UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

UM DETECTOR DE ANOMALIAS DE TRÁFEGO DE REDE BASEADO EM *WAVELETS*

DISSERTAÇÃO DE MESTRADO

Tiago Perlin

Santa Maria, RS, Brasil

2010

UM DETECTOR DE ANOMALIAS DE TRÁFEGO DE REDE BASEADO EM *WAVELETS*

por

Tiago Perlin

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do grau de
Mestre em Computação

Orientador: Prof. Dr. Raul Ceretta Nunes

Co-orientador: Profª. Drª. Alice de Jesus Kozakevicius

Santa Maria, RS, Brasil

2010

Universidade Federal de Santa Maria
Centro de Tecnologia
Programa de Pós-Graduação em Informática

A Comissão Examinadora, abaixo assinada,
aprova a Dissertação de Mestrado

UM DETECTOR DE ANOMALIAS DE TRÁFEGO DE REDE BASEADO EM *WAVELETS*

elaborada por
Tiago Perlin

como requisito parcial para obtenção do grau de
Mestre em Computação

COMISSÃO EXAMINADORA:

Prof. Dr. Raul Ceretta Nunes
(Presidente/Orientador)

Prof. Dr. Christian Emilio Schaerer Serra (UNA)

Prof^{a}. Dr^{a}. Roseclea Duarte Medina (UFSM)

Santa Maria, 07 de outubro de 2010.

Dedicado aos meus pais Sérgio e Ivone.

AGRADECIMENTOS

Ao Programa de Pós-Graduação em Informática (PPGI) da UFSM pela oportunidade de realização deste trabalho.

À Cappes (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo apoio financeiro fundamental para a realização dos trabalhos.

Ao Instituto Nacional de Pesquisas Espaciais (INPE) que, por meio de uma parceria com GMicro da UFSM, disponibilizou uma sala e infraestrutura para o Laboratório de Gerência e Segurança de Redes de Computadores.

Aos integrantes dos grupos de pesquisa Gestão e Tecnologia em Segurança da Informação (GTSeg) e Grupo de Microeletrônica (GMicro) pela acolhida.

Em especial ao orientador deste trabalho Prof. Dr. Raul Ceretta Nunes pela confiança depositada, pela orientação e pelo esforço em garantir os meios para a realização deste trabalho.

Em especial à co-orientadora neste trabalho Profª. Drª. Alice de Jesus Kozakevicius pela orientação e sugestões em todas as fases do trabalho.

Ao ex-colega de grupo de pesquisa Bruno Lopes Dalmazo pela colaboração no início do projeto, pelo companheirismo e amizade.

Ao colega Renato Preigschadt de Azevedo pela ajuda na coleta de dados e pela revisão de artigos e deste trabalho.

Aos colegas Francisco Vogt e Érico Hoff Amaral pela ajuda nas atividades de pesquisa.

Ao amigo Dr. Sayed Mohammad Salman pela amizade e pela ajuda no idioma inglês.

Aos meus pais Sergio e Ivone e à minha namorada Glaucia pelo carinho, compreensão e apoio durante todo o curso.

“The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards.”

Gene Spafford

RESUMO

Dissertação de Mestrado
Programa de Pós-Graduação em Informática
Universidade Federal de Santa Maria

UM DETECTOR DE ANOMALIAS DE TRÁFEGO DE REDE BASEADO EM *WAVELETS*

Autor: Tiago Perlin
Orientador: Prof. Dr. Raul Ceretta Nunes
Co-orientador: Prof[a]. Dr[a]. Alice de Jesus Kozakevicius
Local e data da defesa: Santa Maria, 07 de outubro de 2010.

Ataques em redes de computadores comprometem a segurança do sistema e degradam o desempenho da rede causando prejuízos aos usuários e às organizações. Sistemas Detectores de Intrusões de Rede são usados para a detecção de ataques ou outras atividades maliciosas por meio da análise do tráfego. A detecção de anomalias é uma abordagem de análise usada na detecção de intrusão onde se assume que a presença de anomalias no tráfego, desvios em relação a um comportamento padrão, é indicativo de um ataque ou defeito. Uma das principais dificuldades dos Sistemas de Detecção de Intrusão de Rede baseados em anomalias está na construção do perfil devido à complexidade do tráfego de rede. Métodos derivados da Análise de Sinais, dentre os quais, a Transformada *Wavelet*, têm recentemente demonstrado aplicabilidade na detecção de anomalias de rede. Neste trabalho propõe-se um novo mecanismo baseado em *wavelets* para a detecção de intrusões de rede, por meio da análise dos descritores do tráfego. O mecanismo de análise proposto é baseado na Transformada *Wavelet* Discreta de Daubechies do sinal formado a partir dos descritores do tráfego, o cálculo de *thresholds* e análise direta dos coeficientes *wavelet* para a indicação de anomalias. Assume-se que um ataque gera uma anomalia (alteração) no padrão de tráfego, perceptível nos coeficientes *wavelet*. O mecanismo de detecção é genérico, para trabalhar com diferentes descritores, e apresenta baixa complexidade computacional, o que potencializa a análise em tempo real. Nos experimentos, o mecanismo demonstrou boa taxa de detecção de ataques, com poucos falsos positivos e baixo custo de processamento.

Palavras-chave: Segurança; Ataques; *Wavelets*.

ABSTRACT

Dissertação de Mestrado
Programa de Pós-Graduação em Informática
Universidade Federal de Santa Maria

WAVELET-BASED ANOMALY DETECTION IN NETWORK TRAFFIC

Author: Tiago Perlin
Advisor: Prof. Dr. Raul Ceretta Nunes
Coadvisor: Prof[a]. Dr[a]. Alice de Jesus Kozakevicius

Attacks on computer networks compromises the security of the system and degrade the performance of the network causing problems to users and organizations. Network-based Intrusion Detection Systems are used to detect attacks or malicious activity by analyzing the network traffic. The anomaly-based detection approach is used for intrusion detection. It is assumed that the presence of traffic anomalies, deviations from standard behavior, is indicative of an attack or malfunction. A major difficulty of an anomaly-based Intrusion Detection System is the construction of the profile due to the complexity of network traffic. Methods derived from Signal Analysis, among which, the Wavelet Transform, have recently demonstrated applicability in detecting anomalies in network. This work proposes a new wavelet-based mechanism to detect network intrusions, through the analysis of descriptors of traffic. The mechanism proposed is based on Discrete Wavelet Transform of signal formed from the traffic descriptors, the calculation of thresholds and direct analysis of wavelet coefficients for detection of anomalies. We assume that an attack generates an anomaly (change) in the traffic pattern, visible in the wavelet coefficients. The detection mechanism is generic, to work with different descriptors, and has low computational complexity, which enhances the real-time analysis. In the experiments, the mechanism demonstrated good detection rate of attacks with few false positives and low processing time.

Keywords: Security; Attacks; Wavelets.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

ACK	*Acknowledgment*
ARX	*AutoRegressive with eXogenous input*
CUSUM	*CUmulative SUM*
DARPA	*Defense Advanced Research Projects Agency*
DoS	*Denial of Service* (ataque de negação de serviço)
FIN	*Finalization*
FTP	*File Transfer Protocol*(Protocolo de Transferência de Arquivo)
DIbW	Detector de Intrusões de rede baseado em Wavelets
EWMA	*Exponentially Weighted Moving Average* (Médias Móveis Exponencialmente Ponderadas)
HTTP	*Hypertext Transfer Protocol* (Protocolo de Transferência de Hipertexto)
ICMP	*Internet Control Message Protocol*
IP	*Internet Protocol* (Protocolo Internet)
LRD	*Long-Range Dependence* (Dependência de Longa Duração)
MAD	*Median absolute deviation*
MIB	*Management Information Base*
MRA	*MultiResolution Analysis* (Análise em Resolução Múltipla)
PDF	*Probability Density Function* (Função Densidade de Probabilidade)
ROC	*Receiver Operatoring Characteristic*
SDI	Sistema de Detecção de Intrusão
SDIH	Sistema de Detecção de Intrusão de *Host*
SDIR	Sistema de Detecção de Intrusão de Rede
SDIR-A	Sistema de Detecção de Intrusão de Rede baseado em Anomalias
SMTP	*Simple Mail Transfer Protocol*
SNMP	*Simple Network Management Protocol*
SYN	*Synchronization*

SRD	*Short-Range Dependence* (Dependência de Curta Duração)
TCP	*Transmission Control Protocol* (Protocolo de Controle de Transmissão)
TW	Transformada Wavelet
TWC	Transformada Wavelet Contínua
TWD	Transformada Wavelet Discreta
TWP	Transformada *Wavelet Packet*
UDP	*User Datagram Protocol* (Protocolo de Datagrama de Usuário)

LISTA DE SÍMBOLOS

$r(\tau)$	Função de autocorrelação - (ACF - *AutoCorrelation Function*)
μ	Média
σ	Desvio Padrão
ϕ	Função escala
ψ	Função *Wavelet*
$\delta_{k,l}$	Função Kronecker delta
λ	Valor do *Threshold*
Δt	Intervalo de amostragem
$L^2(\mathbb{R})$	Espaço vetorial cujos elementos são funções de quadrado integrável
V_j	Subespaço da Análise em Multi-Resolução
W_j	Subespaço *Wavelet*
N	Número de elementos no vetor
H	Vetor de coeficiente do filtro passa-alta (filtro *wavelet*)
G	Vetor de coeficiente do filtro passa-baixa (filtro escala)
D	Suporte dos coeficientes dos filtros *wavelet*
P	Número de momentos nulos $P = D/2$
$\mathbb{N}$	Conjunto dos Números Naturais
$\mathbb{R}$	Conjunto dos Números Reais
$\mathbb{Z}$	Conjunto dos Números Inteiros Relativos
$\mathcal{O}(N)$	Complexidade computacional de ordem linear
$\mathcal{O}(N \log N)$	Complexidade computacional de ordem *loglinear*
$y(t)$	Sinal original
t	Tempo
h_k	Constante do vetor H
g_k	Constante do vetor G
j	Escala (inteiro)

$c_{j,k}$	Coeficiente de aproximação ou escala
c_j	Vetor de coeficientes escala no nível j
$d_{j,k}$	Coeficiente de detalhe ou *wavelet*
d_j	Vetor de coeficientes de detalhe ou *wavelet* no nível j
w	Vetor de todos os coeficientes da Transformada *Wavelet*
$supp()$	Suporte
$Tresh_\lambda$	Operação de corte dos coeficientes (*Threshold*)
C	Constante usada para calcular o valor do *threshold*
$\int$	Integral
$\| \|$	Norma
$\langle , \rangle$	Produto Interno
$\{ \}$	Chaves; Objetos dentro delas são elementos de um Conjunto
$( )$	Parênteses; Objetos dentro deles são elementos de um Vetor
$\subset$	Subconjunto
$\perp$	Perpendicular
$\oplus$	Soma ortogonal
$\bigcup$	União
$\bigcap$	Intersecção
$\lceil x \rceil$	O menor inteiro maior que x
$\lfloor x \rfloor$	O maior inteiro menor que x
$\langle x \rangle_q$	Operador módulo $x \ mod \ q$

SUMÁRIO

1 INTRODUÇÃO

A expansão da Internet aumenta a exposição das redes de computadores à ameaças, como ataques aos sistemas computacionais e a infraestrutura, o acesso indevido às informações dos usuários e abusos de privilégios. Neste cenário de interconexão global de dispositivos computacionais, medidas preventivas e ferramentas de detecção são essenciais para garantir a segurança de todo o ambiente computacional pessoal e empresarial.

Considerando-se a pesquisa em segurança da informação e dos sistemas computacionais, esta dissertação trata especificamente da detecção de anomalias no tráfego de rede e propõe um mecanismo para análise do tráfego de rede em tempo real.

1.1 Contexto e Motivação

Medidas preventivas devem ser incluídas prioritariamente em qualquer plano para garantir a segurança de um sistema. Estas medidas são constituídas principalmente por (KIZZA, 2005): controles de acessos físico e lógico, ferramentas de *software* como *firewalls*, dispositivos de *hardware* e configurações. Concomitantemente ao desenvolvimento e implementação de medidas preventivas, atacantes têm explorado vulnerabilidades (principalmente de *software*, mas também de *hardware* e protocolos) e brechas na configuração de sistemas, para obter acesso e efetivar os ataques. As medidas preventivas, assim, apesar de essenciais, possuem limitações e muitas vezes são contornáveis pelos atacantes.

Sistemas de Detecção de Intrusão (SDI) (NORTHCUTT; NOVAK, 2002) são ferramentas que visam melhorar a segurança em um sistema computacional. Detecção de Intrusão são técnicas usadas para detectar ataques ou perturbações à um sistema computacional ou rede de computadores (KIZZA, 2005). O SDI usa as informações coletadas do sistema monitorado (computador, rede ou segmento de rede) para detectar intrusões. Enquanto as medidas de prevenção ativamente buscam evitar que ataques aconteçam, os sistemas de detecção procura identificar ataques pela análise passiva do tráfego da rede ou

os *logs* do sistema. Após a detecção de uma ataque, um SDI deve gerar uma resposta, que pode ser uma intervenção automatizada no sistema ou um alerta para intervenção humana.

Especificamente para redes de computadores, tem-se os Sistemas de Detecção de Intrusão de Rede (SDIR) (NORTHCUTT; NOVAK, 2002) que usam informações coletadas em uma rede ou segmento de rede para identificar ataques tenham ocorrido ou estejam ocorrendo. Para a análise dos dados coletados da rede, os SDIR usam, principalmente a abordagem baseada em assinaturas e a abordagem baseada em anomalias. Ambas abordagens apresentam suas peculiaridades e limitações. A abordagem baseada em assinaturas requer um conhecimento prévio a respeito da forma como cada ataque a uma rede ocorre, ou seja, sua assinatura. Por isso, são menos eficientes na identificação de ataques que usam técnicas ainda desconhecidas. Já a abordagem baseada na detecção de anomalias, que procura detectar alterações no padrão do tráfego em relação ao perfil da rede, pode gerar um excesso de falsos positivos, dificultando a intervenção automatizada ou acarretando a geração de muitos falsos alertas para a intervenção humana. A pesquisa na área de Detecção de Intrusão Rede, entre outras coisas, busca tratar destes problemas.

O constante aumento do volume de dados trafegados nas redes de computadores, gerado pela inclusão de novos computadores e dispositivos e pelo desenvolvimento de novas aplicações baseadas na Internet, dificulta a coleta de dados por um SDIR. Da mesma forma a complexidade das redes, o volume elevado de tráfego, bem como, características intrínsecas do tráfego de rede (STOEV et al., 2005) (SCHERRER et al., 2007) dificultam a análise e consequente Detecção de Intrusões. Neste contexto, métodos de análise de dados de rede que possam trabalhar em tempo real (*online*) são desejáveis.

Este trabalho é motivado pela necessidade de detecção correta e em tempo hábil de anomalias de rede. Como forma de complementar outros mecanismos de segurança, a detecção de anomalias de rede em tempo real é importante para que o administrador seja notificado e possa providenciar os ajustes necessários no sistema para mitigar possíveis ataques.

A detecção de anomalias de rede é uma área de pesquisa bastante ativa, com alguns trabalhos recentes (FARRAPOSO, 2009) (BOLZONI, 2009). Na pesquisa e desenvolvimento de um SDIR baseado em anomalia um dos pontos essenciais é a construção de um perfil da rede. A construção do perfil da rede depende do método de análise usado e implica no conhecimento das características específicas do tráfego de rede. Neste sentido, há diversos métodos de detecção de anomalias no tráfego de rede, como: métodos baseados em análise estatística (SAMAAN; KARMOUCH, 2008) e estatística *bayesiana* (LIU et al., 2008); métodos de mineração de dados, como algoritmos de agrupamento

(LI; LEE, 2003) e lógica *fuzzy* (YAO; ZHITANG; SHUYU, 2006); métodos de inteligência artificial, como sistemas imunológicos artificiais (GUANGMIN, 2008) e algoritmos genéticos (SELVAKANI; RAJESH, 2007); e métodos baseados na análise de sinais (BARFORD et al., 2002) (THOTTAN; JI, 2003). No contexto dos métodos baseados na análise de sinais, a Transformada *Wavelet* (NIELSEN, 1998) mostra-se adequada para a modelagem do tráfego de rede em alguns trabalhos (BARFORD et al., 2002) (SOULE; SALAMATIAN; TAFT, 2005) (HUANG; THAREJA; SHIN, 2006) (GAO et al., 2006) (LU; TAVALLAEE; GHORBANI, 2008) (KIM; REDDY, 2008).

1.2 Objetivos e contribuições

Na Detecção de Anomalias de Rede, o método de análise é de vital importância, pois influencia diretamente no desempenho e eficiência do detector. A abordagem em tempo real, ainda, apresenta alguns desafios, por precisar de resposta a um determinado evento suspeito em tempo reduzido. O tempo de resposta (tempo de reação) reduzido visa minimizar o impacto causado pela possível Intrusão. Quanto menor o tempo de resposta, no entanto, menos informações sobre as consequências do evento são coletadas. O método de detecção precisa ser computacionalmente eficiente para permitir tempos de resposta reduzidos.

Este trabalho propõe um mecanismo de análise do tráfego de rede para a detecção de Intrusões. Por meio da análise dos descritores do tráfego de rede, busca-se encontrar anomalias de tráfego, considerando-se anomalias como possíveis Intrusões. O mecanismo de análise proposto é baseado na Transformada *Wavelet* discreta do sinal formado a partir dos descritores do tráfego padrão de rede e a definição de *thresholds* para a indicação de anomalias, assumindo-se que um ataque ou intrusão gera uma anomalia (alteração) no padrão de tráfego, perceptível nos coeficientes da Transformada *Wavelet*. O mecanismo possui baixa complexidade computacional, permitindo a utilização em análises em tempo real, e é genérica para trabalhar com diferentes variáveis descritivas do tráfego de rede.

Este trabalho busca contribuir para a detecção de anomalias de rede ao propor um novo mecanismo de detecção de anomalias de rede em tempo real baseada na transformada *wavelet* discreta; demonstrar os requisitos e desafios na implementação da abordagem de detecção de anomalias de rede; propor uma arquitetura adaptada para o uso em detecção de anomalias de rede, considerando sequência de amostras de descritores de rede; e por fim demonstrar que a abordagem proposta pode ser empregada em Detectores de Intrusão de tempo real.

1.3 Escopo e organização do texto

Na detecção de anomalias de rede são importantes a escolha do conjunto de variáveis de observação e o método de análise. Normalmente do método depende a eficiência do detector, enquanto que da escolha das variáveis depende o sucesso na detecção de classe específicas de ataques. Este trabalho tem como foco principal o método de análise no qual a questão do desempenho computacional recebe atenção especial. Foge do escopo do trabalho a investigação de quais variáveis são adequadas para a detecção de classes de ataques específicos. Neste trabalho as variáveis são escolhidas e extraídas diretamente do tráfego de rede de forma semelhante ao trabalho em (DAINOTTI; PESCAPE; VENTRE, 2006).

Dentre os métodos de detecção presentes na literatura, os métodos derivados da Análise de Sinais, como a Transformada *Wavelet*, possuem melhor desempenho computacional comparando-se com os métodos baseados em conhecimento e aprendizagem de máquina (GARCíA-TEODORO et al., 2009). Trata-se especificamente, neste trabalho, da detecção de anomalias de rede usando a Transformada *Wavelet* Discreta.

O segundo capítulo apresenta uma revisão bibliográfica sobre os Sistemas Detectores de Intrusões de Rede. Trabalhos anteriores e relacionados são discutidos neste capítulo.

O terceiro capítulo oferece uma descrição da teoria matemática referente a Transformada *Wavelet* Discreta e a algumas técnicas de truncamento da série de dados. A Transformada *Wavelet* Discreta é usada, neste trabalho, no mecanismo de análise dos dados de rede para a detecção de anomalias.

O quarto capítulo apresenta o mecanismo de detecção de anomalias de rede proposto.

O quinto capítulo apresenta as ferramentas usadas no desenvolvimento do protótipo e a sua implementação.

O sexto capítulo descreve o ambiente usado nos experimentos e os resultados alcançados na pesquisa.

Por fim, o sétimo e último capítulo apresenta as conclusões e sugere trabalhos futuros.

2 DETECÇÃO DE INTRUSÃO EM REDES DE COMPUTADORES

A detecção de intrusão é uma área de pesquisa em expansão na segurança em redes de computadores. Com o grande crescimento da interconexão de computadores em todo o mundo, é verificado um consequente aumento nos tipos e no número de ataques a esses sistemas, gerando uma complexidade muito elevada para a capacidade dos tradicionais mecanismos de prevenção. Para a maioria das aplicações atuais, é praticamente inviável a simples utilização de mecanismos que diminuam a probabilidade de eventuais ataques. Um ataque, em casos extremos, pode causar a interrupção total de um serviço ou deixá-lo extremamente lento. O processo de auditoria e posterior restauração manual, normalmente, é lento e oneroso. Isso justifica o estudo e desenvolvimento de mecanismos mais eficientes que a simples prevenção. Neste capítulo é apresentada uma revisão bibliográfica acerca da detecção de intrusão em redes de computadores abordando sua classificação e sistemas mais usados.

Neste Capítulo é feita uma revisão bibliográfica a cerca dos sistemas e abordagens de detecção de intrusão usados em redes de computadores.

Na Seção 2.1 serão descritas algumas características do tráfego de rede padrão e a sua implicação na detecção de anomalias.

Na Seção 2.2 serão descritos os Sistemas Detectores de Intrusão, apresentada uma classificação quanto a localização e abordagem e discutido algumas vantagens e desvantagens de cada abordagem.

Na Seção 2.3 serão descritas algumas técnicas usadas na detecção de intrusões de rede baseadas em anomalias, organizando-as conforme a forma de coleta de dados, manipulação e identificação de anomalias.

Na Seção 2.4 serão apresentados alguns trabalhos na área que usaram *wavelets* na detecção de anomalias.

Por fim na Seção 2.5 serão discutidos os pontos principais do Capítulo.

2.1 Características do Tráfego de rede

O tráfego de rede corresponde a sequência de mensagens (pacotes) trocados entre diferentes dispositivos de rede. Padrão ou anômalo, o tráfego de rede é naturalmente irregular, variando sua intensidade e forma durante decorrer do tempo. A irregularidade do tráfego de rede dificulta a análise e detecção de anomalias. Como característica, o tráfego de rede possui complexas correlações temporais, caracterizadas por Dependências de Curta Duração (SRD - Short-Range Dependence), Dependências de Longa Duração ou longo alcance (LRD - Long-Range Dependence) e auto-similaridade (LELAND et al., 1994).

Inicialmente tratado por (LELAND et al., 1994), e mais recentemente por vários outros (SCHERRER et al., 2007) (BORGNAT et al., 2008), o tráfego de rede foi explicado com dependências de longa duração (LRD) e auto-similaridade. Para o entendimento da LRD é necessário a introdução da Função de Autocorrelação (ACF - *Autocorrelation Function*) (WEISSTEIN, 2010). A Função de Autocorrelação $r(\tau)$, para uma série (vetor) y, é definida como:

$$r(\tau) = \frac{\sum_{t=\tau+1}^{n}(y[t]-\mu)(y[t-\tau]-\mu)}{\sum_{t=1}^{n}(y[t]-\mu)^2} \ , \tag{2.1}$$

sendo τ o atraso em relação ao tempo e μ, o valor médio da série:

$$\mu = \frac{1}{n}\sum_{t=1}^{n} y[t] \ . \tag{2.2}$$

A Função de Autocorrelação mede a relação que um elemento da série tem com outro. Normalmente a função é avaliada para diferentes valores de τ e seu valor está compreendido no intervalo $[-1, 1]$. Um valor negativo ($r(\tau) < 0$) para a Função de Autocorrelação implica em uma relação inversa, um valor positivo ($r(\tau) > 0$) implica em uma relação direta e quando a função é igual a zero ($r(\tau) = 0$) tem-se relação nula ou independência estatística. A LRD significa que pontos distantes da série possuem alto grau de correlação. Matematicamente, a LRD é expressada como (LELAND et al., 1994):

$$r(\tau) \approx \tau^{-\beta} \text{ quando } \tau \to \infty, 0 < \beta < 1, \sum r(\tau) \to \infty \ , \tag{2.3}$$

sendo τ o atraso em relação ao tempo. A Expressão (2.3) significa que a Função de Autocorrelação $r(\tau)$ decai hiperbolicamente, ao invés de exponencialmente, conforme τ aumenta, implicado que a função não é somável (dependência de longa duração, LRD). Por outro lado, na dependência de curta duração (SRD):

$$r(\tau) \approx \rho^{\tau} \text{ quando } \tau \to \infty, 0 < \rho < 1, \sum r(\tau) < \infty \ , \tag{2.4}$$

a Função de Autocorrelação $r(\tau)$ decresce exponencialmente, ou seja, a função é somável.

A Função de Autocorrelação (Função (2.1)) numa série temporal SRD decai exponencialmente (Expressão (2.4)) e numa uma série temporal LRD decai hiperbolicamente (Expressão (2.3)) (LELAND et al., 1994). A presença de LRD em uma série temporal de tráfego de rede, dificulta a análise estatística, pois pontos distantes no tempo estão fortemente correlacionados, sendo necessário muitos pontos para a análise. Quando poucos pontos estão disponíveis o erro é grande (FARRAPOSO, 2009).

O tráfego de rede exibe uma mistura de SRD e LRD, ou seja, a ACF comporta-se semelhante a um processo LRD em escalas longas de tempo e como SRD em escalas pequenas de tempo (SCHERRER et al., 2007). Outras características do tráfego de rede, que estão relacionadas com a presença de LRD, são auto-similaridade (*self similarity*) e a distribuição de probabilidade não normal (não Gaussiana) dos valores das amostragens.

Embora a LRD e a Auto-similaridade sejam conceitos matemáticos distintos e nem sempre relacionados, no contexto de análise de tráfego de rede, ambos são tratados como relacionados e muitas vezes de forma indiferente. Uma forma de definir a Auto-similaridade é como (LELAND et al., 1994):

$$r^{(m)}(\tau) \approx r(\tau) \text{ quando } m \to \infty \ , \tag{2.5}$$

sendo (m) o nível de agregação (escala de tempo) da série. A Expressão (2.5) significa que os valores da Função de Autocorrelação (Função (2.1)) $r^{(m)}(\tau)$ tendem a se manterem inalterados, independente do nível de agregação da série (m). A propriedade da Auto-similaridade do tráfego de rede é melhor explicada graficamente, conforme a Figura 2.1.

Na Figura 2.1, em (LELAND et al., 1994), é apresentada uma prova gráfica da existência de auto-similaridade do tráfego de rede em diferentes níveis de agregação. Considerando-se diferentes níveis de agregação (escala de tempo), $10\,s$, $1\,s$, $0,1\,s$ e $0,01$, o tráfego de rede (pacotes por unidade de tempo), na coluna da esquerda, é semelhante em sua forma em todos os níveis de agregação, não alterando a sua variabilidade (presença de picos). Em contraste o tráfego gerado sinteticamente pelo modelo *Poisson*, na coluna da direita, torna-se mais suave em níveis de agregação mais largos.

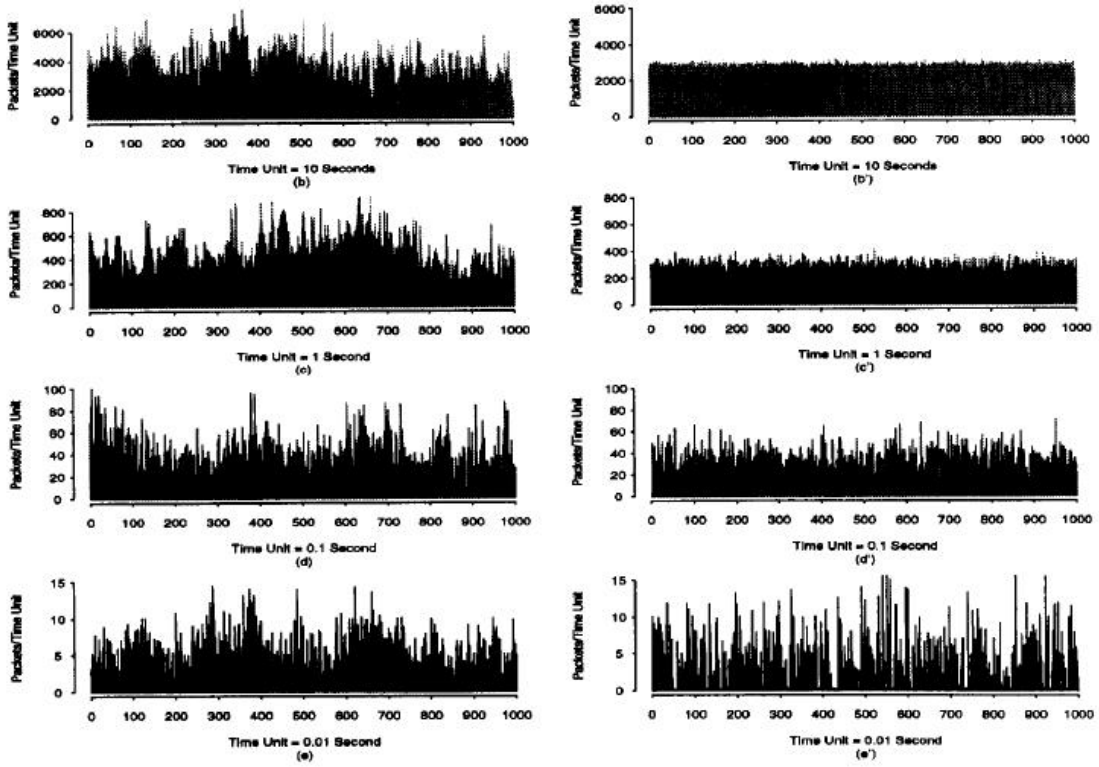

Figura 2.1: Prova gráfica da auto-similaridade do tráfego de rede. O tráfego de rede (pacotes por unidade de tempo), coluna da esquerda, possui forma semelhante independente do nível de agregação (escala de tempo), em contraste com o tráfego sintético, da direita. Fonte: (LELAND et al., 1994).

Embora a LRD e a Auto-similaridade sejam os dois maiores conceitos nos estudos do tráfego de rede, ambos são difíceis de se obter e trabalhar diretamente. O tráfego de rede não é estacionário, o que significa que a sua forma ou estrutura varia conforme o tempo (FARRAPOSO, 2009). A caracterização do tráfego de rede padrão é uma tarefa difícil.

2.2 Sistemas Detectores de Intrusão

Detecção de Intrusão são técnicas usadas para detecção de ataques e intrusões a um computador ou rede de computadores. A partir do trabalho inicial de (DENNING, 1987), que propôs um Sistema de Detecção de Intrusão (SDI), vários outros sistemas foram criados e diversos métodos de detecção foram desenvolvidos. Um SDI (KIZZA, 2005) (NORTHCUTT; NOVAK, 2002) é uma ferramenta usada para detectar intrusões e ataques a um sistema computacional. O SDI usa para a detecção as amostras coletadas do sistema monitorado e, ao mesmo tempo, um método de análise é usado para identificar intrusões.

Definição 2.2.1. Uma Intrusão é qualquer tentativa ilegal e deliberada, bem sucedida ou

não, de manipulação, quebra ou perturbação do funcionamento de um sistema (KIZZA, 2005).

Algumas vezes são feitas distinções entre os termos “Intrusão” e “Ataque”. Enquanto Ataque refere-se à tentativa de perturbação de um sistema, Intrusão representa um ataque que bem sucedido. Para este trabalho o termo “Ataque” é mais adequado, de acordo com a definição, porém algumas vezes ambos os termos são usados sem distinção.

O processo de detecção de ataques, realizado por um SDI, compreende normalmente três atividades fundamentais (NORTHCUTT; NOVAK, 2002): Coleta, Análise e Resposta.

A Coleta corresponde a obtenção dos dados do sistema monitorado. A Coleta de informações pode ser feita diretamente ou por meio de uma ferramenta de *software* ou *hardware* chamado Coletor. A Fonte de Informação costuma ser um computador, uma rede ou um segmento de rede.

A Análise consiste no processamento dos dados coletados procurando identificar a ocorrência de uma Intrusão. Há duas abordagens principais diferentes para a análise dos dados, a abordagem baseada em assinaturas e a abordagem baseada em anomalias, com diferentes métodos em cada abordagem, como serão vistas adiante.

A Resposta é o conjunto de ações que o SDI realiza quando detecta uma intrusão. Como ação típica tem-se a geração de Alarmes e relatórios, mas o SDI também pode ser programado para fazer uma intervenção automatizada no sistema em caso de Intrusão.

Na análise dos dados, detecção e geração de alarmes por um SDI podem ocorrer as seguintes situações: Verdadeiro Positivo, Falso Positivo, Verdadeiro Negativo ou Falso Negativo.

Definição 2.2.2. Um Verdadeiro Positivo (VP) é um alarme verdadeiro, gerado como resposta pelo SDI a uma intrusão.

Definição 2.2.3. Um Falso Positivo (FP) é um falso alarme, gerado pelo SDI a um evento não malicioso.

Definição 2.2.4. Um Verdadeiro Negativo (VN) ocorre quando corretamente não é gerado alarme a um evento não malicioso.

Definição 2.2.5. Um Falso Negativo (FN) ocorre quando não é gerado alarme, porém há uma intrusão.

A quantidade de VP, FP, VN e FN é usada na avaliação de desempenho na detecção de intrusões por um SDI, sendo que números reduzidos de FP e FN são considerados desejáveis. Falsos Positivos influenciam mais que Falsos Negativos a experiência do usuário a um SDI. Enquanto que é indesejável que alguns ataques passem despercebidos, por outro lado, um sistema que frequentemente emite falsos alarmes acaba por ser ignorado (BOLZONI, 2009).

O SDI pode ser classificado, conforme a fonte de informação, em duas categorias (NORTHCUTT; NOVAK, 2002): Sistemas Detectores de Intrusão baseados em *Host* (SDIH) e Sistemas Detectores de Intrusão de Rede (SDIR).

Um SDIH coleta e analisa informações relativas a um *host*, como quantidade de recursos (memória, processamento, disco) utilizada, número de processos, variáveis de ambiente, etc. É necessária, para o funcionamento de um SDIH, a instalação do coletor no *host* a ser analisado, o analisador pode estar na mesma máquina ou em outro computador da rede.

Já um SDIR utiliza para análise informações coletadas em uma rede de computadores, como volume de tráfego, número de conexões, fluxos, pacotes perdidos, etc. É preciso ter um coletor acoplado a uma rede, capturando os pacotes que passarem por ela, ou outros equipamentos capazes de coletarem informações de tráfego de rede. A coleta de informações internas de um *host* é dificultada ou não desejável em alguns ambientes, por razões de segurança e privacidade individuais, este trabalho, por isso, preocupa-se em discutir o desenvolvimento de SDIR.

Também, quanto a forma de analisar os dados, há duas abordagens principais que podem ser usadas nos SDI em geral, bem como em um SDIR: uma baseada em conhecimento, ou assinaturas e outra baseada em comportamento, ou anomalias.

2.2.1 Sistemas Detectores de Intrusão baseados em *Host*

Sistemas Detectores de Intrusão baseados em *Host* (SDIH) são ferramentas usadas para detectar atividades maliciosas em um único computador (KIZZA, 2005). Um SDIH é desenvolvido para um único computador e usa um *software* que monitora as atividades do Sistema Operacional e dos programas que rodam sobre o sistema, como acesso a arquivos, chamadas de sistema, e *logs* do sistema. Quando há uma alteração em um arquivo ou parâmetro monitorado, o SDIH compara o evento com as assinaturas de ataques pré-definidas e, caso haja uma correspondência, o SDIH sinaliza o evento como ilegal. O SDIH também pode ser usado para monitorar uma rede ou segmento de rede, embora este uso apresenta alguns problemas, como o fato de só ser possível analisar o tráfego de rede

que passa pelo computador.

A Detecção de Intrusão baseada em *Host* apresenta algumas vantagens (KIZZA, 2005):

- Capacidade de verificar o sucesso ou falha de um ataque rapidamente pela análise de *logs* do evento. Um SDIH possui informações mais precisas sobre um evento e menos propensa a Falsos Positivos. Neste caso o SDIH pode ser usado como complemento de um SDIR para verificação do sistema;
- Monitoração em baixo nível. Pelo fato de monitorar um *host*, um SDIH, é capaz de analisar atividades de baixo nível, como acesso a arquivos, mudanças nas permissões de um arquivo, execução de arquivos e tentativas de mudanças de privilégios. Muitos ataques são tão discretos que apenas um SDIH é capaz de detectar;
- Detecção quase em tempo real. O SDIH tem a capacidade de detectar eventos no *host* rapidamente e alertar o administrador;
- Capacidade de analisar tráfego criptografado. Um SDIH pode acessar as informações antes e após a encriptação;
- Custo reduzido. Não é necessário *hardware* dedicado ou adicional para a instalação de um SDIH.

O grande problema com o uso de SDIH é o processamento extra necessário apenas para para analisar os dados coletados no computador. Em alguns casos esta sobrecarga pode comprometer o desempenho de todo o sistema computacional e inviabilizar a detecção. Os SDIH ainda apresentam outras desvantagens (KIZZA, 2005):

- Visão limitada. Um SDIH possui uma visão limitada da rede;
- Sujeito a fraudes. Pelo fato de estarem mais perto do usuário os SDIH são mais sujeitos a fraudes.

2.2.2 Sistemas Detectores de Intrusão de Rede

Os Sistemas Detectores de Intrusão de Rede (SDIR) (KIZZA, 2005) são SDI usados para monitorar toda uma rede, com o objetivo de detectar anomalias, ataques ou ações ilegais. Os SDIR usam para análise informações coletadas de uma rede, como volume de tráfego, número de conexões, fluxos e pacotes perdidos. É preciso ter um coletor acoplado a uma rede, capturando os pacotes que passarem por ela, ou outros equipamentos capazes de coletarem informações de tráfego de rede. Um SDIR é constituído, normalmente, por

alguns subsistemas (KIZZA, 2005): Coletor, Analisador, Banco de Dados, Notificador, Atuador e Monitor.

