Universidade Estadual de Londrina

GILBERTO FERNANDES JUNIOR

CARACTERIZAÇÃO DE TRÁFEGO E DETECÇÃO DE ANOMALIAS UTILIZANDO A ANÁLISE DE COMPONENTES PRINCIPAIS E FLUXOS IP

LONDRINA - PR
2014

GILBERTO FERNANDES JUNIOR

CARACTERIZAÇÃO DE TRÁFEGO E DETECÇÃO DE ANOMALIAS UTILIZANDO A ANÁLISE DE COMPONENTES PRINCIPAIS E FLUXOS IP

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação Departamento de Computação da Universidade Estadual de Londrina.

Orientador: Mario Lemes Proença Jr.

LONDRINA - PR
2014

Catalogação elaborada pela Divisão de Processos Técnicos da Biblioteca Central da Universidade Estadual de Londrina

Dados Internacionais de Catalogação-na-Publicação (CIP)

F363c Fernandes Junior, Gilberto.
Caracterização de tráfego e detecção de anomalias utilizando a análise de componentes principais e fluxos IP / Gilberto Fernandes Junior. – Londrina, 2014.
96 f. : il.

Orientador: Mário Lemes Proença Junior.
Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Ciência da Computação, 2014.
Inclui bibliografia.

1. Redes de computadores – Medidas de segurança – Teses. 2. Sistemas de transmissão de dados – Tráfego – Teses. 3. Análise de componentes principais – Teses. 4. Redes de computadores – Administração – Teses. 5. Redes de computadores – Anomalias – Teses. I. Proença Junior, Mário Lemes. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDU 519.68.022

GILBERTO FERNANDES JUNIOR

CARACTERIZAÇÃO DE TRÁFEGO E DETECÇÃO DE ANOMALIAS UTILIZANDO A ANÁLISE DE COMPONENTES PRINCIPAIS E FLUXOS IP

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação Departamento de Computação da Universidade Estadual de Londrina.

BANCA EXAMINADORA

_____
Prof. Dr. Mario Lemes Proença Jr.
Universidade Estadual de Londrina

_____
Prof. Dr. Taufik Abrão
Universidade Estadual de Londrina

_____
Prof. Dr. Rodolfo Miranda de Barros
Universidade Estadual de Londrina

_____
Prof. Dr. Sylvio Barbon Jr.
Universidade Estadual de Londrina

Londrina, _____de _____de _____.

DEDICATÓRIA

A minha Família, que nunca deixou de estar ao meu lado, e a Deus, pois é graças à Ele que estou aqui.

AGRADECIMENTO

Aos meus Pais, que sempre me deram algo que é o mais importante de tudo. Mais importante que o dinheiro. Mais importante que um computador com internet. Mais importante que roupas e calçados. Eles me deram Amor. E isso é tudo pra mim. Obrigado.

Agradeço ao meu orientador e amigo, prof. Dr. Mario Proença Jr., pela amizade, pelo constante suporte, pela motivação, pelo profissionalismo, e pelos valores que aprendi ao longo desta orientação, que se extendem não somente à minha vida profissional, mas à toda mnha vida.

Aos meus colegas de turma, Luiz Fernando, Marcos e Eduardo, pela amizade e por todos os momentos vividos nestes dois anos de mestrado.

Agradeço a Universidade Estadual de Londrina juntamente ao Departamento de Computação, que me acolheram nos anos de graduação, e agora, no mestrado.

Agradeço ao Prof. Joel J.P.C. Rodrigues pela parceria e suporte em diversos trabalhos.

Agradeço ao amigo Alexandro M. Zacaron, pelas contribuições durante o desenvolvimento deste trabalho.

Agradeço também a todos os demais doscentes que contribuíram para minha formação profissional/acadêmica.

Agradeço a CAPES, pela concessão de uma bolsa de estudos.

Ao Circulo Católico Estrela da Manhã, o CCEM, um grupo de oração da Paróquia Nipo-Brasileira, ao qual também agradeço, por ser como uma segunda família para mim, onde Deus age em cada dia de minha vida através das grandes amizades que ali fiz.

E acima de tudo, agradeço a Deus, que me deixou provar de todo o seu Amor e mudou minha vida. Sem Ele, eu não teria forças para concluir mais esta etapa, pois nos momentos de fraqueza, de desânimo e de medo, Ele estava comigo, assumindo o controle e me dando forças para atravessar as tempestades, com a certeza de que a alegria e a vitória viriam pela manhã. À todos citados nesses agradecimentos, desejo que Deus os abençoe e dê muita paz, saúde e luz a cada um.

“ASSIM MESMO

Muitas vezes as pessoas são egocêntricas, ilógicas e insensatas.
Perdoe-as assim mesmo.

Se você é gentil, as pessoas podem acusá-lo de egoísta, interesseiro.
Seja gentil, assim mesmo.

Se você é um vencedor, terá alguns falsos amigos e alguns inimigos verdadeiros.
Vença assim mesmo.

Se você é honesto e franco, as pessoas podem enganá-lo.
Seja honesto assim mesmo.

O que você levou anos para construir, alguém pode destruir de uma hora para outra.
Construa assim mesmo.

Se você tem Paz e é Feliz, as pessoas podem sentir inveja.
Seja Feliz assim mesmo.

Dê ao mundo o melhor de você, mas isso pode nunca ser o bastante.
Dê o melhor de você assim mesmo.

Veja que, no final das contas, é entre você e DEUS.

Nunca foi entre você e as outras pessoas.”
Madre Teresa de Calcutá

“...É no alto que está a vossa meta, e não na terra...”
Col. 3,1

FERNANDES JR., Gilberto. **Caracterização de Tráfego e Detecção de Anomalias Utilizando a Análise de Componentes Principais e Fluxos IP**. 2014. 96. Dissertação de Mestrado em Ciência da Computação – Universidade Estadual de Londrina, Londrina, 2013.

RESUMO

Diversas técnicas e métodos distintos têm sido amplamente utilizados na área da detecção de anomalias em redes de computadores. Ataques, invasões ou falhas internas não detectadas de modo veloz e eficaz podem danificar seriamente todo um sistema de rede. Por este motivo, neste trabalho, será apresentado um sistema inteligente para detecção de anomalias baseado na Análise de Componentes Principais (PCA), um método estatístico para redução de dimensionalidade. A abordagem proposta gera um perfil de rede denominado *Digital Signature of Network Segment using Flow Analysis* (DSNSF), o qual descreve o comportamento normal do tráfego de rede por meio de uma análise de dados históricos extraídos de fluxos IP. Esse perfil é utilizado como um *threshold* para a detecção de anomalias de volume. O sistema proposto utiliza sete atributos presentes em fluxos IP, tais como bits, pacotes, número de fluxos, endereço IP de origem, endereço IP de destino, Porta TCP/UDP de origem e Porta TCP/UDP de destino, com o objetivo de detectar problemas e, em seguida, fornecer ao administrador de rede informações necessárias para resolvê-los. Com o uso de técnicas de avaliação que utiliza dados de tráfego de redes reais, os resultados indicaram uma previsão de tráfego consistente dos DSNSFs gerados pelo sistema e taxas de falso-positivo promissoras.

Palavras-chave: Caracterização de Tráfego; Detecção de Anomalia; DSNSF; PCA; Fluxos.

FERNANDES JR., Gilberto. **Traffic Characterization and Anomaly Detection Using Principal Component Analysis and IP Flows**. 2014. 96. Masters Dissertation in Computer Science – State University of Londrina, Londrina, 2013.

ABSTRACT

Distinct techniques and methods have been widely used in the subject of automatic anomaly detection in computer networks. Attacks, problems and internal failures not early detected may badly harm an entire Network system. For this intention, we propose an autonomous anomaly detection system based on the dimensionality reduction method Principal Component Analysis (PCA). Our approach creates a network profile called Digital Signature of Network Segment using Flow Analysis (DSNSF) that denotes the predicted normal behavior of a network traffic activity through historical data analysis extracted from IP flows. That digital signature is used as a threshold for volume anomaly detection to detect disparities in traffic normal trend. The proposed system uses seven traffic flow attributes, such as bits, packets, number of Flows, source IP address, estination IP address, source TCP/UDP Port and destination TCP/UDP Port, in order to detect problems, and then provide the network administrator necessary information to solve it. Via evaluation techniques performed in this paper using real network traffic data, results showed good traffic prediction by the DSNSF and promissing false-positive rates.

Keywords: Traffic Characterization; Anomaly Detection; DSNSF; PCA; Flows.

LISTA DE ILUSTRAÇÕES

Figura 3.1 - Comunicação gerente/agente: modo de notificação (traps) 26
Figura 3.2 - Comunicação gerente/agente: modo pedido/resposta (polling) 26
Figura 3.3 - Criação de um fluxo 29
Figura 3.4 - Formato básico de um datagrama NetFlow para as versões 1, 5, 7 e 8 30
Figura 3.5 - Exemplo de uma rede monitorada com IPFIX 33
Figura 3.6 - Exemplo de uma mensagem IPFIX 34
Figura 3.7 – Arquitetura de um dispositivo IPFIX 36
Figura 3.8 - Operação do sFlow dentro de um dispositivo de rede 37
Figura 3.9 - Esquema da comunicação entre o coletor e os agentes sFlow 38
Figura 3.10 - Formato de um datagrama sFlow 39
Figura 4.1 - Descrição geral do funcionamento do sistema de detecção de anomalias 42
Figura 4.3 - Exemplo de autovetores no plano 47
Figura 4.4 - Descrição do modelo para caracterização de tráfego 50
Figura 4.5 - Exemplo de fluxo 52
Figura 4.6 - Exemplo do formato de saída dos fluxos após processamento 52
Figura 4.7 - Comparação entre DSNSFs gerados com autovetores de variância máxima (a) e variância de valor intermediário (b) 56
Figura 4.8 - Índices NMSE calculados entre os DSNSFs e o tráfego real dos dias analisados utilizando de uma a dez semanas de treinamento para o tráfego de bits/s 58
Figura 4.9 - Índices NMSE calculados entre os DSNSFs e o tráfego real dos dias analisados utilizando de uma a dez semanas de treinamento para o tráfego de pacotes/s e número de fluxos/s 58
Figura 4.10 - Módulo de detecção de anomalias 61
Figura 5.1 – Exemplo de caracterização de tráfego na rede da Universidade Estadual de Londrina 66
Figura 5.2 - NMSE calculado entre o DSNSF e o tráfego real no período de 15 de Outubro a 9 de Novembro de 2012 67
Figura 5.3 - Coeficientes de Correlação calculados entre o DSNSF e o tráfego real no período de 15 de Outubro a 9 de Novembro de 2012 67
Figura 5.5 - Exemplo de alarmes gerados no tráfego de dois dias na rede da Universidade Estadual de Londrina para os três atributos quantitativos estudados 70

Figura 5.6 - Gráfico ROC mostrando os *trade-offs* entre os valores de TPR e FPR 71
Figura 5.7 - Taxa de acurácia para quatro semanas de testes 71
Figura 5.8 - Módulo de informação em uma simulação de Flash Crowd 73
Figura 5.9 - Módulo de informação em uma simulação de DDoS 73
Figura 5.10 - Exemplo de caracterização de tráfego para a rede da UTFPR, Campus Toledo 75
Figura 5.11 - NMSE calculados entre o DSNSF e o tráfego real no período de 29 de Abril a 17 de Maio de 2013 76
Figura 5.12 - Coeficientes de Correlação calculados entre o DSNSF e o tráfego real no período de 29 de Abril a 17 de Maio de 2013 76
Figura 5.14 - Exemplo de alarmes gerados no tráfego de dois dias na rede da Universidade Tecnológica Federal do Paraná - Campus Toledo para os três atributos quantitativos estudados 78
Figura 5.15 - Gráfico ROC mostrando os *trade-offs* entre os valores de TPR e FPR 78
Figura 5.16 - Taxa de acurácia para as três semanas de testes 79
Figura 5.17 - Módulo de informação em uma simulação de DoS 79
Figura 5.18 - Comparativo entre os DSNSFs produzidos pelos métodos PCADS e KMDS para uma semana 83
Figura 5.19 - Comparativo entre os métodos PCADS e KMDS utilizando o NMSE 84
Figura 5.20 - Comparativo entre os métodos PCADS e KMDS utilizando o teste de Correlação 85

LISTA DE TABELAS

Tabela 2.1 - Trabalhos relacionados 21
Tabela 3.1 - Áreas da Gerência 23
Tabela 3.2 - Principais RFC's do protocolo IPFIX 32
Tabela 5.1 - Definição dos três cenários utilizados na avaliação do sistema proposto 64
Tabela 5.2 - Parâmetros da simulação de anomalias no cenário 1 utilizando a ferramenta *Scorpius* 69
Tabela 5.3 - Parâmetros da simulação de anomalias no cenário 2 utilizando a ferramenta Scorpius 77

LISTA DE ABREVIATURAS E SIGLAS

AS	*Simulated Annealing*
ASIC	*Application-Specific Integrated Circuit*
CERN	*The European Organization for Nuclear Research*
CC	*Correlation Coefficient*
CoS	*Class of Service*
CPD	*Change-Point Detection*
DoS	*Denial of Service*
DDoS	*Distributed Denial of Service*
DSNSF	*Digital Signature of Network Segment using Flow Analysis*
DT	*Decision Tree*
EL	*Limite de Autovalor (Eigenvalue Limit)*
FPR	*False-Positive Rate*
IANA	*Internet Assigned Numbers Authority*
ID	*Identification*
IE	*Information Element*
IETF	*Internet Engeneering Task Force*
IP	*Internet Protocol*
IPFIX	*IP Flow Information Export*
ISP	*Internet Service Provider*
K-L	*Kullback-Leibler*
KM	*K-Means*
KMDS	*K-Means for Digital Signature*
KMIE	*K-Means algorithm based on Information Entropy*
MIB	*Management information base*
NMS	*Network Management System*
NMSE	*Normalised Mean Square Error*
PC	*Principal Component*
PCA	*Principal Component Analysis*
PCADS	*Principal Component Analysis for Digital Signature*
PSO	*Particle Swarm Optimization*
RFC	*Request for Comments*

ROC	*Receiver Operating Characteristics*
RTFM	*Realtime Traffic Flow Management*
SCTP	Stream Control Transmission Protocol
SNMP	*Simple Network Management Protocol*
SSO	*Simplified Swarm Optimization*
SVD	*Singular Value Decomposition*
SVM	*Support Vector Machine*
TCP	*Transmission Control Protocol*
TPR	*True-Positive Rate*
TTL	*Time to Live*
UDP	*User Datagram Protocol*
UEL	*Universidade Estadual de Londrina*
URL	*Uniform Resource Locator*
UTFPR	*Universidade Tecnológica Federal do Paraná*
VoIP	*Voice over IP*
XDR	*External Data Representation*

LISTA DE SÍMBOLOS

A	*Matriz quadrada de ordem n x p*
n	*Número de linhas da matriz A (instâncias)*
p	*Número de colunas da matriz A (variáveis)*
a	*Elementos da matriz A*
$\tilde{A}$	*Matriz A normalizada com média zero*
$\tilde{a}$	*Elementos da matriz $\tilde{A}$*
C	*Matriz de Covariância de ordem p*
c	*Elementos da matriz de covariância C*
$X1$	*Variável Aleatória*
$X2$	*Variável Aleatória*
$PC1$	*Componente principal (autovetor)*
$PC2$	*Componente principal (autovetor)*
B	*Matriz quadrada de ordem 2*
b	*Elementos da matriz B*
I	*Matriz identidade de ordem 2*
λ	*escalares não-nulos*
Q	*Matriz de autovetores ortonormais*
D	*Matriz diagonal de autovalores*
δ	*autovalores*
I_p	*Matriz identidade de ordem p*
$\tilde{Q}$	*Vetor característico*
k	*Número reduzido de variáveis da matriz A após o cálculo do PCA*
Z	*Vetor final dos dados iniciais projetados em k dimensões*
μ	*DSNSF resultante do Algoritmo 1*
t	*Intervalo de tempo*
W	*Matriz aleatória de ordem 5x3*
X	*Medida observada*

Y	*Medida prevista*
x	*elementos do vetor X*
y	*elementos do vetor Y*
EL_{sup}	*Limiar superior do DSNSF*
EL_{inf}	*Limiar inferior do DSNSF*
α	*Porcentagem obtida do autovalor associado ao autovetor*
ρ_{XY}	*Coeficiente de correlação normalizado calculado para as variáveis X e Y*
TP	*Instâncias positivas classificadas corretamente*
P	*Total de instâncias positivas*
TPR	*Taxa de verdadeiro-positivo*
TN	*Instâncias negativas classificadas corretamente*
N	*Total de instâncias negativas*
FPR	*Taxa de falso-positivo*
Acc	*Taxa de acurácia*
n	*Porcentagem a ser adicionada ou subtraída do EL*
J	*Distancia euclidiana*
S	*Número de pontos*
k	*Cluster*
P_S^k	*Valor dos pontos pertencentes ao cluster k*
c^k	*Centro do cluster k*
γ	*Número mínimo de pontos em um cluster*
$O(np^2)$	*Notação assintótica da complexidade do PCA*

SUMÁRIO

1 INTRODUÇÃO 14

2 TRABALHOS RELACIONADOS 17

3 GERÊNCIA DE REDES 23

3.1 ÁREAS DA GERÊNCIA 23

3.2 O PROTOCOLO SNMP 25

3.3 A UTILIZAÇÃO DE FLUXOS IP 26

3.3.1 NetFlow 29

3.3.2 IPFIX: *Internet Protocol For Information Export* 31

3.3.3 sFlow 37

4 SISTEMA DE DETECÇÃO DE ANOMALIAS 41

4.1 DESCRIÇÃO GERAL DO SISTEMA 41

4.2 MÓDULO DE CARACTERIZAÇÃO DE TRÁFEGO 43

4.2.1 Análise de Componentes Principais: O Método 43

4.2.1.1 Matriz de dados de entrada 44

4.2.1.2 Matriz de covariância 44

4.2.1.3 Autovetores e autovalores 45

4.2.1.4 Dados finais 48

4.2.2 PCADS: Análise de Componentes Principais para Assinatura Digital 49

4.2.2.1 Dados de entrada 51

4.2.2.2 Descrição do funcionamento do modelo 53

4.2.2.3 Parâmetros para criação do DSNSF 57

4.3 MÓDULO DE DETECÇÃO DE ANOMALIA 59

5 AVALIAÇÃO DO SISTEMA 62

5.1 MÉTRICAS DE AVALIAÇÃO 62

5.2 CENÁRIOS PARA AVALIAÇÃO DO SISTEMA 63

5.2.1 Cenário 1 – Universidade Estadual de Londrina (UEL) 64

5.2.1.1 Avaliação da caracterização de tráfego 65

5.2.1.2 Avaliação da detecção de anomalia 68

5.2.2 Cenário 2 – Universidade Tecnológica Federal do Paraná (UTFPR) 74

5.2.2.1 Avaliação da caracterização de tráfego 74

5.2.2.2 Avaliação da detecção de anomalias 77
5.2.3 Cenário 3 – Análise Comparativa Entre os Modelos PCADS e KMDS 80
5.2.3.1 KMDS: *K-Means for Digital Signature* 80
5.2.3.2 Análise comparativa 82
5.3 COMPLEXIDADE COMPUTACIONAL 85

6 CONCLUSÃO 87

1 INTRODUÇÃO

Atualmente, as redes de computadores são alvos diários de ataques e de atividades maliciosas que visam interromper ou inabilitar o tráfego e os seus serviços. Esses ataques comprometem a segurança do sistema e degradam o desempenho da rede, causando prejuízos aos usuários e às organizações [1] [2]. Um ataque de negação de serviço (DoS – *Denial of Service*) pode, por exemplo, causar sobrecarga no servidor e congestionar toda a rede com tráfego e requisições indesejadas. *Worms*, *spam*, *spoofing* e *cyber-crime* são outros exemplos de ameaças que podem comprometer as redes de computadores. No entanto, não são apenas os ataques e invasões que prejudicam o comportamento natural de uma rede.

À medida que as redes crescem em tamanho e ficam cada vez mais complexas, outros problemas como quedas de servidor, *bugs*, congestionamento de link, falhas de software e até mesmo a aleatoriedade do trafego, podem ser capazes de gerar anomalias no tráfego da rede [3] [4] [5] [6].

Anomalias são eventos inesperados no comportamento do tráfego de rede que diferem do comportamento padrão. Anomalias de volume se referem a picos na série temporal do tráfego de dados, ocasionados geralmente por *flash crowds* e *outage events*, problemas que podem acarretar em congestionamento, reduzindo o *throughput* da rede e aumentando o atraso nas comunicações. Visto isso, uma detecção de anomalias rápida e precisa é decisiva para garantir a operabilidade e a disponibilidade de grandes redes de computadores [7] [8].

Primeiramente, é necessário que o comportamento do tráfego de rede seja constantemente monitorado, visando uma detecção antecipada das divergências de comportamento e possibilitando uma rápida tomada de decisões e ações contra o problema. Entretanto, a realização de uma análise e de um monitoramento completos em sistemas de rede de larga escala é uma tarefa quase impraticável manualmente por um administrator de rede. As altas velocidades de conexão aliadas ao grande e crescente número de *links* e segmentos para se analisar ainda tornam essa tarefa ainda mais difícil. Desta forma, um sistema de monitoramento de tráfego para detecção de anomalias deve operar de forma autonômica, tendo em vista um gerenciamento de rede proativo.

Patcha et al. [9] classifica a detecção de anomalias de dois modos: baseado em assinatura (*signature-based*) e baseado em um perfil de rede (*profile-based*). Uma detecção baseada em assinatura identifica eventos anômalos com base em um conjunto

predefinido com os padrões (assinaturas) de diversos tipos de ataques. Uma detecção baseada em um perfil de rede se resume em criar um perfil de rede, *baseline*, que descreve o comportamento normal do tráfego, onde toda atividade que se difere deste *baseline* é considerada uma possível anomalia.

Neste trabalho, é proposto um sistema de gerenciamento de rede baseado em um modelo para caracterização de tráfego capaz de criar um perfil de rede que represente o comportamento normal do mesmo, visando identificar possíveis anomalias. Uma abordagem similar é proposta em [10], chamada de *change detection* (detecção de alterações), a qual detecta anomalia de tráfego por meio de um modelo que captura o comportamento normal de uma rede baseado em dados históricos.

A proposta geral do sistema baseado em perfil para detecção de anomalias proposto é composto por dois módulos: i) criação de um modelo que caracteriza o comportamento normal de tráfego por meio de dados históricos e ii) detecção de desvios comportamentais com acionamento de alarmes. Esta abordagem é capaz de trabalhar de forma proativa ao detectar o tráfego de rede anômalo de forma automática, sem supervisão humana. A isto dá-se o nome de Gerenciamento Autônomo de Rede (*Autonomic Network Management*), uma solução emergente que tem como finalidade antecipar, diagnosticar e contornar qualquer problema de funcionalidade em um sistema de rede de maneira independente e autônoma [11].

