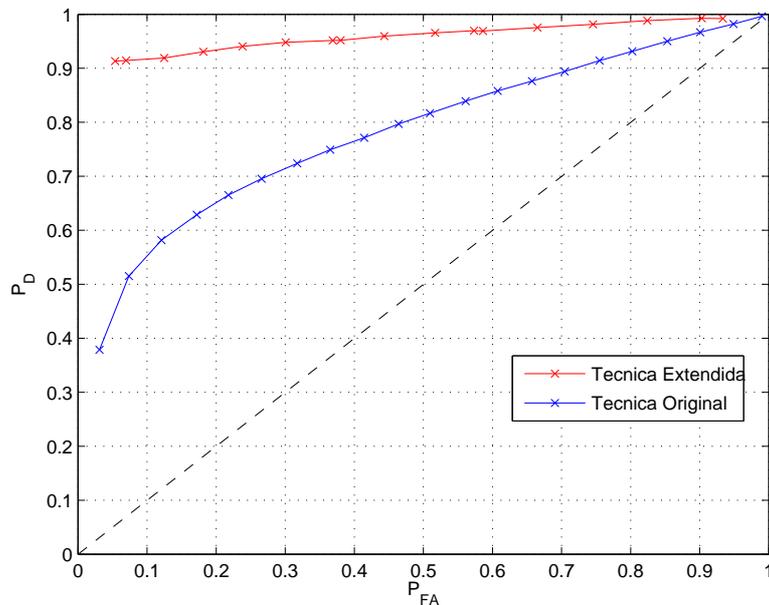


Figura 4.9: Curva ROC da técnica original e da técnica estendida proposta

### 4.3 Detecção de Apnéia Obstrutiva do Sono

A Síndrome de Apnéia Obstrutiva do Sono (SAOS) é uma síndrome caracterizada por episódios de diminuição e ou ausência de fluxo aéreo durante o sono ocasionando dessaturação de oxigênio durante o sono. Estudos realizados em Young et al. (1993) comprovam que essa síndrome está presente em cerca de 4% de homens adultos e 2% de mulheres adultas e está associada com aumento do risco de hipertensão arterial, infarto do miocárdio e derrames.

De acordo com estudos realizados em AASM (1999), o diagnóstico de SAOS é confirmado se o paciente tem 5 ou mais eventos de apnéia por hora de sono durante uma noite. Um evento de apnéia é caracterizado por uma pausa na respiração durante 10 segundos.

O exame padrão para diagnóstico de SAOS é a polisonografia. Esse exame consiste no monitoramento das atividades cerebrais, movimentos dos olhos e mandíbula, movimentos dos músculos da perna, fluxo de ar, esforço respiratório, eletrocardiograma e saturação de oxigênio durante uma noite de sono. Esse exame é caro e exige que o paciente passe uma noite no hospital (Penzel et al. (2002)).

O estudo realizado em Guilleminault et al. (1984) reporta que os eventos de apnéia são caracterizados por variações cíclicas na série temporal de intervalos RR. Essas variações são causadas por bradicardias progressivas, seguidas de taquicardias abruptas na recuperação do fôlego. Esses eventos são altamente não lineares e não estacionários diferentemente do comportamento da série observada na ausência destes, em que se assume que possui dinâmica linear e estacionária. A figura 4.10 ilustra uma série de intervalos RR em um período normal e em um período com eventos de apnéia. Analisando esse gráfico percebem-se claramente as variações cíclicas durante os eventos de apnéia.

Se um método automático de detecção de eventos de apnéia fosse desenvolvido através do monitoramento dos intervalos RR, o diagnóstico inicial de SAOS poderia ser realizado utilizando-se um aparelho de monitoramento de eletrocardiograma portátil e de baixo custo até mesmo na casa do paciente. Esse método acarretaria em uma redução de custos e incômodos causados pelo exame padrão. A figura 4.11 apresenta uma foto de um paciente preparado para o exame de polisonografia<sup>1</sup> e outro paciente utilizando um aparelho de monitoramento de ECG portátil<sup>2</sup>.

Dessa forma, utilizou-se o algoritmo de detecção de novidades para realizar detecção de eventos de apnéia através do monitoramento da série temporal de intervalos

---

<sup>1</sup><http://www.sleepapneacenter.com>

<sup>2</sup>[http://www.hc.ufpr.br/hosp/clinica\\_medica/cardiologia/holter.htm](http://www.hc.ufpr.br/hosp/clinica_medica/cardiologia/holter.htm)

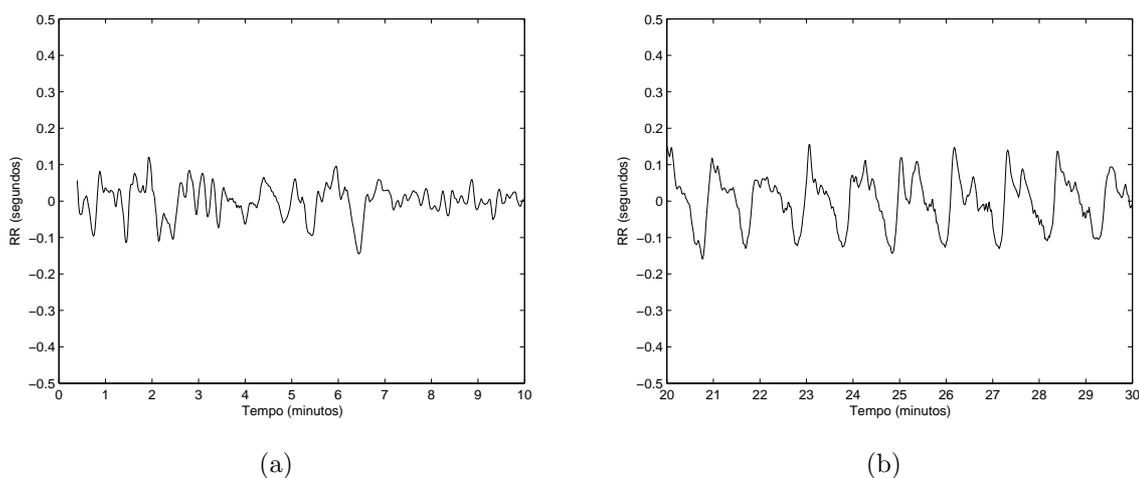


Figura 4.10: A figura (a) ilustra a série temporal de intervalos RR em um instante de tempo em que não foram detectados eventos de apnéia e a figura (b) ilustra a mesma série durante eventos de apnéia.

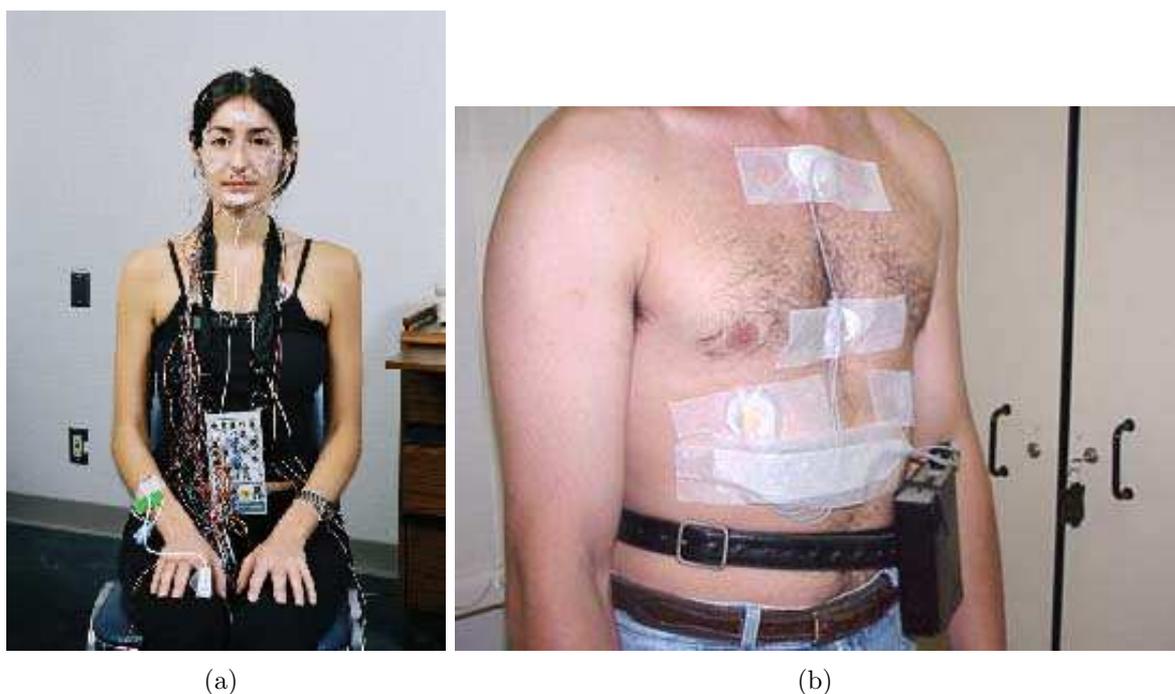


Figura 4.11: A figura (a) apresenta a foto de um paciente preparado para o exame de polisonografia e a figura (b) um paciente utilizando um aparelho de monitoramento de ECG portátil

RR. Utilizou-se a instanciação do algoritmo baseada em modelos autoregressivos e as novidades encontradas foram associadas a eventos de apnéia.