O Coletor é um *software* que roda em uma máquina dedicada e usa um sensor ligado a uma fonte de informação, como uma rede ou um segmento de rede. O sensor pode estar em equipamento de rede ou computador ligado a rede. Normalmente usa-se algum *hardware* de rede em modo "promíscuo" capturando todos os pacotes que passam pela rede independentemente da origem ou destino. A biblioteca LIBPCAP (TCPDUMP, 1998) em conjunto com uma interface de rede em modo "promíscuo" têm sido amplamente usados (HUANG; THAREJA; SHIN, 2006) (Sá SILVA, 2008). Noutros trabalhos (THOTTAN; JI, 2003) (WU; SHAO, 2005) (ZARPELãO et al., 2009) acessam-se diretamente as informações armazenadas em uma base MIB (*Management Information Base*) (PRESUHN, 2002), acessada via protocolo SNMP (*Simple Network Management Protocol*) (HARRINGTON; PRESUHN; WIJNEN, 2002), em equipamentos de rede que disponibilizam este serviço. O desempenho do Coletor depende dos equipamentos de rede usados para a coleta, principalmente em redes de grande tráfego. Alguns *firewalls* atuam, também, como coletor, armazenando informações para um SDI (NORTHCUTT; NOVAK, 2002, p. 273).

O Analisador verifica os dados coletados buscando por eventos que indiquem uma intrusão ocorrida ou que esteja ocorrendo. Há diferentes abordagens para a análise dos dados, como a baseada em assinaturas e a baseada em anomalias, com vários métodos diferentes, como métodos estatísticos, aprendizagem de máquina e baseados em conhecimento (GARCíA-TEODORO et al., 2009).

O Banco de Dados é o repositório de informações do SDI, onde são guardadas informações sobre o sistema monitorado e os eventos suspeitos. As informações guardadas no Banco de Dados dependem do método de detecção usado e da necessidade de se manter um histórico do sistema.

O sistema Notificador é responsável pelo envio de alertas ao administrador do sistema. A notificação pode ser um alerta na tela de um monitor, um aviso sonoro ou uma mensagem eletrônica. Alertas frequentes, com vários Falsos Positivos, são prejudiciais pois banalizam a detecção e acabam desacreditando a ferramenta. O desempenho de um SDI depende da relação entre Falsos Positivos e Falsos Negativos, então é importante que o sistema possa ser ajustado (KIZZA, 2005).

O Atuador possui a capacidade de executar ações automatizadas conforme a Intrusão detectada. Tipicamente a resposta a um evento intrusivo é a reconfiguração do roteador, alteração de regras no *firewall* ou a desconexão de algum usuário ou serviço.

O Monitor ou Terminal de comando tem o objetivo de ser a ligação entre o administrador e o SDI. O Monitor pode ser usado para configurar o sistema, verificar o funcionamento do SDI e a ocorrência de Alarmes.

As principais vantagens de um SDIR são (KIZZA, 2005):

- Habilidade de detectar ataques que SDIH não consegue porque monitora no nível de transporte da arquitetura de rede. Neste nível, o SDIR pode analisar pacotes não apenas por endereços, mas também por números de porta. O SDIH, que monitora pacotes em baixo nível, pode não ser capaz de detectar alguns tipos de ataque;

- Dificuldade de remover evidências. Geralmente um SDIR está em uma máquina dedicada e protegida, o que dificulta a remoção de evidências por um atacante;

- Detecção e Resposta em tempo real. Porque o SDIR está em pontos estratégicos da rede, ele pode detectar intrusões e, tão rápido quanto possível, notificar o administrador;

- Habilidade de detectar mesmo ataques mal sucedidos. Muitos ataques são parados por *firewalls* ou outros motivos, mesmo assim informações referentes a estes ataques são importantes ao administrador.

O principal desafio no desenvolvimento de um SDIR é escolher um método eficiente que identifique uma intrusão de maneira correta sem gerar um número excessivo de falsas detecções. Os SDIR apresentam algumas desvantagens (KIZZA, 2005):

- Pontos cegos. Normalmente os sensores de um SDIR são colocados nas bordas da rede, com isso, algumas vezes alguns segmentos da rede não são vistos pelo SDIR;

- Informações criptografadas. O SDIR não consegue analisar tráfego de rede criptografado, porém, algumas vezes é possível analisar as informações dos cabeçalhos dos pacotes.

Como exemplos de SDIR mais conhecidos tem-se o Bro[1] (BRO, 2009) e o Snort[2] (SNORT, 2009), ambos disponibilizados como *software* livre. Tanto o Bro quanto o Snort são baseados em assinaturas que por meio de ferramentas são compatíveis entre si. Há ainda *plugins*[3], em ambos os sistemas, para a inclusão da capacidade de detecção baseada em anomalias.

[1] Disponível em: http://www.bro-ids.org/
[2] Disponível em: http://www.snort.org/
[3] *Software* que adiciona alguma funcionalidade a um programa.

2.2.3 Detecção de Intrusões de Rede baseada em assinaturas

Os SDIR baseados em assinaturas (KIZZA, 2005), como o (BRO, 2009) e o (SNORT, 2009), comparam os dados coletados da rede com uma base de dados de assinaturas de ataques conhecidos ou regras pré-definidas e quando os eventos analisados são compatíveis com alguma das assinaturas da base de dados um alarme é disparado. Novas formas de ataques ou variações de ataques conhecidos surgem constantemente, por isso, para o bom funcionamento de um SDIR baseado em assinaturas, é necessário manter a base de assinaturas de ataques atualizada. Porém, mesmo com uma base de assinaturas atualizada, tais SDIR têm dificuldade em detectar ataques desconhecidos, ataques mutantes ou camuflados. Os SDIR baseados em assinaturas são, portanto, bastante precisos em suas detecções, apresentando baixo número de falsos positivos, porém, devido a sua dificuldade em detectar ataques novos, podem apresentar um grande número de falsos negativos, o que pode representar uma brecha de segurança.

Resumidamente, as desvantagens da abordagem de Detecção de Intrusões baseada em assinaturas (KIZZA, 2005):

- O sistema não é capaz de detectar ataques desconhecidos, ou seja, que não possuam uma assinatura arquivada;
- O sistema não é capaz de prever e detectar novos ataques.

2.2.4 Detecção de Intrusões de Rede baseada em anomalias

A Detecção de Intrusão usando a abordagem baseada em Anomalias apoia-se na ideia que um ataque gera um desvio do comportamento padrão do sistema (DENNING, 1987) (KRUEGEL; VIGNA, 2003). Assume-se que a atividade maliciosa difere do comportamento padrão do sistema e que esta diferença pode ser expressada quantitativamente (KRUEGEL; VIGNA, 2003). Os SDIR baseados em anomalias (GARCíA-TEODORO et al., 2009) (KRUEGEL; VIGNA, 2003) (THOTTAN; JI, 2003), SDIR-A, constroem um perfil do comportamento padrão da rede com base em informações do histórico, quando o comportamento observado desvia-se significativamente deste perfil, ou seja, uma anomalia é detectada, um alarme é disparado. Os SDIR-A são conhecidos também como Sistemas Detectores de Anomalias de Rede (PLONKA; BARFORD, 2009).

Definição 2.2.6. Uma Anomalia é um evento que causa um desvio (alteração) em relação ao perfil (padrão) do sistema.

Assume-se que uma Anomalia é indicativo de um ataque. De um modo amplo, uma Anomalia de rede pode ocorrer devido a um Ataque, falha de equipamento, problemas de configuração, sobrecarga ou uso abusivo ou inadequado de algum serviço ou recurso da rede. Embora o foco principal de um SDIR seja a detecção de Ataques, no caso de um SDIR baseado em anomalias, a possibilidade de detecção de outras anomalias de rede é interessante. A Detecção de Anomalias é a tarefa de determinar o que é normal e esperado para um sistema e encontrar ou diferenciar as anomalias.

Pelo fato de buscar por comportamentos anômalos, um SDIR baseado em anomalias é capaz de detectar ataques sem seu conhecimento prévio, sendo uma alternativa a abordagem baseada em assinaturas. O tráfego de rede, de modo geral, apresenta como característica alta variabilidade, dificultando a construção de um perfil para a rede e a definição de intervalos confiáveis de variação. Em algumas situações, mudanças do padrão de tráfego de uma rede podem ser erroneamente identificadas, pelo SDIR, como indício de um ataque ou falha, gerando um falso alarme. Os SDIR baseado em anomalias são capazes de detectar ataques desconhecidos, no entanto, uma das limitações ainda é a ocorrência de um grande número de falsos positivos.

Uma das dificuldades de SDIR baseados em anomalias está em construir um perfil da rede devido a algumas características específicas do tráfego de rede. As características do tráfego de rede, de modo geral, foram estudadas em alguns trabalhos (ROHANI et al., 2008) (STOEV et al., 2005) (SCHERRER et al., 2007) que apontam que algumas variáveis descritivas, como número de pacotes ou tamanho dos arquivos transmitidos, apresentam distribuição de probabilidade com cauda pesada, ou seja com decaimento mais lento que a distribuição normal. Distribuição de probabilidade de cauda longa nas variáveis do tráfego de rede normalmente são devido principalmente a dependência de longa duração (LRD). A LRD, em uma variável, significa que a função de auto-correlação decai lentamente. A auto-similaridade ou característica fractal está relacionada a dependência de longa duração e refere-se à característica de uma variável em possuir a mesma distribuição de probabilidade em qualquer nível de agregação ou resolução. O tráfego de rede é muito variável, sendo constituído basicamente por picos, e devido as características de dependências de longa duração, auto-similaridade e distribuição de probabilidade com cauda pesada, é estatisticamente difícil identificar valores extremos e definir intervalos de confiança.

Resumidamente, as desvantagens da abordagem de Detecção de Intrusões baseada em anomalias (KIZZA, 2005):

- **Falsos Positivos**: Muitas atividades anômalas, porém não intrusivas, são equivocadamente sinalizadas como Intrusões;
- **Falsos Negativos**: Intrusões podem não ser detectadas, caso não produzam alguma anomalia perceptível;
- São computacionalmente complexos, pela necessidade de criação e atualização de um perfil.

Abordagens de detecção por assinaturas podem ser adequadas para casos distintos de formas de ataques, enquanto que a abordagem baseada por anomalias é mais indicada para a detecção de ataques desconhecidos. Levando-se em conta a grande variedade de ataques existentes e o rápido surgimento de novos ataques, é possível o uso de um SDI híbrido que incorpore os dois métodos, unindo as vantagens de ambos.

Alguns projetos de SDIR baseados em anomalias conhecidos são: o EMERALD (*Event Monitoring Enabling Responses to Anomalous Live Disturbances*)[4], o Prelude IDS[5], o POLVO-IIDS (Sistema de Detecção de Intrusão Inteligente Baseado em Anomalias) (MAFRA et al., 2008). A maioria usa algum método de aprendizagem de máquina (GARCíA-TEODORO et al., 2009).

2.3 Detecção de anomalias de rede

A detecção de anomalias em redes de computadores é uma área de estudo bastante ativa e várias técnicas são usadas. A classificação das técnicas de detecção de anomalias de rede, presentes na literatura, é uma tarefa difícil devido a diversidade e ao desenvolvimento constante de novas técnicas. Em (GARCíA-TEODORO et al., 2009), o autor classificou os métodos de detecção de anomalias de rede em métodos baseados: Conhecimento, Aprendizagem de Máquina e Análise Estatística. Neste trabalho usa-se a classificação conforme:

- **Conhecimento**: (THOTTAN; JI, 2003) Máquina de estados finitos; Sistemas especialistas ou baseado em regras; Busca por padrões (*Pattern Matching*);
- **Aprendizagem de Máquina**: Redes *bayesianas* (LIU et al., 2008); Cadeias de Markov (GARCíA-TEODORO et al., 2009); Redes Neurais (MAFRA et al., 2008); Lógica difusa (*Fuzzy*) (YAO; ZHITANG; SHUYU, 2006); Algorítimos genéticos

[4]http://www.csl.sri.com/projects/emerald/
[5]http://www.prelude-technologies.com/en/welcome/index.html

(SELVAKANI; RAJESH, 2007); Algorítimos de agrupamento (*Clustering*) (LI; FANG, 2007); Sistemas imunológicos artificiais (GUANGMIN, 2008);

- **Análise de Sinais**: Análise Estatística (SAMAAN; KARMOUCH, 2008) filtros de Kalman (SOULE; SALAMATIAN; TAFT, 2005); CUSUM (*CUmulative SUM*) (THOTTAN; JI, 2003); Séries Temporais (WU; SHAO, 2005); *Wavelets* (HUANG; THAREJA; SHIN, 2006);

Em relação a classificação adotada em (GARCíA-TEODORO et al., 2009), neste trabalho acrescentou-se, na classificação dos métodos de detecção, as técnicas derivadas Análise de Sinais, separando-se algumas das técnicas de Análise Estatística. Na Análise de Sinais são usadas técnicas mais elaboradas para a modelagem dos dados e criação de um perfil que as baseadas na Análise Estatística básica.

Os métodos baseados em Conhecimento, ou baseados em regras, fazem uso de um conjunto de regras e parâmetros elaborados e classificados por um especialista, usando algum formalismo, como máquina de estados finitos por exemplo. Tais métodos são muito robustos, apresentando poucos falsos positivos, e flexíveis. A principal desvantagem, no entanto, está na dificuldade e demora em se obter o conhecimento de qualidade necessário (GARCíA-TEODORO et al., 2009).

A abordagem de Aprendizagem de Máquina baseia-se no estabelecimento de um modelo implícito ou explícito que permite que padrões sejam analisados e classificados. São usadas diversas técnicas, como Redes Neurais e Algorítimos de agrupamento, com diferentes propriedades. Contudo, a principal característica da abordagem está na necessidade de uma fase de treinamento com dados rotulados para a diferenciação do comportamento aceitável do não aceitável pelo sistema. As principais vantagens destes métodos estão na flexibilidade, adaptabilidade e capacidade de capturar interdependências desconhecidas nos dados. Porém esta abordagem depende da determinação (rotulagem) do comportamento aceitável pelo sistema e os métodos empregados demandam muito de recursos computacionais (GARCíA-TEODORO et al., 2009).

Métodos derivados da Análise de Sinais têm sido propostos para a detecção de anomalias de rede (BARFORD et al., 2002). Nos métodos baseados na Análise de Sinais, um perfil é criado representando o comportamento passado da rede. O perfil usando métricas de tráfego, como número de pacotes por protocolo, número de conexões e outras. Um alerta de anomalia é disparado quando o comportamento atual da rede difere significativamente do encontrado no perfil, ultrapassando algum limite (*threshold*) estabelecido. A principal vantagem destes métodos está em não precisar de algum conhecimento predefi-

nido do comportamento padrão da rede, pois são capazes de se adaptar ao comportamento da rede. A principal dificuldade, no entanto, está na definição dos parâmetros, o que influencia na taxa de detecções e de falsos positivos.

Tendo como a vantagem não necessitar de conhecimento predefinido ou de uma etapa de treinamento, as abordagens baseadas na Análise de Sinais tornam-se interessantes para uso na detecção de anomalias devido a variabilidade do tráfego de rede. Neste sentido, a Transformada *Wavelet*, método de Análise de Sinais, demonstrou aplicabilidade para a análise do tráfego e detecção de anomalias de rede (BARFORD et al., 2002) (THOTTAN; JI, 2003) (HUANG; THAREJA; SHIN, 2006) (GAO et al., 2006) (LU; TAVALLAEE; GHORBANI, 2008) (KIM; REDDY, 2008) por permitir a análise em diferentes escalas de tempo (DONOHO; JOHNSTONE, 1995). A maioria dos métodos baseados na Análise de Sinais para detecção de anomalias de rede, presentes na literatura, apresenta ao menos três etapas diferentes: Seleção de Variáveis, Transformação dos dados e Geração de Alarmes.

2.3.1 Seleção de variáveis

A detecção de anomalias é uma atividade complexa. A seleção do conjunto de variáveis usadas pelo processo de análise de dados influencia na capacidade de detecção do SDI e o número de variáveis usadas impacta no desempenho computacional da ferramenta. No entanto, a seleção de variáveis normalmente é guiada por critérios empíricos (ABDOLLAH et al., 2008). As variáveis selecionadas dependem também do tipo de SDI usado e dos tipos de ataques ou anomalias de interesse, por exemplo para um SDIR normalmente se está interessado nos endereços de origem e destino, portas e protocolos dos pacotes de rede. Quanto aos dados coletados em uma rede, um SDIR pode utilizar os dados do *payload* do pacote, como em (KRUEGEL; VIGNA, 2003), ou apenas as informações do *header*, como em (LONGCHUPOLE; MANEERAT; VARAKULSIRIPUNTH, 2009) e (KIM; REDDY, 2008).

A seleção de variáveis consiste na escolha das características (ou descritores) de rede a serem utilizadas para a análise. Normalmente faz-se a distinção entre as características referentes a uma única conexão TCP daquelas referentes a múltiplas conexões. Conforme (ONUT; GHORBANI, 2007) as características do tráfego de rede são classificadas como básicas e derivadas:

- Características Básicas: são características que representam a uma única conexão TCP/IP. Estas características são extraídas diretamente dos pacotes de tráfego de rede. Diferentes nomes também são usados para nomear estas características, como:

Características Básicas; Atributos Essenciais; Características Básicas de uma conexão TCP; Características TCP Básicas. Ainda pode incluir as Características de Fluxo, que engloba também os protocolos não orientados a conexão (exemplo: UDP, ICMP).

- Características Derivadas: representam múltiplas conexões TCP/IP ao mesmo tempo. Também são conhecidas como Características de Tráfego.

Ainda segundo (ONUT; GHORBANI, 2007), as Características Derivadas destinam-se a encontrar similaridades entre diferentes conexões de rede. Para a coleta dessas características podem ser usadas dois tipos de janelas de observação. O primeiro tipo é baseado em uma janela com intervalo de tempo (por exemplo, 5 segundos), enquanto que no segundo tipo é usada uma janela com intervalo de conexões (por exemplo, as últimas 100 conexões). O uso desses dois tipos diferentes de janelas separa as Características Derivadas em: Características baseados no Tempo e Características baseadas em Conexões:

- Baseadas no tempo: são computadas com respeito a um determinado intervalo de tempo passado. Esse tipo de características são boas para a detecção de ataques que geram anomalias de volume de tráfego como ataques do tipo DDoS.

- Baseadas em conexão: são computadas considerando-se o número de conexões passadas. Essas características são usadas apenas com protocolos de rede orientados a conexão, como TCP, e são boas na detecção de ataques que aconteçam em um grande intervalo de tempo.

Devido a diversidade protocolos e serviços de rede existentes, a quantidade de características possíveis é imensa. Embora seja possível no desenvolvimento de um SDI considerar um número grande de características de rede para a detecção de anomalias, tem-se restrições de desempenho computacional. Portanto as características de rede são escolhidas conforme a necessidade do SDI.

Alguns trabalhos recentes (ZAMAN; KARRAY, 2009) (GHALI, 2009) (CHOU; YEN; LUO, 2008) demonstram preocupação com a escolha das características de rede por um SDI e buscam por formas automatizadas de seleção. Em (ZAMAN; KARRAY, 2009), o autor usou uma técnica de aprendizagem de máquina conhecida como *Support Vector Machines*(SVM) para a classificação e seleção das características de rede. Em (GHALI, 2009) propôs um algorítimo baseado em Rede Neural para a seleção de variáveis. Já em (CHOU; YEN; LUO, 2008), o autor propôs um algorítimo baseado em métodos de

agrupamento (*clustering*), mais especificamente *k-nearest neighbor* (k-NN) e lógica difusa (*fuzzy*). Todos os trabalhos citados fizeram uso da base de dados do DARPA KDD 99 (HETTICH; BAY, 1999) e demonstraram uma redução do número de características consideradas importantes para detecção de ataques em um SDI.

Uma variável (contador) armazena uma amostragem de determinada característica de rede. O conjunto de amostragens, ordenadas no tempo, de uma variável forma uma série temporal, que é usada pela maioria dos métodos baseados na análise de sinais. Neste caso fala-se especificamente das características de rede baseadas no tempo. Neste texto ainda, faz-se uma diferenciação entre Variáveis Primárias e Variáveis Derivadas.

As Variáveis Primárias relacionam-se a características extraídas diretamente dos pacotes TCP/IP computadas conforme o intervalo de tempo pré-determinado, como por exemplo: número de pacotes trafegados; tamanho médio dos pacotes; quantidade em *bytes* de dados trafegados; número de pacotes referentes a determinado protocolo, como TCP, UDP ou ICMP; número de pacotes por porta ou serviço. Já, as Variáveis Derivadas são composições ou relações de duas ou mais Variáveis Primárias, como por exemplo: a diferença entre pacotes SYN e FIN; ou a relação entre diferente portas ou serviços.

2.3.2 Transformação dos dados

A transformação dos dados consiste na representação matemática das séries de dados de rede, de modo a remover tendências e tornar evidente as singularidades. Na Transformada *Wavelet*, as séries de dados, no domínio do tempo, são representados no domínio do tempo e escala (DONOHO; JOHNSTONE, 1995). Algumas abordagens utilizam apenas a Transformada *Wavelet*, outras a utilizam em conjunto com outros modelos matemáticos (LU; TAVALLAEE; GHORBANI, 2008).

Por ser o foco principal deste trabalho, os algoritmos baseados em *wavelets* para a análise dos dados presentes na literatura são tratados juntamente com os trabalhos relacionados na Seção 4.3.

2.3.3 Geração de alarmes

Para que a detecção de anomalias ocorra é necessário a geração de alarmes, ou qualquer outra forma de aviso ou intervenção automatizada, toda vez que as medidas estatísticas dos dados mais recentes afastam-se consideravelmente de um modelo de tráfego padrão, construído com base no histórico da rede. Normalmente esta análise é realizada sobre os dados transformados ou resíduos (LU; TAVALLAEE; GHORBANI, 2008) e várias métricas estatísticas podem ser utilizadas, como média ou variância (GAO et al.,

2006). Para acomodar variações insignificantes, devido a algum componente estocástico do modelo, são definidos valores de *threshold*, que podem ser fixos (GAO et al., 2006) ou dinâmicos (KIM; REDDY, 2008).

2.4 Trabalhos relacionados

O estudo realizado neste trabalho foi motivado por um conjunto de trabalhos nos quais *wavelets* foram usadas em alguma das fases da modelagem do detector de anomalias.

No trabalho de (DAINOTTI; PESCAPE; VENTRE, 2006) foi proposto um mecanismo de detecção de anomalias de volume de tráfego de rede com o objetivo de detectar ataques do tipo DoS. O sistema combina uma abordagem tradicional, baseado em Somas Cumulativas (CUSUM - *CUmulative SUM*) (BASSEVILLE; NIKIFOROV, 1993, p. 35) e Médias Móveis Exponencialmente Ponderadas (EWMA - *Exponentially Weighted Moving Average*) com uma nova abordagem baseada na Transformada *Wavelet* Continua (TWC) e *Threshold*. A arquitetura é baseada em dois estágios. O primeiro estágio é usa EWMA e *Thresholds* e destina-se a a fazer a detecção "grosseira" de ataques. O segundo estágio, usa a TWC, destina-se a refinação e detecção "fina" dos ataques, para diminuir o número de falsos alertas. A *Wavelet* Mãe usada foi a *Morlet*.

No trabalho de (GAO et al., 2006) foi proposto um detector de anomalias de rede baseado na Transformada *Wavelet Packet* (TWP) (COIFMAN; WICKERHAUSER, 1992). Os dados de rede são transformados utilizando-se a transformada direta *wavelet packet*, com bases *wavelet* da família Daubechies, e reconstruído a partir dos coeficientes *wavelet* para cada nível da transformada. Medidas estatísticas, como média e variância, foram usadas para caracterizar uma anomalia, como a razão da média ou da variância entre a janela de detecção e a janela histórica foram mensuradas e comparadas com valores de *threshold* predefinidos para identificar uma anomalia.

No trabalho de (LU; TAVALLAEE; GHORBANI, 2008) foi usada uma abordagem para detecção de anomalias de rede baseada na Transformada *Wavelet* e séries auto-regressivas. No sistema proposto foram selecionadas variáveis descritoras de tráfego, usando-se o modelo de agregação por fluxos origem-destino. O sinal original é transformado usando *wavelets* (Transformada *Wavelet* discreta) e os coeficientes *wavelet* $d_{j,k}$ aproximados usando um modelo de predição auto-regressivo do tipo ARX (*AutoRegressive with eXogenous input*) e o resíduo da predição é usado para a detecção de anomalias utilizado o GMM (*Gaussian Mixture Model*). A estratégia de detecção de anomalias consiste na identificação de *outliers* (valor significativamente diferente dos demais),

assumindo-se, que a presença destes no resíduo indica a existência de anomalias no tráfego da rede.

No trabalho de (KIM; REDDY, 2008) foi proposto um detector baseado na análise da correlação dos endereços IP de destino no tráfego de saída de um roteador. A principal diferença deste trabalho em relação aos demais é, justamente, a forma como os dados são agrupados. No primeiro estágio, as informações nos cabeçalhos dos pacotes TCP/IP ou vindos de uma base do NetFlow, como endereço IP e porta de destino, são selecionadas e agrupadas para reduzir o volume de informação. Em seguida, num segundo estágio, as séries são submetidas a uma Transformada *Wavelet* Discreta direta e posteriormente são reconstruídos, com a Transformada *Wavelet* inversa, conforme a escala selecionada. No último estágio é verificada a regularidade das informações comparando-se o histórico dos dados, por meio de *thresholds*. A presença de *outliers* no sinal é considerada como indicador de anomalias. *Thresholds* são estabelecidos com auxílio da desigualdade de Chebyshev e com um intervalo de confiança predefinido.

2.5 Considerações Finais

Neste Capítulo foi abordado como tema o estudo dos Sistemas de Detecção de Intrusão. Inicialmente foram descritas as características básicas do tráfego de rede e posteriormente foram apresentadas uma taxonomia simplificada dos SDI e uma revisão de alguns trabalhos na área. A discussão central deu-se em relação aos Sistemas de Detecção de Intrusões de Rede baseados em anomalias.

Um SDIR baseado em assinaturas é bastante preciso, porém não se adaptam automaticamente a novos ataques e não é capazes de detectar ataques que não estejam presentes no banco de assinaturas. Por outro lado, um SDIR baseado em anomalia pode detectar ataques novos e desconhecidos, no entanto, gera bastante Falsos Positivos. A abordagem baseado em anomalias pode não ser vista como substituta da abordagem baseada em assinaturas em todos as situações, mas como uma alternativa especialmente quando se procura detectar ataques novos ou desconhecidos.

A detecção de anomalias em redes de computadores é uma área de pesquisa bastante ativa. A Transformada *Wavelet*, método baseado na Análise de Sinais, demonstrou aplicabilidade para a análise do tráfego e detecção de anomalias de rede (HUANG; THAREJA; SHIN, 2006) (GAO et al., 2006) (LU; TAVALLAEE; GHORBANI, 2008) (KIM; REDDY, 2008). A Transformada *Wavelet* permite uma análise do tráfego de rede em diferentes escalas de tempo.

Motivado por trabalhos prévios (HUANG; THAREJA; SHIN, 2006) (GAO et al., 2006) (LU; TAVALLAEE; GHORBANI, 2008) (KIM; REDDY, 2008) que usaram a Transformada *Wavelet* em alguma das fases da modelagem de tráfego para a detecção de anomalias, o trabalho desenvolvido nesta dissertação explora o uso da Transformada *Wavelet* Discreta Daubechies para a análise do tráfego e o uso de *Thresholds* para a detecção de anomalias.

3 *WAVELETS* E *THRESHOLD*

Neste Capítulo são apresentados os conceitos básicos referente às funções *Wavelet* ortonormais da família de Daubechies, suas transformadas discretas e as técnicas de truncamento (*threshold*) dos coeficientes *wavelet*. O objetivo é apresentar uma visão geral sobre o assunto de modo a propiciar uma base para o entendimento do mecanismo de detecção de anomalias de rede, tema central deste trabalho. As funções *Wavelet*, por meio dos algoritmos para o cálculo da Transformada *Wavelet* são usadas no processo de análise dos dados do tráfego de rede. As abordagens de *threshold* servem para a criação de estratégias de detecção de anomalias de rede.

Na Seção 3.1 é apresentada a fundamentação matemática das funções *Wavelet*, a Transformada *Wavelet* Discreta e os algoritmos para o cálculo.

Na Seção 3.2 são apresentadas as abordagens de *threshold*.

3.1 *Wavelets*

A literatura sobre as funções *wavelets* é bastante extensa. A referência clássica para a construção da família de funções ortonormais é o texto de Ingrid Daubechies (DAUBECHIES, 1992). No entanto, esta referência tem um enfoque bastante específico e apresenta os conceitos com alto rigor matemático, essa dissertação baseia-se no texto de Ole Møller Nielsen (NIELSEN, 1998, Cap. 2), seguindo suas definições e nomenclaturas. Recomenda-se também a referência (MALLAT, 1998), que trata das funções *wavelet* do ponto de vista da Análise de Sinais.

3.1.1 Propriedades da função Escala e da função *Wavelet*

Ao longo deste trabalho todos os sinais utilizados serão considerados como elementos do espaço vetorial $L^2(\mathbb{R})$, cujos elementos são funções de quadrado integrável:

$$L^2(\mathbb{R}) = \left\{ y(t) : \int_{-\infty}^{\infty} y(t)dt < \infty \right\} . \tag{3.1}$$

As funções Escala e *Wavelet* têm a propriedade de formarem uma base para o espaço vetorial $L^2(\mathbb{R})$. Ou seja, qualquer elemento deste espaço pode ser decomposto como uma combinação linear das funções Escala $\phi(t)$ e *Wavelet* $\psi(t)$ e suas dilatações e translações (DAUBECHIES, 1992).

Relação de Escala. No espaço $L^2(\mathbb{R})$ as funções Escala $\phi(t)$ e as funções *Wavelet* $\psi(t)$ satisfazem a seguinte relação, denominada Relação de Escala:

$$\phi(t) = 2^{1/2} \sum_{k=0}^{D-1} g_k \phi(2t - k) \ , \tag{3.2}$$

$$\psi(t) = 2^{1/2} \sum_{k=0}^{D-1} h_k \phi(2t - k) \ , \tag{3.3}$$

sendo g_k e h_k constante $\forall k$. A Relação de Escala e significa que a função $\phi(t)$ pode ser gerada por uma combinação linear dela mesma quando dilatada $\phi(2t)$ e transladada $\phi(2t-1), \phi(2t-2), ..., \phi(2t-D-1)$. O parâmetro D é determinado de acordo com o tipo de função *Wavelet*. Os coeficientes g_k e h_k são chamados filtros e estão associados às funções $\phi(t)$ e $\psi(t)$ consideradas. Há várias tipos de função Escala e para cada uma existe uma função *Wavelet* associada. Exemplos de funções *Wavelet* serão tratadas na Seção 3.1.2.

As funções geradas pelas dilatações e translações das funções $\phi(t)$ e $\psi(t)$ são denotadas por:

$$\phi_{j,k}(t) = 2^{-j/2} \phi(2^{-j}t - k) \ , \tag{3.4}$$

$$\psi_{j,k}(t) = 2^{-j/2} \psi(2^{-j}t - k) \ . \tag{3.5}$$

O parâmetro j é chamado parâmetro de escala ou dilatação porque dilata ou comprime a função Escala $\phi(t)$ ou a função *Wavelet* $\psi(t)$ e k é o parâmetro de translação porque ele desloca a função $\phi(t)$ ou a função $\psi(t)$ na escala fixada.

Denota-se, ainda:

$$\phi_k(t) = \phi_{0,k}(t) = \phi(t - k) \ , \tag{3.6}$$

$$\psi_k(t) = \psi_{0,k}(t) = \psi(t - k) \ . \tag{3.7}$$

Pela relação (3.4) a função Escala $\phi(t)$ forma um conjunto de novas funções $\phi_{j,k}(t)$ que correspondem a própria função dilatadas (ou encolhidas) conforme a escala j e deslocadas conforme o parâmetro k.

Pela relação (3.5) são geradas as funções $\psi_{j,k}(t)$ por dilatações e deslocamentos da $\psi(t)$ conforme j e k.

Cada conjunto de funções Escala $\phi(t)$ e *Wavelet* $\psi(t)$ e suas dilatações e translações satisfazem uma série de propriedades fundamentais:

1) Energia Finita. Como ambas são elementos de $L^2(\mathbb{R})$, as funções Escala $\phi(t)$ e as funções *Wavelet* $\psi(t)$ possuem energia finita:

$$\int_{-\infty}^{\infty} |\phi(t)|^2 \, dt < \infty \tag{3.8}$$

$$\int_{-\infty}^{\infty} |\psi(t)|^2 \, dt < \infty \; . \tag{3.9}$$

2) Suporte compacto (MALLAT, 1998). O suporte das funções Escala e *Wavelet* está relacionado à sua localidade. As funções $\phi(t)$ e $\psi(t)$ são localizadas no tempo, em um intervalo limitado e fechado da reta:

$$supp(\phi) = supp(\psi) = [0, D-1] \; , \tag{3.10}$$

o que significa que a função é toda nula fora de um intervalo $[0, D-1]$.

3) A função Escala ϕ e a função *Wavelet* ψ possuem norma igual a 1:

$$\|\phi\|_2 \equiv \left(\int_{-\infty}^{\infty} |\phi(t)|^2 \, dt \right)^{1/2} = 1 \; , \tag{3.11}$$

$$\|\psi\|_2 \equiv \left(\int_{-\infty}^{\infty} |\psi(t)|^2 \, dt \right)^{1/2} = 1 \; . \tag{3.12}$$

4) Ainda, as funções *Wavelet* $\psi(t)$ são oscilatórias (MALLAT, 1998):

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \; , \tag{3.13}$$

ou seja, a integral da função *wavelet* ψ é igual a zero, o que significa que o valor médio da função também é zero e, consequentemente, deve ser oscilatória. O nome *Wavelet* (onda pequena) é consequência desta natureza oscilante e do suporte compacto (JANSEN, 2000).

5) As funções $\phi_{j,k}(t)$ e $\psi_{j,k}(t)$ possuem uma propriedade importante que é ortogonalidade entre si:

$$\int_{-\infty}^{\infty} \phi_{j,k}(t) \phi_{j,l}(t) dt = \delta_{k,l} \tag{3.14}$$

$$\int_{-\infty}^{\infty} \psi_{i,k}(t)\psi_{j,l}(t)dt = \delta_{i,j}\delta_{k,l} \tag{3.15}$$

$$\int_{-\infty}^{\infty} \phi_{j,k}(t)\psi_{j,l}(t)dt = 0, \quad j \geq i \tag{3.16}$$

sendo $i, j, k, l \in \mathbb{Z}$ e $\delta_{k,l}$ é o **Kronecker delta** definido como:

$$\delta_{k,l} = \begin{cases} 0, & k \neq l \\ 1, & k = l \end{cases},$$

o que significa as funções ψ são mutuamente ortogonais conforme a relação para diferentes escalas j e diferentes deslocamentos k.

6) Momentos Nulos

As *wavelets* de Daubechies, possuem todos os momentos nulos até a ordem P:

$$\int_{-\infty}^{\infty} y^p \psi(t)dt = 0, \ y \in \mathbb{R}, \ p = 0, ..., P-1 \ , \tag{3.17}$$

sendo y^p um polinômio de ordem p e P o número de momentos nulos da base *wavelet*.

O número de momentos nulos P da função está relacionado com o suporte $[0, D-1]$:

$$D = 2P \ . \tag{3.18}$$

3.1.2 Exemplos de Funções *Wavelet*

As *wavelets* da família ortonormal de Daubechies são especificadas conforme o número de momentos nulos e, pela relação (3.18), pelo suporte. Seguindo a nomenclatura adotada em (NIELSEN, 1998), as *wavelets* de Daubechies são nomeadas conforme o tamanho do suporte D. Assim tem-se: D2, D4, D6, D8, e assim por diante conforme a escolha do parâmetro D. A *wavelet* de Haar, que é um caso especial da família, corresponde a Daubechies D2 ($D = 2$).

Dentre as *wavelets* de Daubechies, a *wavelet* de Haar é a mais simples, com apenas um momento nulo $P = 1$ e suporte no intervalo $[0, 1]$, $D = 2$. A função Escala $\phi(t)$ para a *wavelet* de Haar é definida como:

$$\phi(t) = \begin{cases} 1, & 0 \leq t < 1 \\ 0, & \text{Caso contrário} \end{cases}, \tag{3.19}$$

assim como a função *Wavelet* $\psi(t)$ de Haar é definida como:

$$\psi(t) = \begin{cases} 1, & 0 \leq t < 1/2 \\ -1, & 1/2 \leq t < 1 \\ 0, & \text{Caso contrário} \end{cases}. \tag{3.20}$$

Excetuando-se a *wavelet* de Haar, para as demais *wavelets* não se conhece a forma explícita da função $\phi(t)$ e $\psi(t)$.

Graficamente, para a *wavelet* de Haar, a função $\phi(t)$ e a função $\psi(t)$ são representadas na Figura 3.1.

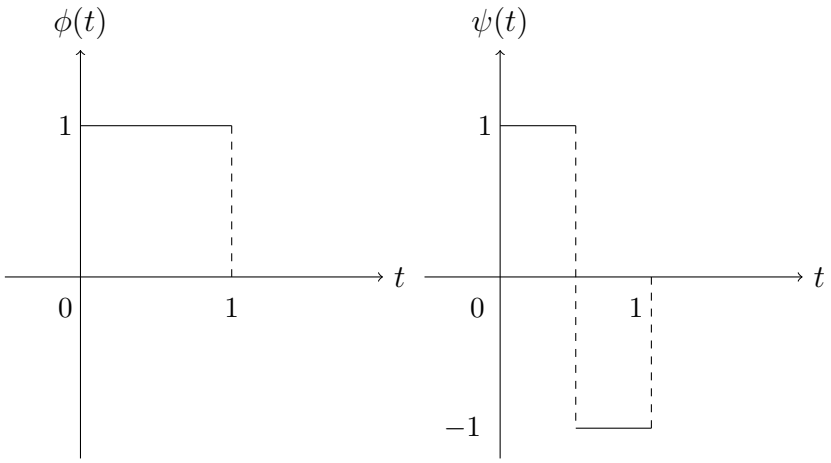

Figura 3.1: Função Escala, $\phi(t)$, e *Wavelet*, $\psi(t)$, de Haar.