O sistema de monitoramento e detecção baseado em perfil proposto tem sua eficácia totalmente relacionada com a caracterização do comportamento do tráfego. Conforme relatado por Proença et al. [12], o tráfego de rede é composto por ciclos que consistem em rajadas (*traffic bursts*) que têm características particulares da sua utilização. Esses comportamentos são diretamente afetados pelas horas de trabalho e pelo período que os usuários acessam a rede. Portanto, o primeiro módulo tem o objetivo de, por meio do reconhecimento destes comportamentos e suas características, criar uma assinatura digital de tráfego chamada de Assinatura Digital de Segmento de Rede utilizando análise de Fluxo (DSNSF, *Digital Signature of Network Segment using Flow analysis*). Essa assinatura é responsável por conter informações sobre o comportamento do tráfego normal, sendo adotada para detecção de anomalias por reconhecer desvios comportamentais que diferem do usual. O modelo para caracterização de tráfego é chamado de *Principal Component Analysis for Digital Signature* (PCADS). A caracterização é realizada por meio da Análise de Componentes Principais (PCA) para identificar os intervalos de tempo do tráfego que possuam uma variância média dentre os dados históricos de treinamento.

A detecção de anomalias ocorre no segundo módulo, cujo perfil de rede criado é comparado e analisado juntamente com o tráfego que é transmitido pela rede em tempo real. A partir dessa análise, desvios comportamentais podem ser detectados como possíveis anomalias. Ainda, para minimizar o ocorrência de alarmes falsos, são extraídas informações da análise de componentes principais realizada pelo PCADS para serem utilizadas na fase de detecção como limiares adicionais ao DSNSF.

Todo esse trabalho foi realizado por meio da análise e da extração de sete atributos presentes em fluxos IP de um banco de dados histórico de uma rede real: bits, pacotes, número de fluxos, endereço IP de origem, endereço IP de destino, porta TCP/UDP de origem e porta TCP/UDP de destino. Padrões do comportamento do tráfego podem ser extraídos a partir da análise desses atributos, tornando possível identificar mudanças que ocorram no tráfego causado por comportamentos anômalos. De acordo com Zhou et al. [13], atividades de monitoramento que utilizam apenas um atributo tornaram-se cada vez menos eficazes, devido a crescente presença de ataques *stealth*. Certos tipos de ataques, como *Denial of Service* (DoS), causam alterações nos tráfegos de pacotes e número de fluxos e não em bits, e Flash Crowds, por sua vez, em bits, pacotes e fluxos.

Na literatura, destaca-se a utilização da Análise de Componentes Principais para detecção de anomalias como uma forma de separar os dados de tráfego em subespaços normais e anômalos, ou ainda para a extração de métricas visando reduzir ainda mais a dimensionalidade de métricas de desempenho. Como contribuição obtida neste trabalho, destaca-se a utilização de uma nova interpretação do PCA para detecção de anomalias, onde este é empregado para criar uma assinatura digital de segmento de rede (DSNSF) a fim de identificar desvios comportamentais no tráfego.

A divisão do trabalho vigente é a seguinte:

- Capítulo 1: introdução.
- Capítulo 2: revisão bibliográfica.
- Capítulo 3: conceitos sobre gerência de redes, abordando o protocolo SNMP e os protocolos de exportação de fluxos IP.
- Capítulo 4: descrição do sistema de detecção de anomalias proposto.
- Capítulo 5: avaliação do sistema por meio do uso de dados reais e métricas de avaliação.
- Capítulo 6: conclusões e trabalhos futuros.

2 TRABALHOS RELACIONADOS

Segurança é um tópico de grande importância e de ampla pesquisa dentro da área de gerenciamento de redes. Atualmente, existem diferentes tipos de métodos e de modelos para detecção de anomalia e de intrusão que se baseiam em vários tipos de algoritmos e técnicas.

Xin Xu [14] apresenta um método sequencial para detecção de anomalia utilizando aprendizagem com diferenças temporais, e também, funções de recompensa projetadas em modelos Markovianos sequenciais. O método de detecção proposto por Xin Xu proporciona um novo *framework* para a detecção de anomalias em cyber ataques de múltiplos estágios, podendo ser aplicado para detecção em dados sequenciais.

Lin et al. [15] fazem uso de *Support Vector Machine* (SVM), Árvore de Decisão (DT) e *Simulated Annealing* (SA), a fim de propor um algoritmo inteligente para detecção de intrusão, com seleção de características e regras de decisão. A técnica de *Simulated Annealing* fornece melhores definições de parâmetros para as Árvores de Decisão e SVM, enquanto que a combinação das técnicas SVM e SA possibilita a localização da melhor seleção de características e recursos para elevar a acurácia do sistema de detecção de anomalias.

Em [16], os autores aperfeiçoam a eficiência de um modelo de detecção com o uso de novas técnicas de amostragem de fluxo, como a Amostragem Inteligente (*Smart Sampling*), focando na seleção de pequenos fluxos, os quais são geralmente a origem do tráfego anômalo (ataques DDoS, *portscans*, propagação de *worms*). Para a detecção de anomalias, é utilizado o modelo de detecção *Change-Point* (CPD, *Change-Point Detection*), cujo objetivo consiste em determinar se a série temporal observada é estatisticamente homogênea e, caso contrário, localizar os intervalos de tempo onde as mudanças ocorrem. Os resultados indicaram um aumento significante na taxa de detecção ao se utilizar técnicas de amostragem, em comparação ao não uso de amostragem.

Em [17], os autores analisam sinais para quatro classes de anomalias (interrupções, flash *crowds*, ataques e falhas de medição), utilizando fluxos IP e medições do SNMP com filtros *wavelet* para separar o tráfego normal da porção anômala. Ainda, é apontado que um modo efetivo para encontrar anomalias se dá por meio da detecção de aumentos acentuados na variância local dos dados.

Li Han [18] propõe o modelo KMIE (*K-Means algorithm based on Information Entropy*), uma combinação do algoritmo de clusterização K-Means com entropia. O KMIE filtra os *outliers* presentes nos dados com a finalidade de reduzir o impacto negativo causado por eles e identifica os centros da clusterização utilizando entropia. Assim, o KMIE utiliza os centros para calcular e classificar de forma iterativa os registros observados em diferentes clusters.

Yasami et al. [19] desenvolve um sistema estatístico de detecção de anomalia utilizando Autômato de Aprendizagem Estocástica. A abordagem constrói um autômato de aprendizagem a partir de dados de séries temporais de tráfego *broadcast* de rede extraído no processo de aprendizagem. A partir disso, as séries temporais do tráfego de *broadcast* observados são comparadas ao modelo normal criado e qualquer desvio é marcado como tráfego anormal.

O trabalho de Kind et al. [20] descreve uma nova abordagem para detecção de anomalias baseado em atributos (*features*), em que são gerados histogramas a partir das características detalhadas de cada atributo do tráfego como endereços IP e Portas TCP/UDP. Um histograma pode ser criado a partir de um atributo isolado – como o número de uma porta – a partir de um conjunto de valores de atributos – como uma faixa de endereços IP –, ou ainda, a partir de valores conjuntos de múltiplos atributos, como o número de uma porta e uma faixa de endereços IP. Ao observar histogramas de atributos, é possível identificar padrões que refletem no comportamento normal da rede, e ainda, observar que anomalias podem distorcer o padrão normal de um ou mais atributos. Desse modo, os histogramas são utilizados em comparações com o comportamento online da rede para identificar desvios e gerar alarmes.

Um novo framework para detecção de intrusão utilizando mineração de dados e lógica *fuzzy* é proposto em [21]. Ainda, uma nova versão do algoritmo PSO (*Particle Swarm Optimization*), chamada de *Simplified Swarm Optimization* (SSO), é aplicada na criação de um sistema híbrido de detecção utilizando busca local e mineração de comportamentos anômalos. O algoritmo PSO também é utilizado por Lima et al. [22], e aplicado em uma abordagem com clusterização, visando encontrar comportamentos anômalos no volume de tráfego. O DSNS (*Digital Signature of Network Segment* – Assinatura Digital de Segmento de Rede), criado a partir de objetos SNMP, é utilizado para produzir *thresholds* do comportamento normal do tráfego de rede. Embora tenha alcançado ótimos resultados, o trabalho de Lima et al. possui certa limitação devido ao uso de objetos SNMP. A utilização de um gerenciamento de redes baseado em análise de fluxos, como é abordada neste trabalho,

possibilita uma análise de tráfego mais detalhada, uma vez que os fluxos possuem uma grande variedade de informações a respeito do comportamento da rede.

No trabalho de Rawat et al. [23], a metodologia empregada no desenvolvimento de um sistema de detecção de intrusão é similar à metodologia utilizada neste trabalho. Os autores tem como base o pressuposto de que, em uma rede, atividades anormais (intrusões) diferem substancialmente das atividades normais. Deste modo, toda execução normal de um processo segue um padrão, o qual pode ser adotado como um perfil de rede, em que qualquer desvio observado pode ser sinalizado como uma intrusão. Assim, para medir a similaridade entre processos, os autores propõem o uso da distância *Kendall Tau*, considerando três questões chaves (ocorrências individuais de chamadas de sistema (*system calls*), frequência de determinada chamada de sistema em um processo e a posição desta chamada de sistema no processo), as quais possibilitam resultados promissores em relação a taxas de falsos positivos.

A utilização de técnicas de mineração de dados, combinadas com sistemas inteligentes na elaboração de sistemas de detecção de intrusão eficientes, é apresentado em Sodiya et al. [24]. O objetivo dos autores é extrair padrões de dados de auditoria para um comportamento de usuário consistente e útil para, em seguida, manter esses comportamentos em perfis normais. A detecção de eventos anômalos ocorre a partir da comparação dos registros atuais com os perfis armazenados pelo sistema em um banco de dados. Essa base de conhecimento é obtida por meio de modelos prévios e problemas existentes em sistemas de detecção presentes na literatura. Um alarme é gerado se as atividades e parâmetros expressados pelos registros atuais não se igualam ao perfil normal, informando a ocorrência de anomalia e interrompendo instantaneamente o evento. Entretanto, se a atividade não possui natureza intrusiva, o perfil normal é alimentado com esse registro, podendo ser utilizado em detecções futuras.

Grande parte dos pesquisadores empregam estratégias baseadas em assinaturas ou aprendizado supervisionado para construir modelos de detecção de anomalias. Essas abordagens possuem algumas desvantagens como a necessidade de dados rotulados e de supervisores externos, o que acarreta em resultados insatisfatórios quando há a ocorrência de anomalias desconhecidas. Deste modo, Mazel et al. [25] introduz uma abordagem não supervisionada para detectar e caracterizar anomalias de rede. Inicialmente, essa abordagem utiliza uma técnica de clusterização, combinando clusterização de subespaço com Acumulação de Evidência (*Evidence Accumulation*) ou Associação de Resultados de *Inter-Clustering*, para identificar às cegas nos fluxos de tráfegos.

Neste trabalho, a detecção de anomalias é realizada por meio da análise de fluxos IP e de suas características. Em [26], Duffield et al. exploram a correlação entre os pacotes e as informações a níveis de fluxo por meio de uma abordagem utilizando aprendizado de máquina que pode associar alarmes de nível de pacote com um vetor de características derivado dos registros de fluxo no mesmo tráfego. Ainda, Fontugne et al. [27] desenvolve um método de detecção de anomalia baseado em reconhecimento de padrão, em que se analisa a relação entre seus parâmetros definidos e as variações de tráfego e suas características.

Rossi et al. [28] desenvolveram um algoritmo que explora os chamados fluxos comportamentais (*behavioral flows*) para a classificação de tráfego de rede. O algoritmo desenvolvido, chamado *Abacus*, procura identificar uma aplicação usando dois campos dos registros de fluxos – bytes e pacotes. Os resultados indicam uma precisão de 90% no pior caso para o volume de tráfego.

A Análise de Componentes Principais (PCA) é uma técnica amplamente utilizada para a detecção de anomalias em redes de computadores, sendo utilizada em diversas abordagens.

Lakhina et al. [29] [30] são os pioneiros no uso do PCA aplicado à segurança de rede. Em seu trabalho, Lakhina aborda o problema da detecção e identificação de anomalias em redes de larga escala utilizando o PCA para separar de forma eficiente o tráfego em dois subespaços – normal e anômalo. O subespaço anômalo, o qual possui mais ruídos e contém os picos de trafego mais significativo, é separado do tráfego normal da rede, esse que apresenta o comportamento normal previsto para o tráfego da rede. Assim, é possível diagnosticar de forma precisa anomalias de volume quando a magnitude da projecção sobre o subespaço anômalo excede um limiar *Q-statistic*.

O método de detecção, proposto por Pascoal et al. [31] utiliza uma versão robusta do algoritmo PCA combinado com um algoritmo robusto de seleção de características, visando mais adaptabilidade a diferentes ambientes e condições de rede. Ainda, essa abordagem robusta do PCA não requer dados de treinamento perfeitos, condição esta discutida em [32] como uma das limitações do método clássico do PCA.

Em [33], os autores propõem o ADMIRE, uma combinação de *sketches* com PCA baseado em entropia, resultando em melhores taxas de verdadeiro e falso positivo. Isso, pois é possível capturar diferentes tipos de anomalias devido a diferenças na série temporal da entropia do PCA.

Kanda et al. [34] combina o uso de *sketches* (projeções randômicas de tráfego) e a Análise de Componentes Principais (PCA) para criar um novo método a fim de detectar e identificar os endereços IP associados com a origem do tráfego anômalo em *backbones*. Ainda, para garantir que as anomalias identificadas pelo método são realmente provenientes de conexões maliciosas, os autores desenvolveram um método heurístico que utiliza as portas, as *flags* do protocolo TCP e padrões de comunicação, com o intuito de classificar cada endereço IP em três categorias: Especial, Ataque e Desconhecido.

Callegari et al. [35] implementam algumas inovações e melhorias a partir do método PCA desenvolvido por Lakhina et al. [29]. Os autores utilizam medidas de entropia juntamente com a divergência *Kullback-Leibler* (K-L) para construir séries temporais a partir de fluxos agregados e detectar comportamentos anômalos. Essa abordagem do sistema de detecção apresentou resultados mais apurados, além de um desempenho mais estável.

A Tabela 2.1 resume os trabalhos apresentados neste capítulo.

Tabela 2.1 - Trabalhos relacionados

	Autor Principal	**Local de publicação**	**Ano**	**Tipo de trabalho**	**Técnicas utilizadas**
1	Xu, X.	Applied Soft Computing	2010	Detecção de anomalia	Aprendizagem com diferenças temporais e modelos Markovianos sequenciais
2	Lin, S-W	Applied Soft Computing	2012	Detecção de intrusão	*Support Vector Machine* (SVM), Árvore de Decisão e *Simulated Annealing* (SA)
3	Androulidakis, G.	IEEE GLOBECOM	2007	Detecção de anomalia	Técnicas de amostragem e *Change-Point Detection*
4	Barford, P.	ACM SIGCOMM	2002	Detecção de anomalia	Análise de sinais por meio de filtros *wavelet*
5	Han, L.	IEEE MINES	2012	Detecção de Anomalia	Clusterização *K-Means* e medidas de entropia
6	Yasami, Y.	IEEE ICT	2008	Detecção de Anomalia	Autômato de Aprendizagem Estocástica
7	Kind, A.	IEEE Transactions on Network and Service Management	2009	Detecção de Anomalia	Criação de histogramas a partir das características detalhadas de cada atributo do tráfego
8	Tajbakhsh, A.	Applied Soft Computing	2009	Detecção de intrusão	Lógica *fuzzy* e PSO (*Particle Swarm Optimization*)

Tabela 2.1 - Trabalhos relacionados

	Autor Principal	Local de publicação	Ano	Tipo de trabalho	Técnicas utilizadas
9	Lima, M.F.	IEEE GLOBECOM	2010	Detecção de anomalia	PSO (*Particle Swarm Optimization*) com busca local
10	Rawat, S.	Information Management & Computer Security	2004	Detecção de Intrusão	Distância *Kendall Tau*,
11	Sodiya, A.S.	Information Management & Computer Security	2004	Detecção de Intrusão	Mineração de dados e sistemas inteligentes
12	Mazel, J.	IEEE CNSM	2011	Detecção de anomalia	Técnicas de clusterização em uma abordagem não-supervisionada
13	Duffield, N.	IEEE INFOCOM	2009	Detecção de anomalia	Aprendizado de máquina
14	Fontugne, R.	ACM SAC	2011	Detecção de anomalia	Reconhecimento de padrão de atributos de fluxos
15	Rossi, D.	ACM IWCMC	2010	Classificação de tráfego	*Behavioral flows* (Fluxos comportamentais)
16	Lakhina, A.	ACM SIGCOMM	2004	Detecção e diagnostico de anomalias	PCA (*Principal Component Analysis*)
17	Lakhina, A.	ACM SIGCOMM	2005	Detecção de anomalia	PCA (*Principal Component Analysis*)
18	Pascoal, C.	IEEE INFOCOM	2012	Detecção de Anomalia	Versão robusta do PCA combinada com algoritmo de seleção de características
19	Kanda, Y.	Computer Communications	2013	Detecção de Anomalia	Combinação de *sketches* com PCA baseado em entropia
20	Kanda Y	IEEE GLOBECOM	2010	Detecção de Anomalia	Combinação de *sketches* com PCA e método heurístico utilizando *flags* TCP e padrões de comunicação
21	Callegari, C.	IEEE ICC	2011	Detecção de Anomalia	PCA, medidas de entropia e divergência *Kullback-Leibler* (K-L)

3 GERÊNCIA DE REDES

As redes já são de extrema importância nos dias de hoje, principalmente para empresas, não sendo mais infraestrutura dispensável, mas sim de missão crítica, ou seja, não podem parar. Assim, são necessárias diversas boas soluções capazes de manter as redes funcionando bem a todo o momento, e a estas soluções dá-se o nome de Gerência de Redes.

A Gerência de Redes é definida como um processo de controle com uma ampla variedade de funções, incluindo atividades, métodos, procedimentos e utilização de ferramentas para administrar, operar e manter os sistemas de rede com segurança. Esse controle é completo, de modo que se estende tanto aos recursos materiais, como *switches* e roteadores, quanto aos recursos lógicos, como protocolos. Otimização do serviço de comunicação, otimização do uso de recursos e controle de gastos, diminuição do tempo de indisponibilidade, monitoramento de tráfego e detecção de falhas são alguns dos principais objetivos da gerência de redes [36].

3.1 ÁREAS DA GERÊNCIA

A gerência de redes é constituída de cinco áreas para a modelagem do gerenciamento e implementação de redes de computadores. Estas cinco áreas são descritas pelo acrônimo F.C.A.P.S. [37] [36], como pode-se observar na Tabela 3.1.

Tabela 3.1 - Áreas da Gerência

F	*Fault* (Gerenciamento de falhas)
C	*Configuration* (Gerenciamento de configuração)
A	*Accouting* (Gerenciamento de contabilidade)
P	*Performance* (Gerenciamento de desempenho)
S	*Security* (Gerenciamento de segurança)

O Gerenciamento de falhas engloba os processos de detecção, localização e correção de problemas de hardware ou software em uma rede. Entende-se “falha” como uma condição anormal que requer algum tipo de reparo, e é indicada pelo fracasso de alguma operação ou pela ocorrência de vários erros seguidos. O gerenciamento de falhas deve ser capaz de enxergar os dispositivos de rede de forma separada para determinar se estão trabalhando corretamente, ou seja, a capacidade de isolar e corrigir falhas. Também é necessário aos sistemas de gerenciamento que tenham o conhecimento das “características das falhas” de cada dispositivo da rede.

Outra característica importante desta área é a questão da filtragem de falhas, onde o objetivo é priorizar determinadas falhas encontradas para não sobrecarregar a rede. Ainda, temos sistemas de gerenciamento que focam na antecipação de falhas, por meio da programação de diagnósticos rotineiros pré-definidos executados periodicamente em cada dispositivo, correlação de alarmes, etc.

O Gerenciamento de configuração conhece e tem o controle do estado do complexo formado pelas redes de uma instituição. Sua função é fornecer informação segura e atualizada sobre os itens de configuração em uso, contribuindo para a inter-relação dos mesmos. A área engloba três subáreas principais: inicialização e manutenção dos diferentes dispositivos que compõem a rede; manutenção do estado de cada dispositivo da rede; e monitoramento do inter-relacionamento entre os objetos da rede (topologia da rede).

Essa área inclui a ativação de todos os dispositivos de rede, alteração de suas configurações, e também mecanismos de autorização, que são importantes para verificar e registrar qualquer mudança ocorrida na rede, assim como determinar o responsável por elas.

O gerenciamento de contabilidade é a gerência de registros, logs ou bilhetes. Seu objetivo é medir (contabilizar) a utilização dos recursos de uma rede com o uso de vários algoritmos, levando em consideração parâmetros como a duração da conexão, o tráfego gerado, a identificação dos usuários, etc. Também inclui métodos para o cálculo do custo de funcionamento de uma determinada rede, orçamento, verificação e faturamento de serviços. A área de gerenciamento de contabilidade é muito utilizada por provedores de acessos (ISPs) na tarifação de serviços.

O gerenciamento de desempenho trata do uso da rede, visando garantir a qualidade do funcionamento da rede por meio do monitoramento de variáveis chaves, como *throughput*, disponibilidade e tempo de resposta. A partir disso, definir onde e como o desempenho da rede pode ser melhorado. Existem duas categorias funcionais dentro do conceito de gerenciamento de desempenho, são elas: monitoração e ajuste. A função de

monitoração localiza atividades na rede, e a função de ajuste possibilita a realização de alterações que visam melhorar o desempenho da rede.

Por último, o gerenciamento de segurança é a área que trata do regulamento e da administração do acesso aos recursos de rede, incluindo a verificação dos privilégios de acesso à rede, detectar e registrar tentativas de acesso não autorizado. Esta área é fundamental para administradores e usuários hoje em dia.

3.2 O PROTOCOLO SNMP

O SNMP (*Simple Network Management Protocol* – Protocolo de Gerência Simples de Rede) foi desenvolvido pela IETF (*Internet Engineering Task Force*), como o protocolo padrão para gerenciamento de redes. O protocolo é utilizado pelos administradores de rede, para realizar consultas específicas aos equipamentos que fazem parte da rede (computadores, roteadores, etc.), podendo auxiliar na detecção de falhas, no monitoramento do desempenho e no controle de configurações. Ainda, o uso do SNMP permite o acompanhamento em tempo real do estado da rede gerenciada [38].

Ele utiliza um modelo chamado de "gerente/agente", onde o *gerente* é uma aplicação que roda em uma estação (servidor) de gerenciamento (NMS – *Network Management System*), e se comunica com o *agente* (aplicação presente no dispositivo a ser gerenciado) através do protocolo de gerencia. O transporte na comunicação entre *gerente* e *agente* é realizado pelo protocolo UDP.