A série de intervalos RR é extraída a partir de um sinal de eletrocardiograma, por

meio de um algoritmo de detecção e classificação automática de batimentos cardíacos (Goldberger et al. (2000)) selecionando apenas batimentos normais de forma a evitar efeitos de batimentos ectópicos.

Em seguida é realizado um pré-processamento da série para que esta possa ser modelada através de um modelo autoregressivo. O pré-processamento é realizado de forma similar a Mietus et al. (2000) e é composto pelos seguintes passos:

1. um filtro de médias móveis é aplicado à série para remover o ruído gerado por erros de detecção e classificação do algoritmo da etapa anterior: para cada conjunto de 41 pontos gerados, a média do conjunto é calculada excluindo o valor central do conjunto e valores menores que 0.4 s ou maiores que 2 s; o valor central do conjunto é descartado se difere mais que 20% do valor médio;
2. a série é linearmente reamostrada a 1 Hz;
3. a série é suavizada: o valor de cada ponto da série é substituído pelo valor médio de uma janela de 5 pontos centralizada neste ponto;
4. as componentes de tendência da série são removidas: para cada ponto da série, a inclinação de uma reta gerada por regressão linear a partir de uma janela de 81 pontos centralizada no ponto atual é calculada; em seguida, o valor da regressão para o ponto atual é subtraído do valor observado.

A figura 4.12 ilustra uma série de intervalos RR antes e após o pré-processamento.

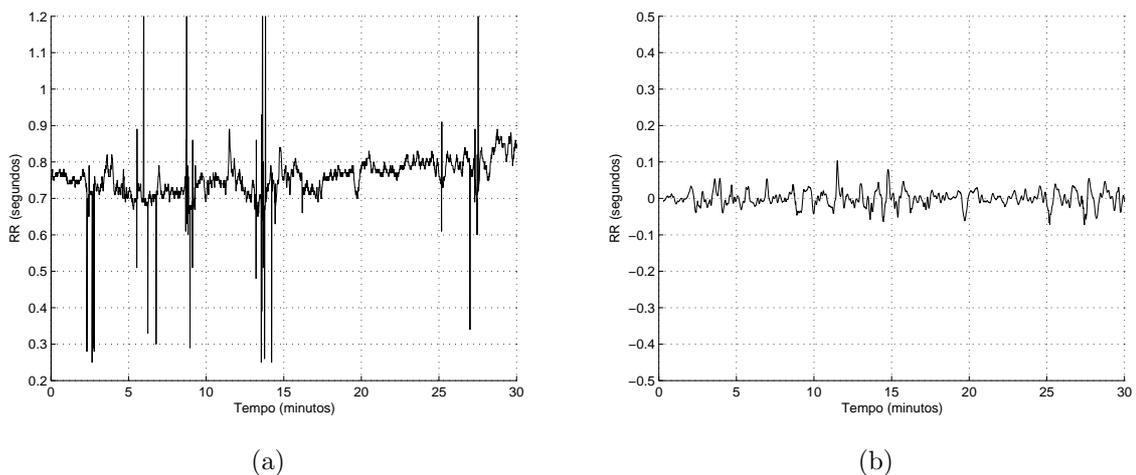


Figura 4.12: A figura (a) apresenta uma série de intervalos RR antes do pré-processamento e a figura (b) a mesma série após o pré-processamento

O conjunto de dados utilizado para realizar os experimentos é proveniente de uma competição realizada pela revista *Computers in Cardiology* em que o desafio consistia

no desenvolvimento e avaliação de detectores de eventos de apnéia baseados em sinais de eletrocardiogramas (Penzel et al. (2002)).

Esse conjunto de dados contém 70 registros com variação de um pouco menos de 7 horas a até quase 10 horas de sinais de eletrocardiogramas digitais. Além disso, cada sinal possui referências de anotações de eventos de apnéia para cada minuto realizadas por médicos especialistas por meio da análise de todos os sinais registrados em um exame de polisonografia.

Todos os registros foram previamente divididos em três grupos distintos:

- grupo A, composto por registros relativos a pacientes com clara evidência de SAOS, contendo mais de 100 minutos de eventos de apnéia;
- grupo B, composto por registros relativos a pacientes com algum grau de SAOS, contendo de 5 a 100 minutos de eventos de apnéia;
- grupo C, composto por registros de pacientes saudáveis, com menos de 5 minutos de apnéia.

Além disso, os 70 registros foram divididos em 2 subconjuntos, um conjunto de treinamento composto e um conjunto de teste ambos com 35 registros.

A competição consistia em dois desafios que deviam ser solucionados através da análise do sinal de eletrocardiograma de cada registro. O primeiro desafio era distinguir entre os registros dos pacientes saudáveis (grupo C) e dos pacientes com clara evidência de SAOS (grupo A). O segundo desafio era determinar se cada minuto de cada registro apresentava eventos de apnéia ou não.

A figura 4.13 ilustra os valores de BIC em função de  $p$  para o registro c01 do conjunto de treinamento. A maior parte dos registros apresentou comportamento semelhante, dessa forma a ordem do modelo autoregressivo foi definida como 17 para todas as séries.

Após definida a ordem do modelo, foram utilizados 10 minutos da série, com ausência de eventos de apnéia, para treiná-lo e os próximos 10 minutos para validá-lo. Porém, não foi possível validar todos os modelos, pois muitas séries geradas não possuíam pontos anotados como normais suficientes para treiná-los e validá-los.

Para detectar os eventos de apnéia, utilizou-se, em cada um dos 70 registros, a instanciação do algoritmo baseada em modelos  $AR(p)$  e intervalos de previsão. Dessa forma, os seguintes passos foram realizados:

1. a série de intervalos RR foi extraída do sinal de eletrocardiograma e pré-processada;
2. 10 minutos da série sem a presença de eventos de apnéia foram utilizados para treinar um modelo  $AR(17)$ ;

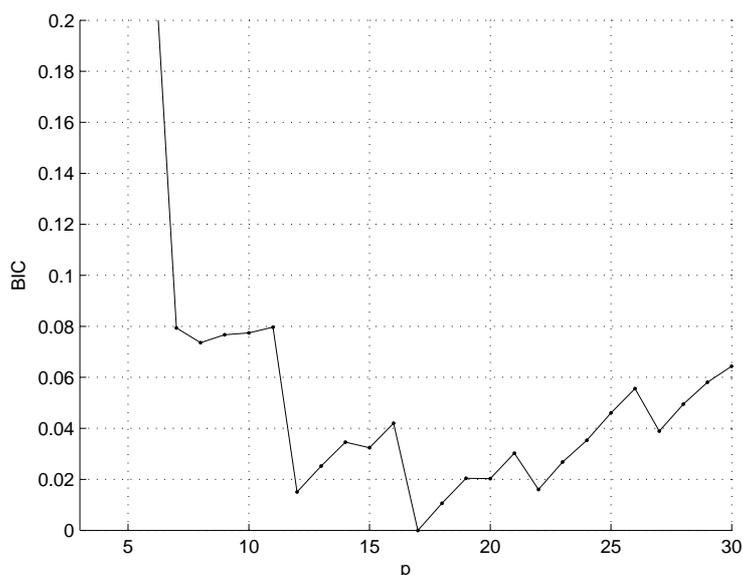


Figura 4.13: Valor de BIC em função de  $p$  para uma série de intervalos RR

3. cada um dos pontos da série foi classificado como normal ou novidade, e os pontos classificados como novidades foram associados à eventos de apnéia. Os parâmetros do algoritmo utilizados para todos os registros foram ajustados para:  $n = 240$  e  $\alpha = 0.01$ .

Como a série de intervalos RR resultante do pré-processamento foi amostrada a 1 Hz, para se chegar à classificação final de cada minuto do registro, para cada janela de 60 pontos da série correspondentes a 1 minuto do registro, utilizou-se a seguinte condição: caso 10 pontos ou mais fossem classificados como novidade, o minuto correspondente era classificado como evento de apnéia.

A distinção entre os registros de pacientes saudáveis e pacientes com SAOS foi realizada da seguinte forma: após a execução do algoritmo, se o resultado apontasse menos de 50 eventos de apnéia, o paciente era considerado saudável (grupo C).

A metodologia proposta foi capaz de distinguir corretamente entre registros de pacientes saudáveis e pacientes com SAOS em 56 dos 60 casos disponíveis (95.00%), 29 de 30 casos (96.67%) do conjunto de treinamento e 27 de 30 casos (90.00%) do conjunto de teste. Os 10 registros restantes fazem parte do grupo B.

Quando utilizada para determinar a presença de eventos de apnéia em cada minuto dos registros (segundo desafio), a metodologia foi capaz de classificar corretamente 28180 de 34313 minutos (80.03%), 13497 de 17045 minutos do conjunto de treinamento (79.18%) e 13962 de 17268 minutos do conjunto de teste (80.85%).

## 4.4 Detecção de Falhas em um Atuador

Nessa seção, a instanciação do algoritmo baseada em redes neurais será utilizada para detectar falhas em um atuador. O atuador em questão controla o fluxo de entrada de água em uma caldeira. Essa caldeira faz parte de um processo de evaporação composto por 5 estágios de uma fábrica de açúcar polonesa. Esse sistema é utilizado como modelo de um *benchmark* para avaliação de métodos de detecção e diagnóstico de falhas em sistemas de controle industrial. Esse *benchmark* foi desenvolvido por um grupo de pesquisa da europa denominado DAMADICS, *Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems* (Bartys et al. (2006)).