Conforme a relação (3.4), para a *wavelet* de Haar o conjunto de novas funções $\phi_{j,k}(t)$ geradas quando $j = 0$ são representadas na Figura 3.2.

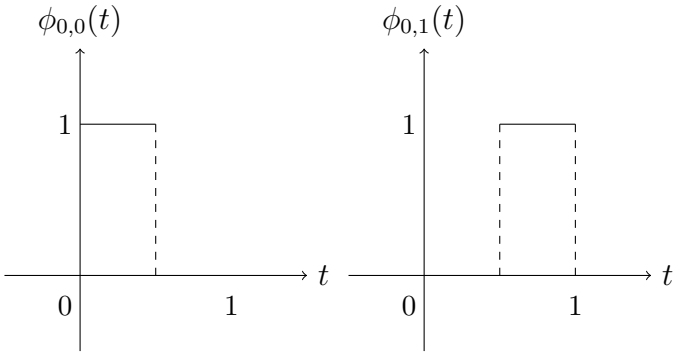

Figura 3.2: Funções Escala de Haar quando $j = 0$, para $k = 0$ e $k = 1$, no intervalo $[0, 1]$.

Em relação a função Escala $\phi(t)$ (Figura 3.1, as funções $\phi_{0,k}(t)$, escala $j = 0$, na Figura 3.2 estão encolhidas (metade da distância no eixo t) e deslocadas conforme os valores de k. A Figura 3.2 representa apenas as funções $\phi_{0,k}$ contidas no intervalo $[0, 1]$ embora o conjunto de funções seja infinito considerando as possibilidade de deslocamentos k para um sinal também infinito.

De forma análoga, pela Relação (3.5), para a *wavelet* de Haar, as funções $\psi_{0,k}(t)$ geradas quando $j = 0$ são representadas na Figura 3.3.

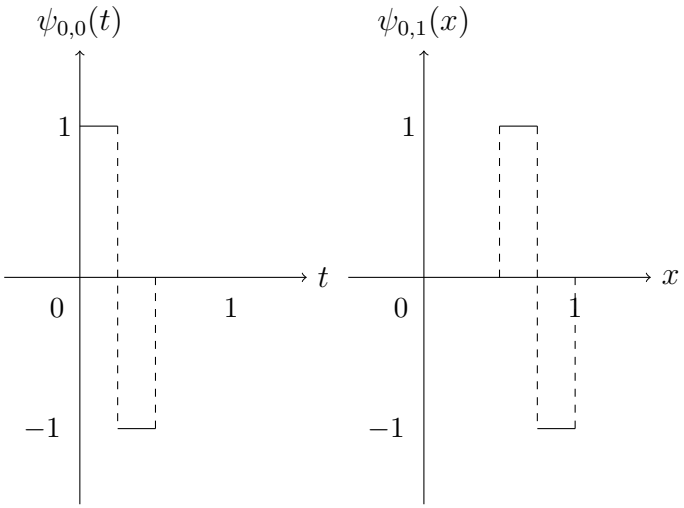

Figura 3.3: Funções *Wavelet* de Haar quando $j = 0$, para $k = 0$ e $k = 1$, no intervalo $[0, 1]$.

Outro exemplo é a *Wavelet* Daubechies D4, que é representada na Figura 3.4. A *wavelet* Daubechies D4 possui dois momentos nulos ($P = 2$) e suporte $D = 4$.

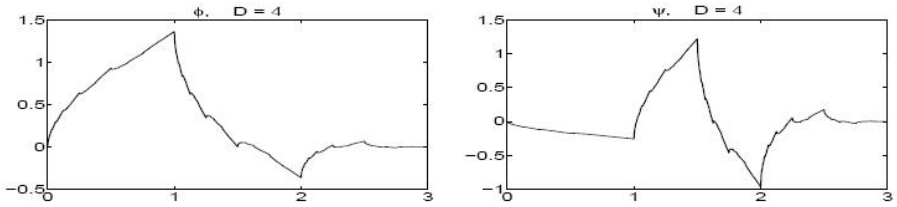

Figura 3.4: Função Escala $\phi(t)$ e função *Wavelet* $\psi(t)$ Daubechies D4 (2 momentos nulos). Fonte: (NIELSEN, 1998)

A Figura 3.5 exemplifica com algumas funções $\phi_{0,k}(t)$ geradas quando $j = 0$ para a *wavelet* Daubechies D4 no intervalo $[0, 3]$. Igualmente, as $\psi_{0,k}(t)$ são vistas na Figura 3.6.

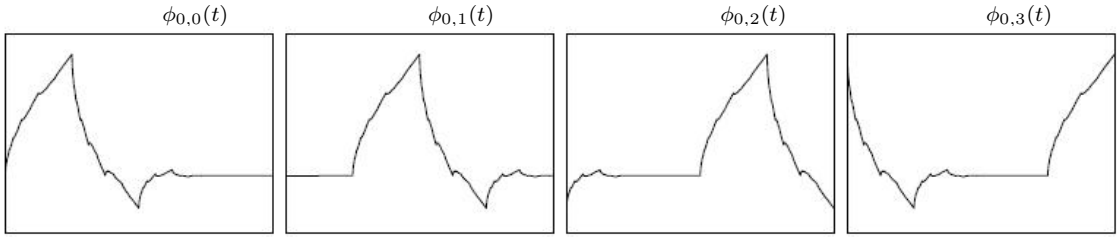

Figura 3.5: Funções Escala de D4 quando $j = 0$, para $k = 0$, $k = 1$, $k = 2$ e $k = 3$. Fonte: (NIELSEN, 1998)

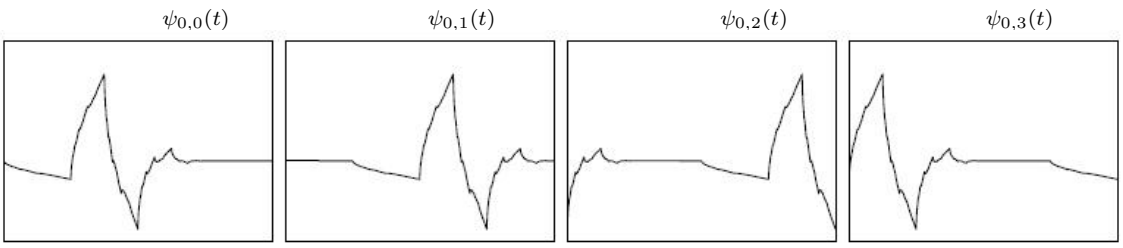

Figura 3.6: Funções *Wavelet* de D4 quando $j = 0$, para $k = 0$, $k = 1$, $k = 2$ e $k = 3$. Fonte: (NIELSEN, 1998)

A *wavelet* Daubechies D6 possui três momentos nulos ($P = 3$) e suporte $D = 6$, Figura 3.7.

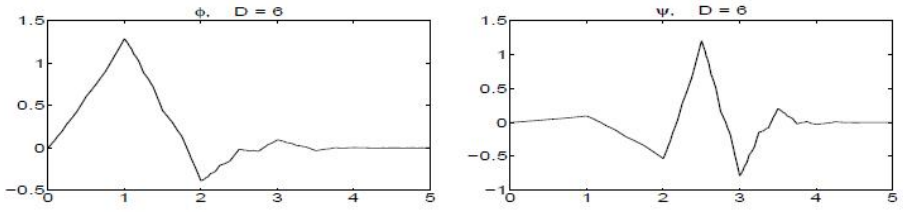

Figura 3.7: Função Escala $\phi(t)$ e função *Wavelet* $\psi(t)$ Daubechies D6. Fonte: (NIELSEN, 1998)

Como não se conhece a forma explícita da função (com exceção da *wavelet* de Haar), através das relações de escala, relação (3.2) e relação (3.3), constroem-se de forma recursiva os valores das funções $\phi(t)$ e $\psi(t)$ para conjuntos de pontos diádicos em escalas cada vez mais finas. As *wavelets* são usadas conhecendo-se esses valores.

3.1.3 *Wavelets* e filtros

A relação de escala (relação (3.4)) diz que a própria função Escala $\phi(t)$ pode ser representada como combinação linear $\{\phi(2t-k),\ k \in \mathbb{Z}\}$ e portanto existem $g_k \in \mathbb{R}$, $k \in \mathbb{Z}$ tais que $\phi(t) = \sqrt{2}\sum_{k=-\infty}^{\infty} g_k \phi(2t-k)$. Como a função $\phi(t)$ tem suporte compacto, o somatório é finito, com apenas um número finito de elementos não zero. Tem-se, então:

$$\phi(t) = \sqrt{2}\sum_{k=0}^{D-1} g_k \phi(2t-k)\ , \phi(t) \in [0, D-1]\ , \tag{3.21}$$

sendo $g_k = \langle \phi(t), \phi(2t-k) \rangle$ (produto interno), que em $L^2(\mathbb{R})$ é dado por:

$$g_k = \int_{-\infty}^{\infty} \phi(t)\phi_{1,l}(t)dx\ . \tag{3.22}$$

De forma similar, tem-se:

$$\psi(t) = \sqrt{2} \sum_{k=0}^{D-1} h_k \phi(2t - k) \ , \tag{3.23}$$

sendo:

$$h_l = \int_{-\infty}^{\infty} \psi(x) \phi_{1,l}(x) dx \ . \tag{3.24}$$

Os g_l são chamados coeficientes do filtro G e os h_l são chamados coeficientes do filtro H. Nos vetores dos filtros G e H, o parâmetro D é um inteiro positivo par que determina o número de coeficientes (constantes) nos respectivos filtros $g_0, g_1, ..., g_{D-1}$ e $h_0, h_1, ..., h_{D-1}$. O parâmetro D depende do suporte da *wavelet* específica e, pela expressão (3.18), do número de momentos nulos da função $\psi(t)$.

Os valores dos coeficientes dos filtros G (expressão (3.22)) e H (expressão (3.24)) são calculados através de um sistema de equações que engloba todas as propriedades a serem impostas à família, além de comportamentos exigidos para suas Transformadas de Fourier. Para obter detalhes da dedução dos filtros sugere-se o livro de (DAUBECHIES, 1992). Neste trabalho, da mesma forma que na maioria das aplicações das *Wavelets*, serão usados apenas os valores dos filtros já calculados para a família, também encontrados em (DAUBECHIES, 1992).

Os vetores dos coeficientes dos filtros possuem norma unitária:

$$\|G\|_2 \equiv \left(\sum_{l=0}^{D-1} |g_l|^2 \right)^{1/2} = 1 \ . \tag{3.25}$$

$$\|H\|_2 \equiv \left(\sum_{l=0}^{D-1} |h_l|^2 \right)^{1/2} = 1 \tag{3.26}$$

As propriedades (3.26) e (3.25) são decorrência de (3.11) e (3.12) respectivamente.

Além disso os filtros são ortogonais entre si:

$$\langle G, H \rangle \equiv \sum_{l=0}^{D-1} g_l h_l = 0 \ , \tag{3.27}$$

em decorrência de (3.16).

Os coeficientes dos filtros G e H estão relacionados entre si:

$$h_l = (-1)^l g_{D-1-l}, \ l = 0, 1, ..., D-1 \ . \tag{3.28}$$

A *wavelet* de Haar, (3.19) e (3.20), por exemplo, possui 2 coeficientes ($D = 2$) para o filtro G:

$$G = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right) \tag{3.29}$$

e pela relação (3.28), que produz $h_0 = g_1$ e $h_1 = -g_0$, encontram-se os coeficientes do filtro H:

$$H = \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right) \ . \tag{3.30}$$

A *wavelet* Daubechies D4 possui os seguintes coeficientes ($D = 4$) para os filtros:

$$G = \left(g_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, g_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, g_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}, g_3 = \frac{1-\sqrt{3}}{4\sqrt{2}} \right) \tag{3.31}$$

e pela relação (3.28) são encontrados os coeficientes do filtro H:

$$H = (h_0 = g_3, h_1 = -g_2, h_2 = g_1, h_3 = -g_0) \ . \tag{3.32}$$

Na Tabela 3.1 são apresentados os coeficientes do filtro G das função *wavelet* de Daubechies D6 juntamente com a D2 (Haar) e D4.

	h_0	h_1	h_2	h_3	h_4	h_5
D2	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$				
D4	$\frac{1+\sqrt{3}}{4\sqrt{2}}$	$\frac{3+\sqrt{3}}{4\sqrt{2}}$	$\frac{3-\sqrt{3}}{4\sqrt{2}}$	$\frac{1-\sqrt{3}}{4\sqrt{2}}$		
D6	0.332670552	0.806891509	0.459877502	-0.13501102	-0.08544127	0.03522629

Tabela 3.1: Coeficientes do Filtro passa baixa G das *wavelets* D2, D4 e D6. Fonte: (GOUD; BINULAL; K.P, 2009)

As *wavelets* por meio dos seus filtros correspondentes são usadas para analisar uma função em diferentes escalas. A Análise em Multirresolução serve como uma maneira de representar o conceito de mudança de escalas e a representação de um sinal por meio *wavelets*.

3.1.4 Análise em multirresolução

Uma Análise em Multirresolução (MRA - Multiresolution Analysis) é caracterizada por:

$$
\begin{aligned}
&\{0\} \ldots \subset V_{-1} \subset V_0 \subset V_{+1} \subset \ldots \subset L^2(\mathbb{R}) && \text{(a)} \\
&\textstyle\bigcup_{j=-\infty}^{\infty} V_j = L^2(\mathbb{R}) \ , \ \bigcap_{j=-\infty}^{\infty} V_j = \{0\} && \text{(b)} \\
&\{\phi(t-k)\}_{k\in\mathbb{Z}} \ \text{ é uma base ortonormal para } V_0 && \text{(c)} \\
&y \in V_j \ \Leftrightarrow \ y(2.) \in V_{j+1} && \text{(d)}
\end{aligned}
\tag{3.33}
$$

A expressão (3.33) (a) descreve uma sequência de espaços encaixados, tal que, em expressão (3.33) (b), a união de todos os espaços $V_j \in L^2(\mathbb{R})$ forma o espaço $L^2(\mathbb{R})$. Quando se passa de um espaço V_j para um espaço V_{j+1}, informações são perdidas, enquanto que quando se passa do espaço V_j para um espaço V_{j-1} mais informações são conhecidas. Pela expressão (3.33) (c), o espaço V_0 tem uma base ortonormal consistindo de translações de uma função ϕ. Projeções de uma função $y \in L^2(\mathbb{R})$ são aproximações de y no espaço V_j. Pela expressão (3.33) (d), quando uma função $y(t)$ move-se de um espaço V_j para um espaço V_{j+1} é reescalonada por dois.

A partir da sequência dos subespaços encaixados (expressão (3.33) (a)), define-se W_j como o complemento ortogonal de V_j em V_{j+1}, $V_j \perp W_j$:

$$V_j = V_{j+1} \oplus W_{j+1} \tag{3.34}$$

sendo, no caso inicial: $V_0 = V_1 \oplus W_1$.

O subespaço W_j corresponde à informação complementar quando se passa do subespaço V_j para V_{j+1}. Aplicando-se (3.34) tem-se:

$$V_0 = V_J \oplus \left(\bigoplus_{j=J}^{1} W_j \right) = V_J \oplus W_J \oplus \ldots \oplus W_1 \ . \tag{3.35}$$

Qualquer função em V_j pode ser expressa como uma combinação linear de funções em V_J e $W_J, \ldots, W_1$.

Pela expressão (3.33) (c), o conjunto $\{\phi(t-k)\}_{k\in\mathbb{Z}}$ é uma base ortonormal em V_0 e por repetidas aplicações de (3.33) (c) segue que o conjunto:

$$\left\{2^{-j/2}\phi(2^{-j}t-k)\right\}_{k\in\mathbb{Z}} \ \text{ é uma base ortonormal para } V_j \ . \tag{3.36}$$

Similarmente para uma função $\psi(t)$, o conjunto:

$$\left\{2^{-j/2}\psi(2^{-j}t-k)\right\}_{k\in\mathbb{Z}} \ \text{ é uma base ortonormal para } W_j \ . \tag{3.37}$$

A Transformada *Wavelet* Discreta (TWD) é a ferramenta que decompõe um sinal em diferentes componentes e possibilita o estudo de cada componente conforme sua escala (DAUBECHIES, 1992). A TWD descreve o sinal em termos de uma forma grosseira, mais diferentes níveis de detalhes, dos mais finos aos mais grossos.

3.1.5 A Transformada *Wavelet* Discreta

A Transformada *Wavelet* Discreta é implementada usando-se um algoritmo rápido baseado em filtros G e H, relacionados a bases *wavelet* ortogonais, ao invés das funções ϕ e ψ diretamente. O algoritmo para o cálculo da TWD é conhecido como Algoritmo Piramidal (MALLAT, 1989) por decompõe um por meio de sucessivos passos usando os filtros H e G de modo recursivo em cada aproximação.

O algoritmo da TWD é dito "rápido" pois possui baixa complexidade e permite uma computação rápida. Do ponto de vista da Análise de Algorítimos, sub-área da Análise da Complexidade Computacional, o cálculo da TWD possuem complexidade computacional teórica de ordem linear no tempo $\mathcal{O}(N)$ (MALLAT, 1998). A complexidade temporal de ordem linear significa que o número de passos para a execução do algoritmo aumenta linearmente conforme o tamanho N dos dados de entrada (WILF, 1994). A linearidade da complexidade no tempo é uma característica desejável para um algoritmo, pois o tempo de processamento está linearmente relacionado ao tamanho da entrada.

Para um sinal $y(t) \in V_0$, sua expansão em série é dada por:

$$y(t) = \sum_{-\infty}^{\infty} \alpha_{0,k}\phi(t-k) \ , \quad \alpha_{0,k} = \int_{-\infty}^{\infty} y(t)\phi(t-k)dt \tag{3.38}$$

No entanto, $y(t)$ pode ser representado em relação a qualquer subespaço V_j e portanto:

$$y(t) = \sum_{-\infty}^{\infty} \alpha_{j,k}\phi_{j,k}(t) \ , \quad \alpha_{j,k} = \int_{-\infty}^{\infty} y(t)\phi_{j,k}(t)dt \tag{3.39}$$

Pela expressão (3.34), $V_j = V_{j+1} \oplus W_{j+1}$, o que implica que todo o elemento representado em V_j pode ser representado como soma direta de suas componentes em V_{j-1} e W_{j-1}. Com isso, a expressão (3.39) equivale a:

$$y(t) = \sum_{k=-\infty}^{k=\infty} \alpha_{j-1,k}\phi_{j-1,k}(t) + \sum_{k=-\infty}^{k=\infty} \beta_{j-1,k}\psi_{j-1,k}(t) \ , \tag{3.40}$$

com

$$\alpha_{j-1,k} = \int_{-\infty}^{\infty} y(t)\phi_{j-1,k}(t)dt \tag{3.41}$$

e

$$\beta_{j-1,k} = \int_{-\infty}^{\infty} y(t)\psi_{j-1,k}(t)dt \ . \tag{3.42}$$

Como (3.34) vale para toda escala j, pode-se seguir este processo de decomposição até um nível bem grosseiro de representação, denominado J. E assim $V_j = V_{j+1} \oplus W_{j+1} = (V_j = V_{j+2} \oplus W_{j+2}) \oplus W_{j+1} = (V_J \oplus W_J) \oplus \ldots \oplus W_{j+1}$. Desta maneira a expansão de $y(t)$ é dada por:

$$y(t) = \sum_{-\infty}^{\infty} \alpha_{J,k}\phi_{J,k}(t) + \sum_{-\infty}^{\infty} \beta_{J,k}\psi_{J,k}(t) + \sum_{-\infty}^{\infty} \beta_{j-1,k}\phi_{j-1,k}(t) \ , \tag{3.43}$$

com $\alpha_{J,k}$ $\beta_{J,k}$ $\beta_{j-1,k}$ seguindo relações (3.41) e (3.42) para a escala correspondente.

Como o sinal analisado, na prática, não é de comprimento infinito, sendo na verdade definido por meio de amostras discretas dadas em relação a um tempo finito de captação, a expressão (3.43) é adaptada para um sinal discreto e de tamanho finito. A expansão *wavelet* de um sinal discretizado $y[t] = (y_0, \ldots, y_{N-1})$ é dada por:

$$y[t] = \sum_{k=0}^{N_J} c_{J,k}\phi_{J,k}(t) + \sum_{j=J}^{1}\sum_{l=0}^{N_j} d_{j,k}\psi_{j,k}(t) \ t \in [0, N_0] \ , \tag{3.44}$$

sendo $N_j = N/2^j - 1$, $c_{J,l}$ os coeficientes escala (ou aproximação) e $d_{j,l}$ os coeficientes *wavelet* (ou detalhes) em todos os níveis de fatoração da transformada, $j = 0, 1, \ldots, J-1$.

A TWD direta do sinal é computada por sucessivas passagens (encadeamento) por filtros H e G. Os filtros dependem (tamanho e valores) da função *wavelet* relacionada (Seção 3.1.3). A Figura 3.8 apresenta graficamente o algoritmo da TWD direta (Algoritmo Piramidal de Mallat).

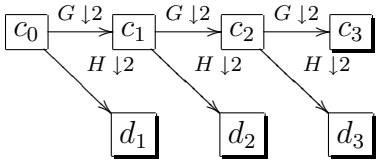

Figura 3.8: Representação gráfica do Algoritmo Piramidal de Mallat, Transformada *Wavelet* Discreta direta.

Na Figura 3.8, G denota o filtro passa-baixa (expressão (3.22)), H denota o filtro passa-alta (expressão (3.24)), $\downarrow 2$ representa a operação de sub-amostragem (o tamanho do vetor resultante possui a metade do tamanho do vetor original), ou seja redução de escala, d_1, d_2 e d_3 são os coeficientes *wavelet* ou detalhes, em cada nível, e c_3 são os coeficientes escala ou aproximação no último nível da transformada.

Transformada *Wavelet* Discreta direta:

$$c_{j+1,k} = \sum_{l=0}^{D-1} g_l c_{j,2k+l} \tag{3.45}$$

$$d_{j+1,k} = \sum_{l=0}^{D-1} h_l c_{j,2k+l} \ , \tag{3.46}$$

para $j = 0, \ldots, J$ e $k = 0, \ldots, N/2^J - 1$. Os coeficientes escala $c_{J,k}$ podem ser interpretados como a média local ponderada do sinal $y[t]$ e os coeficientes *wavelet* $d_{j,k}$ representam a informação complementar ou os detalhes que escapam da média ponderada.

As expressões (3.45) e (3.46) referem-se TWD direta parcial (apenas um nível). Os vetores resultantes c_{j+1} e d_{j+1} possuem a metade do tamanho da aproximação anterior c_j. Os vetores de coeficientes c_{j+1} e d_{j+1} são encontrados pela convolução do vetor da aproximação no nível anterior c_j com os vetores dos coeficientes dos filtros G e H respectivamente.

No primeiro passo da TWD direta, a primeira aproximação corresponde ao sinal inicial $c_{0,t} = y[t]$. A cada iteração do algoritmo a aproximação do sinal c_j é decomposta para a geração de novos c_{j+1} e d_{j+1} e assim sucessivamente até que se tenha a aproximação mais grosseira c_J conforme J desejado e um conjunto de detalhes $d_J, ..., d_1$. Os coeficientes da Transformada *Wavelet* ordenados são representados como:

$$w = \left((c_{J,k})_{k=0}^{N_J} , \left((d_{j,k})_{k=0}^{N_J} \right)_{j=J}^{1} \right) \ , \tag{3.47}$$

ou seja, w é a representação finita (vetor) em termos apenas dos coeficientes da decomposição do sinal na expressão (3.44).

Para um sinal genérico, a TWD direta (decomposição) é representada, sob o ponto de vista dos vetores dos coeficientes, pela Figura 3.9.

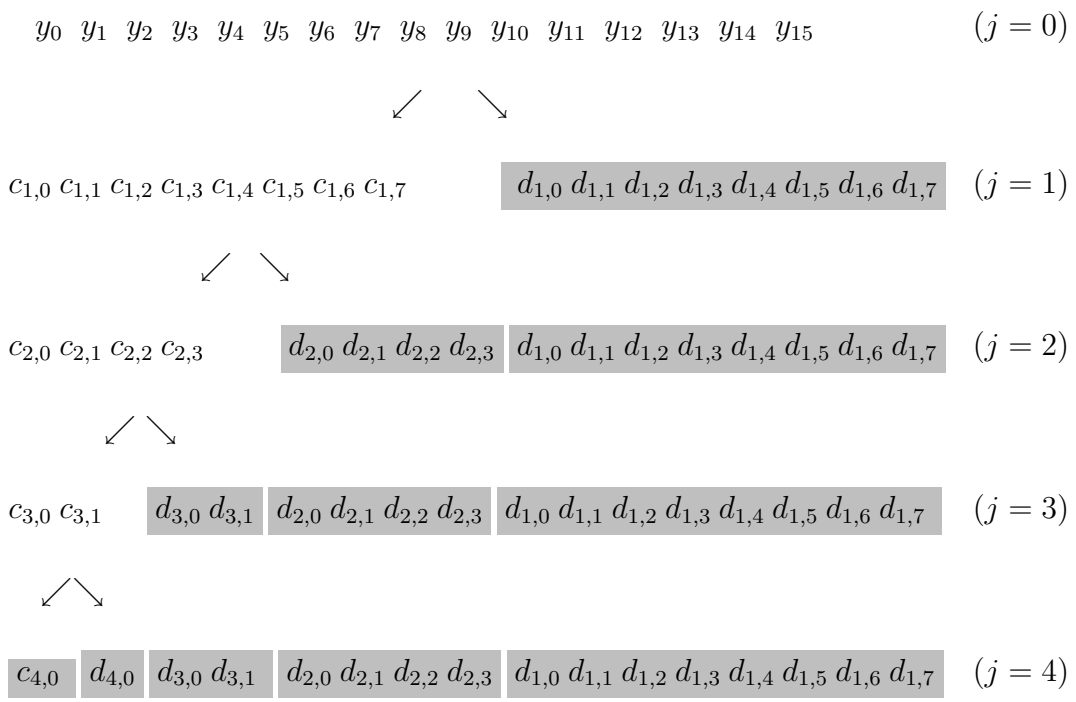

Figura 3.9: Representação da Transformada *Wavelet* Discreta para um sinal genérico y com 16 amostras (2^4). Os coeficientes sombreados, obtidos em cada nível, permanecem inalterados nos próximos níveis. Neste exemplo a transformação vai até o maior nível possível ($j = 4$).

Na Figura 3.9, o sinal $y[t]$, de tamanho $N = 16$ (2^3), é decomposto inicialmente em dois vetores c_1 e d_1, correspondentes ao nível $j = 1$, cada um com a metade do tamanho do vetor do sinal original. No segundo nível $j = 2$, os coeficientes do vetor c_1 (aproximação) é usado para vetor um novo vetor de aproximação c_2 e detalhes d_2. O processo segue até o nível máximo possível.

Ao final do processo da Figura 3.9 os coeficientes da transformada *wavelet*, $c_{j,l}$ e $d_{j,l}$, são agrupados na forma:

$$(c_{3,0}, d_{3,0}, d_{2,0}, d_{2,1}, d_{1,0}, d_{1,1}, d_{1,2}, d_{1,3}) \ . \tag{3.48}$$

A partir dos coeficientes da transformada *wavelet* (expressão (3.47)), o sinal pode ser reconstruído pelo processo inverso. A TWD inversa, Figura 3.10, é o processo inverso da TWD direta e permite a reconstrução do sinal original a partir dos coeficientes da transformada *wavelet*. O processo iterativo de reconstrução do sinal a partir dos coeficientes da transformada *wavelet* é representado na Figura 3.10.

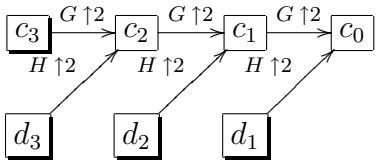

Figura 3.10: Representação gráfica do Algoritmo Piramidal de Mallat, Transformada *Wavelet* Discreta Inversa.

Na Figura 3.10, a aproximação (coeficientes escala) mais grosseira e os coeficientes dos detalhes (coeficientes *wavelet*) de cada nível da transformada passam pelos filtros passa-alta H e passa-baixa G e são reunidos conforme a expressão (3.49). O processo continua até que todos os níveis da transformada sejam processados e o sinal original seja reconstruído.

Transformada *Wavelet* Discreta inversa:

$$c_{j,k} = \sum_{n=n_1(k)}^{n_2(k)} c_{j+1,n} g_{l-2n} + d_{j+1,n} h_{l-2n} \ , \tag{3.49}$$

sendo:

$$\left\lceil \frac{l-D+1}{2} \right\rceil \equiv n_1(l) \leq n \leq n_2(l) \equiv \left\lfloor \frac{l}{2} \right\rfloor \ , \tag{3.50}$$

sendo que $\lceil x \rceil$ significa: O menor inteiro maior que x; e $\lfloor x \rfloor$ significa: o maior inteiro menor que x.

Exemplo de TWD com a *wavelet* de Haar. Usando-se a *wavelet* de Haar (suporte $D = 2$, momentos nulos $P = 1$) como exemplo, as expressões (3.45) e (3.46) da TWD parcial correspondem a:

$$c_{j+1,k} = g_0 c_{j,2k} + g_1 c_{j,2k+1} \tag{3.51}$$

$$d_{j+1,k} = h_0 c_{j,2k} + h_1 c_{j,2k+1} \ , \tag{3.52}$$

com $g_0 = 1/\sqrt{2}$ e $g_1 = 1/\sqrt{2}$ e pela relação (3.28) $h_0 = 1/\sqrt{2}$ e $h_1 = -1/\sqrt{2}$.

A geração dos coeficientes escala $c_{j,k}$ para a *wavelet* de Haar na expressão (3.51) é descrita graficamente pela Figura 3.11.

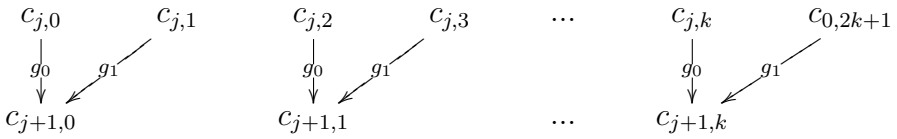

Figura 3.11: Transformada *Wavelet* Discreta, *wavelet* de Haar, geração dos coeficientes escala $c_{j,k}$.

Na Figura 3.11 são gerados os coeficientes escala $c_{j,k}$. Dois elementos da aproximação inicial $c_{j,k}$ são multiplicados com os coeficientes da *wavelet* de Haar, $H = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ e somados para gerar um elemento no conjunto dos coeficientes escala $c_{j+1,k}$. Note que o processo produz uma versão aproximada, no caso da *wavelet* de Haar, aproximação média, com a metade dos elementos da aproximação inicial.

Para a *wavelet* de Haar a geração dos coeficientes $d_{j,k}$ (expressão (3.52)) é exemplificado na Figura 3.12.

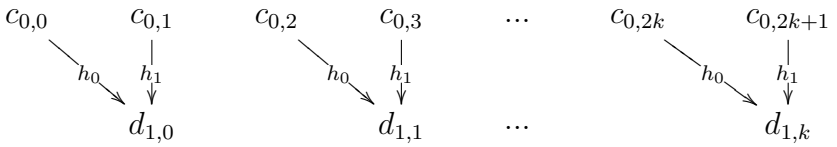

Figura 3.12: Transformada *Wavelet* Discreta, *wavelet* de Haar, geração dos coeficientes *wavelet* $d_{j,k}$.

Na Figura 3.12 são gerados os coeficientes wavelet $d_{j,k}$. Dois elementos da aproximação inicial $c_{j,k}$ são são multiplicados com os coeficientes da *wavelet* de Haar, $H = \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)$ e somados para gerar um elemento no conjunto dos coeficientes escala $d_{j+1,k}$. Note que o processo produz um vetor dos detalhes (coeficientes *wavelet*, no caso da *wavelet* de Haar, diferenças) com a metade dos elementos da aproximação inicial.

Da mesma forma que na TWD com qualquer *wavelet* de Daubechies, no caso da *wavelet* de Haar, o sinal pode ser reconstruído (TWD inversa) a partir do vetor de todos os coeficientes da transformada *wavelet* w (expressão (3.47)). Para a *wavelet* de Haar as Equações para TWD inversa parcial (expressão (3.49)) equivale a:

$$c_{j-1,2k} = g_0 c_{j,k} + h_0 d_{j,k} \tag{3.53}$$

$$c_{j-1,2k+1} = g_1 c_{j,k} + h_1 d_{j,k} \ . \tag{3.54}$$

Graficamente, para a *wavelet* de Haar cada passo da TWD inversa é apresentado na Figura 3.13.

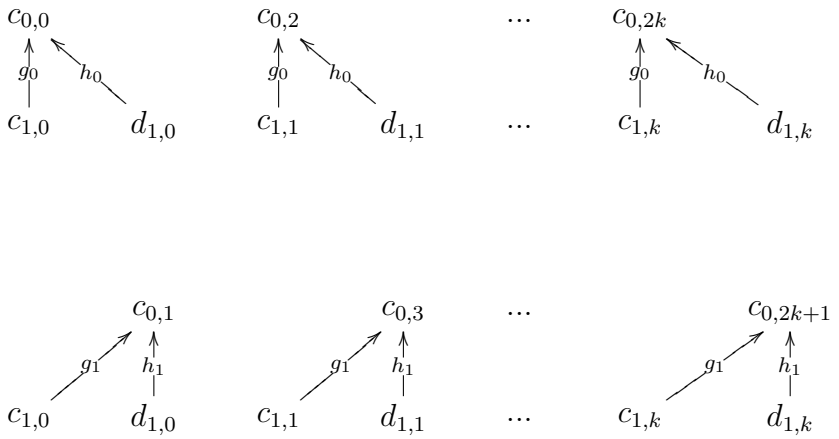

Figura 3.13: Transformada *Wavelet* Discreta inversa, *wavelet* de Haar, reconstrução dos coeficientes *wavelet* $c_{j,k}$.

Na Figura 3.13 cada coeficiente $c_{j,k}$ é reconstruído combinando-se um coeficiente $c_{j-1,k}$ e um coeficiente $d_{j-1,k}$ e fazendo-se uma convolução com os filtros G e H.

Exemplo de TWD com a *wavelet* Daubechies D4. Usando-se agora a *wavelet* Daubechies D4 (suporte $D = 4$, momentos nulos $P = 2$), as expressões (3.45) e (3.46) da TWD correspondem a:

$$c_{j+1,k} = g_0 c_{j,2k} + g_1 c_{j,2k+1} + g_2 c_{j,2k+2} + g_3 c_{j,2k+3} \tag{3.55}$$

$$d_{j+1,k} = h_0 c_{j,2k} + h_1 c_{j,2k+1} + h_2 c_{j,2k+2} + h_3 c_{j,2k+3} \ . \tag{3.56}$$

Para a TWD inversa tem-se:

$$c_{j-1,2k} = g_2 c_{j,k} + h_2 d_{j,k} + g_0 c_{j,k+1} + h_0 d_{j,k+1} \tag{3.57}$$

$$c_{j-1,2k+1} = g_3 c_{j,k} + h_3 d_{j,k} + g_1 c_{j,k+1} + h_1 d_{j,k+1} \ , \tag{3.58}$$

com $G = \left(g_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, g_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, g_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}, g_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}\right)$ e pela relação (3.28) $H = (h_0 = g_3, h_1 = -g_2, h_2 = g_3, h_3 = -g_0)$.

Na TWD com a *wavelet* Daubechies D4, como filtro possui tamanho (suporte) $D = 4$, quatro posições no vetor inicial são multiplicadas com os valores correspondentes nos filtros H ou G para a geração de um coeficiente $c_{j,k}$ (Figura 3.14) ou $d_{j,k}$ (Figura 3.15) correspondentes. Em relação ao vetor inicial, a cada iteração desloca-se duas posições (uma posição em relação ao vetor gerado devido a redução do tamanho ou escala por 2).

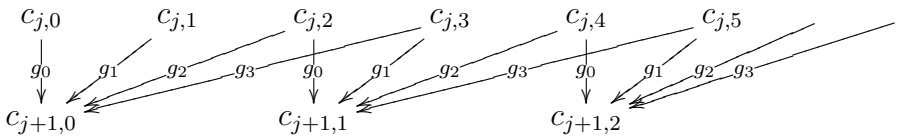

Figura 3.14: Transformada *Wavelet* Discreta, *wavelet* D4, geração dos coeficientes escala $c_{j,k}$.

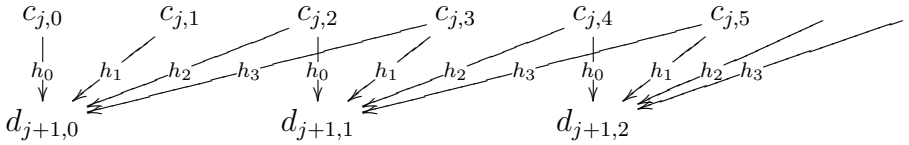

Figura 3.15: Transformada *Wavelet* Discreta, *wavelet* D4, geração dos coeficientes *wavelet* $d_{j,k}$.

O deslocamento a cada duas posições e a necessidade de quatro posições no vetor inicial leva a um problema na última posição do vetor ("fronteira") quando não há mais posições disponíveis no vetor inicial. O problema da fronteira também ocorre na TWD inversa e para todas as *wavelets* de Daubechies com suporte $D > 2$ (Exceto a D2 ou Haar). Como na prática os sinais não são infinitos (vetor com tamanho limitado) contorna-se o problema da fronteira usando-se uma estratégia circular, ou seja, assume-se que o vetor do sinal original é circular e quando faltarem posições no final do vetor usa-se as posições iniciais.

Transformada *Wavelet* Discreta direta para caso circular. Para todas as *wavelets* Daubechies no caso da estratégia circular as expressões da TWD direta (3.45) e (3.46) são dadas por:

$$c_{j+1,k} = \sum_{l=0}^{D-1} g_l c_{j,\langle 2k+l \rangle_{2^j}} \tag{3.59}$$

$$d_{j+1,k} = \sum_{l=0}^{D-1} h_l c_{\langle j,2k+l \rangle_{2^j}} \ , \tag{3.60}$$

sendo que $\langle x \rangle_q$ denota o operador módulo $x \ mod \ q$, ou seja o resto da divisão inteira.