O processo básico da comunicação *gerente*/*agente* consiste no envio de *traps* ao gerente (notificações ou informações de alteração de status), que ao recebê-las, requisita uma consulta ao *agente*, que responde o *gerente* com as informações requisitadas (Figura 3.1). Também é possível ao *gerente* buscar periodicamente informações no *agente* através de um processo chamado *polling* (Figura 3.2). Muitas das informações concedidas aos *gerentes* estão relacionadas a aspectos de segurança e outros aspectos restritos. Por isso, ao requisitar uma consulta, o *gerente* sempre fornece uma espécie de senha ao *agente*, que verifica a identidade do *gerente* sempre que for enviar informações [38].

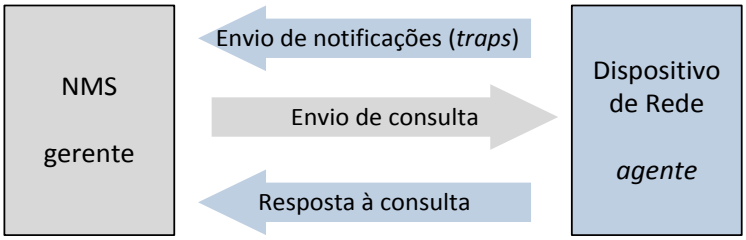

Figura 3.1 - Comunicação gerente/agente: modo de notificação (traps)

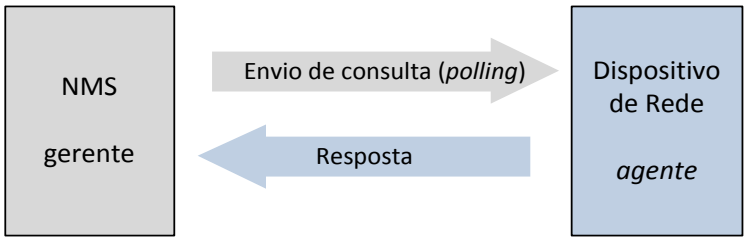

Figura 3.2 - Comunicação gerente/agente: modo pedido/resposta (polling)

O SNMP trabalha com objetos gerenciados, onde cada objeto é a visão abstrata de um recurso real do sistema. Cada objeto possui permissões para leitura (representar o estado real do objeto) e alteração (que se reflete no recurso). Os objetos são descritos em cinco partes: Objeto (nome simbólico do objeto e identificador), Sintaxe (sintaxe abstrata do objeto, que define o tipo de dado), Definição (texto que define o significado do objeto), Acesso (permissões de acesso ao objeto) e Status (necessidade de implementação do objeto: obrigatório, opcional ou obsoleto). O conjunto de todos os objetos de gerência é conhecido como MIB (*Management Information Base*).

A MIB é definida como um repositório de informações necessárias para realizar a gerência de uma rede. Ela possui uma lista com todos os objetos gerenciados, assim como os atributos relacionados a cada um. Cada agente do modelo de gerência SNMP deve implementar uma MIB que represente todos os recursos que estão associados a ele, e os gerentes devem ter conhecimento do mesmo esquema conceitual dos agentes, para que a troca de informações seja possível [38].

3.3 A UTILIZAÇÃO DE FLUXOS IP

O sistema de gerência proposto neste trabalho foi desenvolvido a partir da análise de fluxos IP, uma tecnologia de gerência mais completa que vem sendo utilizada como

alternativa ao protocolo SNMP. O protocolo SNMP é amplamente utilizado em ferramentas tradicionais de gerência de redes para medir parâmetros de desempenho, como por exemplo, contadores de erros em interfaces e volume de tráfego. Porém, com o aparecimento de novos serviços e o aumento da complexidade das redes surgiu a necessidade de se obter um maior detalhamento das informações dos dados trafegados, algo essencial para o entendimento do comportamento de aplicações, dos usuários, dos departamentos de empresas e de outras estruturas que dependem das redes para seu funcionamento. Neste sentido, ao utilizar fluxos a gerência permite formar uma base de dados com informações detalhadas e essenciais do tráfego, possibilitando o entendimento de aspectos mais subjetivos do funcionamento de uma rede [26] [27].

Desta forma, se fez necessário um protocolo que fosse além dos contadores de bits e pacotes do SNMP, de modo que fosse capaz de caracterizar o tráfego de forma mais detalhada, revelando tendências e comportamentos da rede. Então, em 1996, a *Cisco Systems* apresentou o protocolo NetFlow [39], que introduziu uma nova técnica para auxiliar na gerência de redes, pois era totalmente baseado em fluxos IP. Esse novo tipo de medição de tráfego captura o trafego que passa pelos roteadores e *switches* e os agrega em fluxos IP, e em seguida, os exporta a uma entidade responsável por analisar os dados presentes nos fluxos. Os protocolos de extração de fluxos mais comuns são NetFlow, da Cisco [39]; IPFIX, da IETF (*Internet Engineering Task Force*) [40] e sFlow, da InMon [41].

Cada fluxo possui informações detalhadas sobre sua origem e destino, o que possibilita análises que envolvam o comportamento do protocolo de transporte em uma conexão. Monitorar uma rede através de fluxos IP também dá acesso aos administradores de rede a um nível mais agregado de informações (domínio, rede ou subrede), possibilitando uma maior capacidade para caracterizar uma rede, além de proporcionar um planejamento mais preciso e garantir o uso correto de recursos. Tudo isso é possível devido a grande gama de informações que os fluxos são capazes de proporcionar.

Um fluxo IP é, basicamente, uma sequência unidirecional de vários pacotes IP com atributos semelhantes. Os atributos dos pacotes IP utilizados são:

- Endereço IP de origem.
- Endereço IP de destino.
- Porta de origem.
- Porta de destino.
- Tipo de protocolos de transporte (camada 3).
- Tipo de serviço *Class of Service* (CoS).
- Interface do roteador/switch.

Os endereços IP e origem e destino permite saber onde o tráfego originou-se e quem o receberá. Já as portas, revelam a aplicação que gerou os pacotes. *Class of Service* caracteriza a prioridade do tráfego. A interface do roteador/switch mostra como o tráfego está sendo utilizado pelos dispositivos de rede.

Como cada um desses atributos funciona como uma espécie de identidade para os pacotes IP, determinando se ele é único ou não, cada pacote com os mesmos atributos, como endereço IP de origem e destino, portas de origem e destino, CoS, etc, são agrupados em um fluxo. A construção de um fluxo se dá então pelo agrupamento destes pacotes com atributos semelhantes dentro de um intervalo de tempo de 15 segundos (valor *default* na maioria dos roteadores), e caso este tempo expire, o fluxo termine, e um novo se inicia. Também com relação ao término de um fluxo, existem os pacotes RST e FIN, que podem ser utilizados para essa finalidade, mas apenas em conexões TCP.

Além dos atributos já citados acima, algumas informações adicionais podem ser agregadas a um fluxo, como por exemplo, a máscara da subrede, ou os *timestamps* do fluxo, importantes para o cálculo do tempo de vida de um fluxo e para calcular o número de pacotes e bytes por segundo. Por padrão, o tempo de vida máximo de um fluxo é configurado em 30 minutos na maioria das interfaces [42].

A Figura 3.3 ilustra o modo que os fluxos são gerados à medida que um roteador/*switch* configurado com um protocolo de exportação de fluxo recebe pacotes do tráfego gerado por uma rede. Ao receber um pacote, seus atributos são verificados, e caso haja algum fluxo ativo com atributos semelhantes (IP, portas, protocolo, etc.), as informações do pacote são agregadas ao fluxo. Caso contrário, um novo fluxo é criado. Após esse procedimento, o pacote continua seu caminho até seu destino.

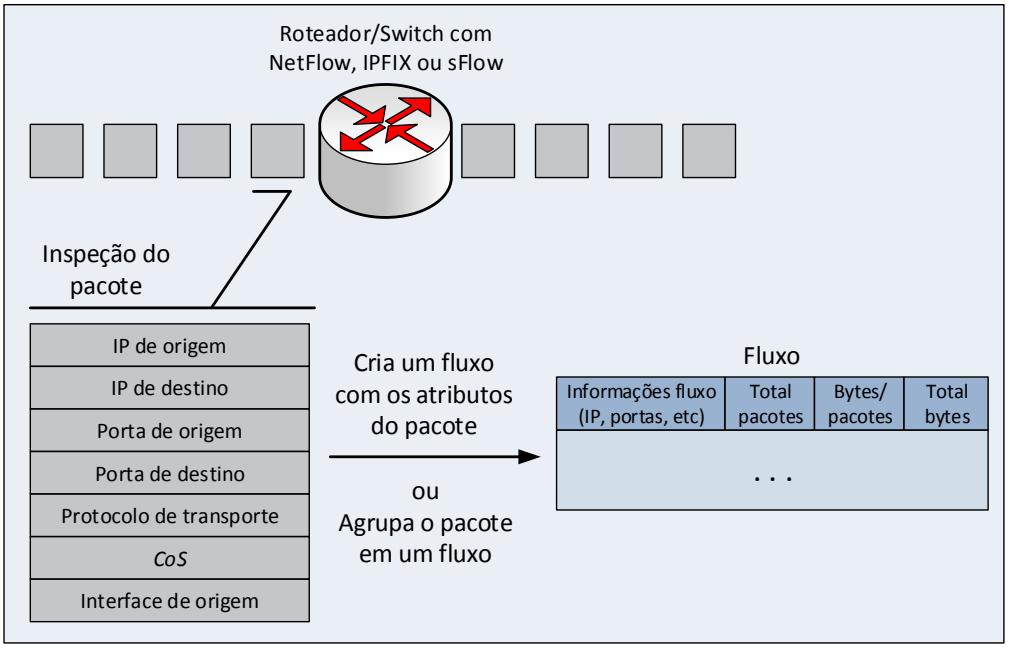

Figura 3.3 - Criação de um fluxo

3.3.1 NetFlow

A fim de coletar e reunir dados importantes e detalhados a respeito do funcionamento de uma rede, já que o protocolo SNMP não oferecia um detalhamento maior dos dados coletados, a Cisco Systems, em 1996, criou o protocolo NetFlow [39] [42], para monitoramento do tráfego IP presente em roteadores e *switches* com o uso da análise de fluxo. Este protocolo atua em switches e roteadores Cisco, onde gera informações sobre cada fluxo IP individual, gerando um registro contendo todas as informações necessárias para uma análise completa da rede, podendo ser utilizado ser utilizado para o gerenciamento e caracterização de uma rede, no combate a ataques *Denial of Service* (DoS), data mining, revelar tendências de rede, etc.

Vale ressaltar que este protocolo permite aos administradores de rede compreender como o tráfego está fluindo, além de possibilitar a análise de produtividade de uma rede, conhecer o uso dos recursos oferecidos por uma rede, o impacto gerado por mudanças na rede, e os pontos vulneráveis de uma rede, auxiliando na segurança e detecção de anomalias [39].

O NetFlow possui dois componentes principais. O primeiro é o que chamamos de NetFlow cache, que é onde as informações dos fluxos IP são armazenadas. O segundo é o mecanismo de exportação (ou transporte), o qual envia os dados a um coletor para relatar os dados.

O uso do NetFlow permite analisar o impacto de novas aplicações à rede, como o VoIP, analisar o tráfego WAN, a fim de entender quem está utilizando a rede e quem são os top users, diagnosticas problemas de desempenho em uma rede analisando a utilização de largura de banda, detectar tráfego WAN não autorizado, monitorar redes, visão detalhada de uma rede, planejamento de rede, segurança, *accounting*, etc.

A partir do conjunto de dados coletados de um fluxo, são construídos os datagramas de exportação, que são as unidade de dados exportadas pelo NetFlow. Essa exportação é realizada pelo protocolo de transporte UDP (por isso é que são utilizados datagramas). Cada datagrama contém um cabeçalho e uma sequência de registros de fluxo. O cabeçalho é composto por números de sequência, contador de registros e o *sisUpTime*. Os registros de fluxo contém informações à respeito do fluxo coletado, como os endereços IP, as portas, informações de roteamento, número de pacotes, sisUpTime, entre outros. A estrutura básica de um datagrama NetFlow é ilustrada na Figura 3.4.

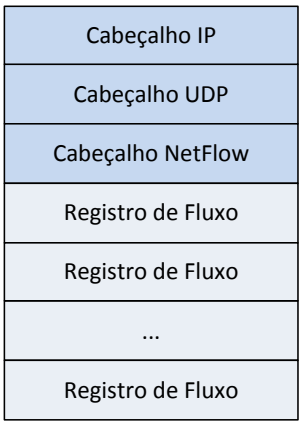

Figura 3.4 - Formato básico de um datagrama NetFlow para as versões 1, 5, 7 e 8

Essa estrutura de datagrama corresponde apenas às versões 1, 5, 7 e 8 do NetFlow, sujeitas apenas a poucas modificações entre cada versão. O formato do datagrama para a versão 9, a mais recente, possui uma estrutura diferente.

Em sua versão 9 [42] [43], o NetFlow traz diferenças no formato dos registros de fluxo, quando comparado às versões anteriores. A novidade está por conta do uso de *templates* para acessar os dados extraídos dos fluxos, assim como no protocolo IPFIX. O

uso de *templates* possibilita observar os dados dos fluxos de maneira flexível, e extensível, já que esta estrutura é definida por um conjunto de campos que descrevem a estrutura e a semântica dos respectivos campos nos registros de dados.

Os pacotes de exportação do NetFlow são encapsulados em datagramas UDP para o envio ao coletor. Isso é feito para aumentar o rendimento em termos de processamento no dispositivo NetFlow quando este lida com um grande volume de pacotes de exportação. Entretanto, essa versão não precisa necessariamente encapsular seus pacotes por meio de protocolos de transporte, como UDP ou SCTP, já que foi projetada para operar independente de protocolos de transporte.

O formato de um pacote de exportação do NetFlow é semelhante ao do IPFIX, que será apresentado na próxima seção. Essa semelhança se deve ao fato do NetFlow v9 ter sido escolhido pela IETF (*Internet Engeneering Task Force*) como base para a criação do padrão de exportação de fluxos IPFIX. O pacote possui um cabeçalho, e em seguida, conjuntos de *templates*, registros de dados e/ou *options templates*.

O conjunto de *templates* presente nos pacotes NetFlow v9 são de extrema importância, pois são eles que garantem a flexibilidade do formato do registro. Isso porque eles permitem ao coletor processar os dados presentes nos registros de fluxo sem a necessidade do coletor saber a interpretação de cada campo, já que isto fica a cargo dos *templates*. São nos conjunto de dados que estão os dados propriamente ditos dos fluxos observados. É aqui que estão os dados descritos pelos *template*s. Entretanto, estes dados só podem ser interpretados se o conjunto de *template*s correspondente também estiver disponível no coletor.

3.3.2 IPFIX: *Internet Protocol For Information Export*

Em 1990, quando a IETF criou o RTFM (*Realtime Traffic Flow Management*) [44], um padrão baseado no SNMP para medição de fluxo, cada proprietário programava em seu roteador um protocolo diferente para exportação de fluxo, e vários protocolos surgiram, como o NetFlow e sFlow. Como solução a esse problema, em 2001, a IETF começou a desenvolver um protocolo que padronizasse a exportação de fluxos, definindo um modelo de dados mais flexível, e que atuasse independente do protocolo de transporte ou do formato das mensagens utilizados [45].

Dentre todos os protocolos de exportação existentes na época, a IETF selecionou a versão 9 do protocolo NetFlow como base para o desenvolvimento do IPFIX pois se adequava melhor aos requisitos descritos na RFC 3917 [46]. Por isso, estes dois protocolos apresentam certas semelhanças, como o uso de *templates* nos registros de fluxo, o que garante maior flexibilidade na construção de uma mensagem IPFIX.

Basicamente, o IPFIX é constituído por 3 (três) tipos de processos: Processo de Medição (*Metering Process*), onde o fluxo é gerado a partir dos cobeçalhos e características dos pacotes coletados; Processos de Exportação (*Exporting Process*), que utiliza o IPFIX para exportar o fluxo ao terceiro processo, os Processos Coletores (*Collecting Process*), que analisam ou armazenam em um banco de dados os fluxos recebidos [47].

Abaixo, a Tabela 3.2 exibe todas as RFCs utilizadas como referência para a elaboração desta seção a respeito do protocolo IPFIX. Em seguida, a Figura 3.5 ilustra uma rede utilizando o protocolo IPFIX.

Tabela 3.2 - Principais RFC's do protocolo IPFIX

RFC 3917	*Requirements for IP Flow Information Export (IPFIX)* [46]	Outubro, 2004
RFC 5101	*Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information* [40]	Janeiro, 2008
RFC 5470	*Architecture for IP Flow Information Export* [48]	Março, 2009
RFC 5982	*IP Flow Information Export (IPFIX) Mediation: Problem Statement* [49]	Agosto, 2010
RFC 6183	*IP Flow Information Export (IPFIX) Mediation: Framework* [50]	Abril, 2011

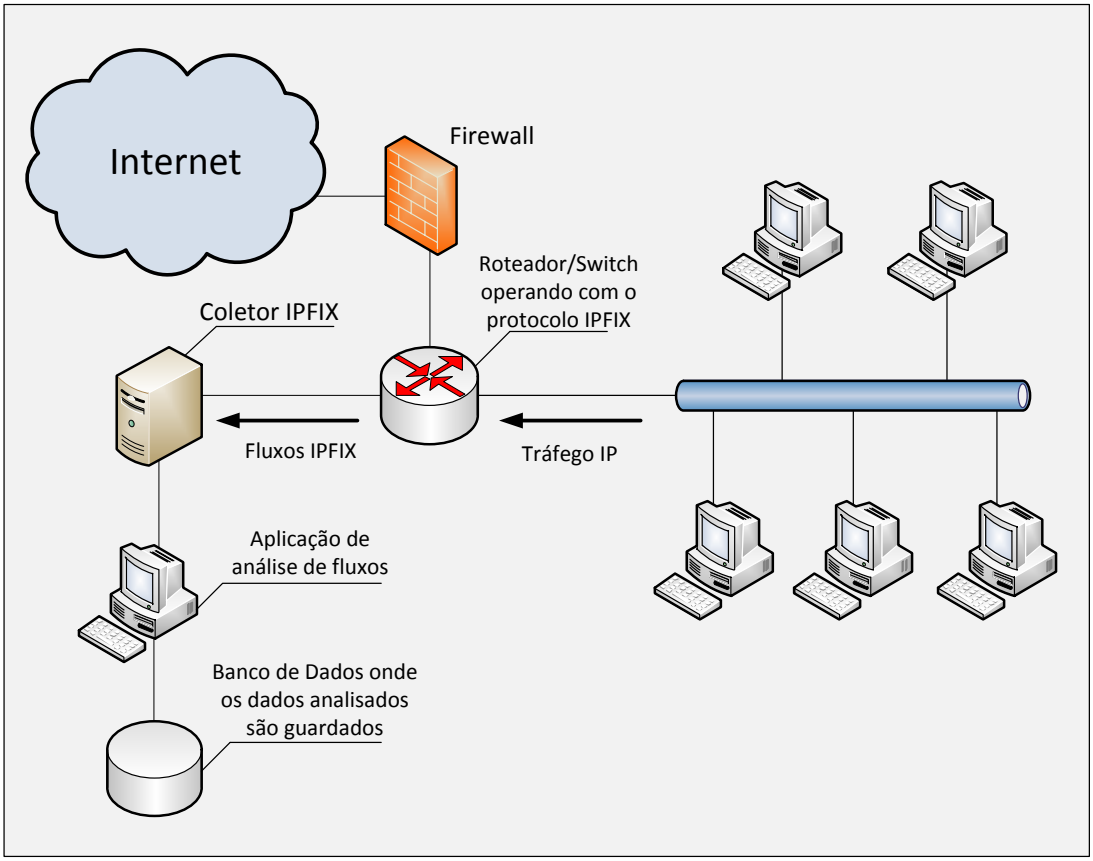

Figura 3.5 - Exemplo de uma rede monitorada com IPFIX

A unidade de dados utilizada pelo IPFIX para o transporte de dados é chamada Mensagem. Esta possui um cabeçalho de 16 bits e vários conjuntos de registros (*Sets*), que podem ser de 3 (três) tipos: *Template Sets* (Conjunto de Modelos), *Options Template sets* (Conjunto de Opções de Modelo) e *Data Sets* (Conjunto de Dados) [47] [40]. A Figura 3.6 ilustra o formato básico de uma mensagem IPFIX.

O cabeçalho de cada mensagem contém o numero da versão, o tamanho da mensagem, o *timestamp* da exportação, um numero de sequência (utilizado para detecção de mensagens perdidas), e o ID do domínio de observação. O primeiro parâmetro corresponde à versão do formato de registro de fluxo utilizado; em seguida, tem-se o tamanho total da mensagem; o timestamp fornece a hora que a mensagem IPFIX deixou o processo exportador. O número de sequência identifica cada mensagem individualmente, e é utilizado pelo processo coletor para detectar mensagens perdidas. Por último, o ID do domínio de observação indica onde o fluxo foi coletado.

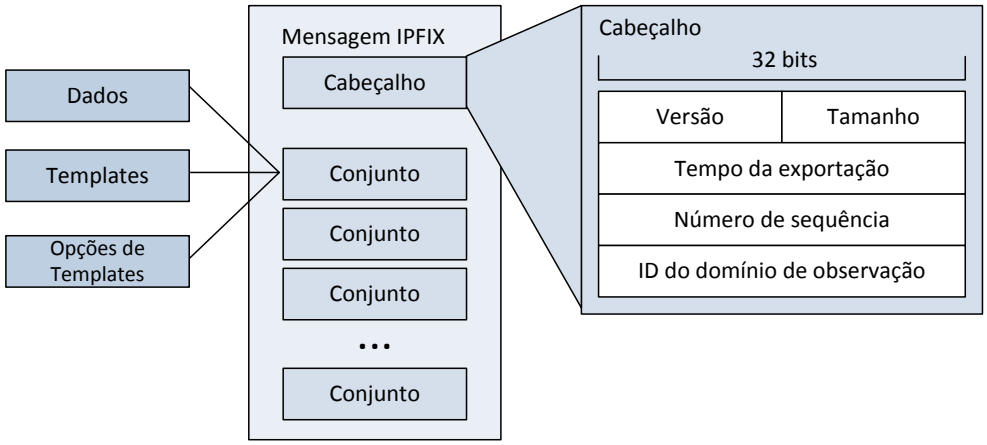

Figura 3.6 - Exemplo de uma mensagem IPFIX

Existem três tipos de conjuntos de registros: *Template Sets* (Conjunto de Modelos), *Options Template sets* (Conjunto de Opções de Modelo) e *Data Sets* (Conjunto de Dados).

Um *Template Set* (Conjunto de modelos) é formado de *Template Records*, os quais são modelos que definem a estrutura e o modo de como cada campo dos conjuntos de dados (*Data records*) serão interpretados, e é utilizado para especificar a estrutura e a semântica do conjunto de informações que é transmitido de um dispositivo IPFIX para um coletor. Cada *Template* consiste em um cabeçalho e uma sequência de *Information Elements* (abreviado para IE). Um *Information Element* é um campo que faz referência a atributos presentes nos registros de dados (*Data records*), contendo a descrição de cada tipo de dado [40]. Os IEs utilizados pelo IPFIX são mantidos em um registro administrado pela IANA (*Internet Assigned Numbers Authority*) [51].