O atuador é composto por três partes distintas: a válvula de controle  $V$ , o servo motor pneumático  $S$  e o posicionador  $P$ . A válvula controla o fluxo de água que passa por um cano. O servo motor pneumático realiza variações no plugue da válvula de controle, de forma a agir na taxa de fluxo de água. O motor é composto por um diafragma preenchido por um fluido, de maneira que, comprimindo esse fluido, é possível realizar o deslocamento da haste do motor. A haste do motor está conectada ao plugue da válvula, controlando sua abertura. O posicionador é um dispositivo utilizado para corrigir posicionamentos incorretos da haste do motor, causados por fontes externas ou internas, tais como: atrito, variações na pressão fornecida ao servo motor, etc. A figura 4.14, retirada de Bartys et al. (2006), ilustra o diagrama esquemático do atuador.

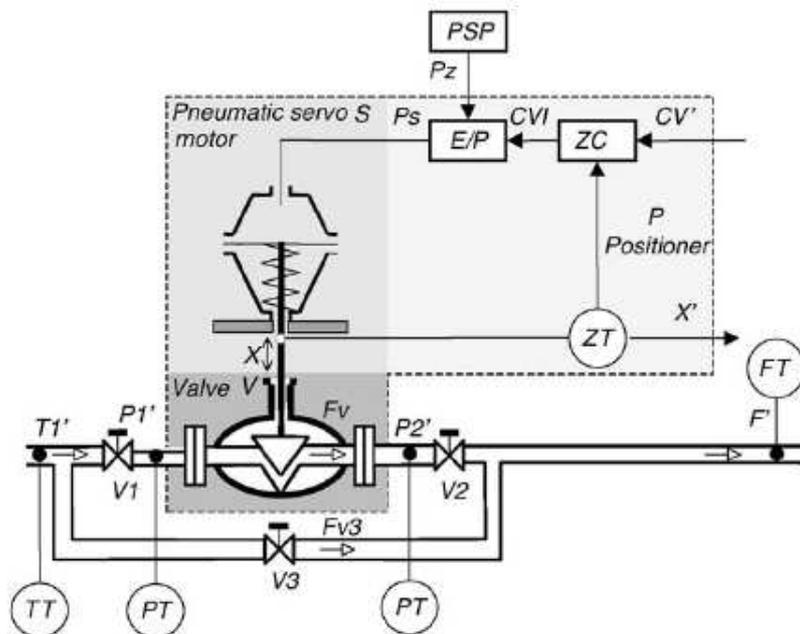


Figura 4.14: Diagrama esquemático do atuador

Utilizou-se um simulador para criar cenários de falha no atuador. Esse simulador faz parte do *benchmark* e foi desenvolvido a partir de descrições analíticas e conhecimentos prévios provindos de um atuador industrial real. Esse modelo foi sintonizado e validado com dados reais.

A figura 4.15 ilustra o modelo do simulador. As variáveis de entrada são: valor de referência do processo  $CV$ , a pressão da água na entrada da válvula  $P1$ , a pressão da água na saída da válvula  $P2$ , a temperatura da água  $T$  e o vetor de falhas  $f$ . As variáveis de saída são: a taxa de vazão da água  $F$  e a posição do tampão da válvula  $X$ . Todas as variáveis descritas são amostradas a 1 Hz e, com exceção do vetor de falhas, são acrescidas de ruído, simulando os erros de medição.

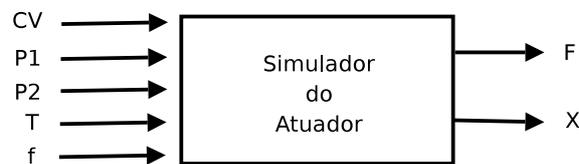


Figura 4.15: Diagrama do simulador do atuador

A tabela 4.4 descreve as 19 falhas que podem ser geradas pelo simulador.

Devido ao número ilimitado de cenários de falhas que podem ser gerados, algumas restrições foram estabelecidas durante o desenvolvimento do simulador. Inicialmente, assumiu-se apenas que cenários de falhas únicas seriam simulados. Além disso, cada cenário de falha é constituído por:

- a intensidade da falha  $f_s$ ;
- o tempo de desenvolvimento da falha  $\Delta_t$ .

Foram definidos quatro valores distintos de intensidade de falha: baixa, média, alta e máxima que correspondem respectivamente a 25%, 50%, 75% e 100% do valor máximo de referência utilizado para simular a falha.

O tempo de desenvolvimento da falha pode ser abrupto ou incipiente. Falhas abruptas ocorrem a partir de um determinado instante temporal, ou seja, dada a intensidade da falha  $f_s$  e seu instante de início  $t_{from}$ , a partir de  $t_{from}$ , o vetor de falha  $f$  passa de 0 para  $f_s$ . Já as falhas incipientes são falhas que são desenvolvidas ao longo do tempo, ou seja, a partir de  $t_{from}$ , o valor de  $f$  aumenta gradativamente até chegar ao valor máximo da intensidade da falha. A figura 4.16 ilustra os dois tempos de desenvolvimento de falha descritos.

As falhas incipientes são divididas em três categorias definidas a partir do tempo de desenvolvimento da falha:

Tabela 4.3: Tipos de Falhas simuladas

Código	Localização	Descrição
f1	Válvula de Controle	Válvula entupida
f2	Válvula de Controle	Sedimentação na válvula ou em sua base
f3	Válvula de Controle	Corrosão na válvula ou em sua base
f4	Válvula de Controle	Aumento do atrito na válvula
f5	Válvula de Controle	Vazamento externo
f6	Válvula de Controle	Vazamento interno
f7	Válvula de Controle	Evaporação média ou fluxo crítico
f8	Servo motor	Servo motor com haste torcida
f9	Servo motor	Agarramento da haste do servo motor
f10	Servo motor	Perfuração do diafragma do servo motor
f11	Servo motor	Falha na mola do servo motor
f12	Posicionador	Falha no transdutor eletro-pneumático
f13	Posicionador	Falha no sensor de posicionamento da haste
f14	Posicionador	Falha no sensor de pressão
f15	Posicionador	Falha na mola do posicionador
f16	Geral ou Externa	Queda no suprimento de pressão para o posicionador
f17	Geral ou Externa	Varição inesperada de pressão ao longo da válvula
f18	Geral ou Externa	Válvula de escape aberta totalmente ou parcialmente
f19	Geral ou Externa	Falha no sensor de medição de vazão

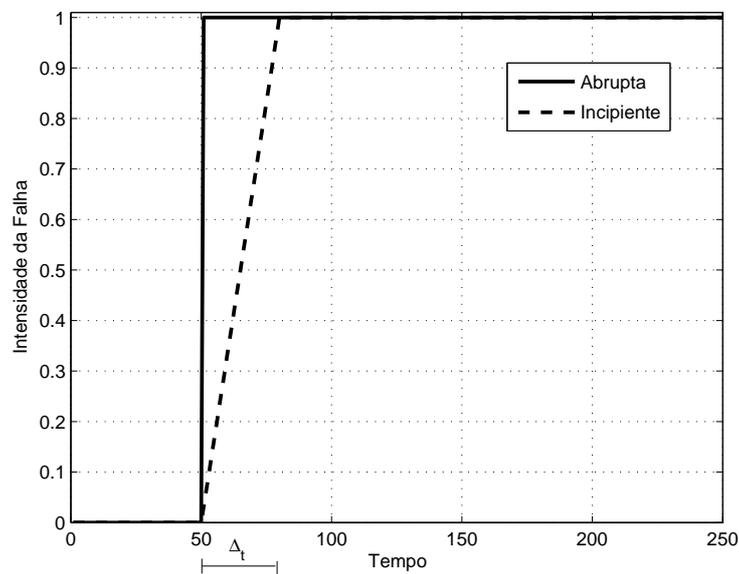


Figura 4.16: Tipos de tempo de desenvolvimento da falha

- $\Delta_{t1} = 15$  minutos
- $\Delta_{t2} = 1$  hora
- $\Delta_{t3} = 1$  dia

A tabela 4.4 descreve todos os possíveis cenários de falhas simulados.

Tabela 4.4: Cenários de Falhas

Código	Abrupta			Incipiente
	Baixa	Média	Alta	
f1	x	x	x	-
f2	-	-	x	$\Delta_{t1}$
f3	-	-	-	$\Delta_{t1}$
f4	-	-	-	$\Delta_{t2}$
f5	-	-	-	$\Delta_{t1}$
f6	-	-	-	$\Delta_{t1}$
f7	x	x	x	-
f8	x	x	x	-
f9	-	-	-	$\Delta_{t1}$
f10	x	x	x	-
f11	-	-	x	$\Delta_{t1}$
f12	x	x	x	-
f13	x	x	x	$\Delta_{t3}$
f14	x	x	x	-
f15	-	-	x	-
f16	x	x	x	-
f17	-	-	x	$\Delta_{t2}$
f18	x	x	x	$\Delta_{t1}$
f19	x	x	x	-

Para se utilizar o algoritmo proposto na detecção de falhas geradas pelo simulador, cada uma das suas variáveis de saída,  $F$  e  $X$ , foram modeladas utilizando-se um comitê de redes neurais, e as falhas foram associadas a novidades encontradas nas séries temporais resultantes de suas amostragens ao longo do tempo.

Optou-se por utilizar a instanciação baseada em redes neurais, pois as séries temporais relativas às variáveis monitoradas não atendiam aos pré-requisitos necessários para utilizar a instanciação baseada em modelos autoregressivos.