Transformada *Wavelet* Discreta inversa para caso circular. A TWD inversa (expressão (3.49)) é similar para o caso circular:

$$c_{j,k} = \sum_{n=n_1(k)}^{n_2(k)} c_{j+1,\langle n \rangle_{2^{j-1}}} g_{l-2n} + d_{j+1,\langle n \rangle_{2^{j-1}}} h_{l-2n} \ , \tag{3.61}$$

com n_1 e n_2 definidos em (3.50).

Exemplos numéricos para a TWD. Como exemplo numérico da TWD direta, a Figura 3.16 apresenta o processo da transformada para um sinal discretizado com 64 amostras $y[t] = (y_0, \ldots, y_{N-1})$, $N = 2^6 = 64$ usando-se a *wavelet* de Haar.

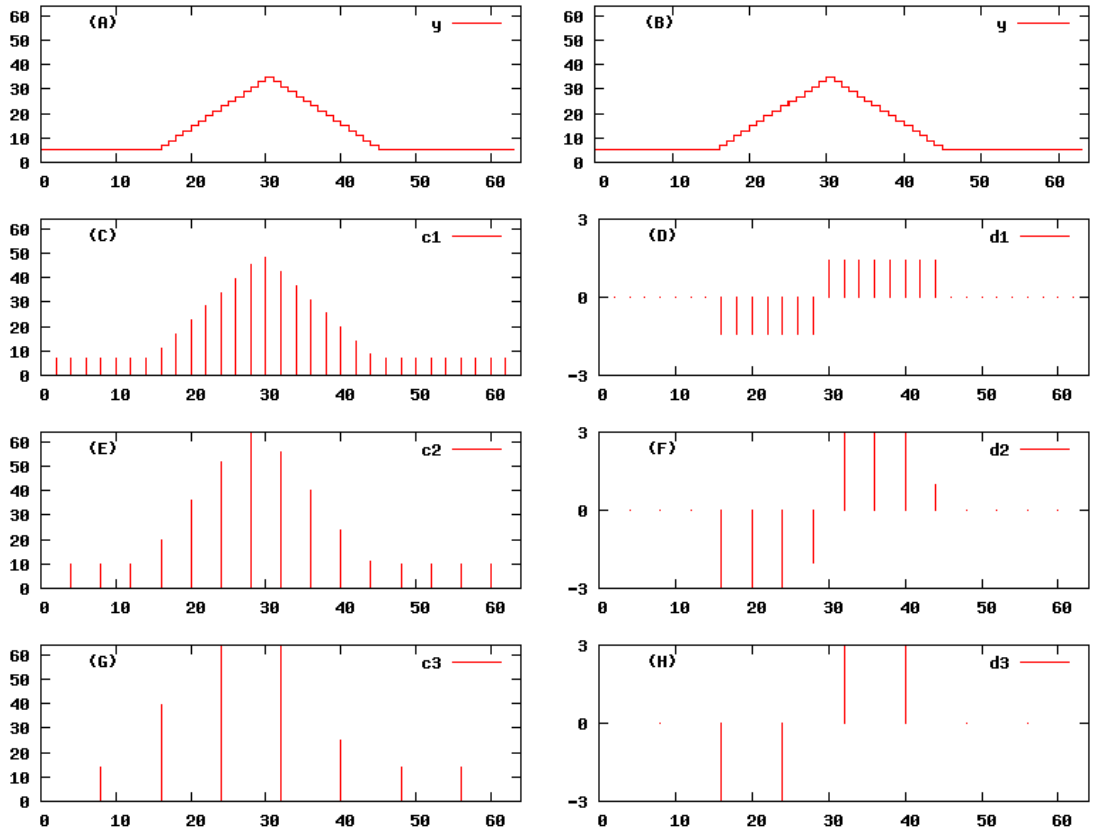

Figura 3.16: Exemplo da Transformada *Wavelet* Discreta, *wavelet* de Haar, para um sinal discreto $y[t]$ gerando os coeficientes transformada c_3, d_3, d_2, d_1. Nos pontos onde a função é constante os detalhes $d_{j,k}$ correspondentes são nulos.

Na Figura 3.16, no primeiro passo da transformada o sinal original $y = (y_0, \ldots, y_{N-1})$ (A) com 64 pontos ($N = 2^6 = 64$,) é decomposto em coeficientes escala $c_{1,k}$ (expressão (3.45)) e coeficientes wavelet $d_{1,k}$ (expressão (3.46)). Repare que os coeficientes $c_{1,k}$ (C) correspondem à uma versão aproximada, com a metade da resolução (metade dos pontos), do sinal original (A). Os coeficientes *wavelet* $d_{1,k}$ (D) correspondem aos detalhes perdidos nesta aproximação e apresentam valores significativos nos pontos onde o sinal apresenta descontinuidade. No segundo passo os coeficientes $c_{1,k}$ são novamente decompostos em coeficientes $c_{2,k}$ (E) e coeficientes $d_{2,k}$ (F) com a metade do tamanho, novamente, os $c_{2,k}$ representam uma aproximação dos $c_{1,k}$ e os $d_{2,k}$ representam os detalhes perdidos na passagem de $c_{1,k}$ para $c_{2,k}$. No terceiro passo são gerados os $c_{3,k}$ (G) e $d_{3,k}$ (H) a partir de $c_{2,k}$.

A TWD com a *wavelet* de Haar (Figura 3.16) consegue representar corretamente nos

coeficientes escala $c_{j,k}$ funções lineares, ou seja, nos pontos onde a função é constante os coeficientes *wavelet* (detalhes), $d_{1,k}$ (D), $d_{2,k}$ (F) e $d_{3,k}$ (H), são nulos nas posições k correspondentes às posições t onde o sinal $y[t]$ (B) é constante.

Usando-se o mesmo sinal $y[t]$ da Figura 3.16 (TWD direta com a *wavelet* de Haar), a Figura 3.17 apresenta um exemplo da TWD direta usando-se a *wavelet* Daubechies D4.

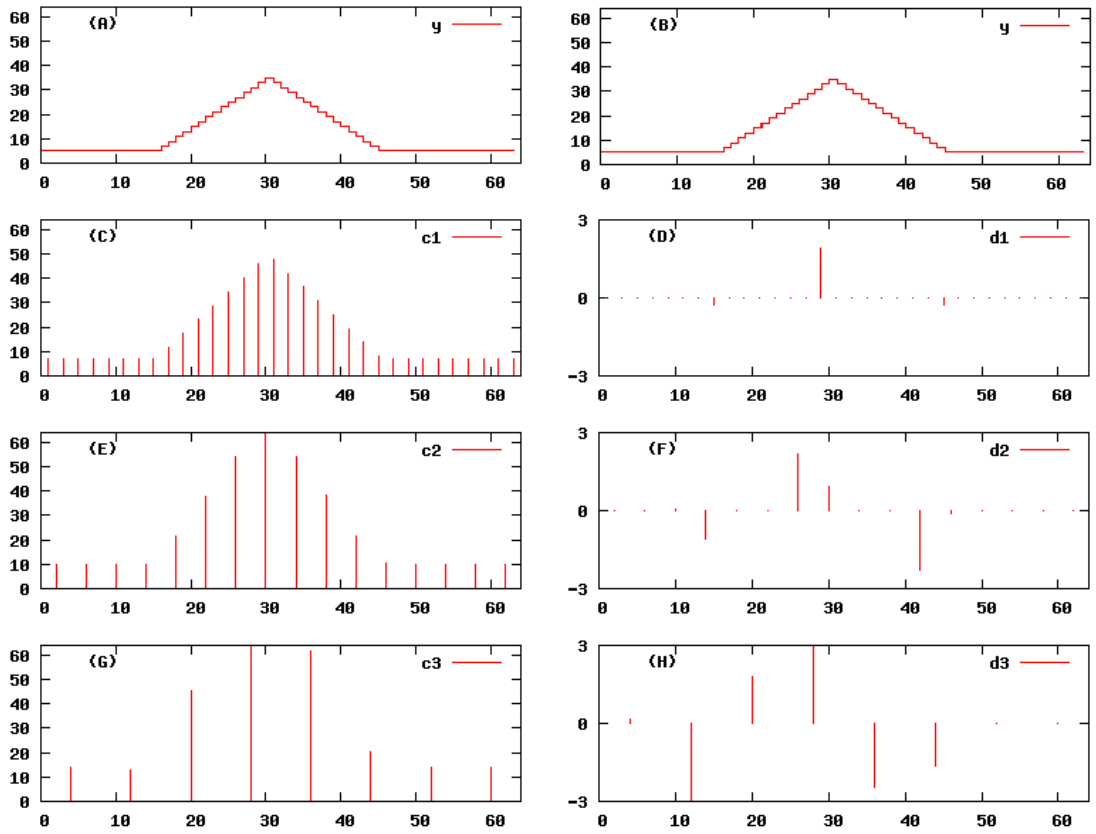

Figura 3.17: Exemplo da Transformada *Wavelet* Discreta, *wavelet* Daubehies D4. Nos pontos onde a função é constante, os detalhes $d_{j,k}$ correspondentes são nulos.

Na Figura 3.17 percebe-se que os coeficientes $d_{j,k}$ (D) (F) (H) nas posições k correspondentes às posições onde o sinal original é constante ou linear são nulos enquanto que os pontos de descontinuidade do sinal produzem valores significativos. Esta é uma característica da *wavelet* Daubechies D4 de representar corretamente nos coeficientes escala $c_{j,k}$ funções constantes e lineares.

Comparando-se os exemplos da TWD usando-se a *wavelet* de Haar (D2) (Figura 3.16) com o exemplo usando-se a *wavelet* Daubechies D4 (Figura 3.17), percebe-se que enquanto a *wavelet* de Haar consegue representar corretamente nos coeficientes escala apenas funções constantes, a TWD usando a *wavelet* Daubechies D4 consegue representar corretamente funções constantes e lineares. Essa constatação condiz com a propriedade dos Momentos Nulos (Seção 3.1.1) das *wavelets* Daubechies, ou seja, a capacidade de re-

presentar corretamente funções polinomiais de ordem p conforme o número de momentos nulos P do filtro. A propriedade dos Momentos nulos também está relacionada a característica da Representação Esparsa de um sinal por meio da TWD. Ou seja, tomando-se como exemplo a TWD com a Daubechies D4 (Figura 3.17), nos pontos onde o sinal é suave (no caso, constante ou linear) os coeficientes $d_{j,k}$ são nulos e onde há singularidades (mudanças) os coeficientes são significativos. Na prática isso significa que alguns coeficientes $d_{j,k}$ podem ser desconsiderados (quando são nulos ou próximos de zero) segundo algum critério sem perda na representação do sinal.

3.1.6 A Transformada *Wavelet* Discreta Packet

A Transformada *Wavelet Packet* (COIFMAN; WICKERHAUSER, 1992) é uma generalização do algoritmo piramidal da Transformada *Wavelet* tradicional. Na TWD *Packet*, contudo, ambos os coeficientes da aproximação e detalhes são decompostos.

Na Figura 3.18 é feita uma comparação entre a TWD tradicional com a TWD *Packet*.

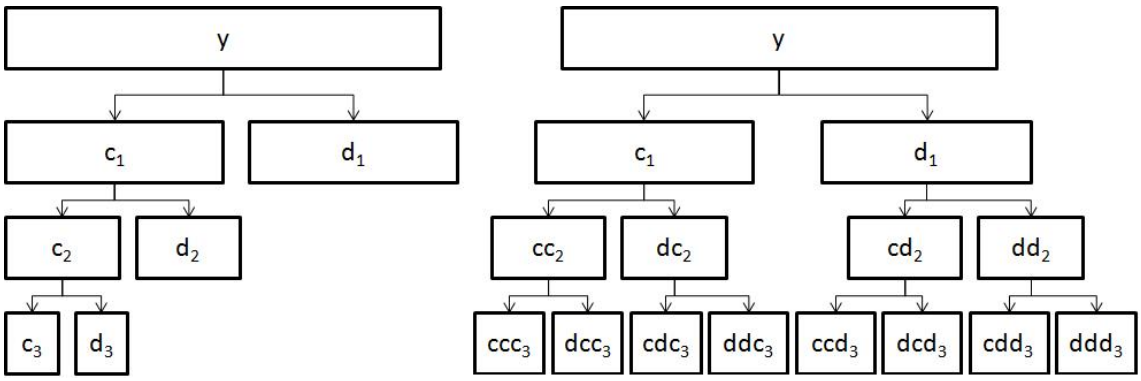

Figura 3.18: Comparação da TWD tradicional (à esquerda) com a arvore de decomposições da TWD *Packet* (à diretita).

Na TWD tradicional (Na Figura 3.18, à esquerda) o sinal inicial é decomposto em coeficientes de aproximação e coeficientes de detalhes. A primeira aproximação (nível 1) é então usada para gerar novos coeficientes de aproximação e novos coeficientes de detalhes (nível 2). A decomposição segue sempre usando os coeficientes da aproximação do nível anterior para a geração de novos coeficientes de aproximação e novos coeficientes de detalhes até o último nível possível ou desejado. Níveis diferentes de coeficientes de detalhes não são mais decompostos. Na TWD *packet*, no entanto, não apenas os coeficientes de aproximações, mas também os coeficientes de detalhes são decompostos, como é ilustrado na Figura 3.18 à direita.

A TWD *Packet* gera 2^N grupos diferentes de coeficientes, em comparação com a TWD tradicional que gera $N + 1$. No entanto, devido a redução da escala em cada passo,

o número total de coeficientes é igual ao sinal original, da mesma forma que a TWD tradicional, e dessa forma não há redundância. Do ponto de vista computacional, a complexidade da TWP é de ordem *loglinear*, $\mathcal{O}(N \log N)$ (COIFMAN; WICKERHAUSER, 1992).

Além da Transformada *Wavelet* Discreta, há ainda a Transformada *Wavelet* Contínua (TWC) (DAUBECHIES, 1992) (MALLAT, 1998). A TWC surgiu como uma alternativa a Transformada de *Fourier*. Como computadores não processam sinais contínuos, a TWC é computada usando-se uma versão discretizada. No entanto, a versão discretizada da TWC não é equivalente a TWD. A TWC discretizada não é realmente uma transformada discreta. A TWC produz informações altamente redundantes e essa redundância requer mais recursos computacionais. A TWD, por outro lado, produz informação suficiente para a análise sem redundância, com uma significativa redução do custo computacional (T.LACHMAN et al., 2010). Devido ao custo computacional, para os propósitos deste trabalho a TWC não será considerada.

3.2 *Threshold*

A redução de ruído é importante nas mais variadas aplicações que envolvem processamento de sinais, assumindo que o ruído está associado à informação de alta frequência, isto é oscilações espúrias. A filtragem de sinais é uma das aplicações da Transformada *Wavelet* e baseia-se no corte ou encolhimento dos coeficientes *wavelet* (detalhes) segundo algum critério. O corte dos coeficientes *wavelet* tem o objetivo de eliminar os componentes ruidosos do sinal considerando que este está representado nos coeficientes *wavelet* (detalhes) menos significativos em relação a um valor de corte (*threshold*). O texto desta Seção segue as definições adotadas por Maarten Jansen em (JANSEN, 2000, Cap. 2).

A filtragem de um sinal usando *wavelets* consiste em três passos:

1. Transformada *Wavelet* Discreta direta do sinal;
2. corte ou encolhimento dos coeficientes *wavelet* conforme a estratégia escolhida; e
3. Transformada *Wavelet* Discreta inversa (reconstrução) usando os coeficientes após o corte.

Dado um sinal discretizado, representado na forma de um vetor $y[t] = (y_0, \ldots, y_{N-1})$, sendo t a posição em relação ao tempo e $N = 2^J$ o número de elementos no vetor. No primeiro passo calcula-se a TWD direta do sinal usando os filtros associados a uma base *wavelet* predefinida:

$$w = W(y) \ . \tag{3.62}$$

A expressão (3.62) representa a TWD em notação matricial, W é a operação de transformação do sinal (transformada *wavelet* discreta direta) e w é o vetor formado pelos coeficientes da transformada *wavelet* (expressão (3.63)), ou seja, coeficientes escala $c_{j,k}$ e coeficientes *wavelet* $d_{j,k}$ ordenados:

$$w = \left((c_{J,k})_{k=0}^{N_J} , \left((d_{j,k})_{k=0}^{N_J} \right)_{j=J}^{1} \right) \ , \tag{3.63}$$

O segundo passo consiste em selecionar os coeficientes *wavelet* $d_{j,k}$ segundo algum critério, que é denominado estratégia de corte:

$$\overline{w} = Thresh_{\lambda}(w) \ . \tag{3.64}$$

Na expressão (3.64), H é a operação de corte ou encolhimento dependendo da escolha da estratégia e determinação do parâmetro λ com o qual os coeficientes $d_{j,k}$ serão comparados, λ é chamado valor de corte (ou *threshold*) e $\overline{w}$ representa os coeficientes da transformada *wavelet* após a operação de corte ou encolhimento:

$$\overline{w} = \left((c_{J,k})_{k=0}^{N_J} , \left((\overline{d_{j,k}})_{k=0}^{N_j} \right)_{j=J}^{1} \right) \ . \tag{3.65}$$

O corte ou encolhimento é realizado apenas nos coeficientes *wavelet* (detalhes) $d_{j,k}$ em todos os níveis j da transformada em todas as posições de cada nível. Há duas estratégias mais usadas para o corte do coeficientes, a *Hard Threshold* (DONOHO; JOHNSTONE, 1995):

$$Thresh_{\lambda}(d_{j,k}) = \begin{cases} 0, & |d_{j,k}| \leq \lambda \\ d_{j,k}, & |d_{j,k}| > \lambda \end{cases} \tag{3.66}$$

e a *Soft Threshold* (DONOHO; JOHNSTONE, 1995):

$$Thresh_{\lambda}(d_{j,k}) = \begin{cases} d_{j,k} - \lambda, & d_{j,l} > \lambda \\ 0, & |d_{j,k}| \leq \lambda \\ d_{j,k} + \lambda, & d_{j,l} < -\lambda \end{cases} \tag{3.67}$$

Na estratégia *Hard Threshold* (função (3.66)), assume-se que os coeficientes wavelets que são menores que o valor do *threshold*, são componentes ruidosos, ficando assim o sinal bem descrito pelos coeficientes wavelets maiores do que o valor de λ. Assim, os coeficientes menores o *threshold* são eliminados. No caso da *Soft Threshold* (função (3.67)) assume-se que os componentes ruidosos estão distribuídos igualmente em

todos os coeficientes wavelets, assim todos os coeficientes *wavelet* são reduzidos pelo valor do *threshold*.

Para a definição do valor do *threshold* λ é proposto por (DONOHO; JOHNSTONE, 1995) a estratégia do *Threshold* Universal:

$$\lambda = \widehat{\delta}\sqrt{2\ln N}\ , \tag{3.68}$$

sendo que λ é o valor do *threshold* encontrado, N o número de amostras no sinal e $\widehat{\delta}$ é estimativa do desvio padrão do ruído.

Ainda em (DONOHO; JOHNSTONE, 1995) é proposto o uso do método baseado no MAD (*Median absolute deviation*) para a estimativa do desvio padrão do ruído:

$$\widehat{\sigma} = 1.4826 * MAD\ , \tag{3.69}$$

sendo que MAD é a mediana dos desvios absolutos em relação à mediana dos dados:

$$MAD = mediana_i(|d_{j,k} - mediana_j(d_j)|)\ . \tag{3.70}$$

A estimativa do desvio padrão do ruído $\widehat{\sigma}$ é calculada usando os coeficientes *wavelet* do primeiro nível $j = 1$ da transformada.

O terceiro e último passo é a reconstrução do sinal a partir dos coeficientes *wavelet* truncados $\overline{w}$, usando a TWD inversa, representada aqui por W^{-1}:

$$\overline{y} = W^{-1}(\overline{w})\ , \tag{3.71}$$

sendo que $\overline{w}$ são os coeficientes da transformada *wavelet* com corte nos coeficientes dos detalhes e $\overline{y}$ é o sinal após a filtragem.

O algoritmo completo para a filtragem do sinal (Algoritmo 3.1) consiste na: Transformada *Wavelet* Discreta direta; estimar o valor do *threshold* usando os coeficientes *wavelet* (detalhes); aplicar a estratégia de corte conforme o *threshold*; e na Transformada *Wavelet* Discreta inversa sobre os coeficientes após o corte.

Algoritmo 3.1: Filtragem de sinal com *Wavelets*.

Entrada: Sinal $y = (y_t)_{t\in[0,1,...,N-1]}$

1 Calcular a transformada *wavelet* discreta direta na série de entrada y, obtendo-se os coeficientes escala $c_{J,k}$ e *wavelets* $d_{j,k}$
2 Estimar o valor do *threshold* λ (nível de corte) com base nos coeficientes *wavelets*
3 Aplicar o corte dos coeficientes *wavelets* menores que o *threshold*, conforme a estratégia de corte (*Hard* ou *Soft*), obtendo-se os coeficientes sem ruído $\overline{d_{j,k}}$
4 Calcular a transformada *wavelet* inversa usando os coeficientes após o corte $\overline{w}$, obtendo-se a estimativa do sinal sem ruído $\overline{y}$

Saída: Sinal filtrado $\overline{y}$

3.3 Considerações Finais

Neste Capítulo inicialmente foram vistas algumas características da Transformada *Wavelet* Discreta e os algoritmos para o cálculo rápido da transformada. Em seguida foram estudas algumas técnicas de corte ou truncamento dos coeficientes *wavelet* (detalhes).

Há um conjunto de bases *wavelet* discretas ortonormais da família Daubechies. As bases *wavelet* de Daubechies são usadas conhecendo-se apenas os coeficientes dos filtros calculados previamente, permitindo o uso de algoritmos rápidos para o cálculo da transformada. O algoritmo rápido para o cálculo da Transformada *Wavelet* Discreta, também conhecido como Algoritmo Piramidal, decompõe o sinal original por meio de sucessivos passos usando os filtros de modo recursivo em cada aproximação do sinal. Após o cálculo da Transformada *Wavelet* Discreta o sinal fica representado como um conjunto de coeficientes, sendo uma aproximação grosseira e vários níveis de detalhes para cada escala. A Transformada *Wavelet* permite dessa forma uma análise do sinal em diferentes escalas de tempo.

O corte ou truncamento de um sinal conforme algum valor de *threshold* busca diminuir ou eliminar coeficientes não significativos para o sinal. A Transformada *wavelet* associada a uma estratégia de corte de coeficientes permite uma representação esparsa do sinal porque possibilita a escolha de apenas os coeficientes mais relevantes para o sinal original.

Do ponto de vista da análise de tráfego de rede, a Transformada *Wavelet* Discreta direta pode ser usada para a modelagem dos dados de rede. O cálculo do valor do *threshold* permite identificar nos coeficientes *wavelet* valores associados a anomalias de tráfego.

4 PROPOSTA DE UM DETECTOR DE INTRUSÕES DE REDE BASEADO EM *WAVELETS*- DIBW

Neste Capítulo é proposto um mecanismo para a detecção ataques de rede baseado na Transformada *Wavelet* Discreta (TWD) e *Thresholds*. O mecanismo proposto, nomeado de Detector de Intrusões em Wavelets (DIbW), é usada no módulo de análise em um Sistema Detector de Intrusões de Rede (SDIR) e destina-se a detecção de ataques por meio da análise dos descritores do tráfego.

A abordagem de análise é baseada na TWD do sinal formado a partir do tráfego padrão de rede e o cálculo de *thresholds* para a indicação de anomalias. Assume-se que um ataque ou intrusão gera uma anomalia (alteração) no padrão de tráfego que é perceptível nos coeficientes da transformada *wavelet*. O mecanismo proposto, portanto, pode ser classificado junto com as abordagens baseadas em anomalias.

Na Seção 4.1 é apresentada a localização do detector e definida uma arquitetura para utilização.

Na Seção 4.2 é apresentado o mecanismo proposto para a detecção de anomalias de rede.

Na Seção 4.3 é feita uma discussão sobre o mecanismo de detecção de anomalias proposto neste trabalho, relacionando-se com as abordagens tratadas nos trabalhos relacionados.

4.1 Arquitetura de um Sistema Detector de Intrusões de Rede

Inicialmente define-se uma arquitetura de um SDIR em que o mecanismo de detecção é usado. A arquitetura (Figura 4.1) é composta por três módulos: Coleta, Análise, e Resposta.

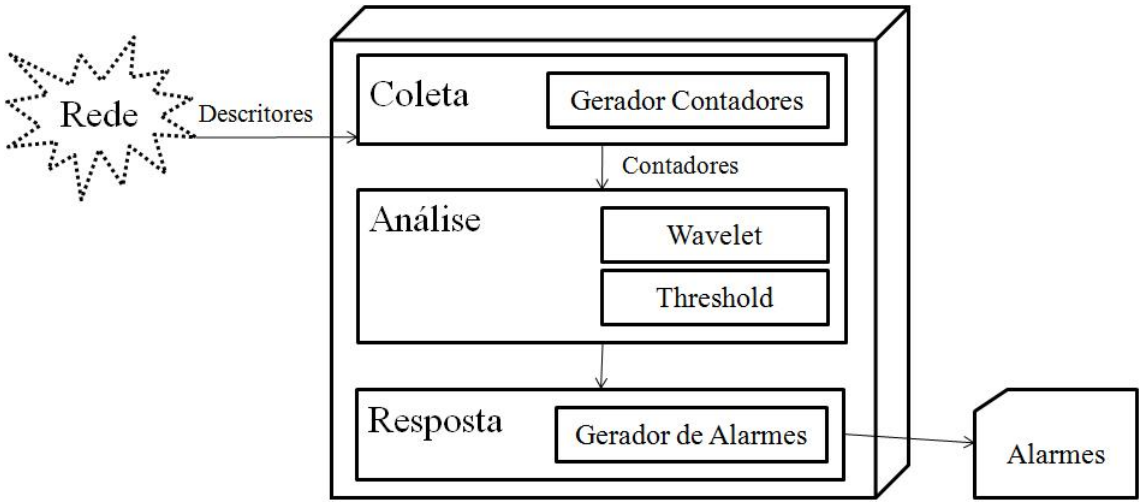

Figura 4.1: Arquitetura do Sistema Detector de Intrusões de rede baseado em *Wavelets*

O módulo de Coleta é responsável pela geração dos dados para a análise. A fonte de dados normalmente é uma sonda de coleta conectada a uma determinada rede ou segmento de rede. A série de dados é gerada pelo processo de amostragem de variáveis descritivas do tráfego de rede. A estatística relacionada a uma determinada variável descritiva, por exemplo o número total de pacotes trafegados, é armazenada em um contador específico. A cada intervalo de tempo Δt (intervalo de amostragem), o valor do contador é lido e repassado para análise.

O módulo de Análise é responsável pela identificação de anomalias nos dados do tráfego de rede. O mecanismo de detecção, tema central deste trabalho, é descrito na Seção 4.2.

No módulo de Resposta são gerados os alarmes. Os alarmes consistem na indicação da ocorrência de algum valor com tamanho absoluto maior que o *threshold* em qualquer um dos níveis de detalhes da transformada *wavelet*. Os alarmes, como informação visual, auxiliam o administrador na tomada de decisão. Ao final os alarmes são salvos, juntamente com sua posição em relação ao tráfego original, em um arquivo de *log* para inspeção *offline*.

4.2 Proposição de um mecanismo de detecção de anomalias de rede baseado em *wavelets*

Neste trabalho é proposto um mecanismo de detecção de anomalias a ser usado no módulo de análise de um Sistema Detector de Intrusões de Rede. O mecanismo proposto, nomeado de Detector de Intrusões baseado em *Wavelets* (DIbW), é baseado na Transformada *Wavelet* Discreta direta dos dados de rede, na determinação de valores de *thresholds*

e a identificação, propriamente dita, da ocorrência de eventuais anomalias nos coeficientes analisados.

A Figura 4.2 apresenta o fluxograma do mecanismo de detecção de anomalias proposto.

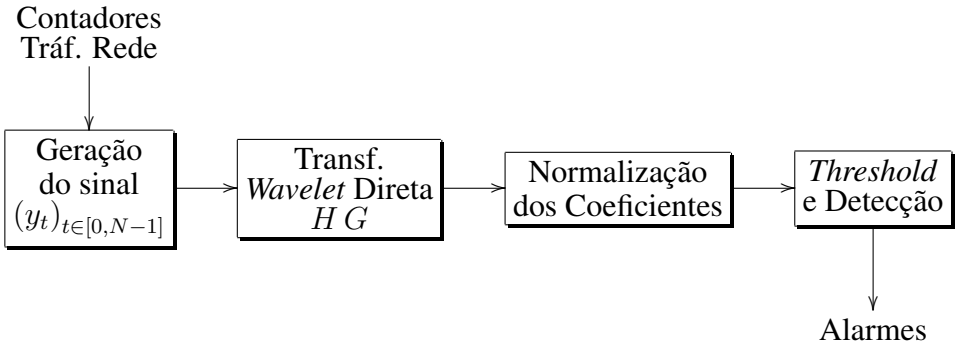

Figura 4.2: Fluxograma do funcionamento do módulo de detecção de anomalias.

O mecanismo de detecção de anomalias (Figura 4.2) é composto de quatro passos: Geração do sinal de análise; Transformada *Wavelet* Discreta Direta; Normalização dos coeficientes *wavelet*; e Detecção usando *Threshold*. Usando os contadores do tráfego de rede, o sinal de análise é gerado por meio de uma janela de observação deslizante. O sinal relativo a janela de observação é transformado usando usando-se as *wavelets* discretas de Daubechies e em seguida os coeficientes *wavelets* são normalizados. A partir dos coeficientes após a normalização é calculado o valor do *threshold* e este é usado para detectar alguma anomalia, indicando na forma de um alarme.

4.2.1 Contadores do Tráfego de Rede

A ideia inicial do método de detecção é que qualquer anomalia de rede gera alguma alteração no comportamento padrão de alguma variável descritiva do tráfego. Por exemplo, variação de volume, abrupta ou progressiva, e mudanças na forma, considerando-se a evolução dos valores passados. Os descritores costumam ser métricas de volume de tráfego, como o número de pacotes e número de *bytes* trafegados, e são modelados como uma série de dados.

Para geração dos dados (contadores) é usado o processo de amostragem de um contador relacionado a alguma variável descritiva do tráfego de rede. A amostragem relacionada a uma determinada variável, por exemplo o número total de pacotes trafegados, é armazenada no contador específico. A cada intervalo de tempo Δt (intervalo de amostragem), o valor do contador é lido, armazenado e repassado para análise.

Há diferentes formas de seleção e agregação de variáveis, porém para este trabalho

utilizam-se características primárias, como número de pacotes por protocolo, porta e *flag*. Por causa de critérios de privacidade neste trabalho são extraídas apenas informações dos *headers* dos pacotes. Na seção 2.3.1 foram apresentadas algumas formas de seleção de variáveis usadas neste trabalho e nos trabalhos relacionados (GAO et al., 2006) (DAINOTTI; PESCAPE; VENTRE, 2006) (KIM; REDDY, 2008) (LU; TAVALLAEE; GHORBANI, 2008).

4.2.2 Geração do sinal

O mecanismo de análise é baseado em uma janela de observação deslizante (GAO et al., 2006), formada por amostras de uma variável descritiva de rede. O conjunto de amostras de uma variável específica, ordenadas no tempo t forma uma série temporal:

$$y[t] = (y_0, y_1, y_2, y_3, \ldots, y_{M-1}) \, , \; M \in \mathbb{N} \, , \tag{4.1}$$

sendo que y_0, quando $t = 0$, corresponde à amostra mais recente, t é o índice[1] da posição da amostra na série e M é o tamanho da série (número de elementos).

A medida que várias amostras são adicionadas a série, a complexidade computacional necessária para a análise de toda a série cresce proporcionalmente. Na hipótese de utilização de todos os valores da série haveria um momento em que a análise se tornaria computacionalmente impraticável. Usa-se aqui, porém, uma janela de observação de tamanho fixo N e menor do que o tamanho da série original, $N < M$. O tamanho N da janela é um número natural, $N \in \mathbb{N}$, potência de 2, requisito para o cálculo da transformada *wavelet*.

$$\overbrace{y_0, y_1, y_2, y_3, \ldots, y_{N-1}}^{Janela}, y_N, \ldots, y_{M-1} \; . \tag{4.2}$$

A cada intervalo de tempo Δt (intervalo de amostragem) uma nova observação é disponibilizada pelo Módulo de Coleta e é incluída na janela observação. A janela de observação é deslizante porque a medida que uma nova amostra y_0 torna-se disponível, esta é incluída na série e a amostra mais antiga y_N é descartada (Figura 4.3). Dessa forma, a janela de observação mantem-se atualizada e de tamanho fixo.

[1] Neste texto as amostras são indexadas de forma crescente para se evitar trabalhar com índices negativos. O índice zero corresponde à amostra mais recente e os índices maiores do que zero as amostras mais antigas.

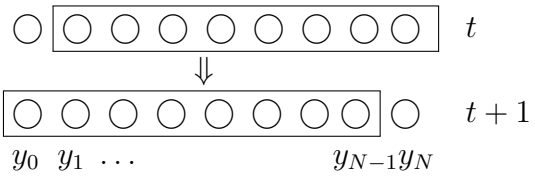

Figura 4.3: Atualização da janela de observação.

A cada atualização da janela de observação um novo sinal y correspondente a esta janela é gerado para a análise:

$$y = (y_0, y_1, y_3, ..., y_{N-1}) \,,\ N = 2^e \ \ e \in \mathbb{N} \,. \tag{4.3}$$

O sinal inicial y é um vetor correspondente à janela de observação, ou seja porção visível da série de dados de entrada. O vetor y é processado pela TWD. Como esta transformada utiliza apenas séries de tamanhos em potência de dois, o tamanho N da janela de observação está restrita à mesma regra:

$$N = 2^e,\ e \in \mathbb{N} \,. \tag{4.4}$$

O intervalo de tempo total da janela de observação é:

$$T = N\,\Delta t \,, \tag{4.5}$$

sendo, N o número de amostras na janela e Δt o intervalo de amostragem.

Considerando que uma série descritiva do tráfego de rede é variável no tempo (não estacionária), justifica-se a utilização de uma janela de observação fixa e limitada para que observações muito antigas não interfiram na análise e sejam consideradas apenas as observações mais recentes. Além disso, limitar o tamanho da janela de observação reduz a carga computacional necessária. A complexidade da transformada *wavelet* discreta é linearmente proporcional ao tamanho do vetor, $O(N)$ (MALLAT, 1998).

Para aplicação em tempo real, o tempo de processamento da janela deve ser menor do que o intervalo de amostragem Δt. Essa é uma restrição para a análise em tempo real e deve ser garantida pelo ajuste correto dos parâmetros: intervalo de amostragem Δt e tamanho da janela N. O atraso (*delay*) máximo teórico, portanto, é o próprio intervalo de amostragem e mínimo o tempo de processamento.

4.2.3 A Transformada *Wavelet*

A Transformada *Wavelet* Discreta (TWD) é uma técnica de análise em multirresolução (MRA - *Multiresolution Analysis*), o que permite que um sinal seja analisado pelos seus componentes localizados em diferentes escalas. Estudos demonstraram que anomalias de rede podem manifestar-se em diferentes escalas de tempo (BARFORD et al., 2002). Em escalas maiores são detectadas anomalias de longa duração e em escalas menores (mais finas) anomalias de curta duração ou variações abruptas (BARFORD et al., 2002).

É sabido que o tráfego de rede possui diversas propriedades estatísticas e exibe dependências curtas (SRD - *Short-Range Dependence*) e longas (LRD - *Long-Range Dependence*) em sua estrutura de correlação (LELAND et al., 1994) (BORGNAT et al., 2008). A estrutura de correlação complexa dificulta a caracterização do tráfego de rede. No entanto, a transformada *wavelet* possui a capacidade de reduzir as complexas relações temporais do tráfego de rede em SRD nos coeficientes *wavelet* (WANG; REN; SHAN, 2003).

A Transformada *Wavelet* Discreta direta é calculada usando-se o sinal corresponde ao tráfego de rede, obtido conforme coleta de determinada variável em um intervalo de amostragem predefinido. A motivação principal na utilização da Transformada *Wavelet* é a sua capacidade em reduzir a correlação temporal dos seus coeficientes (WANG; REN; SHAN, 2003). Desta forma, o tráfego de rede original com SRD e LRD é representado adequadamente pelos coeficientes *wavelet* fracamente correlacionados.

Na TWD, o sinal original:

$$y = (y_0, y_1, ..., y_{N-1}) \ , N = 2^r \ , r \in \mathbb{N} \tag{4.6}$$

é decomposto e fica representado pelos coeficientes da aproximação e dos níveis de coeficientes de detalhes.

A partir do sinal inicial $y[t]$ são gerados os coeficientes na escala 1 para a aproximação $c_{1,k}$ e detalhe $d_{1,k}$, conforme as expressões (3.45) e (3.46) da Seção 3.1.5. Os vetor dos coeficientes c_1 e d_1 gerados possuem a metade do tamanho $N/2$ do sinal inicial:

$$c_1 = \left(c_{1,0}, c_{1,1}, ..., c_{1,N/2-1}\right) \tag{4.7}$$

$$d_1 = \left(d_{1,0}, d_{1,1}, ..., d_{1,N/2-1}\right) \tag{4.8}$$

O vetor w formado por todos os coeficientes c_1 e d_1 possui, portanto, o mesmo tamanho total que o vetor do sinal original:

$$w = \left(\left(c_{1,0}, ..., c_{1,N/2-1}\right), \left(d_{1,0}, ..., d_{1,N/2-1}\right)\right) \ . \tag{4.9}$$

A partir do vetor dos coeficientes c_1 são obtidas as fatorações c_2 e d_2 pela TWD para o próximo nível ($j = 2$). Novamente os coeficientes são armazenados no vetor w:

$$w = \left(\left(c_{2,0}, ..., c_{2,N/4-1}\right), \left(d_{2,0}, ..., d_{2,N/4-1}\right), \left(d_{1,0}, ..., d_{1,N/2-1}\right)\right) \ . \tag{4.10}$$

O processo é executado recursivamente até que o último nível J da TWD seja alcançado, sendo o nível máximo $J \leq \log_2 N$. A Figura 4.4 exemplifica graficamente o processo da TWD para um sinal discretizado.