Options Template Set (Conjunto de Opções de Modelos) são conjuntos de *Options Template Records*, os quais são uma espécie de extensão dos *Templates*, e definem a estrutura e o modo de se interpretar cada campo de um registro de dados, com o diferencial de que este define também a aplicabilidade dos registros de dados [40].

Data Sets (Conjunto de dados) são conjuntos de *Data Records* (Registro de dados), onde cada um destes registros possuem os valores dos parâmetros definidos pelo *Template* correspondente. A ordem e o tamanho de cada campo em um registro de dados são definidos por um *template*, que é referenciado através do campo "Set ID" contido no cabeçalho de um *Data Set* (Set ID = *Template* ID) [40].

Como cada registro de dados é definido por um *Template*, o processo coletor apenas poderá interpretar os dados presentes no registro se o *template* correspondente também estiver disponível. Caso contrário, os dados ficam sem significado.

A arquitetura do IPFIX é apresentada na RFC 5470 [48], onde é especificada por 3 (três) tipos de processos: Processo de Medição (*Metering Process*), Processos de Exportação (*Exporting Process*), e Processos Coletores (*Collecting Process*). A Figura 3.7 ilustra o modo como estes três processos estão relacionados dentro de um dispositivo IPFIX.

O Processo de Medição (*Metering Process*) é o responsável por gerenciar todo o fluxo gerado a partir de um domínio de observação (ponto da rede onde o tráfego é observado). Este processo observa os pacotes que passam no ponto de observação correspondente, realiza o *timestamping*, e classifica os pacotes em Fluxos de acordo com alguns critérios. Estes critérios são regras definidas pelo Processo de Medição para que apenas determinados pacotes sejam selecionados para compor um fluxo, evitando um grande volume de informações com pacotes irrelevantes, como por exemplo, a amostragem de pacotes. Esta função determina quais pacotes serão selecionados para medição de acordo com um critério de amostragem [48]. Exemplo: selecionar 1 a cada 256 pacotes que passarem por um Ponto de Observação.

O Processo de Exportação é o responsável por transmitir os dados para um ou mais softwares coletores de fluxos (Scrutinizer [52], Nfsen [53], ntop [54], etc.) utilizando o protocolo IPFIX. Este processo se comunica tanto com o Processo de Medição, para adquirir os registros de fluxo, quanto com o Processo Coletor presente nos coletores IPFIX. Ainda, as informações exportadas por este processo podem ser classificadas em duas categorias: Informações de Controle e Fluxo de Dados (*Data Stream*). Informações de Controle fornecem aos coletores IPFIX informações necessárias para que estes possam entender os dados enviados pelo processo de exportação. Isso inclui definições de fluxo, o critério para seleção de pacotes e os *templates* que descrevem os dados do fluxo. Já a categoria Fluxo de Dados (*Data Stream*) inclui os registros de fluxo, que possuem os valores dos campos dos varios fluxos observados em cada ponto de observação [48].

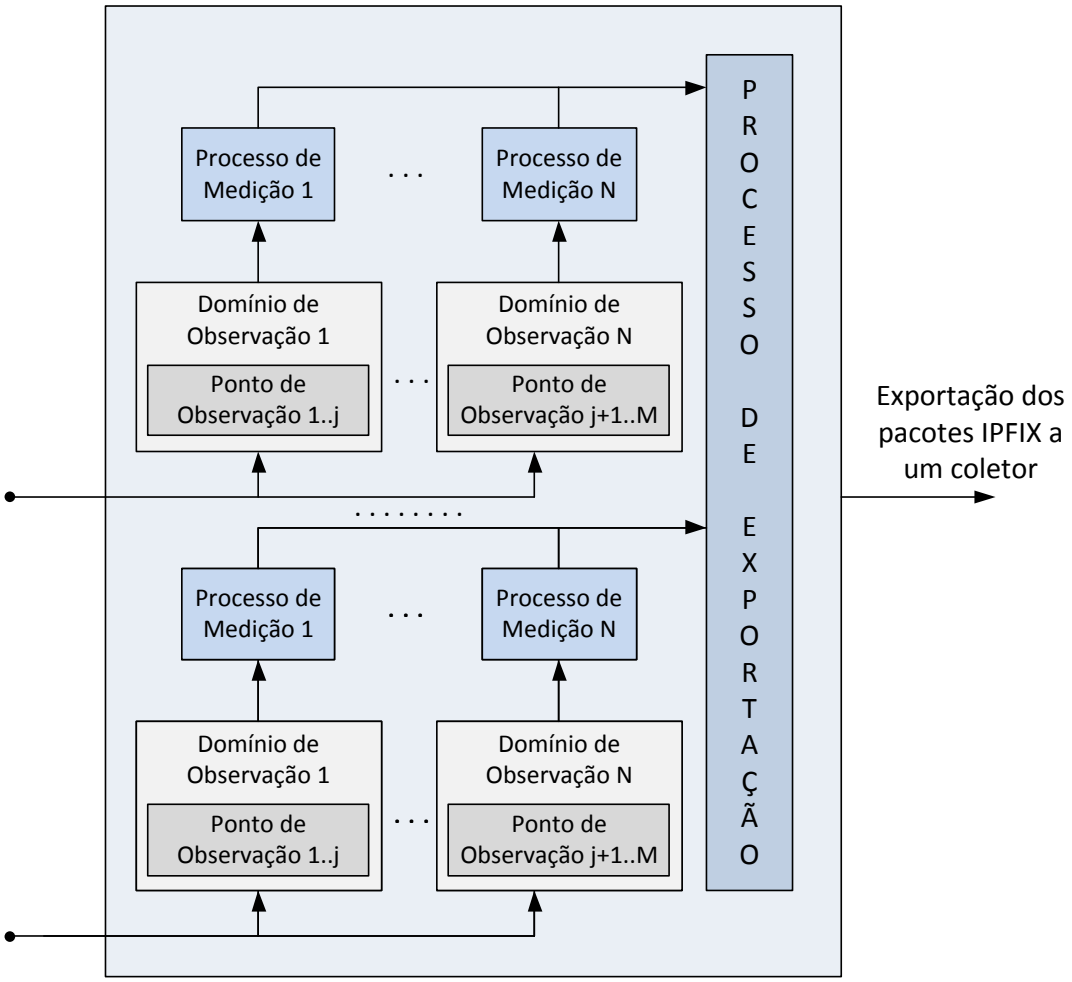

Figura 3.7 – Arquitetura de um dispositivo IPFIX

O Processo Coletor utiliza os ID's dos *Template*s de cada registro de fluxo para interpretar cada Information Element presente nos dados exportados pelo Processo de Exportação. Isso requer que o Processo de Exportação assegure que o coletor IPFIX saiba o ID de todo *Template* que chegar por meio de um registro de fluxo exportado. Ainda, deve-se levar em consideração que o coletor deve ter conhecimento da função de seleção de pacotes usada para cada fluxo pelo Processo de Medição. O Processo Coletor identifica, aceita e decodifica as mensagens IPFIX vindas de cada Processo de Exportação e seu respectivo Domínio de Observação [48].

3.3.3 sFlow

sFlow é uma tecnologia relativamente nova, criada pela InMon Corp. [55] em 2001, para o monitoramento de redes roteadas e comutadas. Diferente do NetFlow, da Cisco, este protocolo utiliza mecanismos de amostragem para elaborar as estatísticas do fluxo coletado de um dispositivo (roteador ou *switch*), e por este motivo, é aplicável a redes de alta velocidade, com velocidades gigabit ou superior. Um sistema de monitoramento utilizando o sFlow consiste em dois módulos: Um agente, que atua em roteadores e/ou switches, e um coletor de dados central. Um datagrama sFlow é usado para transmitir as estatísticas de um tráfego extraídas pelo mecanismo de amostragem para a análise em um sFlow Analyser [41].

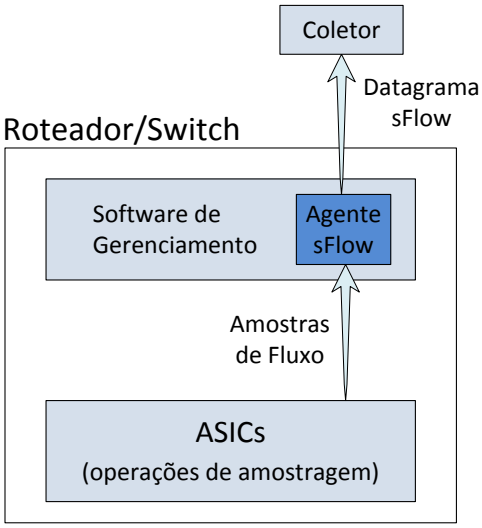

Figura 3.8 - Operação do sFlow dentro de um dispositivo de rede

O agente sFlow é um processo que atua dentro de um dispositivo de rede (roteadores/switches) como parte do software de gerenciamento (Figura 3.8). As técnicas de amostragem são executas por ASICs (*Application-Specific Integrated Circuit*) dentro dos roteadores/*switches*, e após isso, o agente combina os dados das amostras em datagramas sFlow, e os envia para um servidor central, o coletor sFlow, que armazena esses datagramas para realizar análises posteriormente. Em todo o processo, o agente realiza pouco processamento, pois ele simplesmente empacota os dados que recebe, enviando imediatamente. Isso minimiza o uso de recursos. A Figura 3.9 ilustra o funcionamento básico do sFlow. Observe que os agentes sFlow presentes nos dispositivos de rede enviam os

datagramas até um coletor central, que realiza uma análise e armazena os dados do tráfego em um banco de dados.

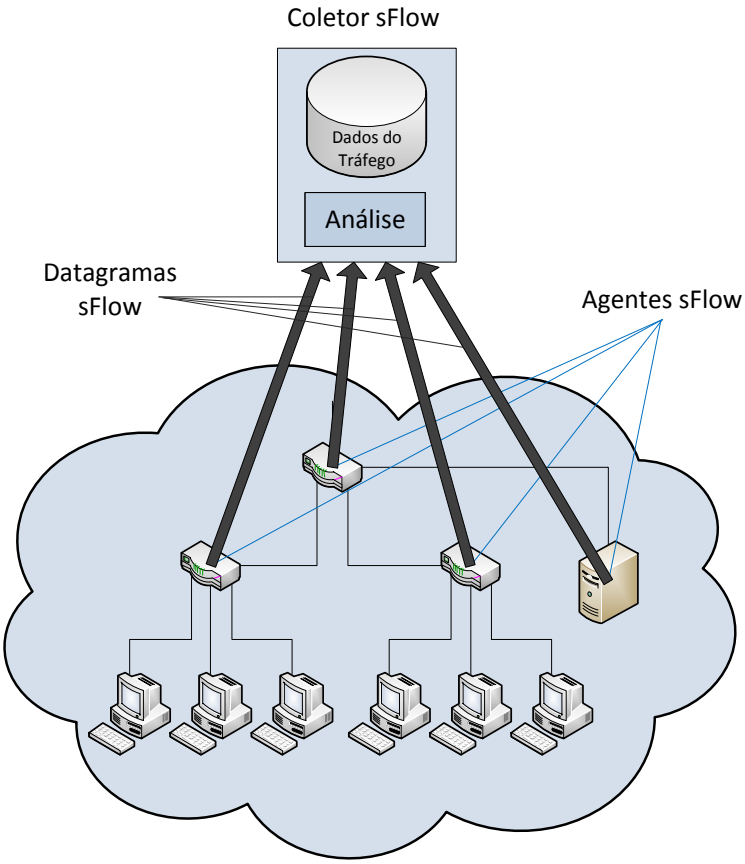

Figura 3.9 - Esquema da comunicação entre o coletor e os agentes sFlow

Ainda, o sFlow possui cinco características principais: é preciso, pois a amostragem é tão simples que é realizada diretamente no hardware; é detalhado, pois o cabeçalho dos pacotes e as informações de roteamento/comutação possibilitam uma análise mais detalhada dos fluxos; possui escalabilidade tanto em relação ao tamanho da rede (um único coletor pode monitorar milhares de dispositivos) quanto à velocidade (é capaz de monitorar redes de 10Gbps, 100Gbps, etc.); a implementação é simples e de baixo custo; e o monitoramento é fiel quando necessário informações em tempo real [41] [55].

A principal característica do sFlow é a amostragem de pacotes. Técnicas de amostragem têm sido utilizadas para monitoramento de tráfego de internet a cerca de 12 anos, tendo a Hewlett-Packard como pioneira no assunto, quando demonstrou o uso de monitoramento utilizando amostragem de pacotes em toda a rede da Universidade de Genebra do CERN (*The European Organization for Nuclear Research*). Desde então, com o

surgimento de redes cada vez mais rápidas e a transição de redes compartilhadas para redes comutadas, as técnicas de amostragem de pacotes no monitoramento de redes estão se tornando cada vez mais aceitas [55]. Com esta tecnologia, é possível ter uma visão geral da rede, além de ser escalável e ter um baixo custo de implementação.

O mecanismo de amostragem de pacotes, como o próprio nome já diz, retira amostras dos pacotes recebidos pelo dispositivo de rede que se encontra o sFlow. Quando um pacote é observado, o dispositivo de rede (*swith*/*router*) inicia um contador, que é decrementado a cada pacote que é observado. Quando este contador chega a zero, o pacote da vez é tido como amostra, e o contador então é reiniciado. Toda vez que um pacote é utilizado como amostra, o contador *Total_Samples* é incrementado. Há também um contador *Total_Packets* que é incrementado independente se o pacote foi ou não utilizado como amostra, ou seja, conta todos os pacotes. A divisão dos contadores (*Total_Packets/ Total_Samples*) indica a taxa de amostragem. Então, as amostras (incluindo as informações dos pacotes e os contadores *Total_Packets* e *Total_Samples*) são enviadas pela entidade de amostragem para processamento no agente sFlow.

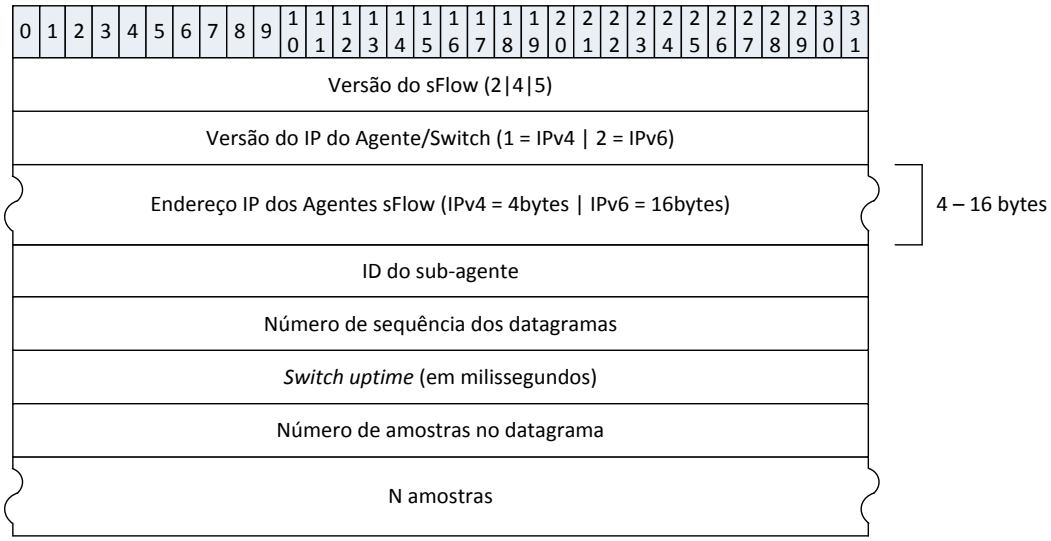

Figura 3.10 - Formato de um datagrama sFlow

Um datagrama sFlow é simples, como se pode observar na Figura 3.10. Cada datagrama é construído usando o padrão XDR (*External Data Representation*) [6], utilizado para a descrição e codificação de dados, pois este torna os processos de codificação (agente sFlow) e decodificação/análise (sFlow Analyser) mais simples. Basicamente, um datagrama sFlow é composto por um cabeçalho, a interface do dispositivo e parâmetros do processo de amostragem, enviados pelos módulos de roteamento/comutação. Além de

informações relevantes a respeito de como o pacote de amostragem foi enviado ao datagrama, como um ID do usuário e uma URL, e um número de sequência para contar também fazem parte de um datagrama sFlow [41].

Os datagramas são enviados utilizando UDP, o que reduz significativamente a quantidade memória necessária para o encapsulamento, e ainda, proporciona um modo de entrega consistente mesmo durante períodos de tráfego intenso, como no caso de um Ataque de Negação de Serviço (DoS – *Denial of Service*).

4 SISTEMA DE DETECÇÃO DE ANOMALIAS

Neste capítulo, o sistema para a detecção de anomalias baseado na Análise de Componentes Principais (PCA) proposto neste trabalho é apresentado. Primeiramente, é exposto o modelo para caracterização de tráfego, denominado PCADS (*Principal Component Analysis for Digital Signature*). Em seguida, é descrita a abordagem para detecção de anomalias. Antes de descrever o modelo, tem-se visto o funcionamento do método PCA utilizado como base em seu desenvolvimento.

4.1 DESCRIÇÃO GERAL DO SISTEMA

A proposta deste trabalho é ilustrada na Figura 4.1, com o sistema de detecção dividido em dois módulos.

O primeiro módulo é o de caracterização de tráfego (PCADS), o qual efetua a extração de características dos atributos de fluxos com a finalidade de gerar um DSNSF. Para este fim, é realizada uma análise multidimensional de fluxos IP, extraindo os atributos quantitativos: bits, pacotes e número de fluxos transmitidos por segundo.

O segundo módulo é o de detecção de anomalias, cujas assinaturas digitais de cada atributo são comparadas com o tráfego real da rede. Assim, desvios comportamentais de grande magnitude, presentes no tráfego de cada atributo em relação ao perfil normal da rede representado pelo DSNSF, são considerados eventos incomuns, podendo ser classificados como anomalias. A detecção de desvios é feita com o auxílio de limiares (*thresholds*), calculados a partir da análise de componentes principais e do DSNSF. Além disso, informações descritivas do tráfego, como endereços IP e portas TCP/UDP de origem e destino, também são utilizadas na análise, as quais são necessárias para fornecer relatórios detalhados ao gerente da rede à respeito das anomalias encontradas.

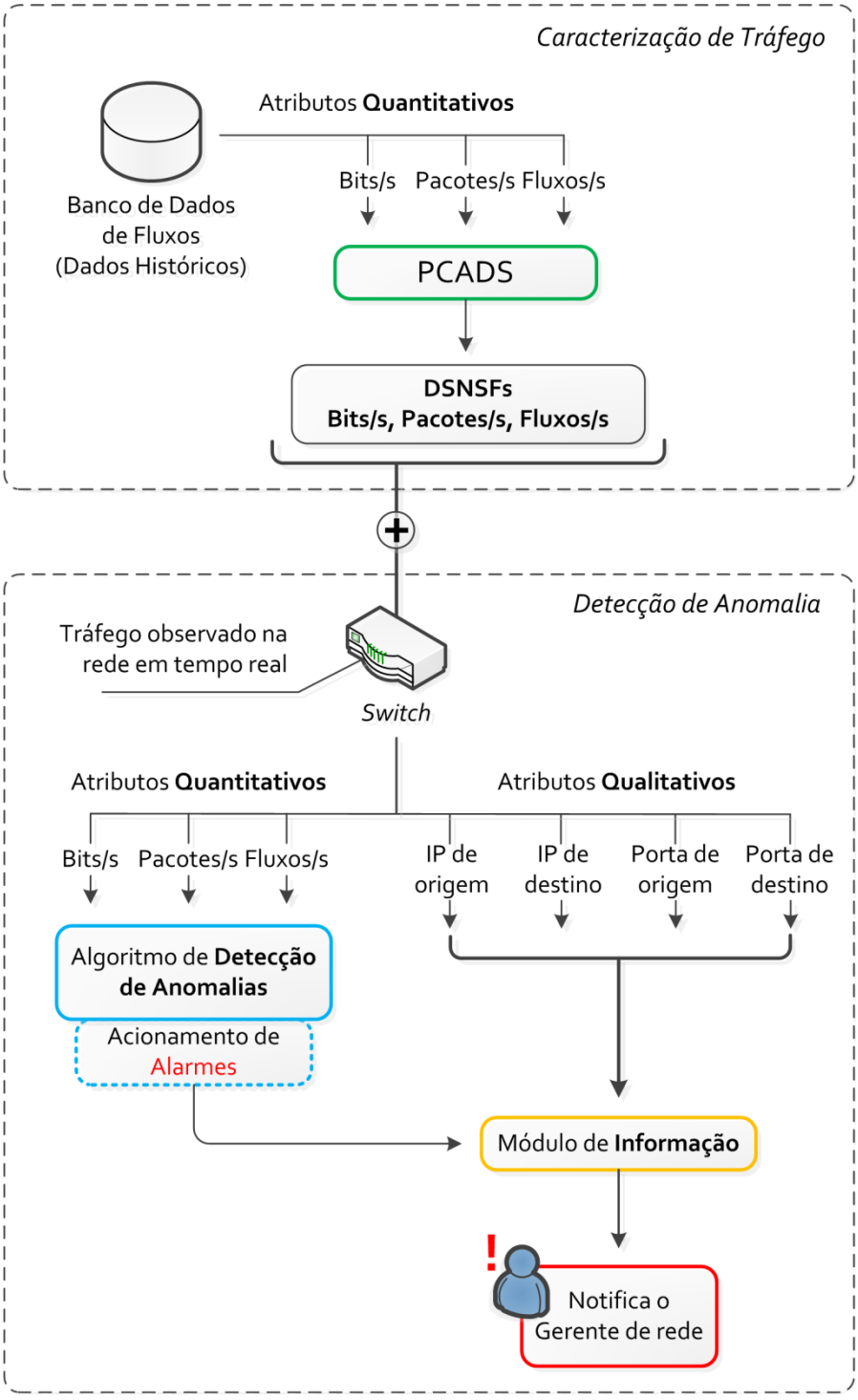

Figura 4.1 - Descrição geral do funcionamento do sistema de detecção de anomalias

4.2 MÓDULO DE CARACTERIZAÇÃO DE TRÁFEGO

Nesta seção, é descrito o funcionamento do PCADS para caracterização do comportamento normal do tráfego de rede. Entretanto, antes de descrever o modelo, é apresentado o funcionamento do método PCA, utilizado como base em seu desenvolvimento.