Para treinar e validar os modelos, foram gerados 2000 pontos pelo simulador, representando o funcionamento normal do sistema. Os primeiros 1000 pontos foram utilizados para treinar as redes e os últimos 1000 para validá-las. A figura 4.17 ilustra os primeiros 1000 pontos das séries.

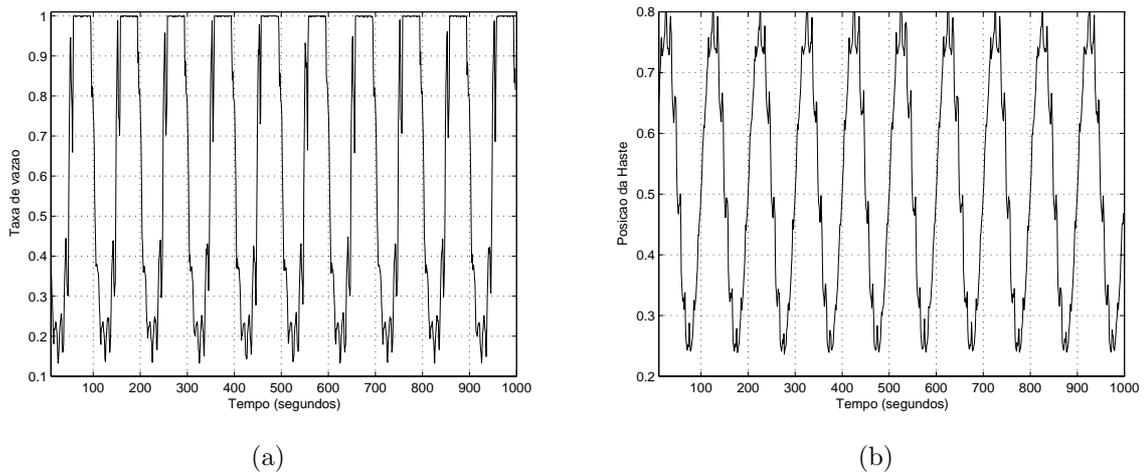


Figura 4.17: As figuras ilustram os primeiros 1000 pontos da variável de saída  $F$  (a) e  $X$  (b) utilizados no treinamento das redes

A série resultante de amostragens da taxa de vazão da água,  $F$ , foi modelada por um comitê composto por 5 redes MLP. Cada rede possui 13 entradas relativas aos 13 últimos pontos da série e 8 neurônios na camada escondida. A figura 4.18 ilustra os valores de BIC em função da topologia da rede e a figura 4.19 ilustra o gráfico de correlação amostral do erro de previsão do conjunto de pontos de validação.

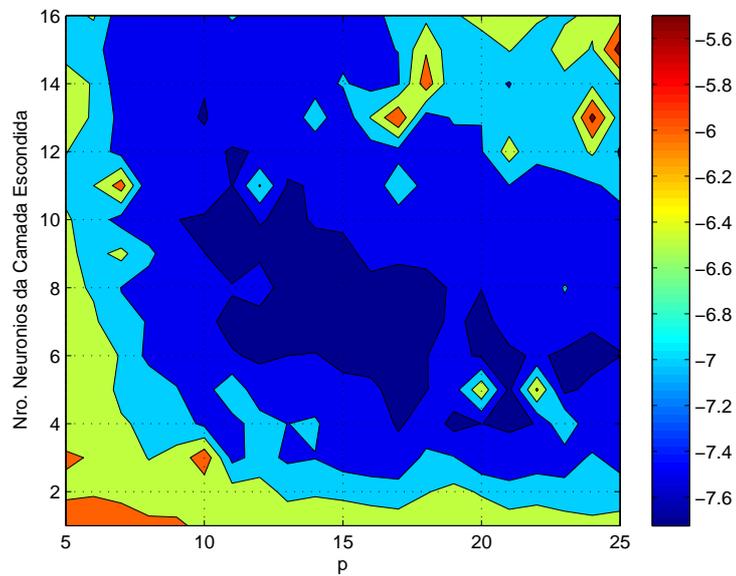


Figura 4.18: Valor de BIC em função da topologia da rede para a variável  $F$

A série resultante de amostragens do posicionamento da haste do servo motor  $X$

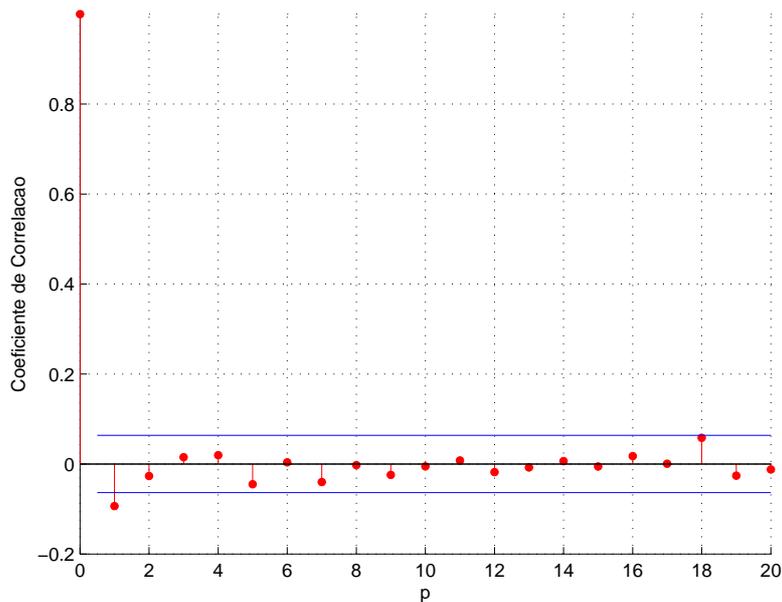


Figura 4.19: Gráfico de correlação amostral do erro de previsão da variável  $F$

foi modelada por um comitê de redes MLP composto por 5 redes. Cada rede possui 23 entradas relativas aos 23 últimos pontos da série e 5 neurônios na camada escondida. Essa topologia de rede corresponde ao segundo menor valor de BIC, uma vez que a topologia correspondente ao menor valor ( $p = 15$  e 11 neurônios na camada escondida) não se mostrou satisfatória ao ser validada. A figura 4.20 ilustra os valores de BIC em função da topologia da rede e a figura 4.21 ilustra o gráfico de correlação amostral do erro de previsão do conjunto de pontos de validação.

Cada uma das redes dos dois comitês foi treinada durante 250 épocas utilizando o algoritmo de treinamento supervisionado *Levenberg-Marquardt* com um erro objetivo de 0.000001. Além disso, para os dois modelos os parâmetros do algoritmo foram ajustados para  $n = 200$  e  $\alpha = 0.05$ .

O resultado final de detecção para cada instante de tempo foi realizado da seguinte forma: caso um dos modelos detecte uma novidade, esse instante é definido como um instante de falha.

Inicialmente, foram realizados experimentos para avaliar o desempenho do método desenvolvido na detecção de falhas abruptas.

Os experimentos foram realizados de forma similar aos experimentos realizados em Previdia e Parisin (2006). Para cada tipo de cenário de falha abrupta, foram gerados 1800 pontos, sendo que as falhas ocorrem no intervalo temporal [900, 1800].

A figura 4.22 ilustra as séries temporais geradas para simular o cenário de falha f1

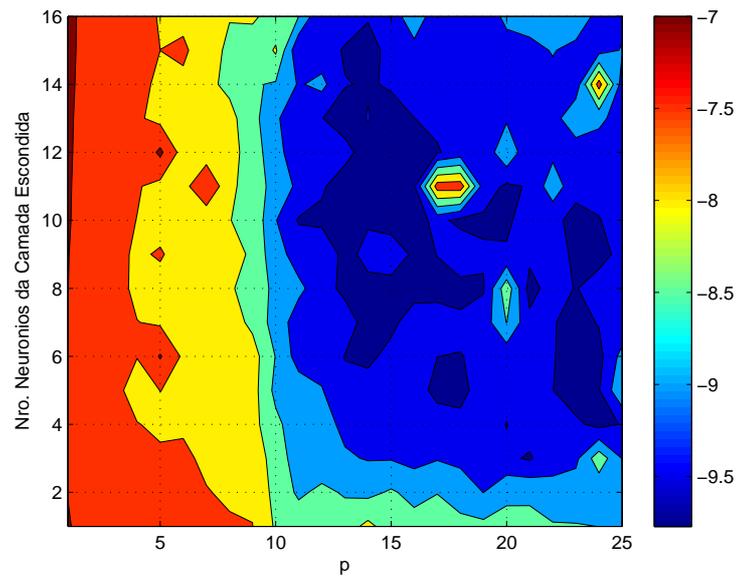


Figura 4.20: Valor de BIC em função da topologia da rede para a variável  $X$

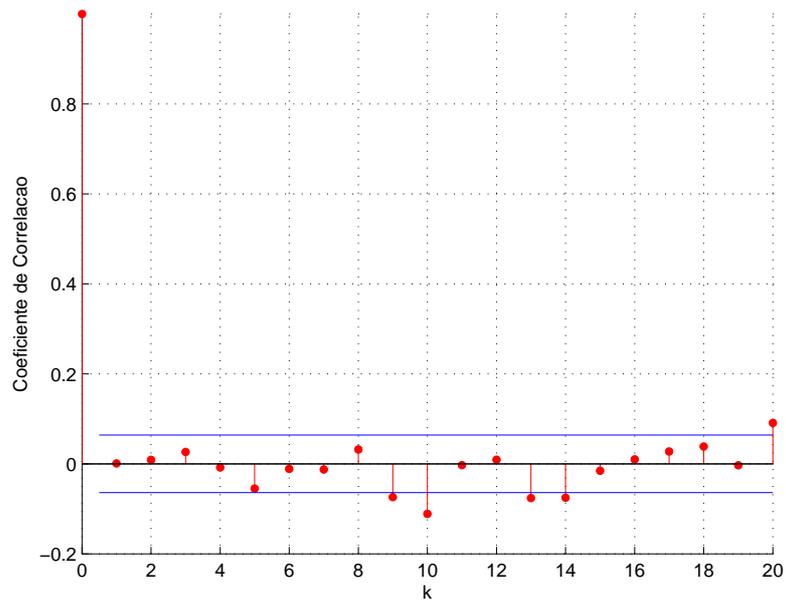


Figura 4.21: Gráfico de correlação amostral do erro de previsão da variável  $X$

de intensidade baixa e a figura 4.23 ilustra o resultado de cada um dos detectores e o resultado final. Valores iguais a 1 indicam instantes temporais em que uma falha foi detectada.