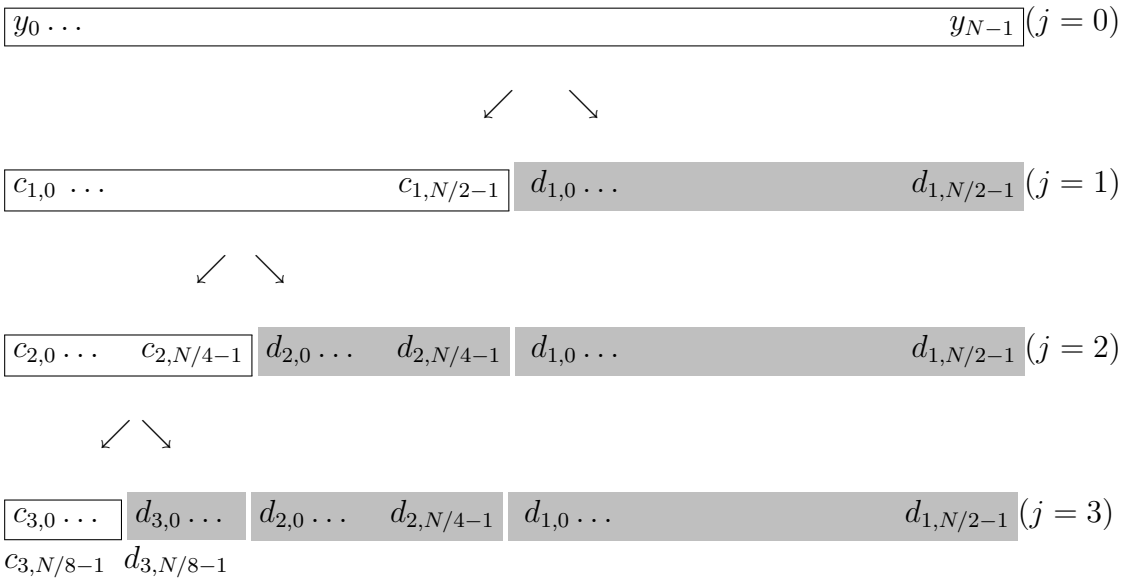

Figura 4.4: Representação da Transformada *Wavelet* Discreta de um sinal y com e 3 níveis de transformação. Os coeficientes *wavelet* sombreados, obtidos em cada nível, permanecem inalterados nos próximos níveis subsequentes.

A cada nível da transformada (Figura 4.4) o tamanho dos vetores resultantes é reduzido pela metade $k = 0 \ldots N/2^j$. São mantidas as informações sobre os detalhes (coeficientes *wavelet*) em diferentes escalas.

4.2.4 Normalização dos coeficientes

O tráfego de rede possui alta impulsividade (LELAND et al., 1994) e não segue uma distribuição normal (SCHERRER et al., 2007). Na Transformada *Wavelet* do tráfego de rede, também os coeficientes *wavelet* (detalhes) não seguem uma distribuição normal (GIBILISCO, 2004, p. 87). A normalização dos coeficientes tem o objetivo de fazer com

que sua distribuição de probabilidade (PDF - *Probability Density Function*) seja mais próxima de uma distribuição normal (gaussiana) (GIBILISCO, 2004, p. 87), permitindo o uso desta característica para a definição de valores de *threshold*.

A distribuição normal é uma das mais importantes distribuições da estatística pela frequência com que ocorre. Assumindo-se a normalidade da distribuição de probabilidade de uma variável pode-se estimar a probabilidade de ocorrência de um determinado valor com base na média e desvio padrão da variável, usando-se uma tabela de distribuição de probabilidades. A Tabela 4.1 apresenta a probabilidade de ocorrência de um valor estar em um determinado intervalo, denominado Intervalo de Confiança, em relação à média e ao desvio padrão para uma distribuição normal.

Intervalo	**Probabilidade**
$\mu \pm 1\sigma$	0.682689492137
$\mu \pm 2\sigma$	0.954499736104
$\mu \pm 3\sigma$	0.997300203937
$\mu \pm 4\sigma$	0.999936657516
$\mu \pm 5\sigma$	0.999999426697
$\mu \pm 6\sigma$	0.999999998027

Tabela 4.1: Probabilidade em relação ao desvio padrão para uma distribuição normal. Construída com base em (GIBILISCO, 2004, p. 161)

Assumindo que uma variável aleatória segue uma distribuição de probabilidade normal cerca de 68% dos valores estão a menos de uma vez o desvio padrão de distância em relação a média, cerca de 95% dos valores estão a menos de duas vezes o desvio padrão de distância em relação à média e cerca 99.7% dos valores estão a menos de 3 vezes o desvio padrão de distância em relação a média, conforme mostra a Tabela 4.1.

Há um conjunto de técnicas usadas para tornar os dados normais (SAKIA, 1992), sendo a raiz quadrada e o logaritmo as técnicas mais usadas. Neste trabalho, é usada a raiz quadrada ou o logaritmo, por necessitarem de menos cálculos e empiricamente terem se mostrado adequadas. A normalização é realizada através da extração da raiz quadrada de cada elemento do vetor dos coeficientes:

$$z_i = sgn(w_i)\sqrt{|w_i|} \tag{4.11}$$

ou pela operação logaritmo:

$$z_i = sgn(w_i)ln(|w_i| + 1)\ , \tag{4.12}$$

sendo $sgn(w_i)$ a Função Sinal:

$$sgn(w_i) = \begin{cases} -1, & \text{se } x < 0 \\ 0, & \text{se } x = 0 \\ 1, & \text{se } x > 0 \end{cases},$$

sendo w_i um valor qualquer do vetor w dos coeficiente da Transformada *Wavelet* e z_i o valor após a normalização.

A escolha da função para a normalização dos coeficientes será discutida na Seção 6.2.

4.2.5 Cálculo do valor do *Threshold*

Para a detecção de anomalias de tráfego de rede propõe-se o cálculo de valores de *threshold* para análise e comparação com os coeficientes *wavelet* (detalhes).

Considera-se que a Transformada *Wavelet* para uma série formada por uma variável descritiva do tráfego de rede é capaz de capturar a tendência (aproximação grosseira) nos seus coeficientes escala $c_{J,k}$ e consegue captar variações dessa tendência em cada nível da transformada via coeficientes *wavelet* $d_{j,k}$ onde há mudanças (singularidades) no padrão do sinal. Propõe-se, então, o uso de *threshold* para detecção destes pontos de singularidades. Para tal proposição, considera-se, também, que os pontos de singularidades do sinal representam anomalias.

Assumindo-se a normalidade dos coeficientes *wavelet* (após a normalização), propõe-se o cálculo do valor do *threshold* conforme:

$$\lambda = \widehat{\mu} + C\,\widehat{\sigma}\ , \tag{4.13}$$

sendo que λ é o valor do *threshold* a ser encontrado, $\widehat{\mu}$ é a média da amostra, $\widehat{\sigma}$ é a estimativa do desvio padrão para a amostra e C é uma constante correspondente ao Intervalo de Confiança (GIBILISCO, 2004, p. 161) desejado conforme a Tabela 4.1. Pela expressão (4.13), o valor do *threshold* λ encontrado considera apenas a parte positiva em relação à Tabela 4.1, no entanto, na aplicação do *threshold* são considerados os valores positivos ou negativos em módulo.

O trabalho (KIM; REDDY, 2008) também usa o Intervalo de Confiança para encontrar valores de *threshold*. Embora com dados e em contexto diferente do usado neste trabalho, os autores consideraram o intervalo $\pm 4\sigma$ adequado para a detecção de anomalias.

O sistema usa como estimativa do desvio padrão do ruído o desvio padrão dos coeficientes *wavelet*. O método direto para estimação do desvio padrão $\widehat{\sigma}$ dos coeficientes *wavelet* $d_{j,k}$ em cada nível j é calculado como segue:

$$\widehat{\sigma} = \sqrt{\frac{1}{N/2^j} \sum_{k=0}^{N/2^j-1} (d_{j,k} - \mu)^2} \, , \tag{4.14}$$

sendo $N/2^j$ o número de coeficientes *wavelet* $d_{j,k}$ no nível j e μ a média dos coeficientes $\mu = \frac{1}{N/2^j} \sum_{k=0}^{N/2^j-1} d_{j,k}$ no nível j.

O método para a estimativa do desvio padrão do ruído baseado no MAD (*Median absolute deviation*), proposto por (DONOHO; JOHNSTONE, 1995), não é usado aqui porque este exige a ordenação do vetor de entrada, o que tornaria o processo computacionalmente ineficiente para uma análise em tempo real.

4.2.6 Detecção das anomalias

Considerando que as anomalias de rede estão representadas nos coeficientes *wavelet* e que o *threshold* consegue identificar variações significativas em relação ao comportamento padrão do sinal, os coeficientes maiores que o valor de corte (*threshold*) são considerados indicadores de anomalias. O sistema considera qualquer coeficiente acima do *threshold* λ em qualquer nível da transformada como uma anomalia (função (4.15)). A indicação da ocorrência de uma anomalia é enviada ao módulo de resposta e geração de alarmes.

$$Alarme^{\lambda}(d_{j,0}) = \begin{cases} 1, & \text{se } |d_{j,0}| > \lambda \\ 0, & \text{se } |d_{j,0}| \leq \lambda \, . \end{cases} \, . \tag{4.15}$$

A função (4.15) define como o coeficiente *wavelet* $d_{j,0}$ é avaliado para a detecção de uma anomalia. Caso o coeficiente avaliado seja maior que o valor do *threshold* a função retorna o valor 1 indicando uma anomalia, caso contrário a função retorna 0.

Como mostrado na Figura 4.5 o primeiro coeficiente $d_{j,0}$ de cada nível j é usado para a detecção de anomalias. Por exemplo, para o nível $j = 1$ o coeficiente $d_{1,0}$ é testado conforme a função (4.15). Pelo fato da janela de detecção ser deslizante (expressão 4.2) apenas o primeiro coeficiente $d_{j,0}$ ($k = 0$) para cada nível j precisa ser comparado conforme a função (4.15). Como a janela y é atualizada a cada nova observação (Figura 4.3), o que gera novamente o calculo da TWD e das outras etapas do algoritmo de de detecção, os $d_{j,0}$ para cada nível j correspondem aos detalhes mais recentes do vetor da janela de detecção y e dos dados de entrada.

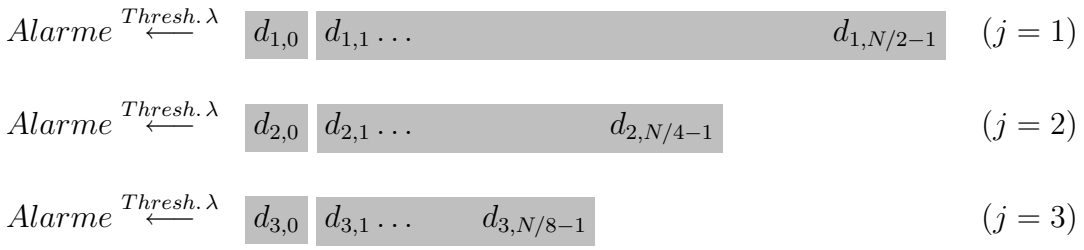

Figura 4.5: Exemplo detecção de anomalias.

A Figura 4.6 exemplifica o processo de detecção em um nível de detalhes da transformada *wavelet*.

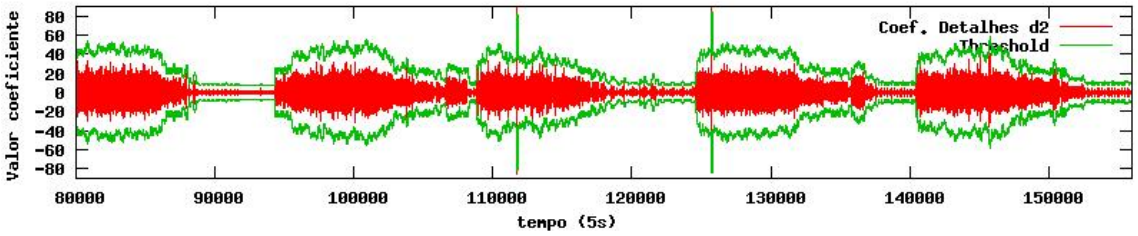

Figura 4.6: Coeficientes *wavelet*, d_2, com respectivos valores de *threshold*.

Na Figura 4.6 estão representados apenas os coeficientes *wavelet* (detalhes) do nível d_2 da transformada *wavelet* do tráfego de rede. Devido a Janela de Detecção ser deslizante (expressão (4.3)), apenas os último coeficiente *wavelet* de cada nível após a transformada e normalização é usado para detecção ou geração de alarme (função (4.15)) comparando-se com o valor do *Threshold* gerado conforme os coeficientes da janela toda (expressão (4.13)). Dessa forma, foram plotados no gráfico (Figura 4.6) apenas os coeficientes do nível d_2, usados em cada execução completa do processo de detecção (Algoritmo 4.1) para cada nova amostra ordenada no tempo, juntamente com o *Threshold* λ correspondente. Para a geração de alarmes, os coeficientes maiores que o valor do *Threshold* em cada nível de detalhes indicam uma anomalia. Todos os coeficientes *wavelet* (detalhes) são usados (embora não representados na Figura 4.6) para teste e a ocorrência de uma anomalia em qualquer um dos níveis de detalhes é suficiente para a geração de um alarme correspondente a respectiva posição em relação ao tempo.

O processo completo para a detecção de anomalias é descrito no Algoritmo 4.1.

Algoritmo 4.1: Algoritmo do mecanismo de detecção de anomalias de rede.

Entrada: Nova observação y_0
Saída: Valor do alarme

1 Atualize a janela de observação: $\overbrace{y_0, y_1, y_2, y_3, \ldots, y_{N-1}}^{Janela}, y_N, \ldots, y_{M-1}$, obtendo-se o vetor $y = (y_0, y_1, y_3, ..., y_{N-1})$
2 Calcule a TWD Direta do vetor y, obtendo-se os coeficientes *wavelet*: $w = ((c_{J,k})_{k=0}^{N/2^j-1}, ((d_{j,l})_{k=0}^{N/2^j-1})_{j=J}^{1}$
`// Normalização os coeficientes`
3 **para** $j \leftarrow J$ **até** 1 **faça**
4 **para** $k \leftarrow 0$ **até** $N/2^j - 1$ **faça**
5 $d_{j,k} = sgn(d_{j,k})\sqrt{|d_{j,k}|}$
6 **fim**
7 **fim**
8 Calcule o desvio padrão dos coeficientes *wavelets*: $\widehat{\sigma} = \sqrt{\frac{1}{N/2^j}\sum_{l=0}^{N/2^j-1}(d_{j,k} - \mu)^2}$
9 Calcule o valor do *Threshold*: $\lambda = \widehat{\mu} + C\widehat{\sigma}$
`// Geração de alarmes`
10 $alarme_t \leftarrow 0$
11 **para cada** *nível* j **faça**
12 **se** $|d_{j,0}| > \lambda$ **então**
13 $alarme_t \leftarrow alarme_t + 1$
14 **fim**
15 **fim**
16 Retorne $alarme_t$

O Algoritmo 4.1 descreve os passos que mecanismo de detecção proposto: atualização da janela de observação (linha 1); Transformada *Wavelet* Discreta direta do vetor da janela (linha 2); normalização dos coeficientes *wavelet* (linha 3); cálculo do desvio padrão dos coeficientes *wavelet* (linha 8); e geração dos alarmes (linha 11). O Algoritmo é executado toda vez que uma nova observação é disponibilizada. A coleta de amostras de um descritor de rede é de responsabilidade do módulo de Coleta do SDIR, que deve realizar esta tarefa de acordo com um intervalo de amostragem Δt. Para uma detecção em tempo real é necessário que o tempo de processamento do algoritmo seja menor que o intervalo amostragem usado pelo módulo de Coleta. Ao final do processamento do algoritmo é retornado um valor para o alarme. O valor zero indica que não foi detectado nenhuma anomalia. A indicação de um alarme, normalmente de forma visual, é de responsabilidade do módulo de Resposta de um SDIR. O tempo de processamento do algoritmo é analisado no Capítulo 6.

4.3 Trabalhos relacionados e considerações finais

Nesta Seção é feito um comparativo entre o mecanismo de detecção proposto neste trabalho com algumas abordagens vistas em alguns trabalhos relacionados (GAO et al., 2006) (DAINOTTI; PESCAPE; VENTRE, 2006) (KIM; REDDY, 2008) e (LU; TAVALLAEE; GHORBANI, 2008). Embora as abordagens dos trabalhos relacionados difiram quanto aos alvos da detecção, a localização e a forma de obtenção dos dados de entrada, o objetivo desta Seção é analisar apenas os algoritmos de detecção de anomalias baseados em *wavelets* usados por tais trabalhos.

No trabalho em (DAINOTTI; PESCAPE; VENTRE, 2006) foi proposto um mecanismo de detecção de anomalias de volume de tráfego de rede com o objetivo de detectar ataques do tipo DoS. O sistema combina uma abordagem tradicional, baseado em Somas Cumulativas (CUSUM - *CUmulative SUM*) (BASSEVILLE; NIKIFOROV, 1993, p. 35) e Médias Móveis Exponencialmente Ponderadas (EWMA - *Exponentially Weighted Moving Average*) com uma nova abordagem baseada na Transformada *Wavelet* Continua (TWC) e *Threshold*. A arquitetura é baseada em dois estágios. O primeiro estágio usa EWMA e *Thresholds* e destina-se a a fazer a detecção "grosseira" de ataques. O segundo estágio, usa a TWC, destina-se a refinação e detecção "fina" dos ataques, para diminuir o número de falsos alertas. A *Wavelet* Mãe usada foi a *Morlet*. Para o trabalho dessa dissertação usa-se a Transformada *Wavelet* Discreta, em um único módulo de detecção, em oposição a Transformada *Wavelet* Contínua (TWC). A TWD é mais adequada para tratar de dados já discretizados, como é o caso deste trabalho que faz uso de dados gerados pela amostragem de contadores de rede (Seção 2.3.1). Além disso, a TWD é computacionalmente eficiente e gera coeficientes sem redundância (T.LACHMAN et al., 2010).

No trabalho em (GAO et al., 2006) foi proposto um detector de anomalias de rede baseado na Transformada *Wavelet Packet* (TWP) (COIFMAN; WICKERHAUSER, 1992). Os dados de rede são transformados utilizando-se a transformada direta *wavelet packet*, com bases *wavelet* da família Daubechies, e reconstruído a partir dos coeficientes *wavelet* para cada nível da transformada. Medidas estatísticas, como média e variância, foram usadas para caracterizar uma anomalia, como a razão da média ou da variância entre a janela de detecção e a janela histórica foram mensuradas e comparadas com valores de *threshold* predefinidos para identificar uma anomalia. A abordagem adotada no trabalho dessa dissertação diferencia-se daquela adotada em (GAO et al., 2006), porque neste trabalho faz-se uso da Transformada *Wavelet* Discreta (TWD) tradicional. Neste trabalho usa-se a TWD direta sem a necessidade da TWD inversa. A TWP possui com-

plexidade computacional maior que a Transformada *Wavelet* Discreta tradicional (MALLAT, 1998). A complexidade da TWP é de ordem *loglinear*, $\mathcal{O}(N \log N)$ (COIFMAN; WICKERHAUSER, 1992), enquanto que a TWD tradicional possui complexidade linear $\mathcal{O}(N)$ (MALLAT, 1998). Este trabalho faz uso da Transformada *Wavelet* Discreta direta com funções ortonormais de Daubechies.

No trabalho em (LU; TAVALLAEE; GHORBANI, 2008) foi usada uma abordagem para detecção de anomalias de rede baseada na Transformada *Wavelet* e séries auto-regressivas. No sistema proposto foram selecionadas variáveis descritoras de tráfego, usando-se o modelo de agregação por fluxos origem-destino. O sinal original é transformado usando *wavelets* (Transformada *Wavelet* discreta) e os coeficientes *wavelet* $d_{j,k}$ aproximados usando um modelo de predição auto-regressivo do tipo ARX (*AutoRegressive with eXogenous input*) e o resíduo da predição é usado para a detecção de anomalias utilizado o GMM (*Gaussian Mixture Model*). A estratégia de detecção de anomalias consiste na identificação de *outliers* (valor significativamente diferente dos demais), assumindo-se, que a presença destes no resíduo indica a existência de anomalias no tráfego da rede. A abordagem adotada no trabalho desta dissertação difere daquela adotada em (LU; TAVALLAEE; GHORBANI, 2008), pois usa-se os coeficientes *wavelet* diretamente sem a necessidade de outra etapa de processamento.

Considerando-se apenas o mecanismo de detecção de anomalias, em (KIM; REDDY, 2008) foi aplicada a Transformada *Wavelet* discreta no sinal de entrada e reconstruído (Transformada *wavelet* Inversa) para cada nível da transformada. Ou seja, com os coeficientes *wavelet* $d_{j,k}$, para cada nível j, é reconstruído o sinal considerando-se apenas estes coeficientes e desconsiderando-se os demais. Dessa forma, o sinal reconstruído para cada nível representa a contribuição daquele nível no sinal original. A partir do sinal reconstruído para cada nível é feita a análise e identificação de anomalias. Diferentemente da abordagem usada em (KIM; REDDY, 2008), neste trabalho os coeficientes *wavelet* (detalhes) são usados diretamente para a detecção de anomalias, sem usar a transformada *wavelet* inversa. Assumindo que as singularidades são devido a anomalias de rede, para os propósitos deste trabalho, considera-se que a análise direta dos coeficientes *wavelet* é adequada para determinar a ocorrência de anomalias. No trabalho de (KIM; REDDY, 2008) a transformada inversa é executada para cada nível dos detalhes. A Transformada *Wavelet* discreta inversa possui a mesma complexidade computacional da transformada direta (MALLAT, 1998), então eliminando-se a transformada inversa, reduz-se a complexidade computacional do método. Portanto, o mecanismo de detecção proposto aqui, é capaz de detectar anomalias que geram mudança significativa na magnitude dos coefici-

entes *wavelet*, sem utilizar a transformada inversa.

A abordagem adotada nessa dissertação baseia-se na Transformada *Wavelet* Discreta, como em (LU; TAVALLAEE; GHORBANI, 2008) e (KIM; REDDY, 2008). Porém na abordagem de (LU; TAVALLAEE; GHORBANI, 2008) os coeficientes *wavelet* precisam, em um segundo estágio, ser modelados usando séries temporais do tipo ARX para a detecção de anomalias. Enquanto que no trabalho dessa dissertação os coeficientes *Wavelet* são usados diretamente para a detecção de anomalias, na abordagem de (KIM; REDDY, 2008) os coeficientes precisam ser reconstruídos (TWD inversa) para cada nível da transformada. Os trabalhos em (LU; TAVALLAEE; GHORBANI, 2008) e (KIM; REDDY, 2008) também diferem quanto em relação as variáveis descritivas de tráfego. Quanto aos objetivos da detecção de anomalias, em termos de ataques, e quanto a forma de coleta dos dados, esse trabalho assemelha-se mais ao trabalho em (GAO et al., 2006), onde foram usadas variáveis descritivas de tráfego baseadas em tempo. Finalmente, este trabalho difere dos demais simplificando, em termos de complexidade do algoritmo, o método de análise, o que potencializa seu uso para detecções em tempo real.

5 DESENVOLVIMENTO DO DETECTOR DE INTRUSÕES DE REDE BASEADO EM *WAVELETS*- DIBW

Com o propósito de avaliar o mecanismo de detecção de anomalias de rede proposto, foi implementado um protótipo denominado Detector de Intrusões de rede baseado em *Wavelets*- DIbW[1]. O DIbW foi desenvolvido na forma de um *framework*, de modo a ser expansível e configurável.

Neste capítulo são apresentados detalhes do desenvolvimento e a implementação do DIbW. O objetivo é apresentar de forma geral alguns aspectos e decisões de projeto no desenvolvimento do DIbW. A descrição do desenvolvimento do mecanismo é relevante na discussão a respeito do seu desempenho computacional.

Na Seção 5.1 é apresentado o ambiente computacional usado no desenvolvimento e implementação do protótipo.

Na Seção 5.2 é descrito como o *framework* foi desenvolvido, seus requisitos e parâmetros.

5.1 Ambiente de desenvolvimento

O DIbW foi desenvolvido na linguagem de programação Java[2] (JAVA, 2010) SDK (*Software Development Kit*) versão 1.6 usando o IDE (*Integrated Development Environment*) NetBeans[3] (NETBEANS, 2010) versão 6.0. Para a geração dos gráficos foi usado o software GNUPLOT[4] (GNUPLOT, 2010), versão 4.2.6. Todas as ferramentas usadas estavam disponíveis sob licença de *software* livre (GPL ou compatíveis).

[1]Para o desenvolvimento deste trabalho foram usadas as instalações do CRS/INPE MCT (Centro Regional Sul Instituto Nacional de Pesquisas Espaciais do Ministério da Ciência e Tecnologia) por meio de uma parceria com o GMicro da UFSM.

[2]http://java.sun.com/

[3]http://www.netbeans.org/

[4]http://www.gnuplot.info/

5.2 *Framework* para detecção de anomalias de rede

O DIbW foi desenvolvido na forma de um *framework* de modo a ser flexível e expansível. O *framework* implementa o módulo de Análise (Figura 4.2) conforme o mecanismo de detecção de anomalias proposto na Seção 4.2 do Capítulo 4. O módulo de Coleta deve ser construído pensando-se na fonte dos dados. Neste trabalho, afim de validar da abordagem de detecção de anomalias, o módulo de Coleta foi adaptado para trabalhar com dados de tráfego de rede disponíveis conforme será tratado no Capítulo 6. O módulo de Resposta foi construído com a funcionalidade básica de disponibilizar visualmente e salvar os *logs* das anomalias ocorridas. A reatividade quando da detecção de uma anomalia de rede não faz parte do escopo do trabalho.

O DIbW foi desenvolvido considerando-se os seguintes requisitos: ser genérico, configurável, ágil e eficiente computacionalmente. A generalidade do DIbW está no fato de ser possível o uso de qualquer descritor de tráfego de rede. A entrada deve estar na forma de amostras de descritores (contadores) de tráfego de rede, referente a características primárias ou derivadas, coletados e disponibilizados em intervalos de tempo pré-definidos.

O DIbW foi desenvolvido para ser configurável a fim de permitir ajustes no mecanismo de detecção conforme a necessidade. Durante a instanciação do *framework* são definidos os seguintes parâmetros: tamanho da janela de observação, base *wavelet*, função de normalização dos coeficientes *wavelet*, função para estimativa do desvio padrão e função para o cálculo do *threshold*.

Pensando-se na agilidade na detecção de anomalias, no desenvolvimento do DIbW procurou-se diminuir o tempo de resposta a um evento, usando-se as informações disponíveis no momento, restringindo o tempo necessário para confirmação de uma anomalia. O tempo máximo de reação depende do intervalo de amostragem Δt dos dados de entrada, definido no módulo de Coleta, e do tempo de análise dos dados. O intervalo de tempo necessário para o processamento (análise) dos dados da série de entrada depende da eficiência computacional dos algoritmos implementados, e será discutida no Capítulo 6.

A Figura 5.1 apresenta o diagrama de classes do DIbW.

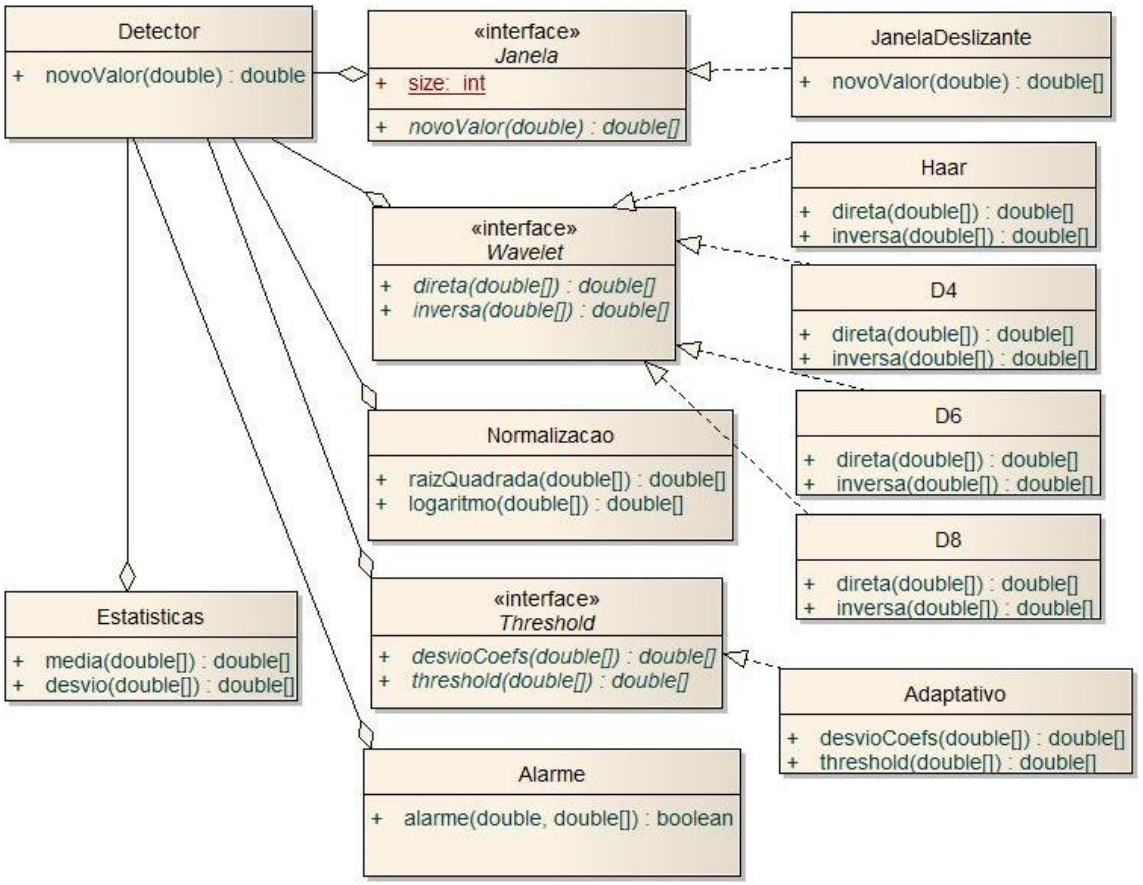

Figura 5.1: Diagrama de Classes do Detector de Intrusões de rede baseado em *Wavelets*-DIbW.

A classe principal do *framework* é *Detector* (Figura 5.1), que deve ser inicializada passando-se como argumento os parâmetros de configuração do sistema. A classe *Detector* possui o método *novoValor* que é invocado toda vez que uma nova observação é disponibilizada. Por meio deste método são invocados métodos de outras classes e todo o processamento necessário é executado e é retornado um valor para o alarme, caso uma anomalia seja detectada.

Cada passo no mecanismo de detecção de anomalias (Figura 4.2) foi implementado em um classe separada. A classe que implementa o mecanismo de atualização da Janela de Observação é usada por meio da interface *Janela*. A interface *Wavelet* permite o uso de uma das classes que implementam a Transformada *Wavelet* Discreta conforme uma base de Daubechies específica. A Classe *Normalizacao* é acessada após a TWD para a normalização dos coeficientes *wavelets* (detalhes). A interface *Threshold* permite, por meio da classe que implementa os seus métodos, o cálculo do valor do *threshold*. O valor do *Threshold* e os coeficientes *wavelet* são usados pela classe *Alarme* para determinar, caso seja detectado, a ocorrência de uma anomalia. As classes ou interfaces *Janela*, *Wa-*

velet , *Normalizacao*, *Threshold* e *Alarme* são tratadas em subseções separadas. A classe *Estatística* é usada para a geração de alguns indicadores estatísticos para análise mas não está envolvida diretamente no mecanismo de detecção.

5.2.1 Janela de Observação

No *framework* é definida a interface *Janela* (Figura 5.1) que especifica o comportamento da Janela de Observação. A interface *Janela* possui como atributo o tamanho N da janela, que deve ser definido na inicialização, e a operação *novoValor* que recebe como argumento um valor numérico (nova observação) e permite a leitura do vetor completo da janela. Na classe *JanelaDeslizante* é implementada a Janela de Observação deslizante (sobreposta) de tamanho fixo. O método de atualização da Janela de Observação implementado como uma fila simples, onde o valor lido (mais recente) é adicionado na fila enquanto que o último elemento (mais antigo) é descartado (Figura 4.3).

O mecanismo da Janela de Observação deslizante permite que apenas as últimas (mais recentes) N amostras sejam usadas e as mais antigas sejam descartadas da análise. Como discutida na Seção 4.2.2, esta postura tem o objetivo de permitir o uso apenas das amostras mais relevantes e evitar a sobrecarga computacional pelo processamento de dados muito antigos.

5.2.2 Transformada *Wavelet* Discreta

Para uso do DIbW foram implementados os algorítimos das transformadas *wavelet* discreta direta e inversa. Foram implementadas as *wavelets* ortonormais da família Daubechies: Daubechies 2 (D2 ou Haar), Daubechies 4 (D4), Daubechies 6 (D6) e Daubechies 8 (D8). Foi definida a interface *Wavelet* (Figura 5.1) que possui dois métodos, um para a Transformada *Wavelet* Direta e outro para a Transformada *Wavelet* Inversa, ambos recebem um vetor como argumento e retornam um vetor com os coeficientes conforme a operação. As diferentes *wavelets* são implementadas em classes separadas e processam a Transformada *Wavelet* conforme a Seção 3.1.5.

A base *wavelet* que será usada no processamento é definida como parâmetro durante a inicialização do DIbW.

5.2.3 Normalização dos coeficientes *wavelet*

Para normalização dos coeficientes *wavelet* (detalhes), usa-se a extração da raiz quadrada ou o logaritmo dos coeficientes *wavelet*, apresentadas na Seção 4.2.4, expressão (4.11) e expressão (4.12), respectivamente.

A classe *Normalização* é usada pela classe *Detector* para a normalização dos coeficientes *Wavelet* (detalhes). Na classe são implementados os métodos *raizQuadrada* para a função Raiz Quadrada e o método *logaritmo* para função Logarítmica.

Algoritmo 5.1: Algoritmo transformada Raiz Quadrada.

Entrada: w : double[N]
para $i \leftarrow 0$ **até** N **faça**
  se $w_i < 0$ **então**
    $w_i = -\sqrt{|w_i|}$
  fim
  senão
    $w_i = \sqrt{w_i}$
  fim
fim
Saída: w

Algoritmo 5.2: Algoritmo transformada Logarítmica.

Entrada: w : double[N]
para $i \leftarrow 0$ **até** N **faça**
  se $w_i < 0$ **então**
    $w_i = -\left(\ln(|w_i|)\right) + 1$
  fim
  senão
    $w_i = \ln(w_i) + 1$
  fim
fim
Saída: w

A função Raiz Quadrada é descrita no Algorítimo 5.1 e o Algoritmo 5.2 descreve a função Logarítmica. A escolha da função para a normalização, *raizQuadrada* ou *logaritmo*, é feita na inicialização.

5.2.4 *Threshold*

O cálculo do valor do *threshold* é feito em dois passos: o cálculo do valor do desvio padrão dos coeficientes *wavelet* e o cálculo do *threshold* propriamente dito.

Para encontrar o desvio padrão dos coeficientes o DIbW usa a fórmula (4.14). O valor do *threshold* é encontrado pela fórmula:

$$\lambda = \widehat{\mu} + C\widehat{\sigma} \ , \tag{5.1}$$

sendo C é uma constante correspondente ao intervalo desejado conforme a Tabela 4.1. O método é implementado em classe separada usando a interface *Threshold* (Figura 5.1).

5.2.5 Geração de alarmes

A geração de alarmes é responsabilidade da classe *Alarme* (Figura 5.1) e implementa o algoritmo da Figura 5.3.

Algoritmo 5.3: Algoritmo para geração de Alarmes.

Entrada: d : double[J]
para cada *nível* j **faça**
  se $|d_{j,N/2^j-1}| > \lambda$ **então**
    $alarme_t \leftarrow alarme_t + 1$
  fim
fim
Saída: $alarme_t$

O algoritmo percorre todos os níveis j dos detalhes transformada *wavelet* d_j e compara o último elemento, posição $N/2^j - 1$, de cada nível com o respectivo valor do *threshold*. Caso o valor do coeficiente testado seja maior, em módulo, que o *threshold* a variável indicadora de alarmes é acrescida de 1. No final o valor do alarme é retornado. Como todo o método é executado toda vez que uma nova amostra de uma variável de rede é disponibilizada, a saída do método corresponde ao nível do alarme naquele instante de tempo t. O valor do alarme é um número inteiro entre zero e o número máximo de níveis na transformada, $0 \leq alarme_t \leq J$. Um valor zero significa que nenhuma anomalia foi detectada, enquanto que um valor igual ou maior que 1 indica a ocorrência de anomalias em um ou mais níveis respectivamente.

5.3 Considerações Finais

Neste Capítulo foi descrito o desenvolvimento e implementação do Detector de Intrusões baseado em *Wavelets* (DIbW). O sistema foi desenvolvido na forma de um *framework* de modo a ser flexível, expansível e configurável. O *framework* DIbW implementa o módulo de análise de Sistema Detector de Intrusões de Rede baseado em anomalias e permite a integração com os módulos de Coleta e Resposta. O mecanismo de detecção de anomalias é baseado na Transformada *Wavelet* Discreta direta, normalização dos coeficientes *wavelet* e cálculo do *threshold*.

Durante a instanciação e inicialização do *framework* são definidos os parâmetros: tamanho da janela de observação, função *wavelet*, função de normalização dos coeficientes *wavelet*, função para estimativa do desvio padrão e função para o cálculo do *threshold*. O DIbW realiza as seguintes tarefas: cálculos das médias e o desvio padrão; da TWD de Daubechies; do valor do *Threshold*; detecção de anomalias; e geração de alarmes.

6 VALIDAÇÃO DA ABORDAGEM DE DETECÇÃO DE ANOMALIAS DE REDE

Neste capítulo são apresentados alguns experimentos usando o SDIR desenvolvido, o Detector de Intrusões de rede baseado em *Wavelets* (DIbW), com o objetivo de avaliar a abordagem de detecção de anomalias de rede proposta.