4.2.1 Análise de Componentes Principais: O Método

Identificar padrões em um conjunto de dados de dimensão muito alta e expressá-los de tal forma que destaque as similaridades e as diferenças entre eles não é uma tarefa fácil. Ainda mais quando não é possível a representação por gráficos (devido à alta dimensionalidade). Para tal fim, é utilizada a Análise de Componentes Principais (PCA), um método estatístico eficaz para redução de dimensionalidade. O método é capaz de comprimir os dados ao identificar seus padrões, ou seja, reduzir o número de dimensões sem grande perda de informação para, por exemplo, facilitar a representação por meio de gráficos. Isso é possível pela transformação dos dados em um novo conjunto de variáveis chamadas de Componentes Principais (PCs – *Principal Components*), as quais são descorrelacionadas e ordenadas de tal forma que as primeiras componentes conservam a maior parte da variação presente em todas as variáveis originais. O uso do PCA se estende em muitas direções sendo uma delas a utilização na compressão de imagens.

O PCA é uma das técnicas de estatística multivariadas mais antigas e sua origem não é associada a apenas um trabalho ou a um autor. Alguns trabalhos, por volta de 1873 e 1834, já possuíam indícios relacionados ao método. No entanto, suas primeiras descrições, do que hoje é conhecido como PCA, estão presentes nos trabalhos de Pearson (1901) [56] e de Hotelling (1933) [57], trabalhos estes que seguem diferentes abordagens. O trabalho de Hoteling se concentra em derivações algébricas, enquanto que o trabalho de Pearson se concentra em encontrar linhas e planos que melhor se ajustam em um espaço p-dimensional, os quais podem ser considerados os próprios componentes principais do PCA de hoje.

4.2.1.1 Matriz de dados de entrada

Seja *A* uma matriz de variáveis aleatórias de ordem *n* x *p*, onde cada coluna *p* representa uma variável, e as *n* linhas as instâncias de cada variável:

$$A_{nxp} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{np} \end{bmatrix}$$

A partir da matriz A, o primeiro passo para a análise de componentes principais consiste em normalizar a matriz A com média zero, ou seja, deve-se subtrair cada elemento de uma coluna pela média da coluna correspondente. Isso é necessário para remover a diferença de escalas entre as amostras das diferentes dimensões, e assim, evitar resultados distorcidos [58]. A matriz normalizada é definida pela matriz Ã, de mesma ordem:

$$\tilde{A}_{nxp} = \begin{bmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \cdots & \tilde{a}_{1p} \\ \tilde{a}_{21} & \tilde{a}_{22} & \cdots & \tilde{a}_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \cdots & \tilde{a}_{np} \end{bmatrix}, \quad \text{sendo } \tilde{a}_{np} = a_{np} - \overline{a_p}.$$

4.2.1.2 Matriz de covariância

Depois de realizado o ajuste dos dados, é necessário calcular a matriz de covariância de Ã. A covariância é uma medida estatística de segunda ordem (dados em duas dimensões) utilizada para determinar se há correlação entre estas dimensões. Uma covariância positiva indica que as duas dimensões crescem juntas, um valor negativo, o oposto, ou seja, uma dimensão aumenta enquanto a outra diminui, enquanto que um valor de covariância zero indica que as variáveis são independentes. A covariância é calculada para dados de duas dimensões de acordo com a equação 2. No PCA, utiliza-se a matriz de covariância porque ela permite sumarizar a covariância dos dados com mais de duas dimensões, onde a covariância é

calculada para cada par de dimensões. O cálculo da matriz de covariância *C*, relativa à matriz Ã é dada por:

$$C_{pxp} = [c_{ij}], \qquad sendo\ c_{ij} = covariância(dimensão_i, dimensão_j) \tag{1}$$

$$c_{ij} = E[(\tilde{a}_i - \bar{a}_i)(\tilde{a}_j - \bar{a}_j)], \qquad sendo\ i, j \in 1 \dots p \tag{2}$$

$$C_{pxp} = \frac{1}{n-1} \tilde{A}^T \tilde{A} \tag{3}$$

A partir da matriz de covariância *C*, são calculados duas estruturas importantes na análise de componentes principais: os autovetores e autovalores. No caso do PCA, a transformação linear é apenas a matriz de covariância, que atua como uma matriz de transformação.

4.2.1.3 Autovetores e autovalores

De acordo com a álgebra de matrizes, duas matrizes podem ser multiplicadas se apresentarem tamanhos compatíveis. Um autovetor se enquadra como um caso específico.

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 11 \\ 5 \end{bmatrix} \tag{4}$$

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 12 \\ 8 \end{bmatrix} = 4 \times \begin{bmatrix} 3 \\ 2 \end{bmatrix} \tag{5}$$

Considere as multiplicações entre uma matriz e um vetor presente nas equações (4) e (5). Na equação (4) é possível observar que o vetor resultante não é múltiplo do vetor original, enquanto que na equação (5), o vetor resultante é exatamente quatro vezes maior que o vetor que multiplicou a matriz. Isso ocorre, pois este vetor se encontra em um espaço bidimensional, representado por uma seta que aponta da origem, (0, 0), ao ponto (3, 2). A matriz em questão é considerada uma matriz de transformação, que quando multiplicada à

esquerda de um vetor, resulta em um vetor escalonado do vetor original. Assim, é desta natureza (transformação linear) que surgem os autovetores.

Algumas das propriedades dos autovetores são:

- Sua obtenção apenas ocorre por meio de matrizes quadradas;
- Nem todas as matrizes quadradas possuem um autovetor;
- Dado uma matriz *n* x *n*, esta resultará em *n* autovetores;
- Todos os autovetores de uma matriz formam uma base ortogonal.

Com relação aos autovalores, temos que esses são totalmente relacionados aos autovetores. Na equação (5) o vetor resultante é quatro vezes maior que o vetor original (autovetor). Esse escalar do autovetor é o que chamamos de autovalor. No exemplo, o escalar 4 é o autovalor associado ao autovetor. Os autovetores e autovalores sempre são obtidos em par. Para cada autovetor, haverá um autovalor associado, e vice-versa.

A fim de destacar a importância dos autovetores e dos autovalores para o PCA, tomemos o exemplo da Figura 4.2, o qual ilustra a projeção de duas variáveis aleatórias quaisquer (*X1* e *X2*) no plano cartesiano. A figura ainda traz dois vetores projetados no plano, *PC1* e *PC2*. Esses são os chamados autovetores ou componentes principais. No exemplo, o autovetor *PC1* é o que representa a maior variância do conjunto de dados. Isto pode ser facilmente observado no gráfico, visto que o *PC1* tem seu sentido na direção em que os dados apresentam maior dispersão. Todavia, o PCA é mais utilizado em conjuntos de dados bem maiores, e, a partir de dados com mais de três dimensões, a representação por gráficos se torna inviável.

Portanto, o que vai indicar se determinado autovetor contém a maior variância dos dados é o seu autovalor correspondente, ou seja, quanto maior o autovalor, maior é a porcentagem da variância e as características dos dados originais representadas pelo autovetor associado àquele autovalor.

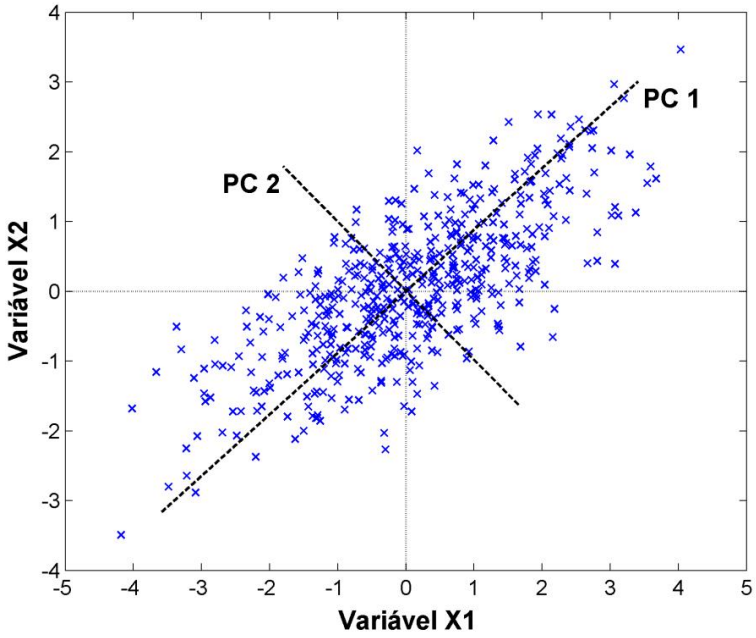

Figura 4.2 - Exemplo de autovetores no plano

O cálculos dos autovetores e autovalores para matrizes de duas dimensões é simples, podendo ser utilizada a equação característica:

$$\det(B - \lambda \cdot I) = 0, \tag{6}$$

$$\det\begin{pmatrix} b_{11} - \lambda & b_{12} \\ b_{21} & b_{22} - \lambda \end{pmatrix} = 0, \tag{7}$$

sendo B uma matriz quadrada de ordem 2, I a matriz identidade de ordem 2, e λ, os escalares não-nulos que solucionam a equação (autovalores).

O cálculo desta equação característica resulta em uma equação de 2º grau, que após o cálculo das raízes, é possível substituí-las no sistema abaixo para descobrir os autovetores correspondentes aos autovalores.

$$\begin{pmatrix} b_{11} - \lambda & b_{12} \\ b_{21} & b_{22} - \lambda \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{8}$$

Porém, o cálculo de autovetores e autovalores para matrizes quadradas de dimensão maior que 3 não é algo simples, sendo necessário utilizar algoritmos numéricos iterativos.

Uma forma de efetuar o cálculo dos autovetores e autovalores para dados de dimensão superior a três se dá por meio da decomposição de valor singular (SVD, *Singular Value Decomposition*) de uma matriz de covariância *C*:

$$C_{pxp} = Q_{pxp} D Q_{pxp}^{T}, \qquad sendo\ Q^{T} Q = I_{p}, \tag{9}$$

sendo as colunas da matriz *Q* os autovetores ortonormais de *C*, $D = diag(\delta_1, \delta_2, \delta_3, \ldots, \delta_n)$ a matriz diagonal de autovalores, e I a matriz identidade de ordem *p*.

Esta decomposição é computada resolvendo o Problema do Autovalor Simétrico (*Symmetric Eigenvalue Problem*) [59] e [60]. O Problema do Autovalor Simétrico, bem como outros modos de calcular autovetores e autovalores, se dão por meio de métodos iterativos complexos, que vão além do escopo deste trabalho.

4.2.1.4 Dados finais

Deste modo, o próximo passo do método consiste em construir o vetor característico $\tilde{Q}$, uma matriz de autovetores de ordem *k*, contendo as *k* componentes principais selecionadas dentre os *p* autovetores de Q, calculados a partir da matriz de covariância C^{pxp}:

$$\tilde{Q} = [autovetor_i], \qquad sendo\ i \in 1 \ldots k. \tag{10}$$

É nesse passo que ocorre a redução de dimensionalidade do problema, pois, aqui, são selecionadas as componentes mais significantes da amostra, e excluindo as menos significantes. O método SVD retorna os autovetores ordenados de forma decrescente de acordo com o seu autovalor correspondente, ou seja, os autovetores são ordenados de acordo com a proporção de variância total do conjunto de dados que cada um representa. O componente com o maior autovalor é o que chamamos de componente principal. Assim, podemos escolher as componentes mais significantes para a amostra (autovetores com os

maiores autovalores) e deixar de lado os menos significantes. Isso resulta na redução de dimensionalidade do problema porque a partir da amostra inicial de *p* dimensões com *p* autovetores e autovalores, ao escolher apenas os primeiros *k* autovetores para compor o vetor característico, os dados finais terão apenas *k* dimensões, onde $k << m$ [61].

Por fim, escolhido as componentes para compor o vetor característico $\tilde{Q}$, basta realizar a operação abaixo:

$$Z_{kxn} = \tilde{Q}_{pxk}{}^{T} \cdot \tilde{A}_{nxp}{}^{T}, \tag{11}$$

sendo Z o vetor final dos dados iniciais projetados em k dimensões, $\tilde{Q}$ o vetor característico e $\tilde{A}$ a matriz de dados de entrada normalizada com média zero.

Esse procedimento resultará nos dados originais projetados no subespaço determinado pelas componentes principais. Um exemplo seria um conjunto de dados tridimensionais, em que temos uma maior concentração de dados na primeira e segunda dimensão (plano bidimensional). Ao utilizar o PCA, é possível identificar tais dimensões e reduzir a dimensionalidade para representar esses dados em um plano bidimensional que explique o significado dos dados por meio das duas componentes mais significativas [58].

4.2.2 PCADS: Análise de Componentes Principais para Assinatura Digital

Nesta seção, é descrito o funcionamento do modelo proposto para caracterização de tráfego com a utilização do PCA, o PCADS (*Principal Component Analysis for Digital Signature*). A principal característica do modelo proposto é a aplicação e contextualização do método PCA descrito na seção 4.2.1 a um ambiente de detecção de anomalias. As principais adaptações estão relacionadas aos dados de entrada e à escolha das componentes principais para compor o conjunto final dos dados, que serão apresentados nas próximas seções.

A Figura 4.3 ilustra o funcionamento do modelo proposto para caracterização do tráfego. Seu funcionamento básico consiste, primeiramente, na exportação de fluxos IP de um *switch* por meio de protocolos padrões de exportação de fluxos como *NetFlow* e IPFIX, armazenando-os em um banco de dados. Esses fluxos são processados para obter os atributos

quantitativos utilizados pelo PCADS. Após o PCADS ser computado, temos os DSNSFs calculados para os três atributos.

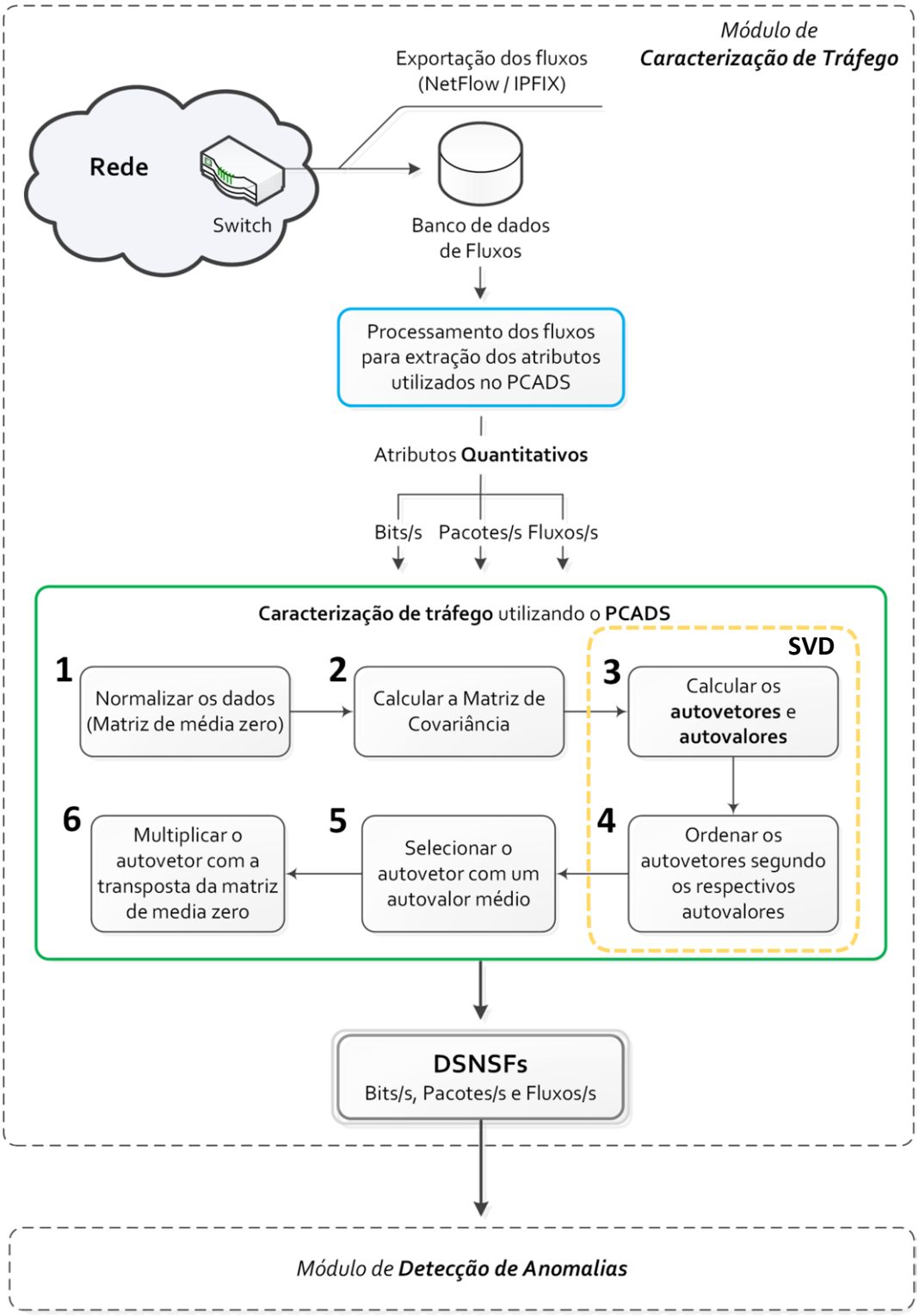

Figura 4.3 - Descrição do modelo para caracterização de tráfego

4.2.2.1 Dados de entrada

Os dados de entrada do PCA clássico são arranjados em uma matriz de dimensão *n* x *p*, onde as *p* colunas representam as dimensões (variáveis) e as *n* linhas as *n* instâncias de cada uma das variáveis. No entanto, para o PCADS, os dados de tráfego coletados dos registros de fluxo são arranjados de tal forma que o tráfego de cada dia é denotado por três vetores, cada um contendo o total de bits, pacotes e número de fluxos transmitidos por segundo, correspondendo às 24 horas do dia. Desta forma, a matriz de entrada *A* do PCADS, de dimensões *n* x *p* é construída, onde *p* representa o tráfego dos *p* dias utilizados como base na geração do DSNSF, e as *n* linhas são os valores de cada atributo extraído dos fluxos.

Após a coleta dos fluxos de um *switch*, esses são processados com a aplicação *nfdump* [62] de modo que para cada dia de tráfego haja apenas um único arquivo contendo todos os fluxos coletados. Este arquivo, agrupado pelo *nfdump,* contém milhares de linhas, e cada linha representa um fluxo, com os devidos atributos (*timestamp*, IPs, portas, bits, pacotes, fluxos, etc.). Como podem existir mais de um fluxo para o mesmo *timestamp* – como para o *timestamp* 10:00:01 podem existir 20 fluxos, ou seja, 20 linhas no arquivo agrupado–, foi desenvolvido um programa em C++ que agrupa todos os fluxos pertencentes ao mesmo *timestamp*, levando em consideração apenas os segundos. A Figura 4.4 ilustra um caso em que o *timestamp* 08:01:01 possui 5 fluxos (os demais atributos dos fluxos, como os IPs e as Portas, foram retirados da imagem para facilitar a visualização). O programa em C++ realiza a leitura de cada fluxo e agrupa o total de bytes que pertençam a um mesmo *timestamp*. No exemplo da Figura 4.4, o *timestamp* 08:01:01 possui um total de bytes de 1.577.472, que é igual à soma do total de bytes de cada um dos fluxos do *timestamp* 08:01:01.

	Date flow start	Duration	Proto	Packets	Bytes	Flows
29961	2012-10-01 08:01:01.550	0.000	TCP	256	389632	1
29962	2012-10-01 08:01:01.550	0.000	TCP	256	389632	1
29963	2012-10-01 08:01:01.550	0.000	TCP	256	389632	1
29964	2012-10-01 08:01:01.550	0.000	TCP	256	389632	1
29965	2012-10-01 08:01:01.550	0.000	TCP	256	18944	1
29966	2012-10-01 08:01:05.550	0.000	TCP	256	389632	1
29967	2012-10-01 08:01:05.550	0.000	TCP	256	34304	1
29968	2012-10-01 08:01:05.550	0.000	TCP	256	176896	1
29969	2012-10-01 08:01:05.550	0.000	TCP	256	389632	1
29970	2012-10-01 08:01:05.550	0.000	TCP	256	389632	1
29971	2012-10-01 08:01:05.550	0.000	TCP	256	17408	1
29972	2012-10-01 08:01:09.549	0.000	TCP	256	389632	1
29973	2012-10-01 08:01:09.549	0.000	TCP	256	17408	1
29974	2012-10-01 08:01:09.549	0.000	TCP	256	389632	1
29975	2012-10-01 08:01:09.549	0.000	TCP	256	367104	1
29976	2012-10-01 08:01:09.549	0.000	TCP	256	389632	1
29977	2012-10-01 08:01:09.549	0.000	TCP	256	367104	1
29978	2012-10-01 08:01:09.549	0.000	TCP	256	17408	1

Figura 4.4 - Exemplo de fluxo

Deste modo, o programa em C++ produz como saída um vetor coluna para cada um dos três atributos estudados, com cada vetor contendo o total de cada atributo para todos os 86.400 segundos presentes em um dia (do *timestamp* 00:00:00 até 23:59:59), como é exemplificado na Figura 4.5.

Timestamp	Bytes
00:00:00	230246
00:00:01	389876
00:00:02	621324
.	.
.	.
.	.
08:01:01	1577472
.	.
.	.
.	.
23:59:59	843652

Timestamp	Pacotes
00:00:00	1024
00:00:01	768
00:00:02	1536
.	.
.	.
.	.
08:01:01	1280
.	.
.	.
.	.
23:59:59	1024

Timestamp	Fluxos
00:00:00	4
00:00:01	3
00:00:02	6
.	.
.	.
.	.
08:01:01	5
.	.
.	.
.	.
23:59:59	4

Figura 4.5 - Exemplo do formato de saída dos fluxos após processamento

4.2.2.2 Descrição do funcionamento do modelo

O funcionamento do modelo para caraterização de tráfego PCADS é apresentado no Algoritmo 1.

Algoritmo 1 – Algoritmo utilizado para a criação do DSNSF

Entrada: A – Conjunto de bits/s, pacotes/s ou fluxos/s coletados de um banco de dados históricos organizado em uma matriz $n \times p$.
Saída: μ – Um vetor representando a estimativa do tráfego de bits/s, pacotes/s ou fluxos/s de um dia, ou seja, o DSNSF.

1: **para** t =1 até 1440 **faça**
2: Normalizar os dados de entrada (A_{nxp}) (matriz de média zero: $\tilde{A}_{nxp}$)
3: Calcular a matriz de covariância C_{pxp} por meio de $\tilde{A}_{nxp}$
4: Calcular os autovetores (Q_{pxp}) e autovalores (D) por meio de C_{pxp}
5: Selecionar o autovetor de Q_{pxp} com um autovalor associado de valor intermediário dentre os autovalores em D ($\tilde{Q}$)
6: $\mu_t \leftarrow \tilde{Q}_{px1}{}_t^T \times \tilde{A}_{nxp}{}_t^T$
7: **fim para**
8: **retorna** μ

No Algoritmo 1, a execução deste se encontra dentro de um laço de 1440 iterações. Como cada período do tráfego possui suas próprias características e, visando prevenir que as características de um período interfiram no outro, a gerar ruído no DSNSF, o algoritmo trabalha com uma janela de tempo de 1 minuto. Assim, o algoritmo é executado para cada minuto do movimento do tráfego independentemente.