Para avaliar o desempenho do método utilizou-se as seguintes métricas: a probabi-

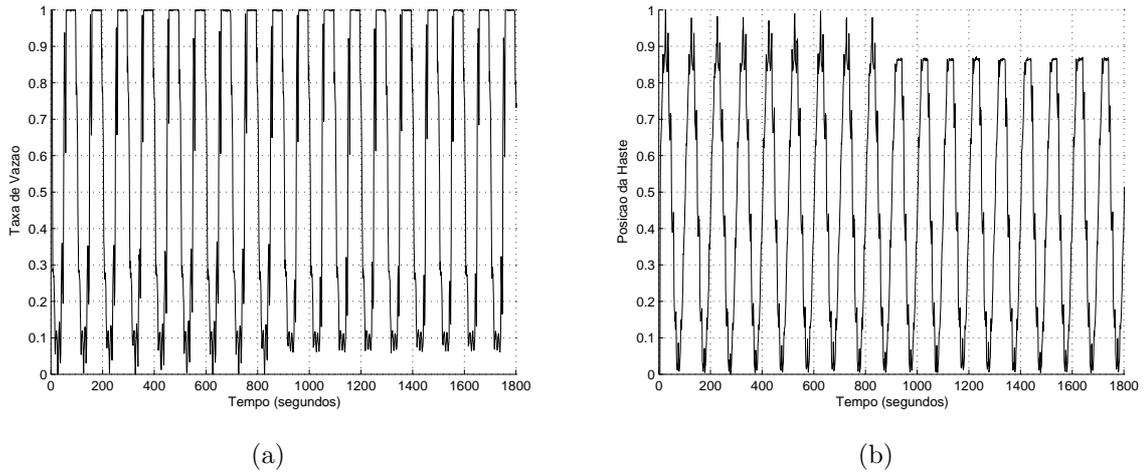


Figura 4.22: As figuras ilustram os valores das variáveis monitoradas  $F$  (a) e  $X$  (b) antes e depois do início da falha f1 abrupta de intensidade baixa ( $t_{from} = 900$ )

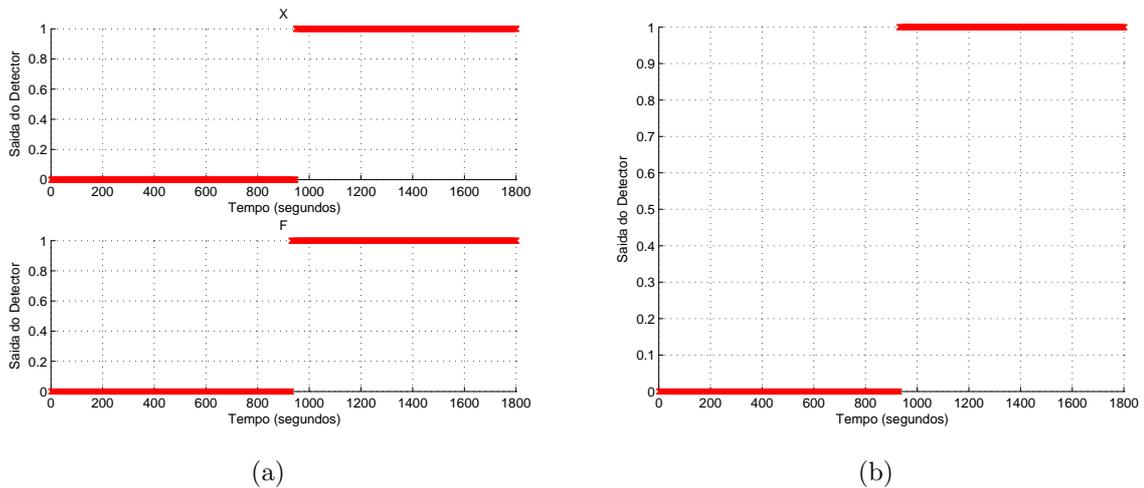


Figura 4.23: A figura (a) ilustra os resultados de cada um dos detectores e a figura (b) ilustra o resultado final ( $t_{from} = 900$ )

lidade de detecção  $P_D$ , a probabilidade de falsos alarmes  $P_{FA}$  e o tempo de detecção  $t_d$ .

A tabela 4.5 apresenta as métricas de desempenho para todos os possíveis cenários de falhas abruptas.

Analisando a tabela 4.5 percebe-se que a metodologia proposta foi capaz de detectar a maior parte dos cenários de falha abrupta sem falsos alarmes. A metodologia não foi capaz de detectar os cenários de falha relativos às falhas f8 e f14, o cenário de falha f16 com intensidade baixa e o cenário de falha f18 com intensidade alta. Nos três primeiros casos, isso ocorreu porque a variação nas duas séries temporais, no período de falha das

Tabela 4.5: Índices de Desempenho para os Cenários de Falhas Abruptas

Código	Baixa			Média			Grande		
	$P_D$	$P_{FA}$	$t_d(s)$	$P_D$	$P_{FA}$	$t_d(s)$	$P_D$	$P_{FA}$	$t_d(s)$
f1	96.78	0	29	98.78	0	11	98.22	0	10
f2	—	—	—	—	—	—	98.78	0	11
f7	98.89	0	10	98.89	0	10	98.89	0	10
f8	0	0	0	0	0	0	0	0	0
f10	98.78	0	11	98.78	0	11	98.80	0	10
f11	—	—	—	—	—	—	98.89	0	10
f12	83.68	0	147	88.68	0	102	92.56	0	67
f13	98.67	0	12	97.89	0	19	98.78	0	11
f14	0	0	0	0	0	0	0	0	0
f15	—	—	—	—	—	—	98.89	0	10
f16	0	0	0	95.34	0	43	98.78	0	11
f17	—	—	—	—	—	—	98.89	0	10
f18	98.67	0	12	98.00	0	18	0	0	0
f19	98.67	0	12	98.34	0	15	98.78	0	11

variáveis monitoradas, não foi suficientemente significativa para que o método pudesse detectar alguma novidade.

A figura 4.24 ilustra os valores das variáveis monitoradas ao longo do tempo antes e depois do início da ocorrência da falha f8 abrupta grande  $t_{from} = 900$ . Analisando os gráficos não é possível notar visualmente nenhuma variação nas séries, conforme descrito, fazendo com que o método não consiga detectar o cenário de falha.

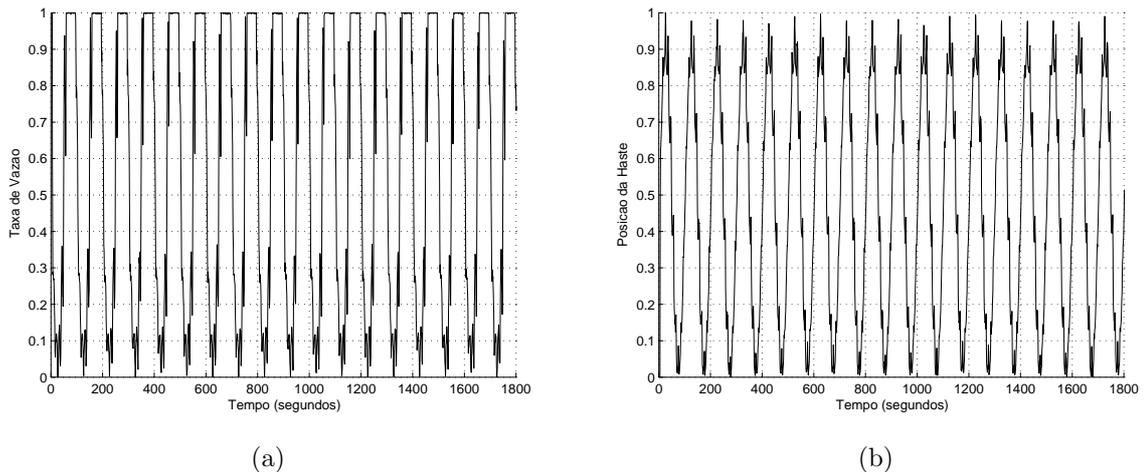


Figura 4.24: As figuras ilustram os valores das variáveis monitoradas  $F$  (a) e  $X$  (b) antes e depois do início da falha f8 abrupta de intensidade alta  $t_{from} = 900$

No caso do cenário de falha f18 abrupto com intensidade alta, o método não foi

capaz de detectar a falha, pois quando a falha ocorre, a série dos valores de  $X$  não sofre alterações, e a série dos valores de  $F$  permanece próxima a 1, já que a taxa de vazão é máxima. Quando isso ocorre, a saída das redes também permanece próxima em 1 e a novidade não é detectada. A figura 4.25 ilustra os valores das variáveis monitoradas ao longo do tempo antes e depois do início da ocorrência da falha. A figura 4.26 ilustra os valores amostrados de  $F$ , a saída do comitê e os intervalos robustos de confiança do erro de previsão. Analisando essa figura percebe-se que o erro de previsão não é suficientemente grande para que a novidade seja detectada.