Para avaliar o desempenho desta proposta é importante a realização de testes de detecção com amostras de tráfego real. Procurou-se por amostras de tráfego de rede padrão e amostras com diferentes tipos de ataques. É desejável, também, que os ataques estejam junto com o tráfego padrão para que o sistema seja testado quanto à capacidade de diferenciar os ataques do tráfego padrão.

Na Seção 6.1 são descritos a base de dados usada como Fonte de Informação para o DIbW, os ataques de rede relevantes para os experimentos e a preparação dos dados e seleção das variáveis descritivas do tráfego de rede.

Na Seção 6.2 são apresentados alguns experimentos realizados com o objetivo de de definir alguns parâmetros de configuração do DIbW.

Na Seção 6.3 são apresentados os experimentos realizados com o objetivo de avaliar a capacidade de detecção de anomalias da abordagem proposta. Os experimentos estão agrupados conforme o protocolo de rede analisado.

Na Seção 6.4 são apresentados os experimentos realizados com o objetivo de avaliar o desempenho computacional do DIbW.

6.1 A base de dados de tráfego de rede

Para a realização de testes de detecção com o mecanismo proposto neste trabalho, utilizou-se a base de dados de tráfego de rede do MIT DARPA (*Massachusetts Institute of Technology* e *Defense Advanced Research Projects Agency*, respectivamente) (DARPA, 1999), conhecida como DARPA 99[1]. A base de dados do DARPA 99 possui informações

[1]Disponível em: http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1999data.html

de tráfego real, coletados em uma rede controlada, e com anomalias causadas por ataques conhecidos, gerados por *scripts*. Os ataques presentes na base são documentados, condição necessária para a contagem de erros e acertos da abordagem de detecção.

A base DARPA 99 recebeu algumas críticas (MAHONEY; CHAN, 2003), por possuir tráfego de uma rede fechada, porque os ataques foram gerados usando-se *scripts* e porque alguns ataques já estarem ultrapassados. No entanto, a base ainda tem sido amplamente usada para testar algorítimos de detecção de anomalias (ZHANG; HAN; REN, 2009) (CHENG; XIE; WANG, 2009) (ZHANG; GU, 2007) (LU; TAVALLAEE; GHORBANI, 2008) (XIA; XU, 2008) (HUANG; THAREJA; SHIN, 2006).

Segundo (BOLZONI, 2009) a DARPA 99 é a única base de tráfego de rede pública compreensível para testes de algorítimos de detecção de anomalias de rede. Porém o fato de já ser antiga dificulta a pesquisa de novos métodos de detecção. Apesar de alguns ataques presentes na DARPA 99 serem antigos, para este trabalho, justifica-se o uso desta base pelo fato do mecanismo de detecção de anomalias proposto ser genérico e não destinado a um tipo específico de ataque. Vale ressaltar também que a escolha da base de dados não influencia na avaliação de desempenho computacional da ferramenta, visto que os dados são transformados em séries de amostragens de contadores específicos da mesma forma independente da fonte de dados.

A base de dados DARPA 99 é composta por 5 semanas de tráfego de rede, contendo 5 dias cada semana. A base contém cerca de 9 Gb de dados coletados pela ferramenta TCPDUMP (TCPDUMP, 1998) na saída e na entrada do roteador da rede. Foram usados somente dados de saída e dados de entrada da rede do *Lincoln Laboratory* e *Air Force Research Laboratory*, capturados pelo programa *tcpdump*. Das 5 semanas disponibilizadas, as 3 primeiras são chamadas de dados de treinamento e contém seus ataques documentados, as semanas 4 e 5 são chamadas de dados de teste e seus ataques não encontram-se documentados.

A primeira e a terceira semanas da fase de treinamento possuem um tráfego normal de rede, ou seja, não possuem nenhum tipo de ataque registrado neste período. A tabela 6.1 possui uma lista com os ataques que ocorreram na segunda semana da fase de treinamento. Esta tabela possui o identificador do ataque, a data, tempo inicial, endereço de origem do ataque, e nome do ataque.

6.1.1 Seleção dos dados para os experimentos

A base de dados do DARPA possui 5 semanas de tráfego, cada semana com apenas cinco dias, sendo os dados coletados na saída e entrada do roteador das 8 horas da manhã

Tabela 6.1: Lista com ataques DARPA. Fonte: (DARPA, 1999).

ID	$Data$	$Tempo$	$Origem$	$Nome$
1	08/03/1999	08:01:01	hume.eyrie.af.mil	NTinfoscan
2	08/03/1999	08:50:15	zeno.eyrie.af.mil	pod
3	08/03/1999	09:39:16	marx.eyrie.af.mil	back
4	08/03/1999	12:09:18	pascal.eyrie.af.mil	httptunnel
5	08/03/1999	15:57:15	pascal.eyrie.af.mil	land
6	08/03/1999	17:27:13	marx.eyrie.af.mil	secret
7	08/03/1999	19:09:17	pascal.eyrie.af.mil	ps attack
8	09/03/1999	08:44:17	marx.eyrie.af.mil	portsweep
9	09/03/1999	09:43:51	pascal.eyrie.af.mil	eject
10	09/03/1999	10:06:43	marx.eyrie.af.mil	back
11	09/03/1999	10:54:19	zeno.eyrie.af.mil	loadmodule
12	09/03/1999	11:49:13	pascal.eyrie.af.mil	secret
13	09/03/1999	14:25:16	pascal.eyrie.af.mil	mailbomb
14	09/03/1999	13:05:10	172.016.112.001-114.254	ipsweep
15	09/03/1999	16:11:15	marx.eyrie.af.mil	phf
16	09/03/1999	18:06:17	pascal.eyrie.af.mil	httptunnel
17	10/03/1999	12:02:13	marx.eyrie.af.mil	satan
18	10/03/1999	13:44:18	pascal.eyrie.af.mil	mailbomb
19	10/03/1999	15:25:18	marx.eyrie.af.mil	perl (Failed)
20	10/03/1999	20:17:10	172.016.112.001-114.254	ipsweep
21	10/03/1999	23:23:00	pascal.eyrie.af.mil	eject (console)
22	10/03/1999	23:56:14	hume.eyrie.af.mil	crashiis
23	11/03/1999	08:04:17	hume.eyrie.af.mil	crashiis
24	11/03/1999	09:33:17	marx.eyrie.af.mil	satan
25	11/03/1999	10:50:11	marx.eyrie.af.mil	portsweep
26	11/03/1999	11:04:16	pigeon.eyrie.af.mil	neptune
27	11/03/1999	12:57:13	marx.eyrie.af.mil	secret
28	11/03/1999	14:25:17	marx.eyrie.af.mil	perl
29	11/03/1999	15:47:15	pascal.eyrie.af.mil	land
30	11/03/1999	16:36:10	172.016.112.001-254	ipsweep
31	11/03/1999	19:16:18	pascal.eyrie.af.mil	ftp-write
32	12/03/1999	08:07:17	marx.eyrie.af.mil	phf
33	12/03/1999	08:10:40	marx.eyrie.af.mil	perl (console)
34	12/03/1999	08:16:46	pascal.eyrie.af.mil	ps (console)
35	12/03/1999	09:18:15	duck.eyrie.af.mil	pod
36	12/03/1999	11:20:15	marx.eyrie.af.mil	neptune
37	03/12/1999	12:40:12	hume.eyrie.af.mil	crashiis
38	03/12/1999	13:12:17	zeno.eyrie.af.mil	loadmodule
39	03/12/1999	14:06:17	marx.eyrie.af.mil	perl (Failed)
40	03/12/1999	14:24:18	pascal.eyrie.af.mil	ps
41	03/12/1999	15:24:16	pascal.eyrie.af.mil	eject
42	03/12/1999	17:13:10	pascal.eyrie.af.mil	portsweep
43	03/12/1999	17:43:18	pascal.eyrie.af.mil	ftp-write

de um dia até as 6 horas do outro dia. A primeira e terceira semanas possuem tráfego de rede padrão sem ataques. A segunda semana de tráfego apresenta ataques de rede, sendo eles identificados na documentação da base de dados, como mostra a Tabela 6.1. A

quarta e quinta semanas não são usadas neste trabalho, porque possuem muitos ataques em intervalos de tempo muito curto entre si e pouco tráfego normal, o que não corresponde a um ambiente real e prejudica a adaptação do algoritmo de detecção ao padrão de tráfego real.

A base DARPA 99 contém todos os pacotes coletados de forma completa, incluindo *headers* e *payload*. Para este trabalho, porém são consideradas apenas as informações contidas nos *headers* dos pacotes, abstendo-se de extrair qualquer informação do *payload* do pacote. Embora esta restrição possa dificultar a detecção de algumas formas de ataques, por exemplo ataques que exploram características específicas dos protocolos HTTP (*Hypertext Transfer Protocol*) (FIELDING et al., 1999), SMTP (*Simple Mail Transfer Protocol*) (KLENSIN, 2008) ou FTP (*File Transfer Protocol*) (POSTEL; REYNOLDS, 1985), garante-se a proteção dos dados e a confidencialidade da informação.

Estão presentes na segunda semana 43 ataques das mais diferentes formas, como ataques a servidores *web*, ataques de negação de serviço e escaneamento de portas. A maioria dos ataques presentes na base exploram vulnerabilidades conhecidas, algumas já solucionadas. Alguns desses ataques não são perceptíveis analisando-se os descritores primários de tráfego. Para os propósitos deste trabalho, foram considerados apenas ataques que podem ser identificados usando somente as informações presentes nos *headers* dos pacotes, sem inspecionar o *payload*, e que geram alguma alteração no volume ou na forma do tráfego de rede, considerando-se o tráfego total, o protocolo TCP, o protocolo UDP e o protocolo ICMP.

Os ataques de rede selecionados consistem dos seguintes tipos:

- *MailBomb*: ataque de negação de serviço, quando tem-se um grande envio de mensagens para entregar, com o intuito de travar ou limitar o funcionamento normal de um servidor.

- *Neptune*: ataque *SYN Flood* para negação de um serviço em uma ou mais portas.

- *Crashiis*: ataque em que é enviado uma *url* muito grande para um servidor Microsoft IIS derrubando-o.

- *PoD*: *denial of service Ping of Death*, são enviados *pings* (pacotes ICMP) malformados para um computador.

- *Satan*: ataque que visa identificar vulnerabilidades no sistema.

- *Portsweep*: faz uma varredura de portas para determinar os serviços rodando em um computador.

O ataques selecionados enquadram-se na categoria de ataques de negação de serviço: *MailBomb*, *Neptune*, *Crashiis*, *PoD*; ou na categoria de escaneamento: *Satan*, *Portsweep*. Ataques da categoria Negação de Serviço (DoS - *Denial of Service*) (PENG; LECKIE; RAMAMOHANARAO, 2007) consistem na tentativa de dificultar o acesso legítimo a um serviço. Geralmente, esses ataques exploram vulnerabilidades dos protocolos de comunicação com o objetivo de desabilitar a capacidade de resposta da vítima. O escaneamento de portas (*portscan*) envolve um *host* remoto escaneando portas TCP na máquina da vítima em busca de serviços vulneráveis.

No ataque *mailbomb* um grande número de mensagens de e-mail é enviada para um servidor por meio de um *host* comprometido, conectado pela porta SMTP (*Simple Mail Transfer Protocol*) do servidor diretamente. Este ataque pode resultar em milhares de mensagens não desejadas para uma conta de algum usuário. Um ataque *mailbomb* típico envia cerca de 10 MB de *emails* não desejados (HUANG; THAREJA; SHIN, 2006).

O ataque *Neptune*, também conhecido como Ataque TCP SYN *flood*, explora a implementação do protocolo TCP/IP. Quando um servidor recebe uma mensagem SYN é reservado recursos, conexão meio aberta, para atender esta requisição e uma mensagem SYN-ACK é retornada ao cliente. O cliente, então, recebe a mensagem SYN-ACK e responde enviando uma mensagem ACK para o servidor. Quando o servidor recebe a mensagem ACK a conexão é estabelecida completamente e os dois computadores podem começar a transmitir informações. No entanto, a tabela que o servidor usa para manter as conexões meio abertas possui tamanho finito e pode ser explorada pelo atacante. Quando o servidor recebe muitos pedidos de conexões, mensagens SYN, a tabela de conexões meio abertas é sobrecarregada e o servidor não consegue estabelecer novas conexões enquanto a tabela estiver cheia. Normalmente há *timeouts* associados a cada conexão meio aberta da tabela, contudo, se o atacante mantiver esta tabela constantemente cheia, o ataque é bem sucedido (HUANG; THAREJA; SHIN, 2006).

Crashiis (RED, 1998) é um ataque em que é enviado uma *url* muito grande para um servidor Microsoft IIS derrubando-o. Embora a vulnerabilidade que permitia o ataque já tenha sido corrigida, o ataque está presente na base DARPA 99.

No ataque *PoD* (*ping of death*) (POD, 1998) são enviados *pings* malformados para um computador. Historicamente muitos sistemas não conseguiam processar pacotes *ping* maiores que 65,535 *bytes* e podiam parar de funcionar. Atualmente, no entanto esse

problema está solucionado.

SATAN (*Security Administrator Tool for Analyzing Networks*) (SATAN, 2010) é uma ferramenta usada para escanear vulnerabilidades em uma rede de computadores. Normalmente a ferramenta é usada por administradores de sistema, mas também é usada por atacantes.

O ataque *Portsweep* (DARPA, 1999) faz uma varredura de portas para determinar os serviços rodando em um computador. Essa informação é útil para um atacante que está a procura de máquinas vulneráveis.

6.1.2 Preparação dos dados para os experimentos

Como os dados da base DARPA 99 estão no formato bruto foi preciso extrair as informações desejadas para uso nesse trabalho. As informações foram coletadas na forma de contadores por meio de amostragem de determinada variável (descritor) em um intervalo de amostragem pré-definido, formando uma série de contadores ordenados no tempo.

Foram selecionados os seguintes descritores: número total de pacotes e número de pacotes dos protocolos de rede, TCP, UDP e ICMP, todos extraídos na saída do servidor agrupados em intervalos de amostragem de 5 segundos. O tempo médio aproximado de um ataque, que foi observado na base de dados do DARPA, é de 10 segundos, então escolheu-se o intervalo de amostragem de 5 segundo de modo que o ataque fosse visível. Como os dados estavam no formato do TCPDUMP (TCPDUMP, 1998), para extração dos descritores, usou-se a ferramenta TCPSTAT (TCPSTAT, 1998), conforme o exemplo:

tcpstat -r outside.tcpdump -o "\%T\n"5 > w1-d1-out-5-tcp.data

Neste exemplo são gerados os contadores referente ao tráfego TCP com intervalo de 5 segundos para o primeiro dia da primeira semana. Após todos os contadores referentes a todos os dias das três primeiras semanas de tráfego são concatenados em um único arquivo. Após a geração conforme as variáveis selecionadas os contadores estavam prontos para o uso nos testes de detecção.

6.2 Definição da Função de Normalização

Nesta Seção são analisadas algumas características estatísticas dos coeficientes da transformada *wavelet* do tráfego de rede. O objetivo é determinar a função para a normalização dos coeficientes *wavelets* (raiz quadrada ou logaritmo) a ser usado no DIbW.

Inicialmente é preciso definir algumas medidas estatísticas que foram usadas: a média, o desvio padrão, a obliquidade e a curtose. O desvio padrão é definido como a raiz

quadrada do valor médio do quadrado da distância entre cada valor e a média. É calculado conforme a fórmula (4.14). A obliquidade (JOANES; GILL, 1998) é a medida da assimetria de uma determinada distribuição de probabilidade de uma variável aleatória. É definida conforme:

$$g_1 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^3}{(\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2)^{\frac{3}{2}}} \ . \tag{6.1}$$

Na fórmula (6.1 x_i representa cada coeficiente do conjunto de entrada e μ a sua média. Um valor negativo para a obliquidade g_1 indica que a distribuição tem uma cauda esquerda (valores abaixo da média) mais pesada. Um valor positivo indica que a distribuição tem uma cauda direita (valores acima da média) mais pesada. A obliquidade igual a zero indica distribuição de probabilidade aproximadamente simétrica.

A curtose (JOANES; GILL, 1998) mede o grau de achatamento de uma distribuição de probabilidade de uma variável aleatória, ou o quanto uma curva de frequência será achatada em relação a uma curva normal. É definida conforme:

$$g_2 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^4}{(\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2)^2} - 3 \ . \tag{6.2}$$

Se o valor da curtose g_2 for igual a zero, então tem o mesmo achatamento que a distribuição normal (mesocúrtica). Se o valor da curtose for menor do que zero então a distribuição é mais concentrada que a distribuição normal (leptocúrtica). Se o valor da curtose for maior que zero, então a função de distribuição é mais achatada que a distribuição normal (platicúrtica).

Usaram-se as medidas estatísticas definidas: média, desvio padrão, obliquidade e curtose, para avaliar as características estatísticas do tráfego de rede padrão (sem ataques). Para isso, escolheu-se a primeira a semana de tráfego padrão (sem ataques) do DARPA 99 para analisar as características estatísticas do tráfego de rede e dos coeficientes da Transformada *Wavelet*. Usou-se o tráfego de rede sem ataques para melhor analisar suas características e configurar o DIbW conforme o tráfego padrão. Os dados do tráfego de rede correspondentes ao número total de pacotes coletados em intervalos de 5 segundos, preparados conforme a Seção 6.1.2, foram transformados usando a *wavelet* Daubechies D8. O tráfego original e os coeficientes *wavelet* foram então analisados. Como observado na Figura 6.1, mesmo o tráfego de rede padrão apresenta alta variabilidade que é capturada pelos coeficientes *wavelet* (detalhes) em todos os níveis da transformada.

Na Figura 6.1 estão representados o tráfego de rede original e os coeficientes da transformada *wavelet*. O tráfego de rede original (A) consiste nos 5 primeiros dias da primeira

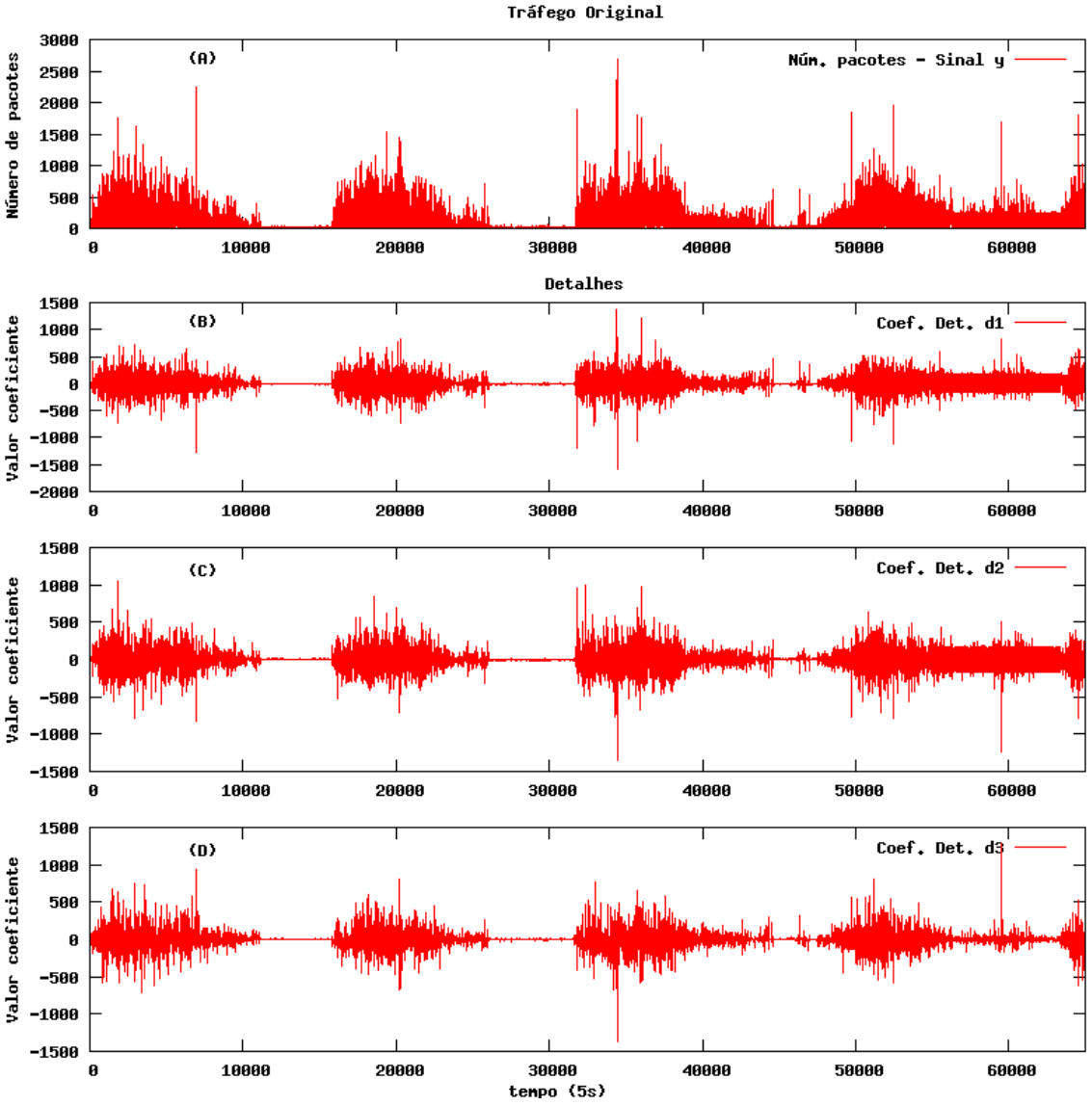

Figura 6.1: Transformada *wavelet* do tráfego de rede - Tráfego original (A) e coeficientes *wavelet* (detalhes), d_1 (B), d_2 (C), d_3 (D). O tráfego de rede apresenta alta variabilidade representada pelas curvas não suáveis, caracterizadas por picos, nos coeficientes *wavelet* (detalhes) em todos os níveis da transformada.

semana de dados da base da DARPA. A série de dados consiste na contagem do número total de pacotes na entrada do roteador da rede, amostrados a cada 5 segundos, totalizando 65536 pontos de amostragem (327680 segundos ou aproximadamente 91 horas). Na série de dados do tráfego de rede original (65536 pontos) foi aplicada a transformada *wavelet* discreta direta usando a base Daubechies D8. Na Figura 6.1 também estão representados os coeficientes da transformada *wavelet* (detalhes) referente aos 3 primeiros níveis, d_1 (B), d_2 (C), d_3 (D).

Como pode ser observado pela Figura 6.1 o tráfego original (A) apresenta como característica alta variabilidade, curva não suave constituída quase que exclusivamente por

Tabela 6.2: Estatísticas dos coeficientes *wavelet* do tráfego de rede padrão.

Coeficiente	**Média**	**Desvio Padrão**	**Obliquidade**	**Curtose**
y	94.98	141.86	2.79	14.24
d_1	-0.13	102.28	0.01	11.52
d_2	-1.44	118.30	0.01	6.84
d_3	-0.95	123.38	0.18	8.85
d_4	1.54	136.18	0.20	5.95
d_5	3.76	145.66	0.48	5.65
d_6	1.54	150.88	-0.21	5.2

picos. Também, os coeficientes *wavelet* (detalhes) apresentam alta variabilidade em todos os níveis da transformada.

Na Tabela 6.2 são apresentados algumas medidas estatísticas para a transformada *wavelet* do tráfego de rede padrão. A Tabela mostra que os coeficientes *wavelet* (detalhes) apresentam média muito pequena, próxima de zero, em comparação com o desvio padrão. A obliquidade próxima de zero para os coeficientes *wavelet* (detalhes) indica que os valores são distribuídos praticamente de forma simétrica em relação à média. A curtose acima de zero para o sinal original, y, e para os coeficientes *wavelet* (detalhes), d_1, d_2, d_3, d_4, d_5, d_6, indica a presença de vários valores altos (picos) em comparação com uma distribuição normal como referência.

Dos dados conclui-se que os coeficientes *wavelet* (detalhes) da transformada *wavelet* para o tráfego de rede padrão apresentam média próxima de zero e são simetricamente distribuídos em relação à média. Estas duas características são importantes para a definição de uma estratégia para a definição de margens (*Threshold*). Como a média é próxima de zero pode-se dispensar uma normalização em relação à média e o desvio padrão torna-se a principal medida estatística da amostra. Sabendo-se que os valores dos coeficientes são distribuídos praticamente de forma simétrica em relação à média a definição de margens (*threshold*) é simplificada, pois o mesmo valor em módulo pode ser utilizado para a margem inferior e a margem superior. Por outro lado a curtose acima de zero indica grande variabilidade com vários picos em relação a distribuição normal, típica de distribuições de probabilidade com cauda longa, o que dificulta no cálculo da margem.

Na Tabela 6.3 são apresentados as características dos coeficientes *wavelet* (detalhes) após a normalização usando Transformada Logarítmica e na Tabela 6.4 são apresentados as características dos coeficientes *wavelet* (detalhes) após a normalização usando Transformada Raiz Quadrada.

Os coeficientes *wavelet* após a aplicação da operação Raiz Quadrada possuem características mais próximas a uma distribuição de probabilidade normal, conforme observado

Tabela 6.3: Estatísticas dos coeficientes *wavelet* (detalhes) da transformada *wavelet* do tráfego de rede padrão após a Transformada Logarítmica.

Coeficiente	Média	Desvio Padrão	Obliquidade	Curtose
d_1	-0.12	3.40	0.08	-1.23
d_2	-0.13	3.71	0.07	-1.41
d_3	-0.10	3.67	0.05	-1.41
d_4	-0.04	3.76	0.04	-1.42
d_5	-0.03	3.81	0.04	-1.42
d_6	0.09	3.85	-0.04	-1.44

Tabela 6.4: Estatísticas dos coeficientes *wavelet* (detalhes) da transformada *wavelet* do tráfego de rede padrão após a Transformada Raiz Quadrada.

Coeficiente	Média	Desvio Padrão	Obliquidade	Curtose
d_1	-0.16	7.62	0.09	0.10
d_2	-0.23	8.60	0.07	-0.44
d_3	-0.17	8.52	0.06	-0.11
d_4	0.00	8.98	0.05	-0.16
d_5	0.04	9.30	0.12	-0.16
d_6	0.19	9.49	-0.05	-0.19

pelos indicadores da obliquidade e da curtose próximos de zero. A partir dessa constatação escolhe-se a função raiz quadrada como método a ser usado no mecanismo de detecção para a normalização dos coeficientes.

Para encontrar o valor do *Threshold* definiu-se empiricamente a constante $C = 4$ baseando-se na Tabela 4.1 da Seção 4.2.5. O tamanho da janela de observação será determinado conforme os testes de detecção.

6.3 Testes de Detecção

Os SDI baseados em assinaturas procuram por ataques correspondentes às assinaturas em sua base de dados, e por definição são capazes de detectar apenas ataques conhecidos. Os SDI baseados em anomalias, porém, são potencialmente capazes de detectar ataques desconhecidos. Para avaliar os algoritmos baseados em anomalias, entretanto, há uma impossibilidade de se gerar ataques desconhecidos. Dessa forma os algoritmos baseados em anomalias são usualmente testados com tipos de ataques conhecidos, sendo que o algoritmo de detecção não tem nenhum conhecimento prévio sobre o ataque (MAHONEY; CHAN, 2003).

Nesta Seção são descritos alguns experimentos com o DIbW realizados com o objetivo de testar a capacidade de detecção de anomalias do mecanismo de detecção. Devido a impossibilidade de se testar a capacidade de detecção de anomalias ou ataques desconhe-

cidos, testa-se a capacidade do sistema em detectar ataques conhecidos e documentados na base DARPA 99.

Na avaliação de desempenho do mecanismo proposto são considerados a quantidade de Verdadeiros Positivos (VP), Falsos Positivos (FP), Verdadeiros Negativos (VN) e Falsos Negativos (FN) gerados pelo sistema. Resumidamente, as possibilidades de classificação dos eventos gerados são apresentados na Tabela de Contingência ou Matriz de Confusão (QIN, 2005) (Tabela 6.5).

		Situação Real	
		positivo	negativo
Atribuído pelo SDI	positivo	VP	FP
	negativo	FN	VN

Tabela 6.5: Matriz de Confusão. Fonte: adaptado de (QIN, 2005)

Considerando-se o número total de Positivos como P = VP + FN (corresponde ao número total de ataques realmente presentes) e o número total de Negativos como N = VN + FP (corresponde ao número total de amostras sem ataques), tem-se:

Definição 6.3.1. A Taxa de Verdadeiros Positivos (TVP %) ou taxa de detecção é igual ao número de Verdadeiros Positivos divido pelo número de Positivos. TVP = (VP / P) * 100.

Definição 6.3.2. A Taxa de Falsos Positivos (TFP %) é igual ao número de Falsos Positivos dividido pelo número de Negativos. TFP = (FP / N) * 100.

Para a realização dos testes de detecção usaram-se os dados de tráfego de rede base DARPA 99 selecionados conforme a Seção 6.1.1. A partir dos dados brutos da base foram gerados contadores, conforme a Seção 6.1.2, considerando-se: o número total de pacotes, número de pacotes TCP, número de pacotes UDP e número de pacotes ICMP. Usou-se uma taxa de amostragem de 5 segundos ($\Delta t = 5\ s$), também usada nos trabalhos em (DAINOTTI; PESCAPE; VENTRE, 2006) e (LU; TAVALLAEE; GHORBANI, 2008) que fazem a extração dos descritores (contadores) de rede de forma semelhante.

O objetivo dos testes é avaliar o comportamento do DIbW na análise de tráfego de rede na presença de ataques. O sistema deve se adaptar aos dados analisados e gerar um alarme quando uma anomalia for encontrada. Como cada tipo de ataque possui características distintas, perceptíveis usando-se descritores de tráfego específicos, e a presença de uma anomalia não caracteriza necessariamente um ataque, cada alarme gerado pelo sistema precisa ser analisado separadamente.

Durante os testes, a primeira, segunda e terceira semanas de tráfego foram agrupadas sequencialmente, porém apenas a segunda semana de tráfego, que possui ataques documentados, foi avaliada. Os contadores estão na forma de uma série de amostras organizadas sequencialmente conforme o *timestamp* (tempo) e armazenados em um arquivo. Para os testes o sistema lê os contadores sequencialmente do arquivo correspondente ao descritor desejado e submete ao mecanismo de análise. Após o processamento e geração de alarmes pelo mecanismo, os alarmes foram avaliados usando-se a documentação da base.

6.3.1 Estudo de caso 1 - Tráfego IP

Para este experimento foram usados os contadores correspondentes ao número total de pacotes IP (*Internet Protocol*) (SOCOLOFSKY; KALE, 1991) coletados com intervalo de amostragem de 5 segundos, referentes a segunda semana de tráfego da base DARPA 99. As *wavelets* usadas foram a Daubechies D8, D4 e D2. Para a normalização dos coeficientes foi usada a função Raiz Quadrada, pois esta se mostrou mais adequada, do ponto de vista estatístico conforme a Seção 6.2. Para o cálculo do valor do *threshold* usou-se o desvio padrão dos coeficientes *wavelet* (detalhes) e a constante $C = 4$ conforme a Tabela 4.1.

A Figura 6.2 (A) apresenta o sinal formado a partir do descritor de tráfego de rede correspondente ao total de pacotes IP trafegados a cada 5 segundos. No segundo gráfico (B) estão representados os alarmes gerados pelo DIbW usando uma janela de detecção de tamanho 128 e *wavelet* D8.

Nos gráficos (Figura 6.2) (A) e (B) os dados estão ordenados no tempo, cada posição equivale a uma amostra de aproximadamente 77000 amostras. Em relação aos dados originais cada amostra equivale a 5 segundos. No gráfico dos alarmes (B) estão representados os alarmes gerados pelo sistema, sendo que cada alarme equivale a uma anomalia de tráfego detectada. Dessa forma o gráfico (A) representa a entrada do mecanismo de detecção de anomalias e o gráfico (B) a saída após o processamento. No gráfico do sinal original (A) duas anomalias (neste caso picos com grande intensidade) são visíveis, as demais não são identificáveis apenas pelo gráfico. Cada alarme gerado pelo sistema foi verificado com a documentação da base para contagem de erros e acertos.

A Figura 6.3 representa os três primeiros níveis de coeficientes *wavelet* (detalhes), d_1 (A), d_2 (B) e d_3 (C), após a normalização usando a função Raiz Quadrada, com os respectivos valores de *threshold*. Os gráficos (A) (B) e (C) referem-se aos dados de entrada usados pela função de geração de alarmes (Algoritmo 4.1) e o gráfico (D) representa a

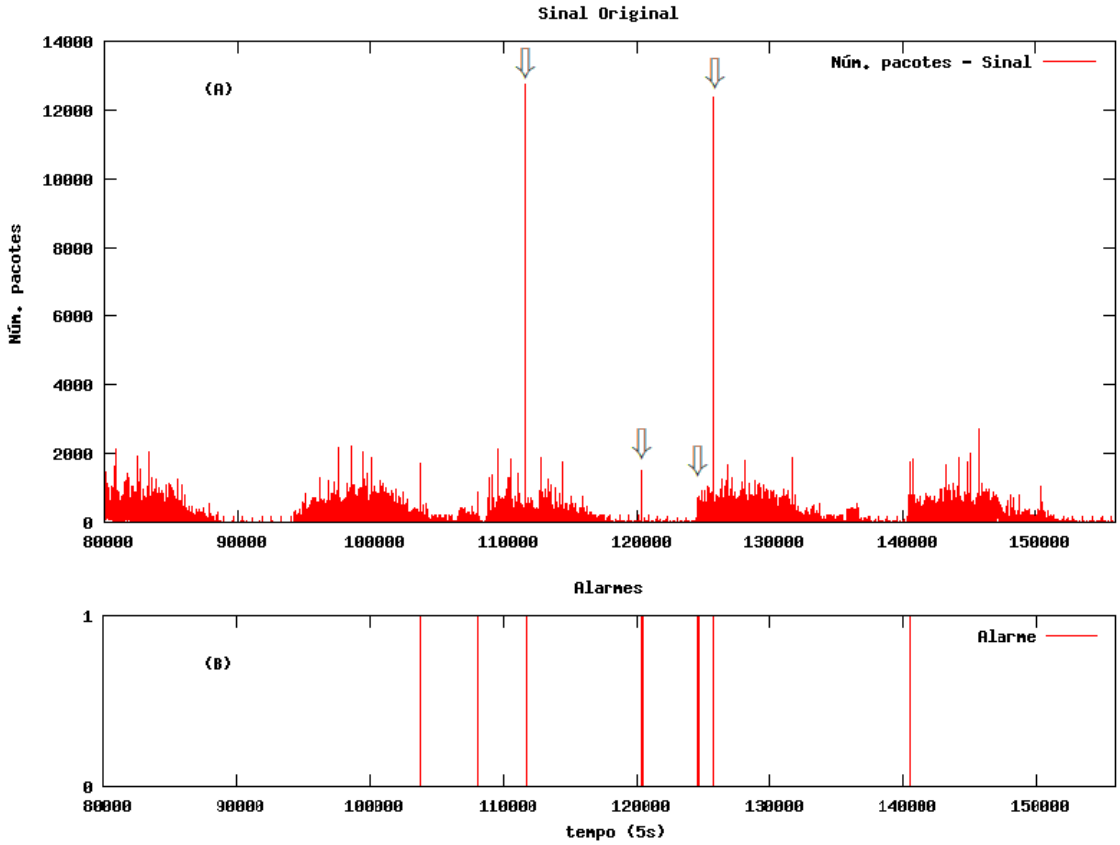

Figura 6.2: Tráfego de rede (A), corresponde ao total de pacotes IP capturados a cada 5 segundos, e os alarmes (B) gerados pelo DIbW. As setas (A) indicam a localização os ataques.

saída, ou seja, os alarmes gerados.

Os coeficientes *wavelet* (Figura 6.3) (A) (B) (C) após a normalização usando a função Raiz Quadrada descrevem curvas mais suaves. A função para cálculo do *threshold* para cada nível consegue se adaptar à curva dos coeficientes *wavelet*, de modo que apenas coeficientes anômalos ultrapassem em valor o *threshold*. Caso o valor de determinado coeficiente *wavelet* em qualquer nível ultrapasse o respectivo valor do *threshold*, um alarme é gerado pelo sistema para aquela posição em relação ao tempo. Para o cálculo do *threshold* a constante $C = 4$ foi escolhida empiricamente, porém baseando-se na Tabela 4.1.

Em relação ao tráfego original (Figura 6.2) (A), variações abruptas do sinal foram detectadas nos primeiros níveis de detalhes da transformada (Figura 6.4) (B), enquanto que variações mais suaves, mas ainda anômalas foram detectadas nos níveis maiores (níveis mais grosseiros) (Figura 6.6) (C).