Conforme o funcionamento do PCA, o primeiro passo é normalizar os dados subtraindo a média de cada coluna da matriz de entrada, resultando em colunas de média zero. Isso resulta em uma matriz centralizada na média, a qual é importante porque assegura que a análise de componentes capture a variância real dos dados, e, consequentemente, evite resultados distorcidos devido a diferenças na utilização da rede [29].

Feito isso, o próximo passo é calcular a matriz de covariância *C*. A matriz de covariância mostra de que forma as variáveis presentes na matriz $\tilde{A}$ estão correlacionadas entre si e fornece uma diagonal principal de variâncias, especificando a direção e a força da correlação linear entre duas variáveis, identificadas por colunas distintas em $\tilde{A}$.

Em seguida, os *p* autovetores e autovalores são obtidos por meio da matriz de covariância segundo especificado na seção 4.2.1. Os autovetores com os maiores autovalores são chamados de componentes principais e são selecionados pelo PCA para compor o novo conjunto de dados de dimensão reduzida. No PCADS, o algoritmo gera a assinatura digital (DSNSF) por meio de apenas uma componente principal, já que o DSNSF é representado matematicamente por um vetor coluna. No entanto, ao invés de selecionar o autovetor que apresenta o maior autovalor, o PCADS seleciona o autovetor que possui um autovalor de valor médio dentre os demais autovalores calculados. Isto é necessário, pois, o maior autovalor representa a dimensão com a maior porcentagem de variância dos dados de entrada e criar uma assinatura digital com esse componente de variância alta pode gerar *outiliers* no comportamento do DSNSF gerado.

No exemplo abaixo, tem-se os autovetores e autovalores calculados por meio da análise de componentes principais (PCA) para W, uma matriz de variáveis aleatórias de ordem 5x3:

$$W = \begin{bmatrix} -0{,}0301 & -0{,}8637 & 1{,}5326 \\ -0{,}1649 & 0{,}0774 & -0{,}7697 \\ 0{,}6277 & -1{,}2141 & 0{,}3714 \\ 1{,}0933 & -1{,}1135 & -0{,}2256 \\ 1{,}1093 & -0{,}0068 & 1{,}1174 \end{bmatrix}$$

Primeiramente, a matriz W é normalizada com média zero:

$$\widetilde{W} = \begin{bmatrix} -0{,}5571 & -0{,}2395 & 1{,}1274 \\ -0{,}6919 & 0{,}7015 & -1{,}1749 \\ 0{,}1006 & -0{,}5900 & -0{,}0338 \\ 0{,}5662 & -0{,}4893 & -0{,}6308 \\ 0{,}5822 & 0{,}6173 & 0{,}7121 \end{bmatrix}$$

Em seguida, a partir de $\widetilde{W}$, a matriz de covariância C é calculada:

$$C = \frac{1}{5-1}\widetilde{W}^T\widetilde{W} = \begin{bmatrix} 0{,}3647 & -0{,}0823 & 0{,}0597 \\ -0{,}0823 & 0{,}3795 & -0{,}0815 \\ 0{,}0597 & -0{,}0815 & 0{,}8894 \end{bmatrix}$$

Como C se trata de uma matriz quadrada de ordem 3, os autovetores e autovalores podem ser facilmente calculados por meio da equação (6), resultando na matriz de autovetores Q e na matriz diagonal D, com os autovalores:

$$Q = \begin{bmatrix} 0{,}1322 & -0{,}6709 & 0{,}7296 \\ -0{,}1700 & 0{,}7098 & 0{,}6835 \\ 0{,}9765 & 0{,}2144 & 0{,}0202 \end{bmatrix}$$

$$D = \begin{bmatrix} 0{,}9117 & 0 & 0 \\ 0 & 0{,}4327 & 0 \\ 0 & 0 & 0{,}2893 \end{bmatrix}$$

Cada autovalor é proporcional à porção de "variância" (soma dos quadrados das distâncias dos pontos à média multidimensional dos mesmos) associada a cada autovetor, e a soma de todos os autovalores é igual à soma dos quadrados dos pontos à média multidimensional dos mesmos, ou seja, a "variância" total do conjunto de dados. Cada autovalor representa a variância de cada variável em relação ao conjunto total, e a soma de todos, a variância total do conjunto. Deste modo, os autovalores representam a porcentagem de informação do conjunto de dados original que cada componente carrega, ou seja, a relevância de cada dimensão. Deste modo, pode-se calcular a relevância de cada componente com base nos autovetores e autovalores exibidos acima:

$$Soma\ dos\ autovalores\ de\ W = \ 0{,}9117 + 0{,}4327 + 0{,}2893 = 1{,}6337 = 100\%$$

$$Relevância\ (em\ porcentagem) para\ cada\ componente:$$
$$Autovalor\ da\ componente\ 1 = \ 0{,}9117 = 55{,}8058\%$$
$$Autovalor\ da\ componente\ 2 = 0{,}4327 = 26{,}4859\%$$
$$Autovalor\ da\ componente\ 3 = 0{,}2893 = 17{,}7083\%$$

Ao analisar os resultados acima, pode-se observar o relevância de cada dimensão no conjunto de dados original (W). Deste mesmo modo ocorre com o cálculo do PCADS, onde o tráfego de cada dia que compõe a matriz de entrada *A* possui um autovetor e um autovalor correspondente, com cada componente representando um grau de relevância. Gerar um DSNSF com um autovetor correspondente ao tráfego de um dia com relevância alta pode implicar na incorporação de *outiliers* presentes no tráfego deste dia no DSNSF. E de

forma similar, utilizar um autovetor correspondente ao tráfego de um dia com relevância baixa pode implicar na incorporação de situações houve queda do servidor ou queda de energia, por exemplo, presentes no tráfego deste dia no DSNSF.

Após diversos testes, foi observado que uma componente cuja variância corresponde a um valor médio (intermediário) entre as componentes de variâncias máxima e mínima produz uma assinatura digital uniforme e mais condizente ao comportamento normal da rede, prevenindo que possíveis disparidades (anomalias) presentes nos dados de treinamento produzam distorções no DSNSF. Na Figura 4.6 podemos observar a diferença entre DSNSFs gerados com a utilização dos autovetores de variância máxima e variância de valor intermediário. Note que os DSNSFs da Figura 4.6 (b) apresentam um comportamento mais próximo ao tráfego real, enquanto que os da Figura 4.6 (a), não obtiveram uma estimativa boa do tráfego.

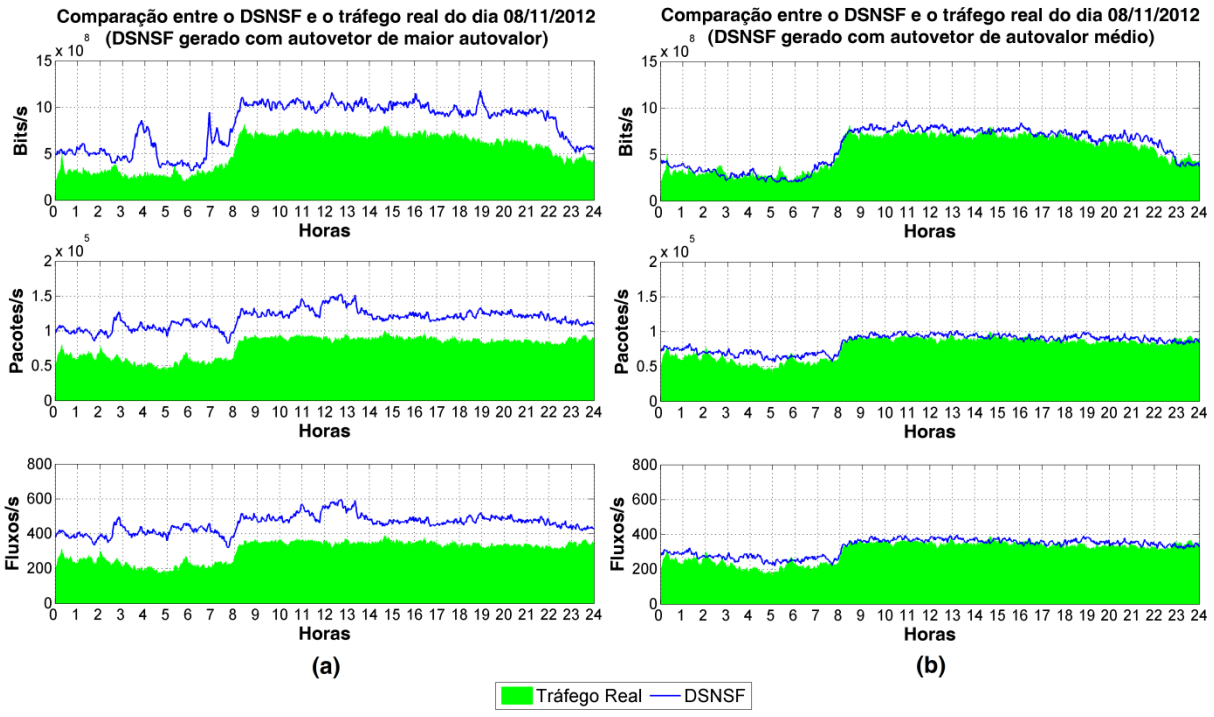

Figura 4.6 - Comparação entre DSNSFs gerados com autovetores de variância máxima (a) e variância de valor intermediário (b)

O último passo consiste em multiplicar o autovetor selecionado pela matriz normalizada para cada intervalo de tempo *t*. Assim, após realizar este cálculo para cada intervalo de tempo *t*, temos como resultado o DSNSF.

4.2.2.3 Parâmetros para criação do DSNSF

Na metodologia utilizada no PCADS, assume-se que é gerado um DSNSF para cada dia de trabalho (de segunda a sexta), baseado em um histórico das atividades da rede nas semanas anteriores [63]. Dessa forma, o DSNSF gerado para uma segunda-feira, por exemplo, necessita dos dados do tráfego das segundas-feiras anteriores para que o algoritmo "aprenda" o comportamento padrão da segunda-feira, e assim para os demais dias da semana. Deste modo, primeiramente deve-se definir o número de semanas anteriores ao dia que se deseja o DSNSF, o qual será utilizado como dado de treinamento para o PCADS, de modo que esse apresente bons resultados. Este parâmetro da caracterização é importante porque afeta diretamente seu tempo de execução, visto que a matriz de entrada *A* do algoritmo do PCADS tem seu tamanho definido de acordo com o número de dias (*p*) anteriores ao que se deseja criar um DSNSF. Portanto, computar a análise de componentes principais para matrizes muito grandes pode ocasionar um tempo de execução elevado. A detecção e identificação de problemas na rede devem ocorrer de forma ágil, já que qualquer pequeno atraso pode causar sérios danos na rede.

A Figura 4.7 e a Figura 4.8 apresentam o cálculo do NMSE de DSNSFs gerados para uma semana em que se utilizam os tráfegos de uma a dez semanas anteriores como dados de treinamento.

O NMSE avalia os desvios gerais entre valores previstos e valores observados (reais), destacando as diferenças mais notáveis entre tais medidas. O resultado ótimo desta métrica consiste no valor zero, o qual indica a situação em que a medida prevista é exatamente igual à medida observada. De modo equivalente, altos valores de NMSE indicam resultados mais distantes do esperado [64]. O NMSE é definido pela seguinte equação:

$$NMSE(X,Y) = \frac{\sum_{i=1}^{n}(x_i - y_i)^2}{\sum_{i=1}^{n}(y_i)^2}, \tag{12}$$

sendo $X = (x_1, x_2, \dots, x_n)$ a medida observada e $Y = (y_1, y_2, \dots, y_n)$ a medida prevista.

Como se pode observar na Figura 4.7, para o tráfego de bits/s, o modelo começou a produzir melhores resultados do NMSE a partir do uso de cinco semanas. Acima

disso, os resultados apresentaram certa estabilidade, de modo que nenhum ganho a mais foi verificado.

Para os tráfegos de pacotes/s e para o número de fluxos/s ilustrados na Figura 4.8, os resultados apresentam grandes divergências entre os períodos de uma a quatro semanas, porém os índices NMSE dos cinco dias analisados se estabilizam com o uso de cinco semanas em diante.

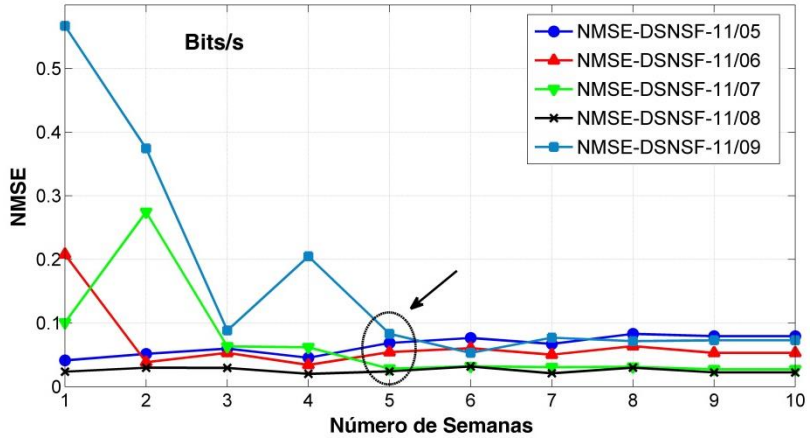

Figura 4.7 - Índices NMSE calculados entre os DSNSFs e o tráfego real dos dias analisados utilizando de uma a dez semanas de treinamento para o tráfego de bits/s

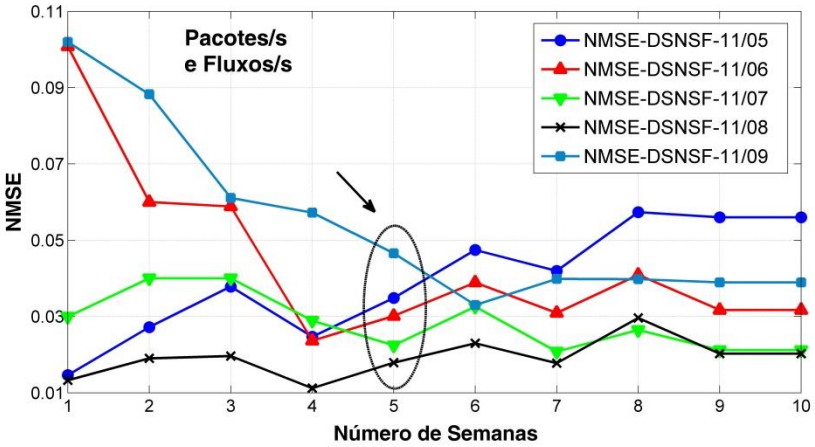

Figura 4.8 - Índices NMSE calculados entre os DSNSFs e o tráfego real dos dias analisados utilizando de uma a dez semanas de treinamento para o tráfego de pacotes/s e número de fluxos/s

4.3 MÓDULO DE DETECÇÃO DE ANOMALIA

O DSNSF é um perfil de rede que representa uma estimativa do comportamento de tráfego de um determinado segmento. Na Figura 4.6 (b), ele é representado pela linha azul, enquanto que o tráfego real observado se encontra em verde. Com base nisso, assume-se que sempre que o tráfego real apresentar um desvio em relação à linha do DSNSF, este determinado intervalo de tempo é classificado como um comportamento anômalo. Deste modo, o sistema de detecção irá disparar um alarme ao gerente de rede para que este possa direcionar seus esforços diretamente no local em que ocorre o problema.

No entanto, como a caracterização de tráfego produz uma estimativa do comportamento normal da rede e como toda previsão não é perfeita, há a possibilidade de ocorrer pequenos desvios no tráfego real em relação ao DSNSF em certos intervalos de tempo que podem não ser propriamente um evento anômalo. Caso todos esses pequenos desvios fossem considerados anomalias, centenas de alarmes seriam desencadeados por dia, contrastando com a característica de sistema de gerenciamento proativo, já que o gerente de rede seria notificado inúmeras vezes para observar e analisar um número excessivo de alarmes, cuja maioria pode não ser relevante.

Logo, visando minimizar o disparo excessivo de alarmes falsos e classificar como anomalias apenas desvios de volume mais críticos, neste trabalho é proposto a criação de limiares (*thresholds*) superiores e inferiores ao DSNSF. Estes *thresholds* são definidos como limites aceitáveis do DSNSF, com o objetivo de que pequenos desvios de tráfego em relação ao DSNSF não desencadeiem uma série de alarmes falsos, visto que tais desvios podem ser decorrentes de variações normais do comportamento da rede.

Os limiares são obtidos por meio do autovalor do autovetor escolhido pelo algoritmo de caracterização na composição do DSNSF. Como o DSNSF é gerado a partir de um autovetor, explícito na Seção 4.2.2.2, o autovalor que representa sua variância pode, por conseguinte, representar a "variância" do próprio DSNSF, a resultar em uma porcentagem aceitável, α, para ser utilizada como limite. A partir de α, os limiares, chamados de Limite de Autovalor (EL - *Eigenvalue Limit*), são calculados conforme descrito nas equações abaixo:

$$EL_{sup} = \mu + \mu \cdot \frac{\alpha}{100} \quad (13)$$

$$EL_{inf} = \mu - \mu \cdot \frac{\alpha}{100}, \quad (14)$$

sendo EL_{sup} e EL_{inf} os limiares superior e inferior respectivamente, μ o DSNSF, e α a porcentagem obtida do autovalor associado ao autovetor utilizado na composição do DSNSF.

Após diversos testes numéricos, foi observado que esta abordagem resultou em bons resultados como é demonstrado na Figura 5.5 e Figura 5.6 da seção 5.2.1.2.

A descrição geral do sistema de detecção proposto é ilustrada na Figura 4.1, com detalhes do modelo de detecção de anomalias mostrados na figura Figura 4.9. Primeiramente, o modelo de caracterização fornece os DSNSFs dos três atributos analisados (bits, pacotes e número de fluxos) juntamente com os respectivos autovalores. Logo, o módulo de detecção gera os limiares inferiores e superiores, denotados por EL_{sup} e EL_{inf}. Feito isso, o sistema extrai do tráfego real os atributos quantitativos e qualitativos e se divide em duas etapas.

Na primeira etapa, de acordo com o algoritmo presente na Figura 4.9, os atributos quantitativos são comparados com os DSNSFs e os limiares gerados. A análise é executada em janelas de tempo de 1 min, onde, para cada intervalo de tempo do tráfego da rede (r_t), o algoritmo verifica se este se encontra dentro dos limites aceitáveis definidos por EL_{sup} e EL_{inf}. Caso esta condição ocorra no mesmo intervalo de tempo de dois ou mais atributos simultaneamente, o algoritmo considera que o comportamento é normal. Caso contrário, um alarme é disparado, acionando o Módulo de Informação. Trabalhos anteriores fazem o uso de janelas de 5 minutos [65], porém, o método de detecção desenvolvido neste trabalho realiza análises de tráfego em janelas de tempo de 1 minuto. Essa redução de tempo da janela visa obter um tempo de resposta mais rápido, haja visto, que recentemente foram introduzidas no mercado redes que operam a 40 Gbps e 100 Gbps, as quais geram um grande volume de tráfego adicional por minuto. Em redes com tais velocidades, gerar alarmes a cada cinco minutos pode significar cerca de 30 TB de trafego pernicioso afetando negativamente a rede antes que o alarme seja disparado.

A ativação do módulo de informação corresponde à segunda etapa do método de detecção. Após os atributos descritivos serem extraídos do tráfego real, eles são organizados conforme o número de ocorrências, para que sejam geradas listas dos Top *N* IPs e Portas de origem e destino que ocorreram no intervalo anômalo identificado na primeira

etapa. Este tipo de estatística auxilia o gerente de rede com relação ao problema encontrado, visto que por meio do IP é possível descobrir a origem do problema, assim como o alvo de ataques, enquanto que portas TCP/UDP somente identificam a aplicação que sofreu um ataque ou teve problemas.

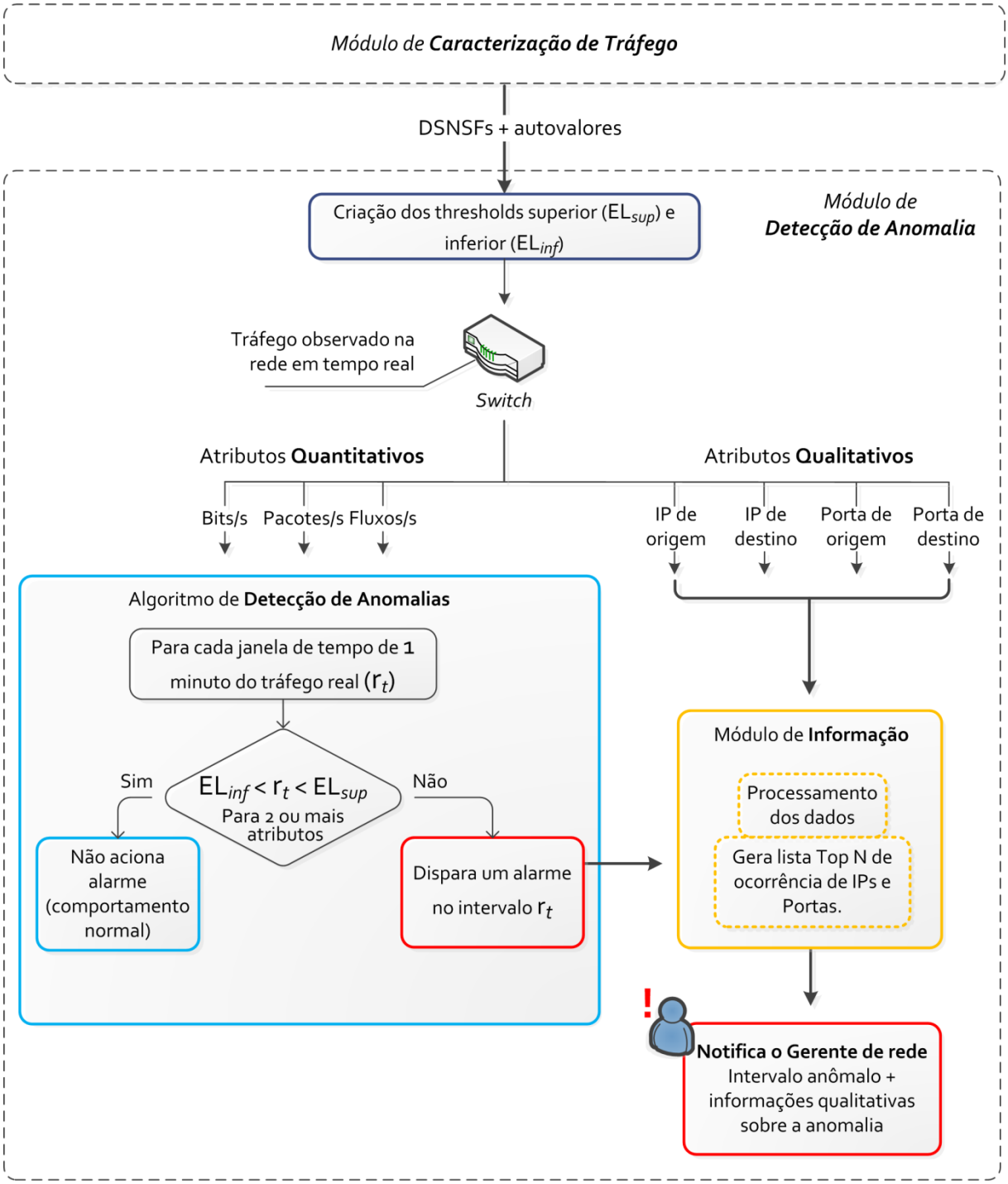

Figura 4.9 - Módulo de detecção de anomalias

5 AVALIAÇÃO DO SISTEMA

Neste capítulo, o sistema de detecção de anomalias baseado no PCA proposto neste trabalho é aplicado ao tráfego de dados gerados a partir de redes reais. Seu desempenho em criar um perfil de rede capaz de caracterizar um segmento de rede e detectar anomalias é avaliado por meio de métodos estatísticos. Todos os testes foram realizados com o uso de dados de fluxos provenientes de redes reais, a verificar a viabilidade do método quando aplicado a um ambiente real.