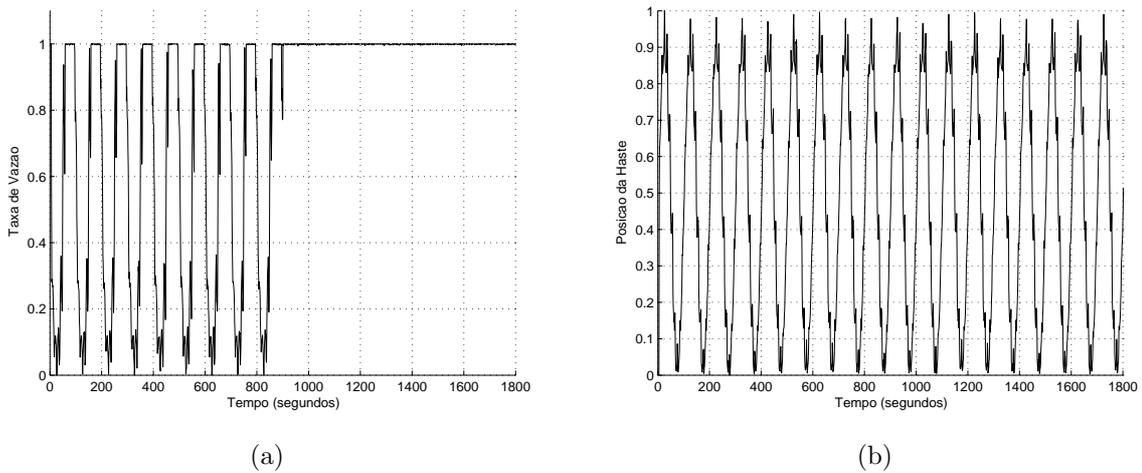


Figura 4.25: As figuras ilustram os valores das variáveis monitoradas  $F$  (a) e  $X$  (b) antes e depois do início da falha f18 abrupta de intensidade alta  $t_{from} = 900$

Em seguida, o método proposto foi utilizado para detectar falhas incipientes. Para cada cenário de falha, o número de pontos gerados variou de acordo da dinâmica temporal do cenário. Para falhas com dinâmica  $\Delta_{t1}$  foram gerados 87300 pontos, para  $\Delta_{t2}$ , 4500 e para  $\Delta_{t3}$ , 1800 pontos. Em todos os cenários  $t_{from} = 900$ .

As mesmas métricas de desempenho utilizadas nos cenários de falha abrupta foram utilizados para avaliar os resultados.

A tabela 4.6 apresenta os índices de desempenho para todos os possíveis cenários de falhas incipientes.

Analisando a tabela 4.6 percebe-se que a metodologia proposta foi capaz de detectar todos os cenários de falhas incipientes sem gerar falsos alarmes.

## 4.5 Análise dos Resultados

A metodologia desenvolvida de apoio ao diagnóstico de SAOS apresentou resultados próximos aos obtidos pelos melhores métodos que participaram da competição da re-

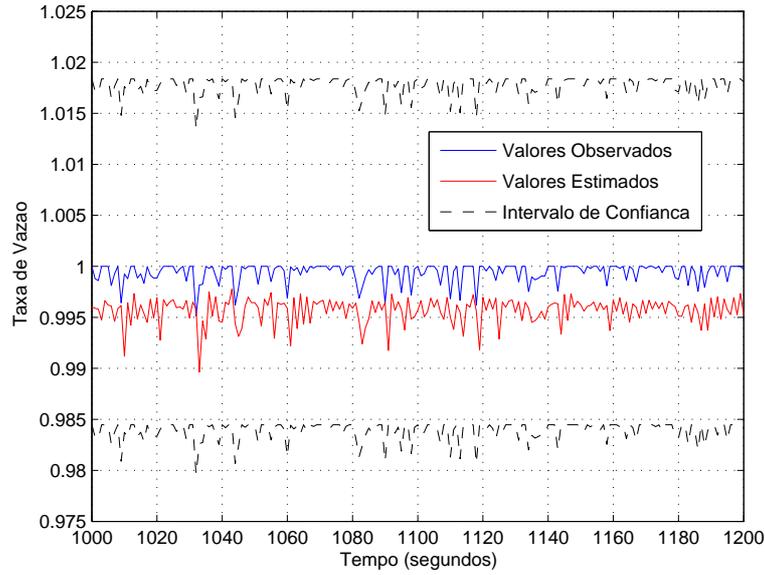


Figura 4.26: Valores observados, estimados e o intervalo robusto de confiança do erro para a série de amostragens de  $F$  no intervalo temporal [1000, 1200]

Tabela 4.6: Índices de Desempenho para os Cenários de Falhas Incipientes

<b>Código</b>	$P_D$	$P_{FA}$	$t_d(s)$
f2	97.50	0	2408
f3	95.30	0	5799
f4	95.17	0	174
f5	74.08	0	35812
f6	97.82	0	2099
f9	45.17	0	79297
f11	88.06	0	14699
f13	96.33	0	52
f17	94.50	0	198
f18	97.52	0	82866

vista *Computers in Cardiology* no ano de 2000. Em Penzel et al. (2002) foram publicados os métodos que obtiveram os melhores resultados na competição analisando o resultado para os registros do conjunto de testes. Quanto à distinção de registros de pacientes saudáveis e pacientes com SAOS, os resultados obtidos comparam-se com os 10 melhores resultados da competição. Já quanto à detecção da presença de apnéia em cada minuto dos registros, os resultados foram um pouco inferiores aos melhores resultados do concurso. Chegou-se a uma taxa de acerto de cerca de 80,2%, os melhores resultados da competição tiveram taxas de acerto variando de 84,5 a 92,6%. Apesar

dos resultados serem inferiores, o método proposto tem como vantagem a simplicidade, uma vez que a detecção é feita por um algoritmo com ordem de complexidade linear no tempo e espaço, podendo ser executado em *hardware* de baixo custo.

Vale ressaltar que esse trabalho se resumiu a detecção de novidades em séries temporais estacionárias. Dessa forma, assumiu-se que a série temporal resultante do pré-processamento dos intervalos RR era estacionária. Espera-se que melhores resultados sejam obtidos utilizando-se uma variação do algoritmo proposto baseado em um modelo autoregressivo adaptativo. Maiores detalhes das modificações necessárias para que esse tipo de modelo seja utilizado são sugeridas na seção 5.1.

A aplicação do algoritmo nos dados de *benchmark* desenvolvido pela DAMADICS também apresentou bons resultados, porém é difícil realizar uma comparação com os métodos encontrados na literatura aplicados a esse problema, pois apesar do *benchmark* definir uma forma padrão de medir o desempenho integrando o código do método com código do simulador, não foi encontrado nenhum trabalho que realize essa integração.

Os dados dos experimentos relativos a falhas abruptas foram gerados da mesma maneira descrita em Previdia e Parisin (2006), porém esse método utiliza apenas alguns dos cenários de falhas abruptas de intensidade alta. A tabela 4.7 apresenta uma comparação das métricas de desempenho dos experimentos realizados e dos resultados do método proposto em Previdia e Parisin (2006) para os cenários de falha abrupta de intensidade alta que os dois métodos foram capazes de detectar. Os dois métodos possuem como parâmetro o nível de significância, dessa forma os resultados apresentados foram gerados para  $\alpha = 0.05$ . Analisando essa tabela, percebe-se que o método desenvolvido apresenta resultados iguais ou melhores para todos os cenários de falhas descritos.

Tabela 4.7: Comparação dos resultados obtidos nos experimentos de detecção de falhas

Código	Método Proposto			Previdia e Parisin (2006)		
	$P_D$	$P_{FA}$	$t_d$	$P_D$	$P_{FA}$	$t_d(s)$
f1	98.22	0	10	83.00	0.02	153
f2	98.78	0	11	81.00	0.02	171
f7	98.89	0	10	83.00	0.02	153
f10	98.80	0	10	74.00	0.02	234
f11	98.89	0	10	99.00	0.02	10
f12	92.56	0	67	66.22	0.02	306
f15	98.89	0	10	92.00	0.02	72
f17	98.89	0	10	99.00	0	9
f19	98.78	0	11	78.00	0.02	63

# Capítulo 5

## Conclusão

O problema de detecção de novidades em séries temporais foi estudado nesse trabalho. Primeiramente, o problema foi definido e foi realizada uma revisão bibliográfica de alguns dos principais métodos encontrados na literatura.

Em seguida, um novo método de detecção de novidades em séries temporais unimodais foi proposto. Esse método foi desenvolvido a partir de conceitos definidos nos métodos revisados.