Na Figura 6.4 (A) está representada uma porção do tráfego de rede (número de pacotes) contendo um ataque do tipo *satan*. O ataque gerou uma alteração abrupta no número de pacotes trafegados. A alteração (anomalia) gerada pelo ataque foi capturada pelos co-

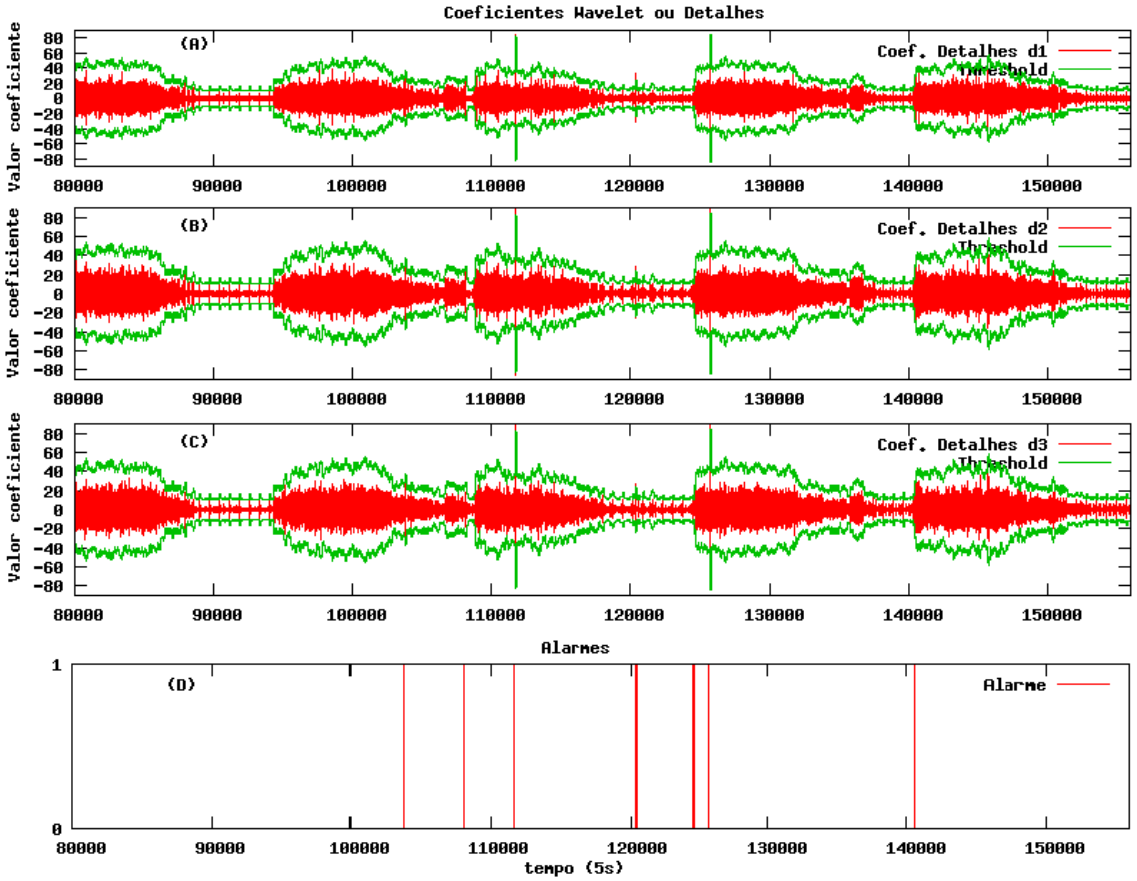

Figura 6.3: Coeficientes *wavelet* (detalhes) d_1, d_2 e d_3 e os respectivos valores de *Threshold*.

eficientes *wavelet* (detalhes) no primeiro nível d_1 (B). Em (B) os coeficientes *wavelet* do nível d_1 foram normalizados usando a raiz quadrada. Como o valor dos coeficientes ultrapassaram o valor do *threshold* um alarme foi gerado para aquela posição (D). O valor do *threshold* adapta-se conforme a variação dos coeficientes *wavelet* (B).

Na Figura 6.5 (A) está representado outro ataque do tipo *satan*. Neste caso a anomalia gerada pelo ataque foi detectada no primeiro e segundo níveis de coeficientes *wavelet* (B) (C), gerando alarmes (D). Para fins de avaliação apenas um alarme foi considerado pois referem-se ao mesmo evento.

Na Figura 6.6 (A) está representado um ataque do tipo *crashiis*. Este tipo de ataque gerou uma alteração de tráfego menos acentuada e mais suave quanto um ataque do tipo *satan*. O ataque foi corretamente detectado no segundo nível *wavelet* d_2 (C).

Falsos positivos ocorreram em posições nos dados de entrada em que houve variação brusca no tráfego e o sistema gerou um alarme (Figura 6.7). O alarme gerado indica a ocorrência de uma anomalia nos dados, porém não havia ataque documentado na base para aquela posição, o que se caracteriza um falso positivo. Para a variável analisada,

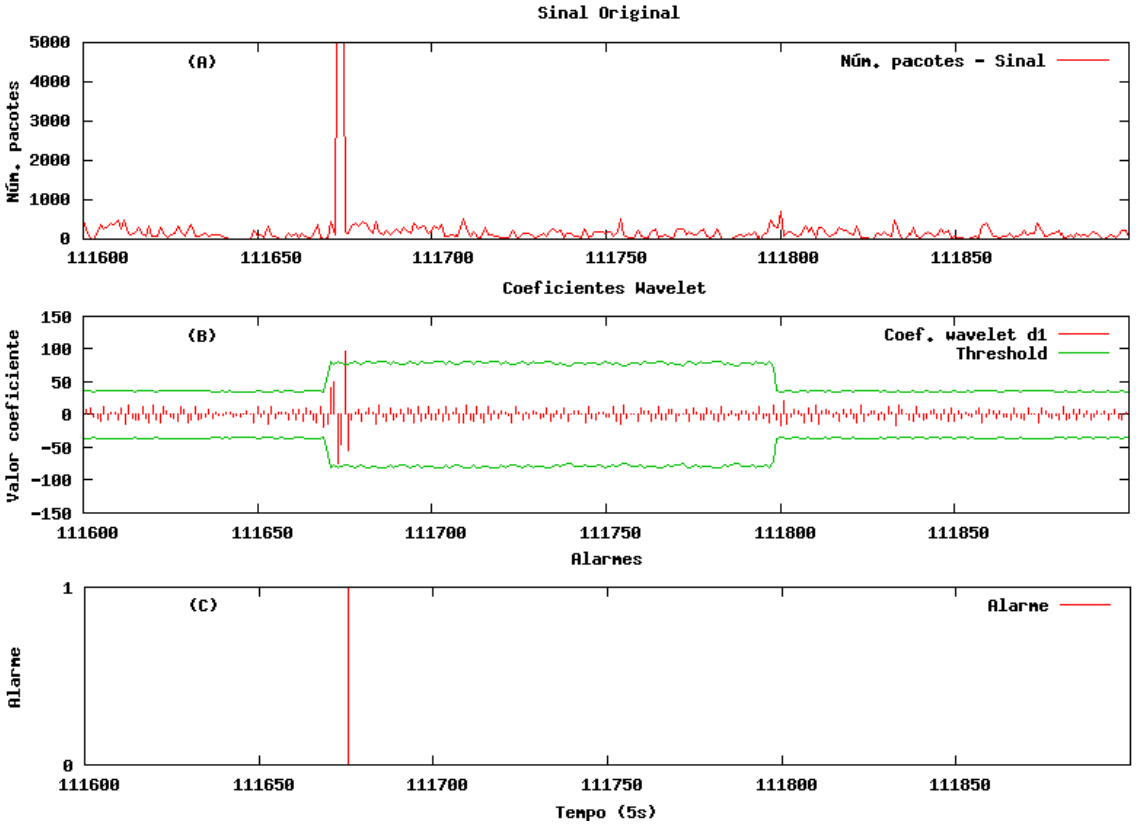

Figura 6.4: Ataque do tipo *Satan* (A) detectado no primeiro nível d_1 (B) dos coeficientes *wavelet*, gerando um alarme (C).

neste caso o tráfego IP total, não percebeu-se diferença na forma da curva gerada por um ataque ou por um falso positivo, apenas na intensidade. O padrão da curva do tráfego de um modo geral é bastante variável e com vários picos e variações. Vale ressaltar que o sistema procura por anomalias, então quando um alarme é gerado significa que há uma anomalia no padrão dos dados, porém nem sempre está relacionado a um ataque, sendo muitas vezes variações normais do tráfego.

Na Figura 6.7 está representada uma porção de tráfego de rede onde o sistema detectou uma anomalia, e consequentemente gerou um alarme. Porém como não há nenhum ataque associado ao evento, trata-se de um falso positivo. Comparando-se visualmente um falso positivo (Figura 6.7) (A) assemelha-se a um ataque de média intensidade (Figura 6.5) (A).

Na Tabela 6.6 estão os resultados da detecção de ataques pelo DIbW na base do DARPA 99 usando o tráfego total na segunda semana. Em todos os testes o número total de amostras foi de 77077 pontos. Foram usadas as *wavelets* D8, D4 e D2, com tamanhos de janela de 64, 128 e 256 pontos. Alarmes consecutivos foram considerados, para a avaliação, como um único alarme.

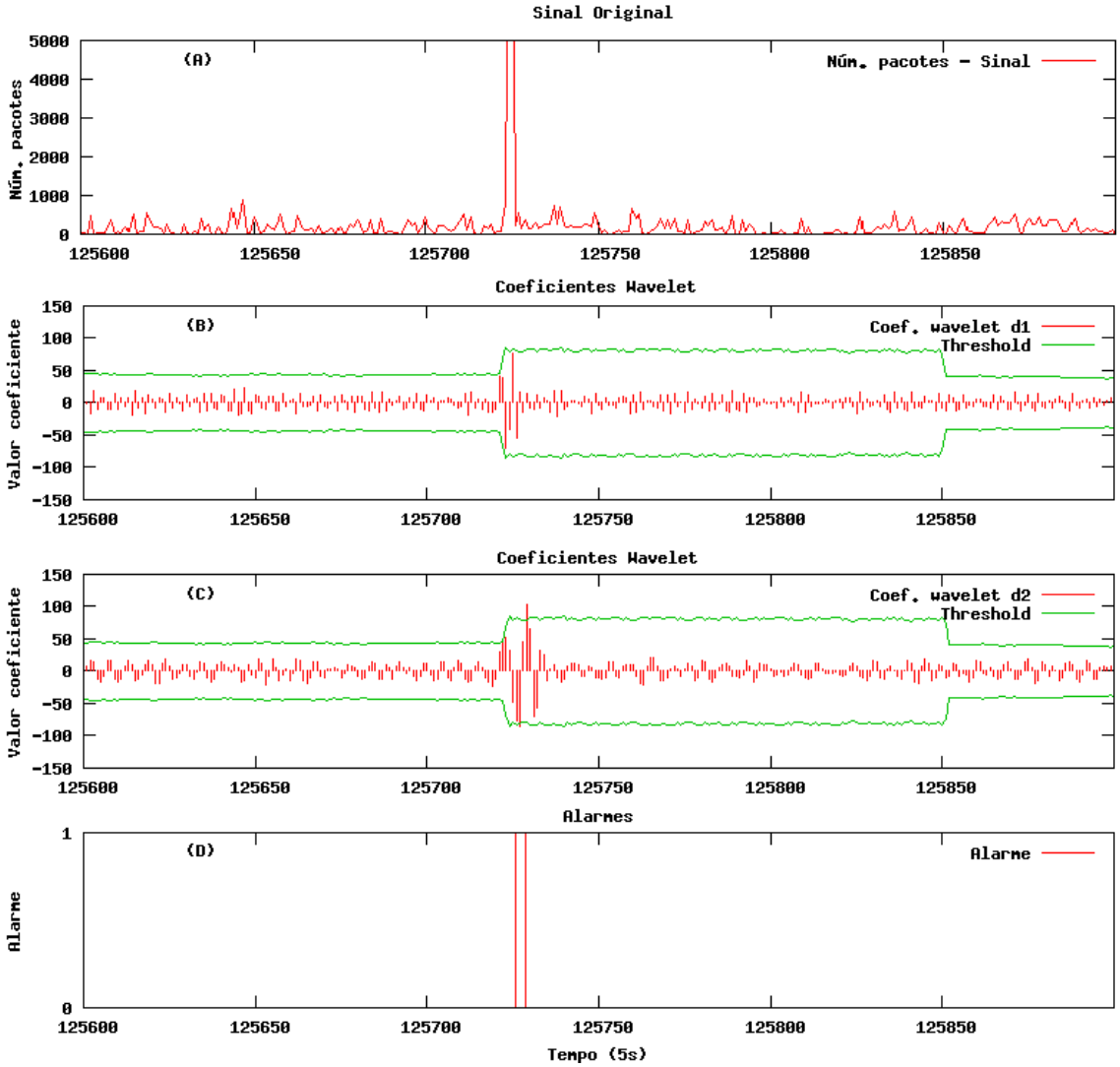

Figura 6.5: Ataque do tipo *satan* (A) detectado no primeiro e segundo níveis, d_1 (B) e d_2 (C) dos coeficientes *wavelet*, gerando alarmes (D).

Tabela 6.6: Resultados da análise de todos os pacotes do tráfego de rede.

Wavelet	**Tam. Jan.**	**Verdadeiros P.**	**Falsos P.**	**Ataques detectados**
D8	64	2	0	satan
D8	128	4	3	satan, crashiis
D8	256	4	5	satan, crashiis
D4	64	2	3	satan
D4	128	4	3	satan, crashiis
D4	256	4	7	satan, crashiis
D2	64	2	3	satan
D2	128	4	3	satan, crashiis
D2	256	4	7	satan, crashiis

O Tamanho da Janela de Detecção influenciou nos resultados em relação ao número de detecções e falsos positivos. Usando-se 128 pontos como tamanho da janela de detecção

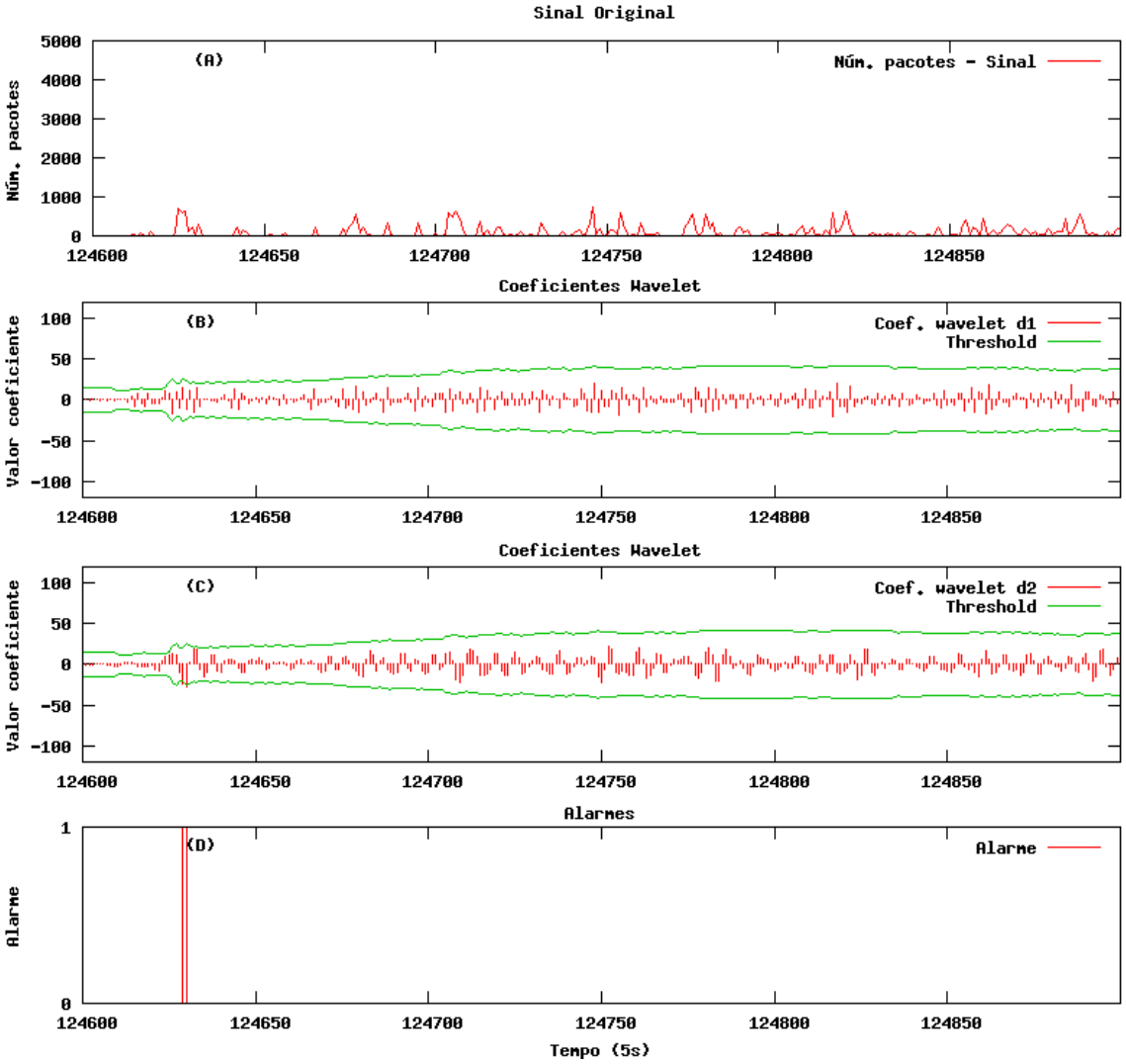

Figura 6.6: Ataque do tipo *crashiis* (A) detectado no segundo nível d_2 (C) dos coeficientes *wavelet*, gerando dois alarmes consecutivos (D).

obteve-se o melhor resultado considerando o número de detecções e o menor número de falsos positivos. Quando usou-se uma janela de detecção de tamanho igual a 64 pontos o sistema identificou menos ataques devido à quantidade reduzida de dados na janela para análise. Por outro lado quando usou-se uma janela de detecção de tamanho igual a 256 pontos o sistema apresentou uma quantidade maior de falsos positivos. A quantidade maior de falsos positivos deve-se a demora maior do sistema em se adaptar às oscilações normais do tráfego de rede.

Usando-se as funções *wavelet* D2, D4 e D8 não se percebeu variação significativa quanto ao número de detecções, apenas uma pequena variação quanto ao número de falsos positivos. O melhor caso foi observado usando-se a *wavelet* D8 e uma janela de detecção de tamanho 128 (Tabela 6.6).

Embora a quantidade de amostras com ataques seja pequena, o sistema apresentou um

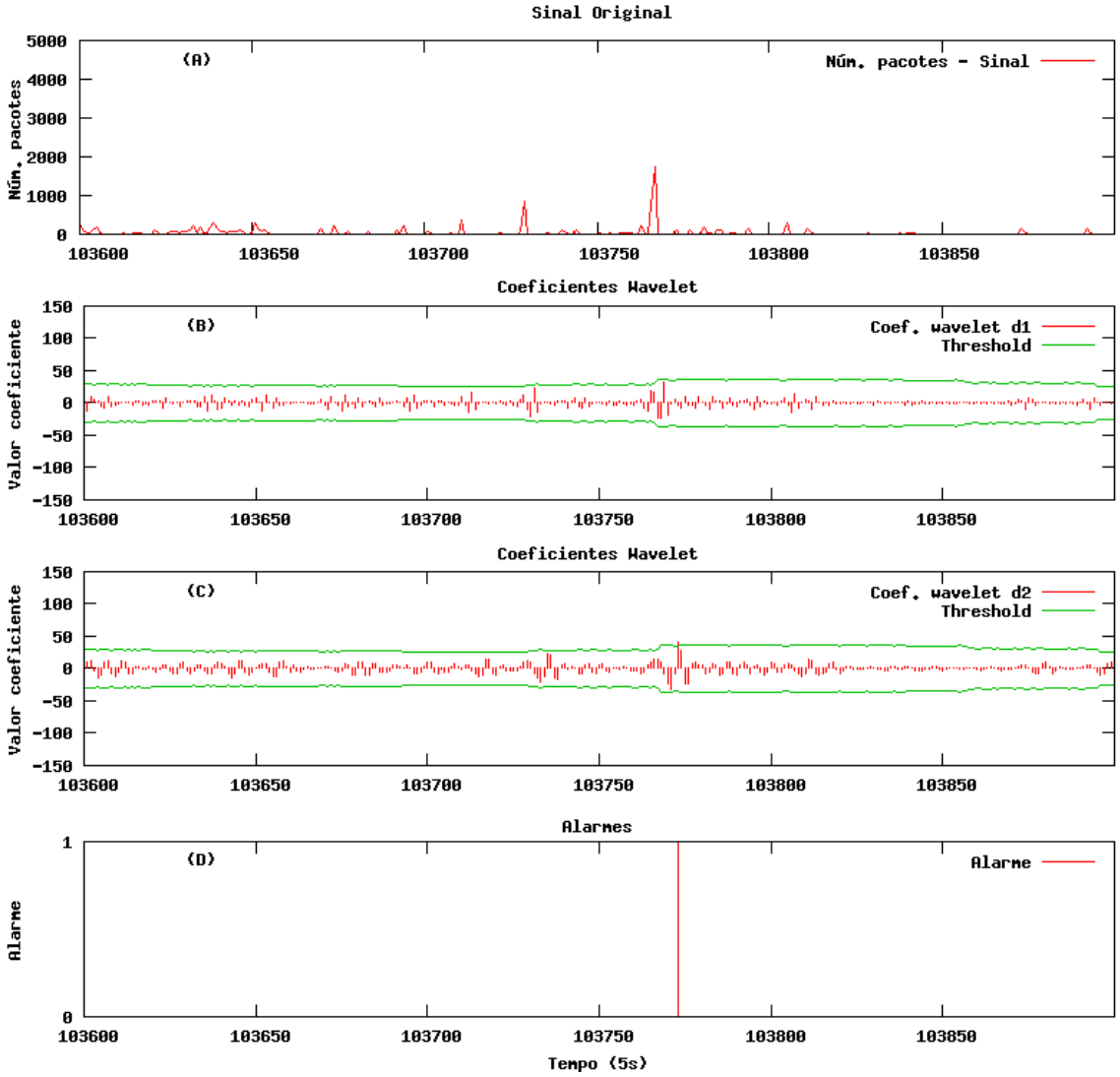

Figura 6.7: Falso Positivo, oscilação normal do tráfego que porém gerou um alarme.

desempenho satisfatório. No melhor caso (Tabela 6.7), com uma de tamanho 128 e função *wavelet* D8, o DIbW identificou corretamente 4 ataques, dos tipos: *satan* e *crashiis*. Neste caso apenas 3 falsos positivos foram gerados pelo sistema.

Tabela 6.7: Ataques detectados usando o tráfego total, janela de tamanho 128 e *wavelet* D8.

Ataque	**Total Ataques**	**T. Ataques Detectados**
satan	2	2
crashiis	3	2

Foram detectados corretamente pelo sistema 2 ataques do tipo *satan*, de 2 presentes, ou seja 100% de acerto para este ataque. SATAN (*Security Administrator Tool for Analyzing Networks*) (SATAN, 2010) é uma ferramenta usada para escanear vulnerabilidades em uma rede de computadores. No tráfego da base o ataque *satan* gerou picos com inten-

sidade maior de tráfego. Nos experimentos o ataque ficou visível nos primeiros níveis de detalhes da transformada *wavelet* do tráfego.

Crashiis (RED, 1998) é um ataque em que é enviado uma *url* muito grande para um servidor Microsoft IIS derrubando-o. Na base de dados o ataque *crashiis* causou uma pequena variação no volume de tráfego. O problema na identificação deste ataque deve-se justamente a essa variação de pequena intensidade que ele causa no tráfego. O sistema detectou 2 ataques do tipo *crashiis*, dos 3 presentes. O primeiro foi identificado no período noturno e de pouco tráfego total. O segundo foi detectado no início do tráfego diurno onde ocorreu uma elevação pequena, porém brusca, do volume de pacotes. O terceiro ataque não foi detectado pois estava em um período com oscilações normais do tráfego, o que mascarou o ataque.

Os falsos positivos foram gerados em períodos em que ocorreram alterações de média intensidade no padrão do tráfego. Embora tratam-se de anomalias de tráfego, como não estão associadas a nenhum ataque documentado foram consideradas como falsos positivos. Visualmente essas anomalias são indistinguíveis de ataques de média intensidade.

Embora o sistema identificou poucos ataques usando o tráfego total, o número de falsos positivos gerados foi baixo, apenas 3 em mais de 77000 amostras. Por outro lado, para tipos específicos de ataques o sistema identificou corretamente quase todos os ataques, ou seja, 2 do tipo *satan*, de 2 presentes, e 2 ataques do tipo *crashiis*, de 3 presentes (Tabela 6.7). O baixo número de falsos positivos deve-se ao fato da transformada *wavelet* e do *threshold* usados ajustarem-se adequadamente ao padrão do tráfego de rede.

6.3.2 Estudo de caso 2 - Tráfego TCP

Neste experimento foram gerados, a partir da base DARPA 99, contadores apenas para o tráfego de rede correspondente ao protocolo TCP (*Transmission Control Protocol*) (POSTEL, 1981a). Usaram-se os mesmos parâmetros do experimento com o tráfego IP (Seção 6.3.1), intervalo de amostragem de 5 segundos, base *wavelet* Daubechies D8, normalização dos coeficientes *wavelet* usando a Raiz Quadrada e $C = 4$. Foram usados tamanhos de janela de detecção de 64, 128 e 256 pontos.

Na Figura 6.8 está representado o sinal formado a partir do descritor de tráfego de rede correspondente aos pacotes TCP trafegados a cada 5 segundos. No segundo gráfico está representado os alarmes gerados pelo DIbW.

Na Tabela 6.8 estão os resultados da detecção de ataques pelo DIbW na base do DARPA 99 usando o tráfego TCP.

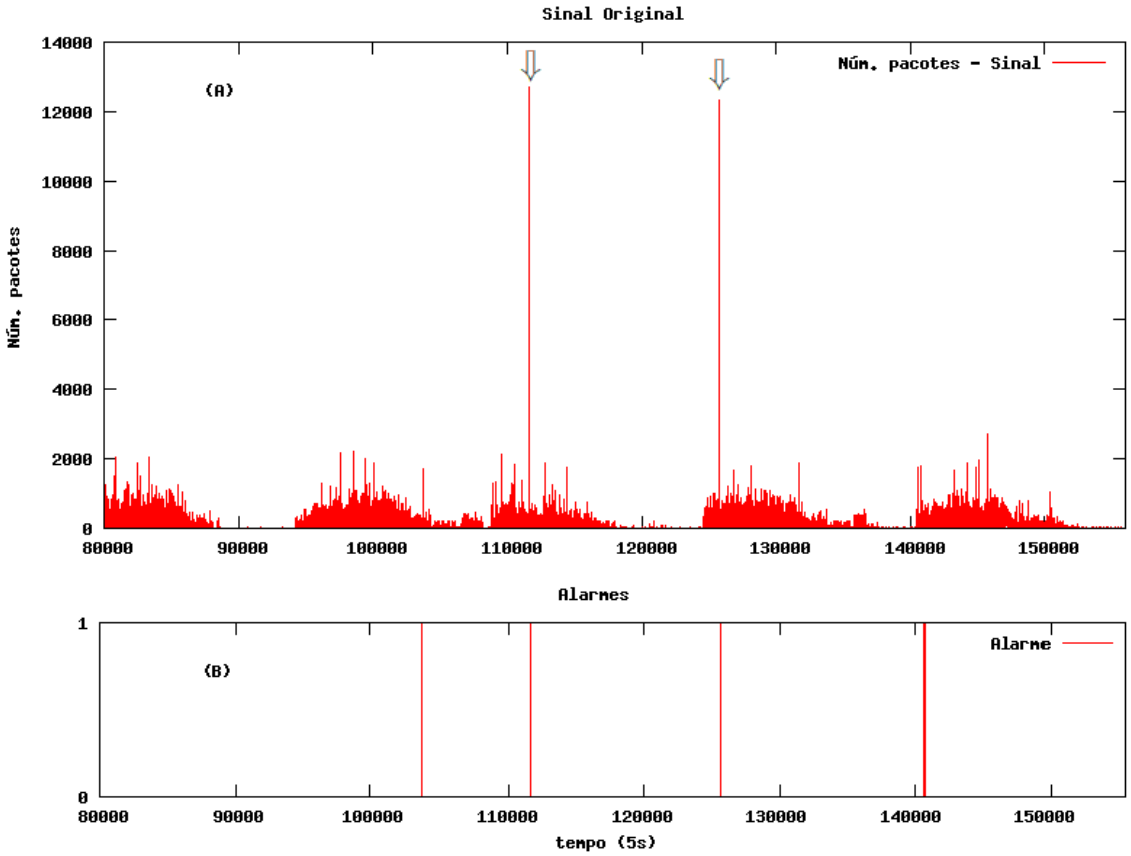

Figura 6.8: Tráfego de rede correspondente aos pacotes do protocolo TCP (A) capturados a cada 5 segundos e os alarmes gerados pelo DIBW (B).

Tabela 6.8: Resultados da análise dos os pacotes TCP do tráfego de rede.

Tam. Jan.	**Verdadeiros P.**	**Falsos P.**	**Ataques detectados**
64	2	1	satan
128	2	2	satan
256	2	9	satan

Tabela 6.9: Ataques detectados usando o tráfego do protocolo TCP, janela de tamanho 128 e *wavelet* D8.

Ataque	**Total Ataques**	**T. Ataques Detectados**
satan	2	2

Quando considerou-se apenas os pacotes do protocolo TCP, com a janela de detecção de tamanho 128, o sistema foi capaz de detectar 2 ataques presentes (Tabela 6.8). Por outro lado, neste caso foi gerado apenas 2 falsos positivos pelo sistema.

6.3.3 Estudo de caso 3 - Tráfego UDP

Neste experimento foram gerados contadores apenas para o tráfego de rede correspondente ao protocolo UDP (*User Datagram Protocol*) (POSTEL, 1980). Usaram-se os mesmos parâmetros do experimento com o tráfego IP (Seção 6.3.1), intervalo de amostragem de 5 segundos, base *wavelet* Daubechies D8, normalização dos coeficientes *wavelet* usando a Raiz Quadrada e $C = 4$. O tamanho para a janela de detecção usado foi de 128 pontos.

Na Figura 6.9 (A) está representado o sinal formado a partir do descritor de tráfego de rede correspondente aos pacotes UDP trafegados a cada 5 segundos. No segundo gráfico (B) estão representados os alarmes gerados pelo DIbW.

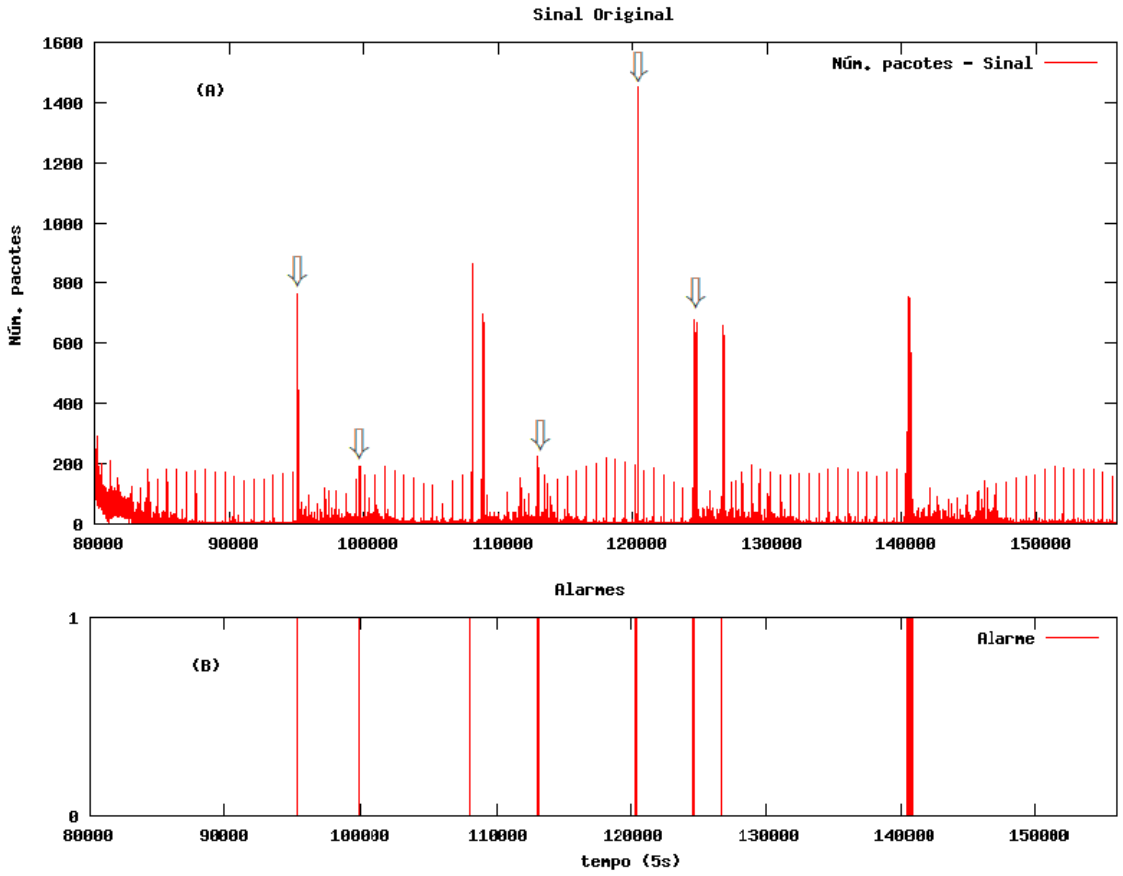

Figura 6.9: Tráfego de rede correspondente aos pacotes do protocolo UDP capturados a cada 5 segundos (A) e os alarmes gerados pelo DIBW (B).

Na Tabela 6.10 estão os resultados da detecção de ataques pelo DIbW na base do DARPA 99 usando o tráfego UDP.

Tabela 6.10: Resultados da análise dos os pacotes UDP do tráfego de rede.

Tam. Jan.	Verdadeiros P.	Falsos P.	Ataques detectados
64	4	2	portsweep, mailbomb, crashiis
128	6	3	portsweep, mailbomb, crashiis
256	6	5	portsweep, mailbomb, crashiis

Tabela 6.11: Ataques detectados usando o tráfego do protocolo UDP, janela de tamanho 128 e *wavelet* D8.

Ataque	Total Ataques	T. Ataques Detectados
portsweep	2	1
crashiis	3	2
mailbomb	2	2

Considerando-se apenas os pacotes do protocolo UDP, com a janela de detecção de tamanho 128, o sistema detectou todos os ataques do tipo *portsweep* e *mailbomb* e quase todos os ataques do tipo *crashiis* (Tabela 6.10). O número de falsos positivos gerados foi pequeno.

6.3.4 Estudo de caso 4 - Tráfego ICMP

Neste experimento foram gerados contadores apenas para o tráfego de rede correspondente ao protocolo ICMP (*Internet Control Message Protocol*) (POSTEL, 1981b). Usaram-se os mesmos parâmetros do experimento com o tráfego IP (Seção 6.3.1), intervalo de amostragem de 5 segundos, base *wavelet* Daubechies D8, normalização dos coeficientes *wavelet* usando a Raiz Quadrada e $C = 4$. O tamanho para a janela de detecção usado foi de 128 pontos.

Na Figura 6.10 (A) está representado o sinal formado a partir do descritor de tráfego de rede correspondente aos pacotes ICMP trafegados a cada 5 segundos. No gráfico (B) estão representados os alarmes gerados pelo DIbW.

Na Tabela 6.12 estão os resultados da detecção de ataques pelo DIbW na base do DARPA 99 usando o tráfego ICMP.

Tabela 6.12: Resultados da análise dos os pacotes ICMP do tráfego de rede.

Tam. Jan.	Verdadeiros P.	Falsos P.	Ataques detectados
64	4	0	pod, satan, portsweep
128	6	4	pod, satan, portsweep, neptune
256	6	5	pod, satan, portsweep, neptune

O tráfego de pacotes do protocolo ICMP, presente na base, é mais irregular do que o tráfego dos outros protocolos. O sistema conseguiu identificar variações no padrão de

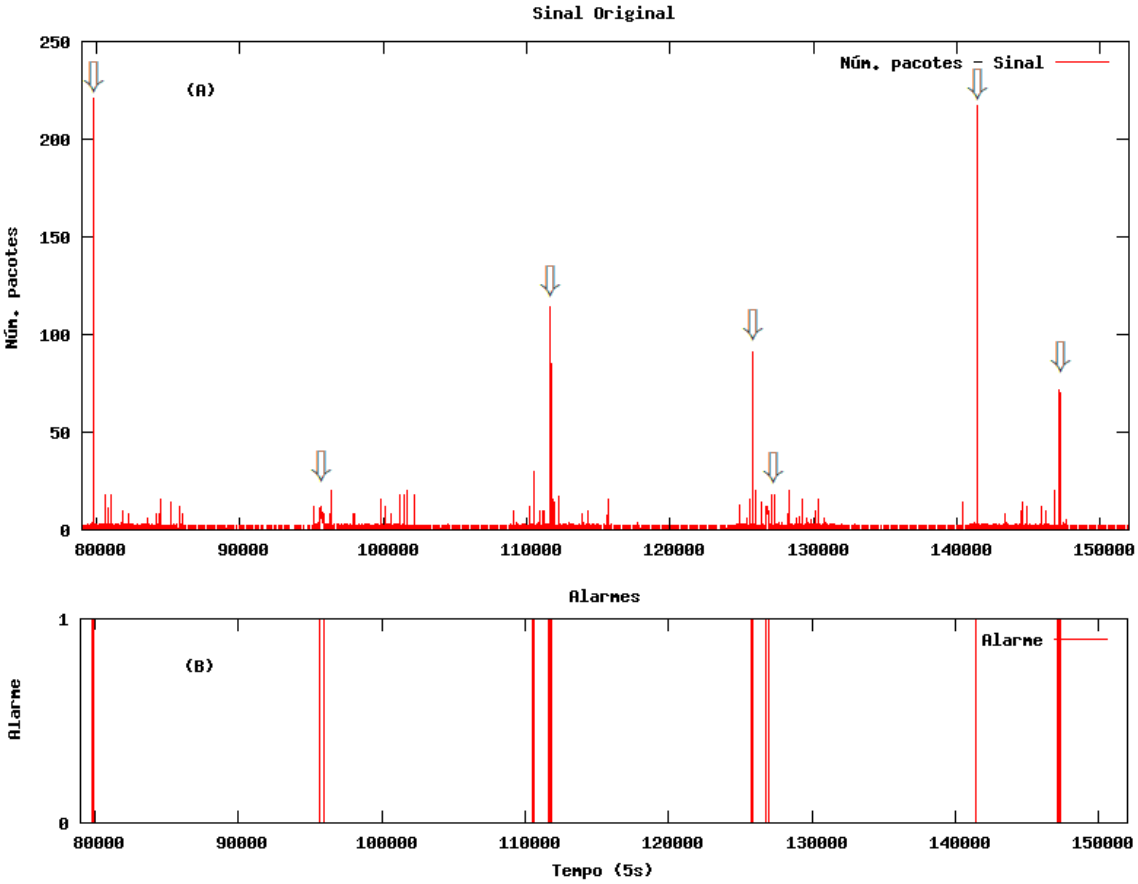

Figura 6.10: Tráfego de rede correspondente aos pacotes do protocolo ICMP capturados a cada 5 segundos (A) e os alarmes gerados pelo DIbW (B).

tráfego (anomalias) deste protocolo, porém algumas dessas anomalias não eram devido a ataques. Vários ataques foram identificados, porém como o padrão de tráfego é naturalmente irregular, alguns falsos positivos foram gerados (Tabela 6.12).