5.1 MÉTRICAS DE AVALIAÇÃO

A eficácia da assinatura digital produzida pelo modelo proposto é mensurada a partir de três testes estatísticos: Erro Quadrático Médio Normalizado (NMSE – *Normalized Mean Square Error*) e Coeficiente de Correlação Normalizado (CC).

O Erro Quadrático Médio Normalizado (NMSE) vai ser usado conforme descrito previamente na seção 4.2.2.3, por meio da equação (12).

O Coeficiente de Correlação normalizado indica o grau de correlação entre duas variáveis, bem como, a direção desta correlação (positiva ou negativa). Os valores obtidos estão dentro do intervalo de -1 a +1, cujo valor 1 indica correlação total, 0 (zero) indica que as duas variáveis não apresentam correlação entre si, e -1 especifica uma correlação total inversa, isto é, sempre que uma variável aumenta, a outra diminui e vice-versa. [66]. O coeficiente de correlação normalizado ρ é definido por:

$$\rho_{XY} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} = \frac{cov(X,Y)}{\sqrt{var(X) \cdot var(Y)}}, \quad (15)$$

sendo $X = (x_1, x_2, \dots, x_n)$ e $Y = (y_1, y_2, \dots, y_n)$ os valores medidos das variáveis X e Y, e ρ o coeficiente de correlação normalizado calculado para X e Y.

Para avaliar as taxas de detecção de anomalias do sistema proposto, é utilizado o método da curva ROC (*Receiver Operating Characteristics*). A ROC é definida como uma técnica a se medir o desempenho de classificadores. É amplamente utilizada na

teoria de detecção de sinais visando descrever o *trade-off* entre as taxas de acerto (taxa de verdadeiro-positivo) e as taxas de alarme falso (taxa de falso-positivo). As taxas de verdadeiro-positivo (TPR – *True-Positive Rate*) descrevem os sinais detectados corretamente enquanto que as taxas de falso-positivo (FPR – *False-Positive Rate*) descrevem a frequência com que sinais são detectados erroneamente [67].

$$TPR = \frac{TP}{P} = \frac{instâncias\ positivas\ classificadas\ corretamente}{total\ de\ instâncias\ positivas} \tag{16}$$

$$FPR = \frac{TN}{N} = \frac{instâncias\ negativas\ classificadas\ corretamente}{total\ de\ instâncias\ negativas} \tag{17}$$

Ainda, é utilizado a medida de acurácia Acc, a qual representa a taxa de acerto geral, ou seja, a taxa de acertos de ambas as classes (resultados positivos e negativos), cuja expressão é definida por:

$$Acc = \frac{TP + TN}{P + N} = \frac{instâncias\ positivas\ e\ negativas\ classificadas\ corretamente}{total\ de\ instâncias\ positivas\ e\ negativas} \tag{18}$$

5.2 Cenários Para Avaliação do Sistema

Para melhor avaliar o validar o sistema proposto, foram definidos cenários de teste, onde são utilizados dados reais provenientes da rede de duas universidades, simulações de anomalias de tráfego, e comparação com outro modelo. As seções subsequentes apresentam a avaliação do modelo em três diferentes cenários, os quais estão definidos na Tabela 5.1, que contém os parâmetros de avaliação utilizados em cada ocasião.

Tabela 5.1 - Definição dos três cenários utilizados na avaliação do sistema proposto

	Cenário 1	**Cenário 2**	**Cenário 3**
Local	Universidade Estadual de Londrina	Universidade Tecnológica Federal do Paraná – Campus Toledo	Universidade Estadual de Londrina
Período de Coleta	10 de Setembro a 9 de Novembro de 2012	25 de Março a 17 de Maio de 2012	10 de Setembro a 9 de Novembro de 2012
Período de Treinamento	10 de Setembro a 12 de Outubro de 2012 (5 semanas)	25 de Março a 26 de Abril (5 semanas)	10 de Setembro a 12 de Outubro de 2012 (5 semanas)
Período de Testes	15 de Outubro a 9 de Novembro de 2012 (4 semanas)	29 de Abril a 17 de Maio (3 semanas)	15 de Outubro a 9 de Novembro de 2012 (4 semanas)
Protocolo de exportação	sFlow	NetFlow v9	sFlow
Amostragem	1:256	1:1	1:256
Caracterização	X	X	X
Detecção	X	X	
Comparação com outro modelo			K-Means
Atributos quantitativos	X	X	X
Atributos descritivos	X	X	
NMSE	X	X	X
CC	X	X	X
ROC	X	X	
Acurácia	X	X	

5.2.1 Cenário 1 – Universidade Estadual de Londrina (UEL)

Neste cenário, os fluxos foram coletados de um *core switch* da Universidade Estadual de Londrina (UEL) no formato da versão 5 do protocolo sFlow [41] do período de 10 de Setembro a 9 de Novembro de 2012. A cada cinco minutos, o coletor salva e exporta os fluxos em arquivos binários, sendo processados em seguida pelo *NFDUMP tools* [62].

Visto que a rede da Universidade possui cerca de sete mil dispositivos interconectados e uma das principais características do protocolo sFlow é a amostragem de

pacotes, os fluxos foram coletados utilizando uma taxa de amostragem de 1:256. Isso significa que, a cada 256 pacotes, apenas um tem seus atributos extraídos para compor um fluxo que represente os demais 255 pacotes, os quais não foram coletados.

5.2.1.1 Avaliação da caracterização de tráfego

A Figura 5.1 ilustra os DSNSFs gerados pelo PCADS para quatro dias – de 5/11/2012 a 8/11/2012 – utilizando os tráfegos de bits, pacotes e número de fluxos transmitidos por segundo. Em cada gráfico, o DSNSF (linha azul) é comparado com o tráfego real da rede (em verde), observado durante as 24 horas de cada dia. Ao analisar as figuras, as assinaturas digitais para os três atributos puderam estimar eficientemente o comportamento normal do tráfego da rede, visto que se pode observar um ótimo ajuste entre a curva do DSNSF e o movimento real. No entanto, em alguns casos, esse ajuste não ocorre no período das 03h30min às 05h30min da manhã. A disparidade entre as séries presente neste intervalo ocorre devido a uma particularidade do tráfego da Universidade. Todos os dias, é realizado um backup durante este período, gerando um excesso de tráfego. À medida que os backups foram realizados nas semanas anteriores, e não durante o período apresentado nas figuras, o algoritmo de caracterização de tráfego do PCADS aprendeu este comportamento. Assim, no momento em que os backups não são mais realizados, os DSNSFs gerados logo se adaptam a um novo comportamento sem esta característica.

Ainda na Figura 5.1, nos gráficos do dia 09/11/2012 há um caso em que há a presença de uma possível anomalia no tráfego observado. Esse comportamento destacado pode ser considerado uma anomalia haja visto que, ao analisar as atividades da rede nos dias anteriores neste mesmo período, esse comportamento não se repete. Com isso, pode-se observar uma grande discrepância entre o DSNSF e o tráfego real. Logo, como o DSNSF representa o comportamento normal e esperado para a rede naquele momento, pode-se afirmar que, possivelmente, ocorreu uma anomalia de volume, a demonstrar a usabilidade do modelo para a detecção de anomalias.

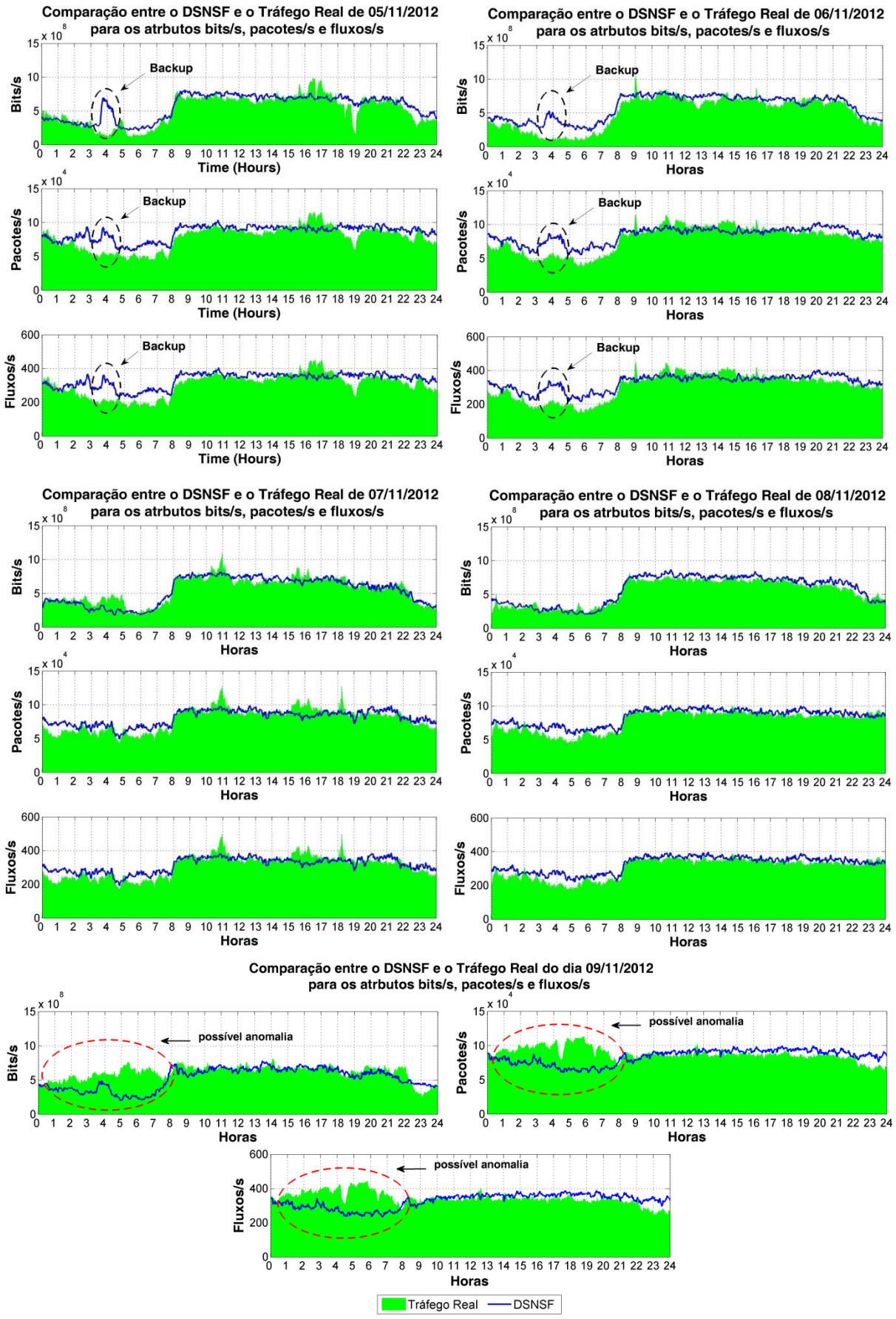

Figura 5.1 – Exemplo de caracterização de tráfego na rede da Universidade Estadual de Londrina

O primeiro teste realizado foi o do Erro Quadrático Médio Normalizado (NMSE). Os resultados estão em amostra na Figura 5.2. O modelo apresentou bons resultados para os três atributos estudados durante as quatro semanas de estudo, de modo que obteve índices de erro sempre próximos a zero (abaixo de 0,1). Em alguns casos, o erro é maior devido ao fato de que os dias 15 de Outubro e 2 de Novembro de 2012 são feriados nacionais, resultando em um comportamento de rede diferente do que é esperado. O método proposto não se aplica à feriados, visto que por se tratar de um caso isolado onde as atividades da rede não correspondem ao que é considerado normal, o DSNSF não é capaz de aprender este comportamento.

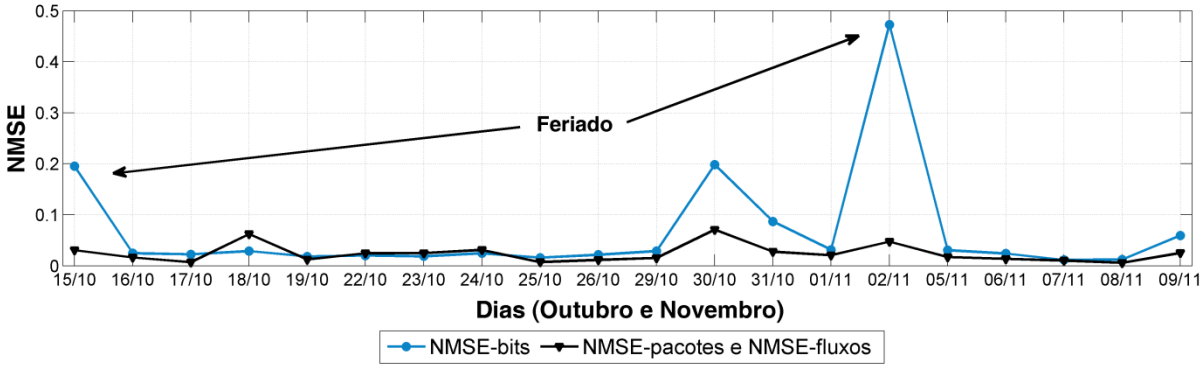

Figura 5.2 - NMSE calculado entre o DSNSF e o tráfego real no período de 15 de Outubro a 9 de Novembro de 2012

A Figura 5.3 apresenta os resultados do cálculo da Correlação entre os DSNSFs e o tráfego real. Nota-se que os resultados de correlação para o atributo bits/s se mostraram melhores que para os demais atributos. Mas, apesar disso, o modelo obteve bons resultados porque, segundo [66], resultados acima de 0,7 indicam uma correlação forte e se pode observar uma média de 0,9 para bits/s e 0,7 para pacotes/s e número de fluxos/s.

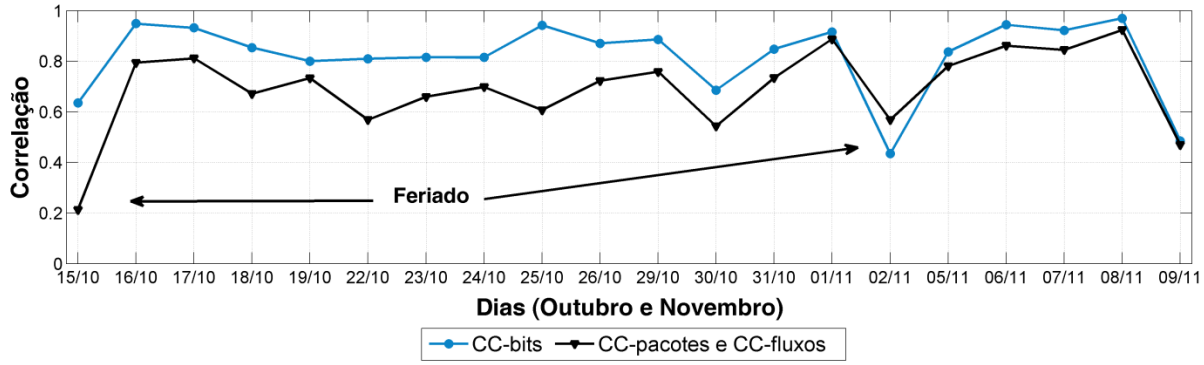

Figura 5.3 - Coeficientes de Correlação calculados entre o DSNSF e o tráfego real no período de 15 de Outubro a 9 de Novembro de 2012

5.2.1.2 Avaliação da detecção de anomalia

Para avaliar corretamente o sistema de detecção de anomalias, foi utilizada uma ferramenta para injetar artificialmente eventos anômalos no tráfego real.

As anomalias foram simuladas nos dados da rede da Universidade Estadual de Londrina por meio de uma ferramenta chamada *Scorpius*, desenvolvida pelo grupo de pesquisas em redes do Departamento de Computação da Universidade Estadual de Londrina [71]. Ela é capaz de simular anomalias de rede tais como ataques DoS e DDoS, *Port Scan* e *Flash Crowds*. As anomalias são injetadas nos fluxos após sua coleta do *switch*, ou seja, o tráfego é alterado sem que ocorra intervenção direta na rede, preservando-a de impactos causados por estas anomalias artificiais.

Assim, foram simulados ataques DoS e DDoS, e *Flash Crowds* no conjunto de dados descrito na seção 5.2.1, a fim de criar um gabarito (*template*) que contenha todos os intervalos de tempo infectados, com o objetivo de compará-lo com os alarmes gerados pelo sistema proposto neste trabalho.

Tem-se na Tabela 5.2 a relação dos dias cujas anomalias foram injetadas, bem como os demais parâmetros utilizados na simulação. Todos os IPs e Portas são fictícios, criados apenas para a finalidade de testes.

Tabela 5.2 - Parâmetros da simulação de anomalias no cenário 1 utilizando a ferramenta *Scorpius*

Dia	Anomalia	Intervalo de Tempo	IP de origem (ou núm. de IP de origem)	Porta de origem	IP de destino	Porta de destino	Protocolo
16/10/12	DoS	9h até 11h	28.235.160.128	617	83.94.15.23	6008	TCP
17/10/12	DDoS	12h até 15h	10	-	60.89.255.157	6724	ICMP
19/10/12	DDoS	5h até 7h	5	-	180.122.5.131	8188	UDP
22/10/12	DDoS	3h até 8h	5	-	220.151.209.46	3486	TCP
23/10/12	Flash Crowd	0h até 8h	15	-	108.57.76.10	7213	TCP
25/10/12	Flash Crowd	17h até 19h	5	-	12.12.12.12	5055	TCP
26/10/12	DDoS	6h até 10h	7	-	125.127.254.176	2222	TCP
29/10/12	DDoS	16h até 19h	5	-	3.40.112.112	9031	TCP
31/10/12	Flash Crowd	4h até 8h	7	-	148.236.85.173	3446	TCP
01/11/12	DDoS	16h até 18h	10	-	218.209.223.219	2717	TCP
05/11/12	Flash Crowd	15h até 17h	10	-	254.234.112.61	2365	TCP
06/11/12	DoS	5h até 8h	130.232.209.93	7530	67.17.25.44	673	TCP
07/11/12	DoS	10h até 13h	74.110.154.14	8006	122.191.184.218	2346	ICMP
08/11/12	DDoS	9h até 10h	5	-	10.10.10.10	4041	TCP

A Figura 5.4 ilustra a geração de alarmes para o tráfego de bits, pacotes e número de fluxos transmitidos por segundo durante dois dias com anomalias injetadas artificialmente.

A linha azul espessa é o DSNSF e as linhas finas são os limites inferiores e superiores (*thresholds*) calculados usando o *Eigenvalue Limit* descrito na seção 4.3. Os intervalos de tempo em que o tráfego real permanece dentro da área entre os *thresholds* são considerados normais e os alarmes (em vermelho) são gerados quando o movimento real se afasta desses limites. Os alarmes em vermelho, indicados na figura, são as possíveis anomalias detectadas para cada atributo distintamente. Porém, segundo a abordagem ilustrada

na Figura 4.9, o sistema só notifica o administrador de rede caso seja gerado um alarme em dois ou mais atributos.

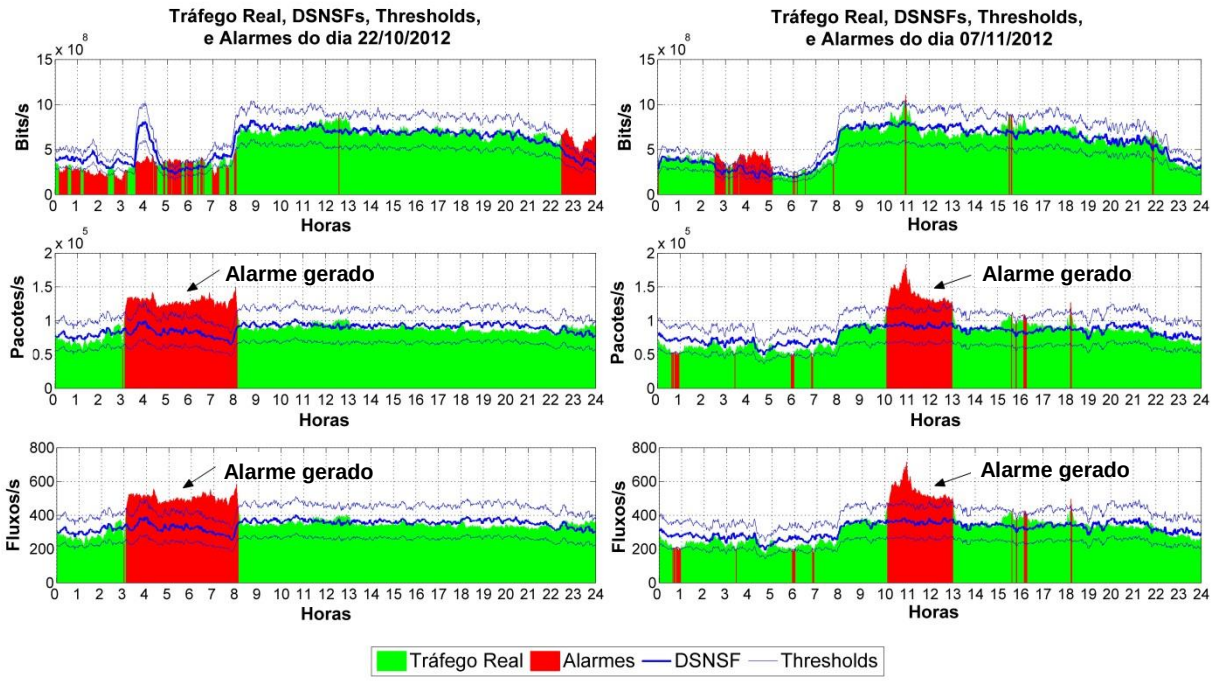

Figura 5.4 - Exemplo de alarmes gerados no tráfego de dois dias na rede da Universidade Estadual de Londrina para os três atributos quantitativos estudados

A Figura 5.5 apresenta a curva ROC para as quatro semanas selecionadas. O gráfico ROC foi gerado por meio do cálculo das taxas de verdadeiro-positivo e falso-positivo de cada dia, tendo como base o conjunto de dados rotulados, o qual define o que é uma anomalia ou não no tráfego. Além disso, foram utilizados outros valores de limiares para validar o Limite de Autovalor. Estes outros valores de limiares foram produzidos variando o limiar calculado por meio do EL de 2% a 4%, para mais e para menos. A equação abaixo descreve a elaboração dos demais limiares variando o EL:

$$EL \pm n = \mu \pm \mu \cdot \frac{\alpha - n}{100}, \tag{19}$$

sendo n a porcentagem a ser adicionada ou subtraída do EL calculado conforme explícito na seção 4.3.