A idéia inicial do trabalho era desenvolver um método de detecção de novidades que utilizasse um modelo de previsão, seja ele estatístico ou neural. Inicialmente, o método proposto em Oliveira e Meira (2006) foi avaliado. Porém, esse método foi desenvolvido para resolver principalmente, o problema de detecção de fraudes em sistemas financeiros, em que uma novidade é definida por apenas um ponto, diferente dos problemas que foram tratados nessa dissertação. Desta maneira, havia a necessidade de realizar uma extensão desse método para que este pudesse ser aplicado nos problemas apresentados ao longo dessa dissertação.

Para tratar uma novidade como uma sequência de pontos, utilizou-se o *framework* proposto em Ma e Perkins (2003a), porém, com as seguintes modificações:

- Em Ma e Perkins (2003a) não se calcula o intervalo de confiança de  $p(|E_n(t)|)$ , ao invés disso, um evento é definido como evento de novidade, caso a sua norma seja maior que o valor esperado de  $|E_n|(t)$  e que  $p(|E_n(t)|) < 1 - \alpha$ . Esse trabalho opta por calcular os intervalos de confiança, pois acredita-se que os intervalos de confiança reflitam mais fielmente, do ponto de vista estatístico, a importância do nível de significância.
- Além disso, em Ma e Perkins (2003a), sempre que um evento é classificado como evento de novidade, todos os pontos que fazem parte do evento também são classificados como novidade. Optou-se por classificar apenas o último ponto de

cada evento como novidade, quando um evento é classificado como evento de novidade, para diminuir a probabilidade de falsos alarmes.

Optou-se por desenvolver um método que não fosse dependente do modelo de previsão escolhido. Então, desenvolveu-se um algoritmo genérico que pudesse ser utilizado com qualquer modelo. Foram apresentadas duas instanciações desse método representando dois cenários distintos, o primeiro, quando a forma da função de distribuição de probabilidade do erro de previsão é conhecida e a segunda quando essa função não é conhecida.

Com isso, as duas instanciações desenvolvidas utilizam um modelo de previsão e calculam limiares de tolerância para o erro de previsão de maneira similar a Oliveira e Meira (2006), porém realizam um processamento desse resultado utilizando uma versão modificada do *framework* proposto em Ma e Perkins (2003a).

O processamento do erro de previsão utilizado nas duas instanciações nada mais é que uma metodologia de classificação de uma classe, baseada na estimação da distribuição de probabilidade. Assim, qualquer outro método de classificação de uma classe poderia ser utilizado, tendo o erro de previsão como padrão de entrada.

Embora o algoritmo da etapa de classificação ilustrado na figura 3.1 possua ordem de complexidade temporal  $O(nm)$ , sendo  $m$  o número de pontos a serem classificados, é possível desenvolver uma versão desse algoritmo com ordem de complexidade linear no tempo e espaço. O apêndice A ilustra essa versão.

Uma vez definido o método, experimentos foram realizados para comprovar sua eficácia e analisar a influência de seus parâmetros.

O método proposto possui dois parâmetros:

- nível de significância,  $\alpha$ ;
- tamanho do evento,  $n$ .

O nível de significância reflete a probabilidade esperada de falsos alarmes. A partir do valor desse parâmetro é possível realizar uma estimação de  $P_{FA}$  através da equação (4.3). Além disso, a probabilidade de detecção é proporcional ao valor de  $\alpha$ .

O tamanho do evento,  $n$ , também tem impacto em  $P_D$  e  $P_{FA}$ . Existe um valor ideal para  $n$  de forma a maximizar  $P_D$ , porém o cálculo do valor ideal ainda é uma questão em aberto.

Finalmente, o método proposto foi validado através de dois experimentos com problemas reais.

Sumarizando, esta dissertação possui as seguintes contribuições:

- uma revisão de algumas das principais técnicas de detecção de novidades em séries temporais presentes na literatura, juntamente com algumas sugestões de melhoria;
- um algoritmo genérico de detecção de novidades baseado em modelos de previsão, assim como duas instanciações desse algoritmo.
- Uma extensão do método proposto em Oliveira e Meira (2006), para que este possa ser utilizado mais eficientemente na detecção de novidades em séries em que uma novidade seja caracterizada por uma sequência de pontos;
- uma metodologia de detecção de eventos de apnéia que pode ser utilizada como diagnóstico inicial de SAO , através de um exame mais barato e prático, se comparado ao exame tradicional (polisonografia);
- uma metodologia de detecção de falhas em sistemas dinâmicos. Essa metodologia utiliza apenas dados de funcionamento normal do sistema durante seu treinamento, apresentando aplicação em sistemas com dados de falha escassos ou ausentes.

Contudo, o método desenvolvido apresenta uma deficiência, pois, se durante o treinamento do detector, houver dados relativos a eventos de novidades disponíveis, não há como utilizá-los, ao contrário de algumas metodologias revisadas.

## 5.1 Trabalhos Futuros

Propõe-se como trabalhos futuros:

- definição dos intervalos de tolerância do erro para outros modelos estatísticos, tais como MA, ARMA e ARIMA (Box e Jenkins (1990));
- aplicação em séries temporais multivariadas;
- modificação da técnica para utilizar modelos de previsão adaptativos.

### 5.1.1 Definição dos Intervalos de Tolerância do Erro de Previsão para Outros Modelos Estatísticos

A primeira instanciação do algoritmo genérico apresentada utiliza um modelo estatístico em que a função de densidade de probabilidade do erro de previsão é conhecida.

O modelo escolhido,  $AR(p)$ , só pode ser utilizado caso a série atenda a alguns pré-requisitos detalhados na seção 3.2.5.

Para que outros modelos estatísticos possam ser utilizados, deve-se calcular os intervalos de previsão para cada modelo e, a partir desses intervalos, o intervalo de tolerância do erro de previsão.

### 5.1.2 Séries Multivariadas

A técnica descrita nesse trabalho limitou-se na detecção de novidades em séries temporais univariadas, porém acredita-se que o algoritmo proposto possa ser utilizado sem modificações na detecção de novidades em séries multivariadas, bastando apenas utilizar um modelo de previsão multivariado e calcular os intervalos de tolerância do erro de previsão para esses modelos.

O problema apresentado na seção 4.4 de detecção de falhas em um atuador poderia ser resolvido utilizando-se um modelo multivariado, de forma que as séries temporais das saídas do simulador,  $X$  e  $F$ , fossem representadas não somente por seus valores atrasados no tempo, mas também por valores das entradas do simulador  $P1$ ,  $P2$ ,  $CV$ ,  $T$ . Acredita-se que esse modelo de previsão seria mais robusto, uma vez que descreveria melhor a dinâmica real do sistema.

### 5.1.3 Modelos Adaptativos

As duas técnicas apresentadas nesse trabalho limitam-se à análise de séries temporais estacionárias, devido aos modelos de previsão propostos. Porém, em muitos casos práticos, não se pode assumir a estacionariedade dos dados, ou se pode assumir estacionariedade apenas em intervalos temporais.

Nesse caso, deve-se utilizar um modelo de previsão adaptativo, em que os parâmetros do modelo são ajustados à medida que os momentos estatísticos da série analisada variem ao longo do tempo. Para se utilizar tais modelos, o algoritmo deve ser modificado de forma que os intervalos de tolerância do erro de previsão passem a ser dinâmicos, sendo atualizados à medida que a estrutura do modelo varie.

Outro ponto importante a ser considerado na modificação do algoritmo para modelos adaptativos seria a utilização somente de novos pontos classificados como normais na atualização do modelo. Pois, caso pontos classificados como novidades forem utilizados para atualizar o modelo, seu desempenho preditivo poderia ser deteriorado uma vez que este não mais estaria ajustado para os dados de funcionamento normal.

## 5.2 Produção Científica

O seguinte artigo foi gerado a partir dos estudos realizados nessa dissertação:

- Lemos A. P., Terra-Crioulo C. J., Caminhas W. M.(2007). ECG Anomalies Identification Using a Time Series Novelty Detection Technique. *IV Latin-American Congress on Biomedical Engineering (CLAIB 2007)*.

Além disso, os seguintes artigos foram submetidos com a aceitação ainda pendente:

- Lemos A. P., Terra-Crioulo C. J., Caminhas W. M.(2007). Algoritmo Genérico de Detecção de novidades em Séries Temporais Utilizando Modelos de Previsão. *I Simpósio Brasileiro de Inteligência Computacional (SBIC 2007)*
- Lemos A. P., Terra-Crioulo C. J., Caminhas W. M.(2008). Screening of Obstructive Sleep Apnea by RR Interval Time Series Using a Time Series Novelty Detection Technique. *International Conference on Bio-Inspired Systems and Signal Processing (BIOSIGNALS 2008)*

# Apêndice A

## Algoritmo com Ordem de Complexidade Linear

Embora o algoritmo da etapa de classificação ilustrado na figura 3.1 possua ordem de complexidade temporal  $O(nm)$ , sendo  $m$  o número de pontos a serem classificados e  $n$  o tamanho do evento, é possível desenvolver uma versão desse algoritmo com ordem de complexidade linear no tempo e espaço. O algoritmo 3 ilustra o pseudo-código do algoritmo modificado.