Tabela 6.13: Ataques detectados usando o tráfego do protocolo ICMP, janela de tamanho 128 e *wavelet* D8.

Ataque	**Total Ataques**	**T. Ataques Detectados**
pod	2	2
satan	2	2
portsweep	2	1
neptune	2	1

Usando-se uma janela de detecção de tamanho 128 com o contadores do tráfego ICMP, o sistema detectou corretamente todos os ataques do tipo *pod* e *satan* e a metade dos ataques do tipo *portsweep* e *neptune* (Tabela 6.13).

Considerando-se todos os descritores de tráfego simultaneamente: IP, TCP, UDP, ICMP, o sistema detectou 11 dos 13 ataques analisados (85%) (Tabela 6.15). O número de falsos positivos foi baixo, 12 em 77077 amostras (0,015%). Diferentes descritores de

tráfego permitiram a detecção de diferentes tipos de ataques (Tabela 6.14).

Tabela 6.14: Ataques detectados usando diferentes descritores de tráfego de rede, janela de tamanho 128 e *wavelet* D8.

Ataque	Total Ataques	IP	TCP	UDP	ICMP
satan	2	2	2	-	2
crashiis	3	2	-	2	-
portsweep	2	-	-	1	1
mailbomb	2	-	-	2	-
pod	2	-	-	-	2
neptune	2	-	-	-	1

Tabela 6.15: Resultado da análise de todos os descritores: IP, TCP, UDP e ICMP com janela de tamanho 128 e *wavelet* D8.

Tam. Janela	T. Amostras	T. Ataques	VP	FP	FN	TVP%	TFP%
128	77077	13	11	12	2	85%	0,015%

De modo geral, a detecção de ataques de rede é influenciada pela especificidade das variáveis selecionadas. Nestes experimentos foram considerados ataques que geram perturbação nos descritores de tráfego analisados: IP, TCP, UDP e ICMP. Para os ataques analisados o sistema alcançou boa taxa de detecção, em vários casos todos os ataques de um tipo específico foram detectados. Os falsos positivos gerados pelo sistema foram devido a mudanças normais no tráfego que assemelham-se a ataques. Vale lembrar que o tráfego de rede é naturalmente irregular. No entanto, o número de falsos positivos foi baixo.

Comparações com os trabalhos relacionados quando ao desempenho na detecção de ataques é uma tarefa difícil devido ao uso diverso dos dados de entrada e de diferentes metodologias de obtenção de dados empregados em cada trabalho.

No trabalho em (DAINOTTI; PESCAPE; VENTRE, 2006) os autores propuseram um mecanismo de detecção de anomalias de rede que combina uma abordagem baseada no método CUSUM e EWMA com uma abordagem baseada na Transformada *Wavelet* Continua (TWC), *wavelet* de Morlet, com o objetivo de detectar anomalias de volume de tráfego de rede causadas por ataques do tipo DoS. Usando dados próprios e a DARPA 99, os autores reportaram uma Taxa de Acertos média de 87 % de ataques do tipo DoS, enquanto que a Taxa de Erros média ficou em 38 % em relação ao total de alarmes. Os autores definiram a Taxa de Acertos como : número de verdadeiros positivos / núm. de amostras X 100; e a Taxa de Erros como: número de falsos positivos / total de alarmes X 100. No entanto, a comparação dos trabalhos não é precisa pois, apesar da base (DARPA

99) e o intervalo de amostragem ($\Delta t = 5\ s$) usados serem os mesmos usados aqui, os autores relatam usarem simulações de ataques, o que diferencia os testes.

No trabalho em (LU; TAVALLAEE; GHORBANI, 2008) os autores propuseram uma abordagem para detecção de anomalias de rede baseada na Transformada *Wavelet* Discreta e séries auto-regressivas do tipo ARX. Na abordagem, as séries de dados são transformadas em um conjunto de coeficientes *wavelet*, usando-se TWD, em seguida os coeficientes são aproximados o modelo ARX e então o resíduo da predição é usado para a detecção de anomalias utilizado o GMM (*Gaussian Mixture Model*), buscando a identificação de *outliers*. No trabalho foi usada a dados da base KDDCUP 99, derivada da DARPA 99, onde foram selecionadas quinze variáveis descritivas de tráfego, usando-se o modelo de agregação por fluxos origem-destino. Considerando-se apenas o melhor caso, a base *Wavelet* "Daubechies1"[2] (Haar), a abordagem corretamente identificou 7 de 10 tipos de ataques DoS presentes. Os autores usaram apenas os dados do primeiro dia da quinta semana da base DARPA 99 e não informaram o número de falsas detecções.

No trabalho em (GAO et al., 2006), os autores usaram a Transformada *Wavelet Packet* (TWP) e a reconstrução do sinal a partir dos coeficientes *wavelet* para cada nível selecionado da transformada. Medidas estatísticas, como média e variância, foram usadas para caracterizar uma anomalia, como a razão da média ou da variância entre a janela de detecção e a janela histórica foram mensuradas e comparadas com valores de *threshold* predefinidos para identificar uma anomalia. O sinal reconstruído para cada nível é usado para a detecção de anomalias. Os autores usaram dados próprios mesclados com simulações.

No trabalho em (KIM; REDDY, 2008) foi usada uma função, definida no trabalho, que calcula a correlação dos endereços IP de origem e destino dos pacotes trafegados para a geração dos dados de entrada do detector. As funções *wavelet* são usadas para decompor o sinal e reconstruir conforme os níveis desejados. A detecção é feita sobre o sinal reconstruído. Os autores usaram para avaliação dados próprios juntamente com dados simulados, o que inviabilizou comparações de resultados.

6.4 Análise de desempenho

Para analisar o desempenho do DIbW em relação ao tempo de execução, foram realizados testes, nos quais uma sequência de amostras (série) do tráfego de rede foi submetida ao sistema e coletado o tempo de execução. O objetivo dos testes é verificar a possibi-

[2]Nomenclatura alternativa para a base Daubechies D2 ou Haar

lidade de uso da ferramenta proposta em análises e detecção de anomalias de tráfego de rede em tempo real (*on line*). Os experimentos foram realizados no seguinte ambiente:

- Computador com processador Intel Core 2 Duo modelo T7300 2.0 GHz, memória DDR2 de 2 GB e disco rígido interface Sata de 250 GB;
- Sistema Operacional Microsoft Windows Vista 32 bits;
- Máquina Virtual Java SDK (*Software Development Kit*) versão 1.6.05.

No primeiro experimento, foi submetido ao DIbW uma sequência de amostras (série) de tráfego de rede correspondente as três primeiras da base de dados DARPA 99. Os contadores foram gerados selecionando-se o número de pacotes total de pacotes de rede trafegados a cada intervalo de 5 segundos. Neste experimento foram avaliados o tempo de execução considerando-se diversas configurações para o tamanho da janela de observação e função *wavelet*. Foram usadas janelas de observação de tamanhos de 64, 128 e 256 pontos estas foram usadas nos testes de detecção, sendo que uma janela de 128 pontos mostrou resultados melhores quanto ao numero de detecções e falsos positivos. As funções *wavelet* usadas são da família Daubechies (DAUBECHIES, 1992): Daubechies 2 (D2 ou Haar), Daubechies 4 (D4), Daubechies 6 (D6) e Daubechies 8 (D8). Para a normalização dos coeficientes *wavelet* (detalhes) foi usada a função Raiz Quadrada.

Tabela 6.16: Teste de desempenho do DIbW. Uma sequência de amostras (230608 amostras) de tráfego de rede foi submetida ao sistema para cada configuração (tamanho da janela de observação e base *wavelet*) e foi avaliado o tempo total de execução e calculado o tempo por amostra.

Wavelet	**Tam. Jan.**	**Tempo Total (s)**	**Tempo Médio (μs)**
D8	256	5.866	0.25437105390966490
D8	128	3.229	0.14002116145146742
D8	64	1.872	0.08117671546520502
D6	256	5.616	0.24353014639561507
D6	128	3.042	0.13191216263095817
D6	64	1.762	0.07640671615902310
D4	256	5.070	0.21985360438493026
D4	128	2.886	0.12514743634219108
D4	64	1.732	0.07510580725733712
D2	256	4.961	0.21512696870880454
D2	128	2.745	0.11903316450426699
D2	64	1.669	0.07237389856379657

A Tabela 6.16 sumariza os principais resultados. O tamanho total da série de entrada é de 230608 amostras. Para cada tamanho da janela de observação e função *wavelet* foi

submetida ao sistema a mesma série e registrado o tempo total de execução (em segundos). O tempo médio de execução de cada amostra (em microssegundos) corresponde ao tempo total de execução dividido pelo número total de amostras (230608). Para cada amostra o sistema realiza o processo de análise e geração de alarme conforme o Algoritmo 4.1 da Seção 4.2.

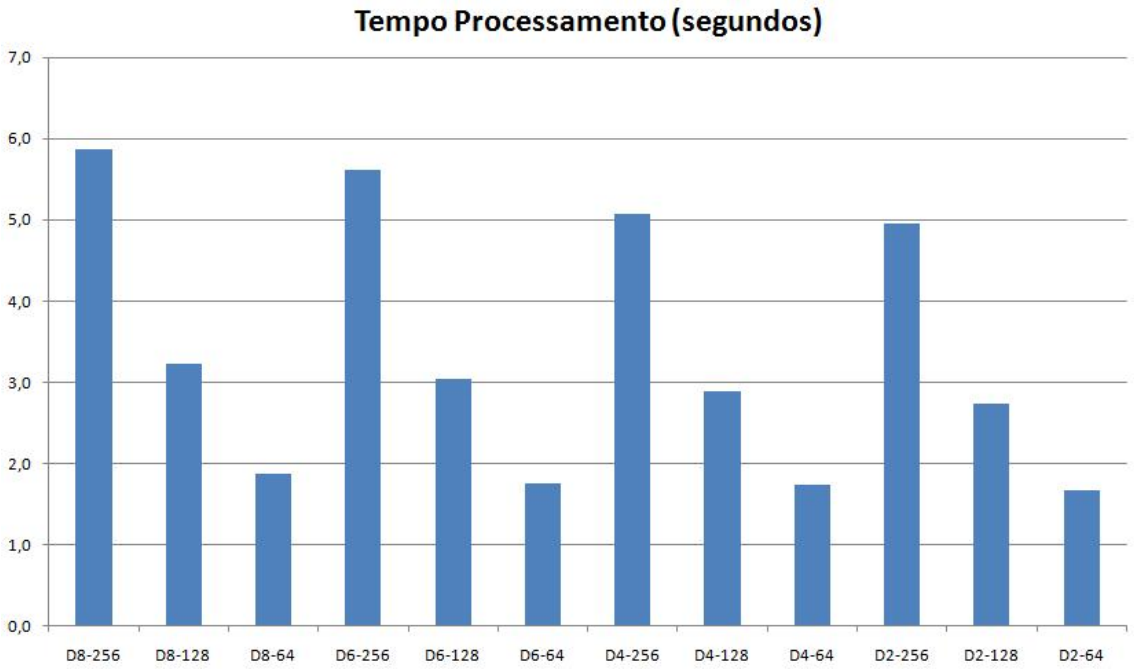

Figura 6.11: Tempo de processamento de 230608 amostras de tráfego de rede usando as funções *wavelet* D2, D4, D6 e D8 com tamanhos de janela de 64, 128 ou 256 pontos.

Na Figura 6.11 está representado o gráfico com as comparações dos tempos totais de execução das 230608 amostras de tráfego de rede pelo DIbW. O tempo de processamento foi influenciado principalmente pelo tamanho da janela de observação. Quanto maior o tamanho da janela de observação mais dados o sistema precisa analisar a cada execução. O uso de diferentes funções *wavelet*, D2, D4, D6 ou D8, não influenciou significativamente o tempo de processamento. Como a função *wavelet* não apresentou impacto significativo no tempo de processamento do mecanismo, pode-se escolher a *wavelet* que melhor se adapta aos dados. Nos testes de detecção a *wavelet* D8 apresentou resultados ligeiramente superiores.

Para os tamanhos de janela de observação e funções *wavelet* testadas, o mecanismo proposto apresentou baixo custo computacional. Para cada amostra o tempo de execução médio ficou em frações de microssegundos (Tabela 6.16). Considerando um intervalo de amostragem de 5 segundos na coleta, o sistema consegue processar os dados sem comprometer o desempenho. O atraso na detecção de uma anomalia devido à analise dos

dados é o tempo de processamento (frações de microssegundos). Na prática, o atraso na detecção do sistema depende também do módulo de coleta, porém não superior ao intervalo de amostragem do módulo.

6.5 Considerações Finais

Neste Capítulo foi analisado o mecanismo de detecção de anomalias do DIbW quanto à capacidade de detecção de ataques e o desempenho computacional. Nos testes usou-se a base de dados DARPA 99 por possuir tráfego de rede real, ataques conhecidos e documentados.

Inicialmente foram analisados estatisticamente os coeficientes da transformada *wavelet*. Aplicando-se a operação Raiz Quadrada, os coeficientes *wavelet* (detalhes) apresentaram características mais próximas de uma distribuição normal. Dessa forma, escolhe-se a raiz quadrada para a normalização dos coeficientes no mecanismo de detecção. O *threshold* é calculado conforme o desvio padrão dos coeficientes *wavelet* e uma tabela de probabilidade para uma distribuição normal.

Para avaliar a performance na detecção de ataques usou-se a segunda semana de tráfego de dados da DARPA 99 que possui ataques rotulados. Foram gerados descritores para os protocolos: IP, TCP, UDP e ICMP, com intervalo de amostragem de 5 segundos. Usando-se a janela de observação de tamanho 128 e a *wavelet* D8 obteve-se os melhores resultados quanto ao número de detecções e falsos positivos. Diferentes descritores de tráfego permitiram a detecção de diferentes tipos de ataques, em alguns casos todos os ataques de um tipo específico foram detectados. Na média o sistema detectou 85% dos ataques com 0,015% de falsos positivos. O mecanismo de detecção mostrou-se adequado para a detecção de ataques que geram alterações (anomalias) no padrão de tráfego de um descritor de tráfego de rede. Considerando-se que o tráfego de rede é natualmente irregular, a quantidade de falsos positivos foi baixa.

Na avaliação de desempenho computacional foram usadas as três semanas (com ataques e sem ataques) de tráfego da DARPA 99. Os descritores de tráfego foram inseridos sequencialmente e o foi coletado o tempo de processamento. Para as janelas de detecção de tamanhos 64, 128 e 256, com *wavelets* Haar, D4 ou D8, o tempo de processamento (em frações de microssegundos para cada amostra) não comprometeu o desempenho do sistema. Como foi usado um intervalo de amostragem de 5 segundos o mecanismo pode ser usada para analises em tempo real.

7 CONCLUSÕES

Este trabalho explorou a área de detecção de intrusão em redes de computadores usando a abordagem baseada em anomalias. A detecção de anomalias em redes de computadores é uma área de estudo bastante ativa e normalmente preocupa-se com a eficiência dos métodos e com problema dos falsos positivos. Neste contexto, o uso da Transformada *Wavelet* mostrou-se viável para a detecção de anomalias de rede devido a capacidade de análise em multirresolução. No entanto, os métodos empregados muitas vezes não são adequados para a análise em tempo real, devido a complexidade computacional envolvida, o que acaba por limitar tais soluções.

A complexidade das redes de computadores, devido à quantidade de dispositivos conectados, a variedade de protocolos e serviços, volume elevado de tráfego, bem como, características intrínsecas do tráfego padrão, dificultam a coleta, análise e detecção de anomalias. Na Detecção de Anomalias de Rede, o método de análise é de vital importância pois impacta diretamente no desempenho e eficiência do detector. A abordagem em tempo real, ainda, apresenta alguns desafios, por precisar de resposta a um determinado evento suspeito em tempo reduzido. Consequentemente, o mecanismo de detecção precisa ser eficiente para permitir tempos de resposta reduzidos.

Neste sentido, este trabalho propôs um novo mecanismo para a detecção de anomalias de rede baseada na Transformada *Wavelet* Discreta. O método mostrou-se eficiente computacionalmente e adequado para análises em tempo real. Por meio da análise dos descritores do tráfego de rede, busca-se identificar anomalias de tráfego, considerando-se anomalias como possíveis Intrusões.

O mecanismo de detecção de anomalias proposto consiste na amostragem de descritores de rede, na geração de um sinal para análise usando uma janela deslizante, na transformação do sinal com *wavelets* discretas ortonormais de Daubechies, na normalização dos coeficientes *wavelet* (detalhes), no cálculo do valor do *threshold* baseado no desvio padrão dos coeficientes e conforme uma tabela de probabilidades da distribuição normal,

e por fim na detecção de anomalias conforme o *threshold* diretamente nos coeficientes *wavelet* (detalhes).

Como os coeficientes *wavelet* não seguem uma distribuição normal, a operação Raiz Quadrada mostrou-se adequada estatisticamente para a normalização. Esta característica foi usada para o cálculo do *threshold*. Usou-se uma janela de observação deslizante de tamanho 64, 128 ou 256 e as *wavelets* de Haar, D4 e D8. A janela de observação de tamanho 128 e a função *wavelet* D8 apresentaram os melhores resultados quanto ao número de detecções e falsos positivos.

Na análise de desempenho na detecção de anomalias o mecanismo proposto apresentou bom desempenho em relação ao número de ataques detectados com poucos falsos positivos. Diferentemente das abordagens tradicionais, no entanto, o mecanismo proposto apresenta um esquema de detecção simplificado. As anomalias de rede são detectadas nos coeficientes *wavelet*, eliminando-se a necessidade da transformada *wavelet* inversa ou de outras etapas de processamento. Esta abordagem possui baixa complexidade computacional, e mostrou-se eficiente em termos de detecção e tempo de execução, permitindo o seu uso em análises em tempo real. Além disso, o método é genérico e pode trabalhar com diferentes descritores do tráfego de rede. As principais contribuições deste trabalho são proposição do mecanismo de detecção de anomalias de rede destinado a análise em tempo real e a demonstração de sua capacidade de detecção de ataques de rede.

7.1 Principais Contribuições

Este trabalho inova ao explorar a eficiência computacional no projeto do mecanismo de detecção de anomalias de rede, como requisito para a análise de tráfego em tempo real, e contribui ao propor um novo método de detecção baseado na Transformada *Wavelet*. As principais contribuições deste trabalho são:

- A proposição de um novo mecanismo de detecção de anomalias de rede baseado na Transformada *Wavelet* Discreta. O mecanismo de detecção usa amostragens de um descritor de rede genérico, possui um projeto eficiente computacionalmente e é capaz de detectar anomalias de tráfego de rede;

- Apresentar a construção de *framework* genérico e expansível para detecção de anomalias de rede usando a abordagem proposta; e

- Demostrar a capacidade de detecção de anomalias do mecanismo proposto e a eficiência computacional do método, na forma de testes de desempenho.

No entanto, este trabalho possui algumas limitações, como a base de dados usada. A base do DARPA 99, apesar de já desatualizada, ainda é bastante usada e possui ataques documentados e possibilitou a realização de avaliações quanto a capacidade de detecção do mecanismo proposto.

7.2 Trabalhos Futuros

A detecção de anomalias de rede é uma área de pesquisa bastante ativa com constante desenvolvimento de novas ferramentas e aplicação de novas técnicas. Este trabalho representa uma pequena contribuição para a detecção de anomalias em redes de computadores.

Como tema para trabalhos futuros sugere-se a pesquisa quanto a seleção de variáveis para análise. A escolha das variáveis e a criação de variáveis derivadas ainda é fracamente explorada na literatura quanto ao impacto na detecção de anomalias. O sistema proposto neste trabalho pode ser usado para a análise de diversas variáveis simultaneamente, permitindo a identificação de diferentes formas de anomalias de rede.

REFERÊNCIAS

ABDOLLAH, M. F.; YAACOB, A. H.; SAHIB, S.; ISMAIL MOHAMAD, M. F. I. Revealing the Influence of Feature Selection for Fast Attack Detection. **IJCSNS International Journal of Computer Science and Network Security**, [S.l.], v.8, n.8, p.107–115, aug 2008.

BARFORD, P.; KLINE, J.; PLONKA, D.; RON, A. A signal analysis of network traffic anomalies. In: ACM SIGCOMM WORKSHOP ON INTERNET MEASURMENT, IMW 2002, 2., 2002, New York, NY, USA. **Anais...** ACM, 2002. p.71–82.

BASSEVILLE, M.; NIKIFOROV, I. V. **Detection of abrupt changes**: theory and application. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

BOLZONI, D. **Revisiting anomaly-based network intrusion detection systems**. 2009. Tese (Doutorado) — University of Twente (Netherlands).

BORGNAT, P.; DEWAELE, G.; FUKUDA, K.; ABRY, P.; CHO, K. Seven Years and One Day Sketching the Evolution of Internet Traffic. **Infocom 2009**, [S.l.], 2008.

BRO. **Bro Intrusion Detection System**. Disponível em: http://www.bro-ids.org/, último acesso em dezembro de 2009.

CHENG, X.; XIE, K.; WANG, D. Network Traffic Anomaly Detection Based on Self-Similarity Using HHT and Wavelet Transform. In: INFORMATION ASSURANCE AND SECURITY, 2009. IAS '09. FIFTH INTERNATIONAL CONFERENCE ON, 2009. **Anais...** [S.l.: s.n.], 2009. v.1, p.710–713.

CHOU, T. S.; YEN, K. K.; LUO, J. Network Intrusion Detection Design Using Feature Selection of Soft Computing Paradigms. **International Journal of Computational Intelligence**, [S.l.], v.4, n.3, 2008.

COIFMAN, R. R.; WICKERHAUSER, M. V. Entropy-Based Algorithms For Best Basis Selection. **IEEE Transactions on Information Theory**, [S.l.], v.38, p.713–718, 1992.

DAINOTTI, A.; PESCAPE, A.; VENTRE, G. NIS04-1: wavelet-based detection of dos attacks. In: GLOBAL TELECOMMUNICATIONS CONFERENCE, 2006. GLOBECOM '06. IEEE, 2006. **Anais...** [S.l.: s.n.], 2006. p.1–6.

DARPA. **Defense Advanced Research Projects Agency**. disponível em: http://www.ll.mit.edu/IST/ideval/index.html. Último acesso em outubro de 2008.

DAUBECHIES, I. **Ten Lectures on Wavelets**. [S.l.]: SIAM, 1992. n.61. (CBMS/NSF Series in Applied Math.).

DENNING, D. E. An intrusion-detection model. **IEEE Transaction on Software Engineering**, [S.l.], v.13, n.2, p.222–232, 1987.

DONOHO, D. L.; JOHNSTONE, I. M. De-noising by soft-thresholding. **IEEE Transactions on Information Theory**, [S.l.], v.41, n.3, p.613–627, 1995.

FARRAPOSO, S. **Contributions on detection and classification of internet traffic anomalies**. 2009. Tese (Doutorado) — Université Paul Sabatier - Toulouse III. 09414.

FIELDING, R.; GETTYS, J.; MOGUL, J.; FRYSTYK, H.; MASINTER, L.; LEACH, P.; BERNERS-LEE, T. **Hypertext Transfer Protocol – HTTP/1.1**. Updated by RFC 2817, RFC 2616 (Draft Standard).

GAO, J.; HU, G.; YAO, X.; CHANG, R. Anomaly Detection of Network Traffic Based on Wavelet Packet. In: ASIA-PACIFIC CONFERENCE ON COMMUNICATIONS, 2006. **Anais...** [S.l.: s.n.], 2006.

GARCíA-TEODORO, P.; DíAZ-VERDEJO, J.; MACIá-FERNáNDEZ, G.; VáZQUEZ, E. Anomaly-based network intrusion detection: techniques, systems and challenges. **Computers and Security**, [S.l.], v.28, n.1-2, p.18–28, 2009.

GHALI, N. I. Feature Selection for Effective Anomaly-Based Intrusion Detection. **IJCSNS International Journal of Computer Science and Network Security**, [S.l.], v.9, n.3, 2009.

GIBILISCO, S. **Statistics Demystified**. 1.ed. [S.l.]: McGraw-Hill Professional, 2004.

GNUPLOT. **gnuplot homepage**. Disponível em: http://www.gnuplot.info/, último acesso em janeiro de 2010.

GOUD, P. A.; BINULAL, G.; K.P, S. Simplified Method of Designing Daubechies Wavelets in Class Room. **International Journal of Recent Trends in Engineering**, [S.l.], v.1, n.4, 2009.

GUANGMIN, L. Modeling Unknown Web Attacks in Network Anomaly Detection. In: THIRD INTERNATIONAL CONFERENCE ON CONVERGENCE AND HYBRID INFORMATION TECHNOLOGY 2008, ICCIT '08, 2008. **Anais...** [S.l.: s.n.], 2008. v.2, p.112–116.

HARRINGTON, D.; PRESUHN, R.; WIJNEN, B. **An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks**. Updated by RFCs 5343, 5590, RFC 3411 (Standard).

HETTICH, S.; BAY, S. D. **The UCI KDD Archive**. Irvine, CA: University of California, Department of Information and Computer Science.disponível em: http://kdd.ics.uci.edu. Último acesso em outubro de 2009.

HUANG, C.-T.; THAREJA, S.; SHIN, Y.-J. Wavelet-based Real Time Detection of Network Traffic Anomalies. In: SECURECOMM AND WORKSHOPS 2006, 2006. **Anais...** [S.l.: s.n.], 2006. p.1–7.

JANSEN, M. **Wavelet Thresholding and Noise Reduction**. 2000. Tese (Doutorado) — Katholieke Universiteit Leuven. Faculteit Toegepaste Wetenschappen.

JAVA. **JAVA Technology**. Disponível em: http://java.sun.com/, último acesso em janeiro de 2010.

JOANES, D. N.; GILL, C. A. Comparing measures of sample skewness and kurtosis. **Journal of the Royal Statistical Society (Series D): The Statistician**, University of Leeds, UK, v.47, n.1, p.183–189, 1998.

KIM, S. S.; REDDY, A. L. N. Statistical techniques for detecting traffic anomalies through packet header data. **IEEE/ACM Transaction on Networking**, Piscataway, NJ, USA, v.16, n.3, p.562–575, 2008.

KIZZA, J. M. **Computer Network Security**. New York, NY: Springer, 2005.

KLENSIN, J. **Simple Mail Transfer Protocol**. 2008, RFC 5321 (Draft Standard).

KRUEGEL, C.; VIGNA, G. Anomaly detection of web-based attacks. In: ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, CCS 2003, 10., 2003, New York, NY, USA. **Anais...** ACM, 2003. p.251–261.

LELAND, W. E.; TAQQU, M. S.; WILLINGER, W.; WILSON, D. V. On the self-similar nature of Ethernet traffic (extended version). **IEEE/ACM Transaction on Network**, Piscataway, NJ, USA, v.2, n.1, p.1–15, 1994.

LI, L.; LEE, G. DDoS attack detection and wavelets. In: INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATIONS AND NETWORKS 2003, ICCCN 2003, 12., 2003. **Anais...** [S.l.: s.n.], 2003. p.421–427.

LI, Y.; FANG, B.-X. A Lightweight Online Network Anomaly Detection Scheme Based on Data Mining Methods. In: IEEE INTERNATIONAL CONFERENCE ON NETWORK PROTOCOLS 2007, ICNP 2007, 2007. **Anais...** [S.l.: s.n.], 2007. p.340–341.

LIU, T.; QI, A.; HOU, Y.; CHANG, X. Method for network anomaly detection based on Bayesian statistical model with time slicing. In: WORLD CONGRESS ON INTELLIGENT CONTROL AND AUTOMATION 2008, WCICA 2008, 7., 2008. **Anais...** [S.l.: s.n.], 2008. p.3359–3362.

LONGCHUPOLE, S.; MANEERAT, N.; VARAKULSIRIPUNTH, R. Anomaly detection through packet header data. In: INFORMATION, COMMUNICATIONS AND SIGNAL PROCESSING, 2009. ICICS 2009. 7TH INTERNATIONAL CONFERENCE ON, 2009. **Anais...** [S.l.: s.n.], 2009. p.1–4.

LU, W.; TAVALLAEE, M.; GHORBANI, A. Detecting Network Anomalies Using Different Wavelet Basis Functions. In: COMMUNICATION NETWORKS AND SERVICES RESEARCH CONFERENCE 2008, CNSR 2008 6TH ANNUAL, 2008. **Anais...** [S.l.: s.n.], 2008. p.149–156.

MAFRA, P. M.; FRAGA, J. S.; MOLL, V.; SANTIN, A. O. POLVO-IIDS, Um Sistema de Detecção de Intrusão Inteligente Baseado em Anomalias. In: VIII SIMPóSIO BRASILEIRO EM SEGURANçA DA INFORMAçãO E DE SISTEMAS COMPUTACIONAIS (SBSEG 2008), 2008. **Anais...** [S.l.: s.n.], 2008. p.201–214.

MAHONEY, M. V.; CHAN, P. K. An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection. In: SIXTH INTERNATIONAL SYM-

POSIUM ON RECENT ADVANCES IN INTRUSION DETECTION, 2003. **Anais...** Springer-Verlag, 2003. p.220–237.

MALLAT, S. G. A theory for multiresolution signal decomposition: the wavelet representation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [S.l.], v.11, p.674–693, 1989.

MALLAT, S. G. **A wavelet tour of signal processing**. [S.l.]: Academic Press, 1998.

NETBEANS. **NetBeans**. Disponível em: http://www.netbeans.org/, último acesso em janeiro de 2010.

NIELSEN, O. M. **Wavelets in scientific computing**. 1998. Tese (Doutorado) — Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby.

NORTHCUTT, S.; NOVAK, J. **Network Intrusion Detection, Third Edition**. [S.l.]: New Riders Publishing, 2002.

ONUT, I.-V.; GHORBANI, A. A. A Feature Classification Scheme For Network Intrusion Detection. **International Journal of Network Security**, [S.l.], v.5, n.1, p.1–15, 2007.

PENG, T.; LECKIE, C.; RAMAMOHANARAO, K. Survey of network-based defense mechanisms countering the DoS and DDoS problems. **ACM Computer Surv.**, New York, NY, USA, v.39, n.1, p.3, 2007.

PLONKA, D.; BARFORD, P. Network anomaly confirmation, diagnosis and remediation. In: COMMUNICATION, CONTROL, AND COMPUTING, 2009. ALLERTON 2009. 47TH ANNUAL ALLERTON CONFERENCE ON, 2009. **Anais...** [S.l.: s.n.], 2009. p.128 –135.

POD. **CERT Advisory CA-1996-26 Denial-of-Service Attack via ping**. Disponível em: http://www.cert.org/advisories/CA-1996-26.html, último acesso em janeiro de 2010.

POSTEL, J. **User Datagram Protocol**. [S.l.]: IETF, 1980. n.768. (Request for Comments).

POSTEL, J. **Transmission Control Protocol**. Updated by RFCs 1122, 3168, RFC 793 (Standard) 1981.

POSTEL, J. **Internet Control Message Protocol**. Updated by RFCs 950, 4884, RFC 792 (Standard) 1981.

POSTEL, J.; REYNOLDS, J. **File Transfer Protocol**. Updated by RFCs 2228, 2640, 2773, 3659, RFC 959 (Standard) 1985.

PRESUHN, R. **Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)**. 2002, RFC 3418 (Standard).

QIN, Z.-C. ROC Analysis for Predictions made by Probabilistic Classifiers. In: FOURTH INTERNATIONAL CONFERENCE ON MACHINE LEARNING AND CYBERNETICS, 2005. **Anais...** [S.l.: s.n.], 2005.

RED, C. **Code Red Worm Crashes IIS 4.0 Servers with URL Redirection Enabled**. Disponível em: http://www.cert.org/incidentnotes/IN-2001-10.html, último acesso em janeiro de 2010.

ROHANI, M.; MAAROF, M.; SELAMAT, A.; KETTANI, H. LoSS Detection Approach Based on ESOSS and ASOSS Models. In: FOURTH INTERNATIONAL CONFERENCE ON INFORMATION ASSURANCE AND SECURITY 2008, ISIAS 08, 2008. **Anais...** [S.l.: s.n.], 2008. p.192–197.

Sá SILVA, L. de. **Uma Metodologia para Detecção de Ataques de Redes baseada em redes Neurais**. 2008. Tese (Doutorado) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, SP, Brasil.

SAKIA, R. M. The Box-Cox Transformation Technique: a review. **Journal of the Royal Statistical Society. Series D (The Statistician)**, [S.l.], v.41, n.2, p.169–178, 1992.

SAMAAN, N.; KARMOUCH, A. Network anomaly diagnosis via statistical analysis and evidential reasoning. **Network and Service Management, IEEE Transactions on**, [S.l.], v.5, n.2, p.65–77, jun 2008.

SATAN. **SATAN (Security Administrator Tool for Analyzing Networks)**. Disponível em: http://www.porcupine.org/satan/, último acesso em janeiro de 2010.

SCHERRER, A.; LARRIEU, N.; OWEZARSKI, P.; BORGNAT, P.; ABRY, P. Non-Gaussian and Long Memory Statistical Characterizations for Internet Traffic with Anomalies. **IEEE Transactions on Dependable and Secure Computing**, [S.l.], v.4, n.1, p.56–70, jan 2007.

SELVAKANI, S.; RAJESH, R. Genetic Algorithm for framing rules for intrusion Detection. **International Journal of Computer Science and Network Security, IJCSNS**, [S.l.], v.7, n.11, nov 2007.

SNORT. **Snort**. Disponível em: http://www.snort.org/, último acesso em dezembro de 2009.

SOCOLOFSKY, T.; KALE, C. **TCP/IP tutorial**. 1991, RFC 1180 (Informational).

SOULE, A.; SALAMATIAN, K.; TAFT, N. Combining filtering and statistical methods for anomaly detection. In: IMC 05 PROCEEDINGS OF THE 5TH ACM SIGCOMM CONFERENCE ON INTERNET MEASUREMENT, 2005, Berkeley, CA, USA. **Anais...** USENIX Association, 2005. p.31–31.

STOEV, S.; TAQQU, M. S.; PARK, C.; MARRON, J. S. On the wavelet spectrum diagnostic for Hurst parameter estimation in the analysis of Internet traffic. **Computer Networking**, New York, NY, USA, v.48, n.3, p.423–445, 2005.

TCPDUMP. **TCPDUMP/LIBPCAP public repository**. Disponível em: http://www.tcpdump.org/, último acesso em dezembro de 2009.

TCPSTAT. **tcpstat Home Page**. Disponível em: http://www.frenchfries.net/paul/tcpstat/, último acesso em dezembro de 2009.

THOTTAN, M.; JI, C. Anomaly detection in IP networks. **IEEE Transactions on Signal Processing**, [S.l.], v.51, n.8, p.2191–2204, Aug. 2003.

T.LACHMAN; A.P.MEMON; T.R.MOHAMAD; Z.A.MEMON. Detection of Power Quality Disturbances Using Wavelet Transform Technique. **International Journal for the Advancement of Science and Arts**, [S.l.], v.1, n.1, p.177–185, 2010.

WANG, X.; REN, Y.; SHAN, X. WDRLS: a wavelet-based on-line predictor for network traffic. In: IEEE GLOBAL TELECOMMUNICATIONS CONFERENCE 2003, GLOBECOM 2003, 2003. **Anais...** [S.l.: s.n.], 2003. v.7, p.4034–4038.

WEISSTEIN, E. W. **Autocorrelation. From MathWorld–A Wolfram Web Resource.** Disponível em: http://mathworld.wolfram.com/Autocorrelation.html, último acesso em fevereiro de 2010.

WILF, H. S. **Algorithms and complexity**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1994. Disponível em: http://www/cis.upenn.edu/wilf.

WU, Q.; SHAO, Z. Network Anomaly Detection Using Time Series Analysis. In: JOINT INTERNATIONAL CONFERENCE ON AUTONOMIC AND AUTONOMOUS SYSTEMS AND INTERNATIONAL CONFERENCE ON NETWORKING AND SERVICES 2005, ICAS-ICNS 2005, 2005. **Anais...** [S.l.: s.n.], 2005. p.42–42.

XIA, H.; XU, W. Research on Method of Network Abnormal Detection Based on Hurst Parameter Estimation. In: INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE AND SOFTWARE ENGINEERING 2008, 2008. **Anais...** [S.l.: s.n.], 2008. v.3, p.559–562.

YAO, L.; ZHITANG, L.; SHUYU, L. A Fuzzy Anomaly Detection Algorithm for IPv6. In: SECOND INTERNATIONAL CONFERENCE ON SEMANTICS, KNOWLEDGE AND GRID 2006, SKG '06, 2006. **Anais...** [S.l.: s.n.], 2006. p.67–67.

ZAMAN, S.; KARRAY, F. Features selection for intrusion detection systems based on support vector machines. In: CCNC'09: PROCEEDINGS OF THE 6TH IEEE CONFERENCE ON CONSUMER COMMUNICATIONS AND NETWORKING CONFERENCE, 2009, Piscataway, NJ, USA. **Anais...** IEEE Press, 2009. p.1066–1073.

ZARPELãO, B. B.; MENDES, L. S.; ABRãO, T.; SAMPAIO, L. D. H.; LIMA, M. F.; JR., M. L. P. Detecção de Anomalias em Redes de Computadores. In: XXVII SIMPóSIO BRASILEIRO DE TELECOMUNICAçõES - SBRT 2009, 2009. **Anais...** [S.l.: s.n.], 2009.

ZHANG, X.-Q.; GU, C.-H. CH-SVM Based Network Anomaly Detection. In: MACHINE LEARNING AND CYBERNETICS, 2007 INTERNATIONAL CONFERENCE ON, 2007. **Anais...** [S.l.: s.n.], 2007. v.6, p.3261 –3266.

ZHANG, Y.; HAN, Z. guo; REN, J. xia. A Network Anomaly Detection Method Based on Relative Entropy Theory. In: ELECTRONIC COMMERCE AND SECURITY, 2009. ISECS '09. SECOND INTERNATIONAL SYMPOSIUM ON, 2009. **Anais...** [S.l.: s.n.], 2009. v.1, p.231–235.