loa 3: Pseudocódigo da etapa de classificação do algoritmo genérico com ordem de complexidade linear no tempo e espaço

```

1: function NOVELTY_DETECTOR( $x, M_x(t), [\varepsilon_1, \varepsilon_2], \gamma, n, \alpha$ )
2:    $m = \text{length}(x)$ 
3:   for  $t \leftarrow 1, m$  do
4:      $e(t) \leftarrow M_x(t) - x(t)$ 
5:     if  $e(t) \in [\varepsilon_1, \varepsilon_2]$  then
6:        $O(t) \leftarrow 1$ 
7:     else
8:        $O(t) \leftarrow 0$ 
9:     end if
10:  end for
11:   $|E_n| \leftarrow 0$ 
12:  for  $t \leftarrow 1, n$  do
13:     $|E_n| \leftarrow |E_n| + O(t)$ 
14:  end for
15:  for  $t \leftarrow 1, m - n + 1$  do
16:    if  $|E_n| > \gamma$  then
17:       $\text{class}(t + n - 1) \leftarrow 1$ 
18:    else
19:       $\text{class}(t + n - 1) \leftarrow 0$ 
20:    end if
21:     $|E_n| \leftarrow |E_n| + (-1)^{O(t)}(O(t) \text{ XOR } O(t + n))$ 
22:  end for
23:  return  $\text{class}$ 
24: end function

```

# Referências Bibliográficas

- AMERICAN ACADEMY OF SLEEP MEDICINE TASK FORCE AASM( 1999). Sleep-related breathing disorders in adults: recommendations for syndrome definition and measurement techniques in clinical research. *Sleep*, 22:667–689.
- Bartys, M.; Patton, R.; Syfert, M.; de las Heras, S. e Quevedo, J.( 2006). Introduction to the DAMADICS actuator FDI benchmark study. *Control Engineering Practice*, 14:577–596.
- Box, G. E. P. e Jenkins, G.( 1990). *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated.
- Cottrell, M.; Girard, B.; Girard, Y.; Mangeas, M. e Muller, C.( 1995). Neural modeling for time series: a statistical stepwise method for weight elimination. *IEEE Transactions on Neural Networks*, 6:1355–1364.
- Cover, T. M.( 1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, 14:326–334.
- Dasgupta, D. e Forrest, S.( 1996). Novelty detection in time series data using ideas from immunology. In *Neural Information Processing Systems (NIPS) Conference*.
- Dasgupta, D. e Gonzalez, F.( 2002). An Immunity-Based Technique to Characterize Intrusions in Computer Networks. *IEEE Trans. Evol. Comput.*, 6:1081–1088.
- D’haeseleer, P.; Forrest, S. e Helman, P.( 1996). An immunological approach to change detection: Algorithms, analysis and implications. In *SP ’96: Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pp. 110–119, Washington, DC, USA. IEEE Computer Society.
- Dingfei Ge, N. S. e Krishnan, S. M.( 2002). Cardiac arrhythmia classification using autoregressive modeling. *BioMedical Engineering OnLine*, 1(5).

- Dorffner, G.( 1996). Neural Networks for Time Series Processing. *Neural Network World*, 6:447–468.
- Ehlers, R. S.( 1996). Análise de séries temporais. Web doc. <http://leg.est.ufpr.br/ehlers/notas>.
- Forrest, S.; Perelson, A.; Allen, L. e Cherukuri, R.( 1995). A change-detection algorithm inspired by the immune system. In *IEEE Transactions on Software Engineering*.
- Forrest, S.; Perelson, A. S.; Allen, L. e Cherukuri, R.( 1994). Self-nonsel self discrimination in a computer. In *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, pp. 202–212, Oakland, CA. IEEE Computer Society Press.
- Goldberger, A. L.; Amaral, L. A. N.; Glass, L.; Hausdorff, J. M.; Ivanov, P. C.; Mark, R. G.; Mietus, J. E.; Moody, G. B.; Peng, C.-K. e Stanley, H. E.( 2000). Physio-Bank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation*, 101(23):e215–e220. Circulation Electronic Pages: <http://circ.ahajournals.org/cgi/content/full/101/23/e215>.
- Gonzalez, F. A. e Dasgupta, D.( 2002). Combining negative selection and classification techniques for anomaly detection. In *Proceedings of the 2002 Congress on Evolutionary Computation CEC2002*, pp. 1081–1088, Honolulu, HI.
- Guilleminault, C.; Connolly, S.; Winkle, R.; Melvin, K. e Tilkian, A.( 1984). Cyclical variation of the heart rate in sleep apnoea syndrome. Mechanisms, and usefulness of 24 h electrocardiography as a screening technique. *Lancet*, 1:126–131.
- Haykin, S.( 1999). *Neural Networks - A Comprehensive Foundation*. Prentice Hall, 2 edição.
- Hornik, K.; Stinchcombe, M. e White, H.( 1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366.
- Kay, S. M.( 1993a). *Fundamentals of statistical signal processing: detection theory*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Kay, S. M.( 1993b). *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Keogh, E.; Lin, J. e Fu, A.( 2005). Hot sax: Efficiently finding the most unusual time series subsequence. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pp. 226–233, Washington, DC, USA. IEEE Computer Society.

- Keogh, E.; Lonardi, S. e Chi' Chiu, B. Y.( 2002). Finding surprising patterns in a time series database in linear time and space. In *ACM Knowledge Discovery and Data Mining*, pp. 550–556.
- Lin, J.; Keogh, E.; Lonardi, S. e Chiu, B.( 2003). A symbolic representation of time series, with implications for streaming algorithms. In *DMKD '03: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pp. 2–11, New York, NY, USA. ACM Press.
- Ma, J. e Perkins, S.( 2003a). Online novelty detection on temporal sequences. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 613–618, New York, NY, USA. ACM Press.
- Ma, J. e Perkins, S.( 2003b). Time-series novelty detection using one-class support vector machines. In *Neural Networks, 2003. Proceedings of the International Joint Conference*, pp. 1741–1745.
- Markou, M. e Singh, S.( 2003a). Novelty Detection: A Review Part 1: Statistical Approaches. *Signal Processing*, 83:2499–2521.
- Markou, M. e Singh, S.( 2003b). Novelty Detection: A Review Part 2: Neural Networks Based Approaches. *Signal Processing*, 83:2481–2497.
- Marquardt, D.( 1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal of Applied Mathematics*, 11:164–168.
- Masters, T.( 1995). *Neural, Novel and Hybrid Algorithms for Time Series Prediction*. John Wiley & Sons, Inc., New York, NY, USA.
- Mietus, J. E.; Peng, C. K.; Ivanov, P. C. e Goldberger, A. L.( 2000). Detection of obstructive sleep apnea from cardiac interbeat interval time series. *Computers in Cardiology*, 27:753–756.
- Oliveira, A. L. I. e Meira, S. R. L.( 2006). Detecting novelties in time series through neural networks forecasting with robust confidence intervals. *Neurocomputing*, 70:79–92.
- Oliveira, A. L. I.; Neto, F. B. L. e Meira, S. R. L.( 2003). Novelty detection for short time series with neural networks. In *The Third International Conference on Hybrid Intelligent Systems*, pp. 66–75.
- Papoulis, A.( 1991). *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 3 edição.

- Parzen, E.( 1962). On the estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33:1065–1076.
- Penzel, T.; McNames, J.; de Chazal, P.; Raymond, B.; Murray, A. e Moody, G.( 2002). Systematic comparison of different algorithms for apnoea detection based on electrocardiogram recordings. *Medical & Biological Engineering & Computing*, 40:402–407.
- Previdida, F. e Parisin, T.( 2006). Model-free actuator fault detection using a spectral estimation approach: the case of the DAMADICS benchmark problem. *Control Engineering Practice*, 14:635–644.
- Schurmann, J.( 1996). *Pattern classification: a unified view of statistical and neural approaches*. John Wiley & Sons, Inc., New York, NY, USA.
- Schwarz, G.( 1978). Estimating the Dimension of a Model. *The Annals of Statistics*, 6:461–464.
- Smola, A. e Schoelkopf, B.( 1998). A tutorial on support vector regression. Technical Report NeuroCOLT2 Technical Report NC2-TR-1998-030.
- Tax, D. M. J.( 2001). *One-class classification; Concept-learning in the absence of counter-examples*. PhD thesis, Delft University of Technology.
- Ubeyli, E. D.( 2007). ECG beats classification using multiclass support vector machines with error correcting output codes. *Digit. Signal Process.*, 17(3):675–684.
- Vasconcelos, G.( 1995). *An investigation of feedforward neural networks with respect to the detection of spurious patterns*. PhD thesis, University of Kent.
- Virone, G.; Wood, A.; Selavo, L.; Cao, Q.; Fang, L.; Doan, T.; He, Z.; Stoleru, R.; Lin, S. e Stankovic, J.( 2006). An assisted living oriented information system based on a residential wireless sensor network. In *Proceedings of the 1st Distributed Diagnosis and Home Healthcare (D2H2) Conference*, pp. 95–100, Arlington, Virginia, USA.
- Young, T.; Palta, M.; Dempsey, J.; Skatrud, J.; Weber, S. e Badr, S.( 1993). The occurrence of sleep-disordered breathing among middle-aged adults. *New England Journal of Medicine*, 328:1230–1235.
- Ypma, A. e Duin, R.( 1998). Support objects for domain approximation. *ICANN'98, Proceedings of the 8th International Conference on Artificial Neural Networks*, 1:161–165.

- Zhang, G.; B., E. P. e Y., M. H.( 1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14:35–62.
- Zhao, Q. e Zhang, L.( 2005). ECG Feature Extraction and Classification Using Wavelet Transform and Support Vector Machines. *Neural Networks and Brain, 2005*, 2:1089–1092.