A análise das curvas demonstrou que não houve melhorias expressivas ou quedas nas taxas de TPR e FPR ao variar o *threshold* calculado com o EL. No geral, o sistema

obteve desempenho satisfatório com 94% de taxas de verdadeiro-positivo e taxas de falso-positivo em 23%.

Na Figura 5.6 é apresentado o cálculo da acurácia para o mesmo período de estudo de quatro semanas e a mesma comparação de *thresholds* realizada na Figura 5.5. Essa medida é a proporção de resultados verdadeiros (verdadeiro-positivo e verdadeiro-negativo) obtidos. Os testes indicam que o sistema atingiu uma acurácia de 85% em média. Mais uma vez, pode-se notar que houve melhorias ou perdas significativas no resultado ao variar o EL.

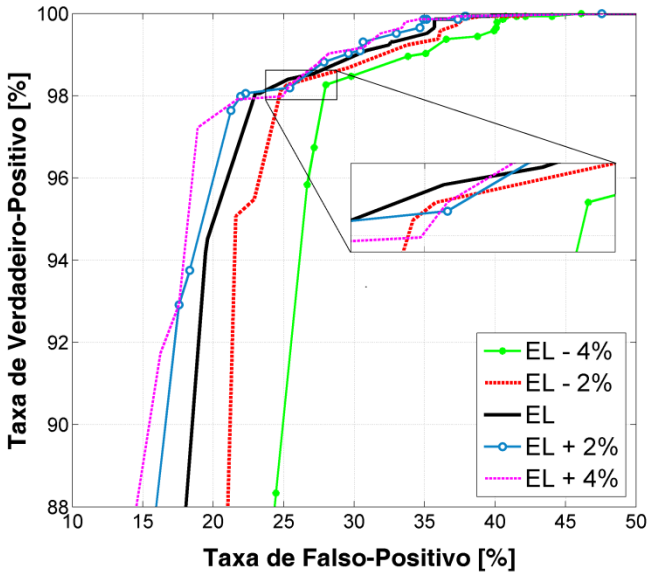

Figura 5.5 - Gráfico ROC mostrando os *trade-offs* entre os valores de TPR e FPR

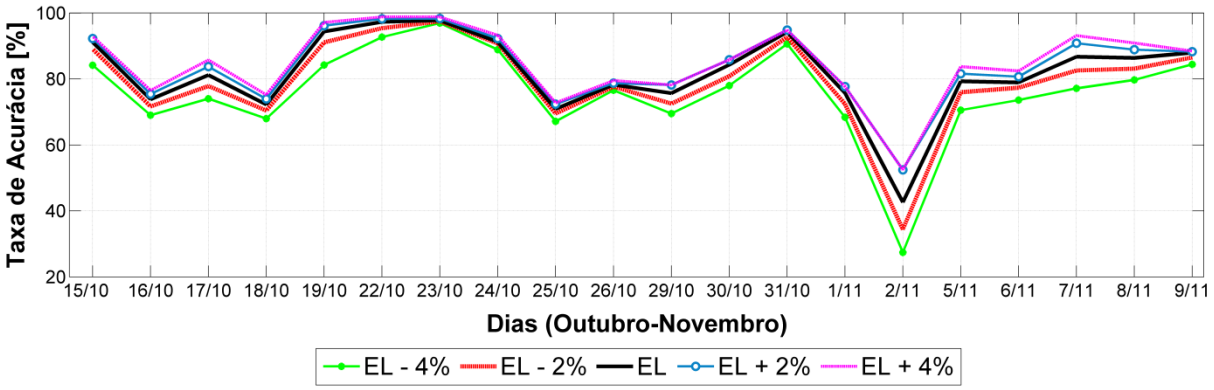

Figura 5.6 - Taxa de acurácia para quatro semanas de testes

Após o sistema detectar um possível evento anômalo, o Módulo de Informação é acionado para disponibilizar informações qualitativas do intervalo anômalo detectado. Este, auxilia o gerente de rede a tomar medidas específicas para a solução do

problema encontrado. Os endereços IP e as Portas TCP/UDP são informações muito úteis e significativas para uma detecção de anomalia rápida e precisa. Esses atributos revelam o local do problema, quem o causou, ou até mesmo, o tipo de aplicação que foi o alvo.

Para exemplificar o funcionamento deste módulo, um estudo de caso foi realizado tendo sido selecionado dois dias com anomalias injetadas artificialmente, usando endereços IP e Portas fictícias, sendo eles: dia 25 de outubro de 2012 e dia 8 de novembro de 2012. As informações a respeito dos ataques simulados para este estudo de caso se encontram na Tabela 5.2.

A Figura 5.7 e Figura 5.8 exibem o Módulo de Informação do sistema. Ambas as figuras mostram os endereços IP e as Portas presentes no intervalo de tempo em que o alarme foi acionado, bem como gráficos ilustrando o comportamento anômalo. O sistema é capaz de gerar estatísticas de todos os atributos presentes no intervalo anômalo onde foi acionado um alarme pelo módulo de detecção, porém, para facilitar a visualização e por ser suficiente para estes exemplos, é exibido apenas os três atributos que ocorrem com mais frequência no intervalo analisado.

Por meio da análise dos resultados na Figura 5.7 e Figura 5.8, os endereços IP e as portas de destino com maior frequência identificada pelo nosso sistema correspondem aos atributos da Tabela 5.2 utilizados na simulação das anomalias. Pode-se observar que uma situação anômala afeta uma grande proporção dos fluxos do tráfego, não apenas relacionado ao volume, mas também aos atributos qualitativos.

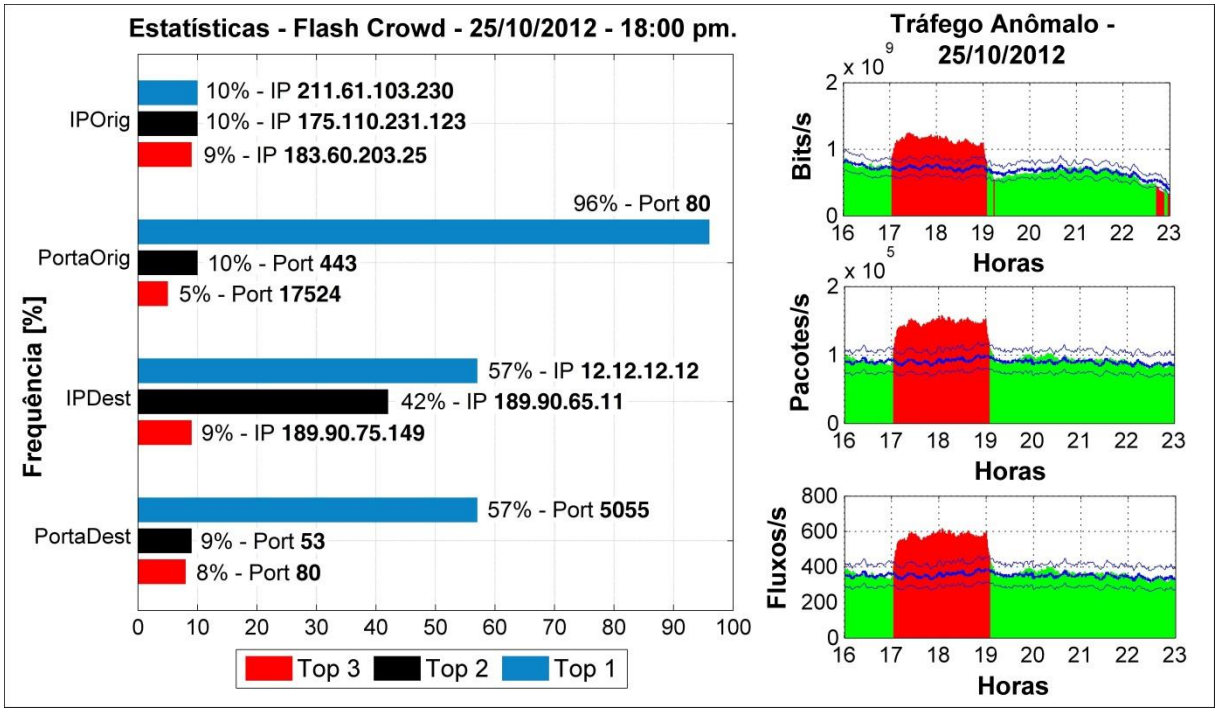

Figura 5.7 - Módulo de informação em uma simulação de Flash Crowd

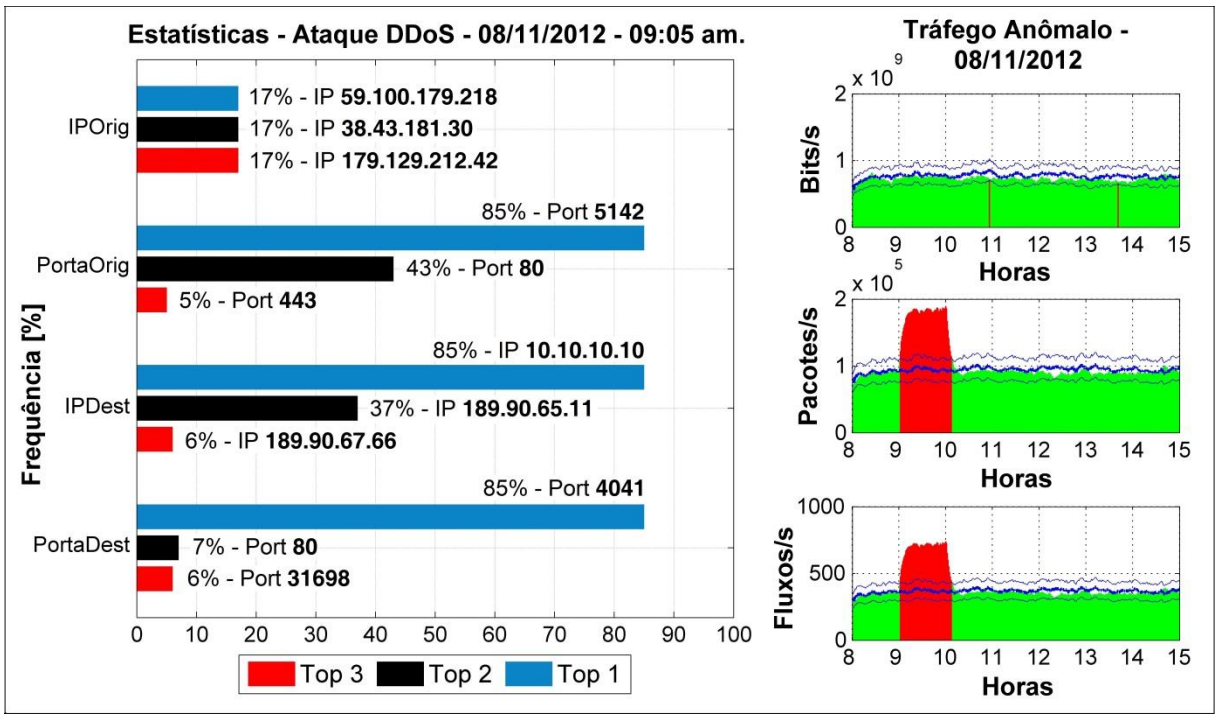

Figura 5.8 - Módulo de informação em uma simulação de DDoS

5.2.2 Cenário 2 – Universidade Tecnológica Federal do Paraná (UTFPR)

Neste cenário, a coleta dos fluxos foi realizada na Universidade Tecnológica Federal do Paraná – Campus Toledo, que possui aproximadamente 400 hosts e um link de dados de 12 Mbps. Os fluxos foram exportados com amostragem 1:1 pelo aplicativo *softflowd* instalado no gateway sem a utilização de nenhuma técnica de amostragem. O *softflowd* é um analisador de rede capaz de exportar fluxos no formato NetFlow através do monitoramento de uma interface de rede ou pela leitura de um arquivo. Neste cenário, os fluxos foram exportados na versão 9 do NetFlow por um servidor CentOS 5.5 com o aplicativo NFDUMP, uma ferramenta para coleta e processamento de dados NetFlow. O período de coleta compreendeu os meses de Março, Abril e Maio de 2013, e o método foi avaliado para três semanas deste período (29 de Abril a 17 de Maio).

5.2.2.1 Avaliação da caracterização de tráfego

Na Figura 5.9, tem-se um exemplo da caracterização de tráfego sob este cenário para os dias úteis de uma semana do mês de maio, compreendendo o período entre 6 de Maio a 10 de Maio de 2013. Ao analisar a figura, pode-se observar que não houve um ajuste adequado do DSNSF (em azul) para a maioria das amostras de tráfego mostradas na figura, resultando em uma efetividade menor do que os exemplos exibidos no Cenário 1.

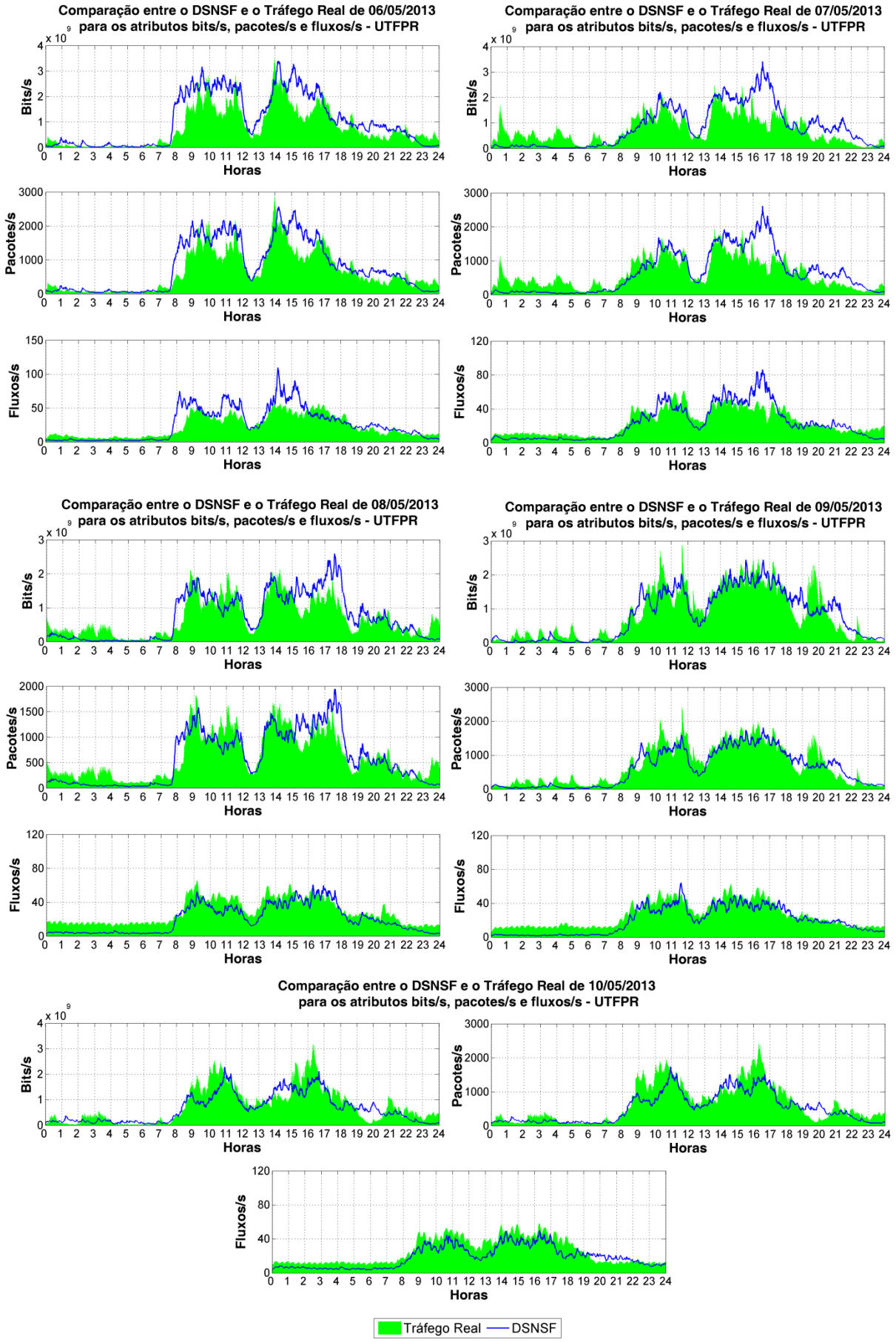

Figura 5.9 - Exemplo de caracterização de tráfego para a rede da UTFPR, Campus Toledo

Os resultados ao aplicar o Erro Quadrático Médio Normalizado (NMSE) neste cenário são exibidos no gráfico da Figura 5.10. Os índices de erros obtidos foram expressivamente maiores do que os observados no mesmo teste no Cenário 1, indicando uma adaptação moderada do modelo de caracterização para os três atributos de volume estudados. Por ser feriado nacional e não haver atividade na universidade, o dia 01 de Maio apresentou um comportamento anormal, resultando em um índice de erro elevado.

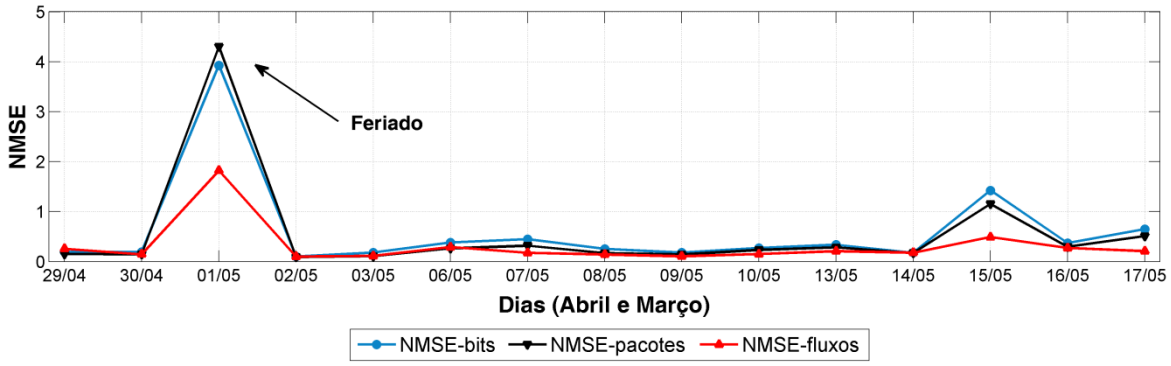

Figura 5.10 - NMSE calculados entre o DSNSF e o tráfego real no período de 29 de Abril a 17 de Maio de 2013

Em seguida, a Figura 5.11 exibe os Coeficientes de Correlação calculados. Com a exceção do dia 01/05, feriado, todos os demais dias analisados indicaram correlação forte entre os DSNSFs e o tráfego real, com coeficientes de correlação acima de 0,7.

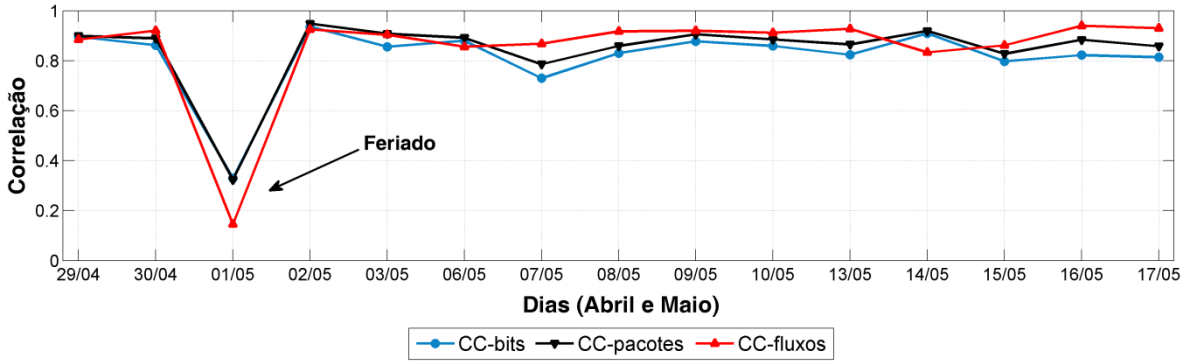

Figura 5.11 - Coeficientes de Correlação calculados entre o DSNSF e o tráfego real no período de 29 de Abril a 17 de Maio de 2013

5.2.2.2 Avaliação da detecção de anomalias

Assim como no Cenário 1, foi criado um gabarito com simulações de ataques e eventos anômalos no tráfego da rede deste cenário utilizando a ferramenta *Scorpius*. A Tabela 5.3 apresenta os dias cujas anomalias foram injetadas, bem como os parâmetros utilizados em cada situação. Assim no Cenário 1, os IPs e Portas utilizados são fictícios, gerados aleatóriamente.

Tabela 5.3 - Parâmetros da simulação de anomalias no cenário 2 utilizando a ferramenta Scorpius

Dia	Anomalia	Intervalo de Tempo	IP de origem (ou núm. de IP de origem)	Porta de origem	IP de destino	Porta de destino	Protocolo
30/04/13	DoS	2h até 4h	82.138.42.211	6195	58.143.61.148	4471	TCP
02/0513	DoS	2h até 5h	187.72.161.80	3453	89.107.142.173	200	TCP
06/0513	Flash Crowd	3h até 6:30h	10	-	240.47.158.218	7206	TCP
09/0513	DDoS	14h até 16h	3	-	172.244.215.131	3776	UDP
10/0513	DDoS	11h até 14h	2	-	149.140.101.67	1483	ICMP
13/0513	DoS	3h até 7h	178.82.128.178	1442	76.198.59.242	1841	UDP
14/0513	Flash Crowd	4h até 7h	6	-	30.7.75.84	8591	TCP
17/0513	Flash Crowd	8h até 11h	5	-	93.70.69.105	6039	TCP

Na Figura 5.12, é exemplificado a geração de alarmes para o tráfego de bits, pacotes e número de fluxos transmitidos por segundo durante dois dias com anomalias injetadas artificialmente neste cenário. A linha azul espessa é o DSNSF, as linhas finas são os *thresholds*, e em vermelho são os alarmes gerados quando o movimento real (em verde) se afasta dos limites definidos pelos *thresholds*.

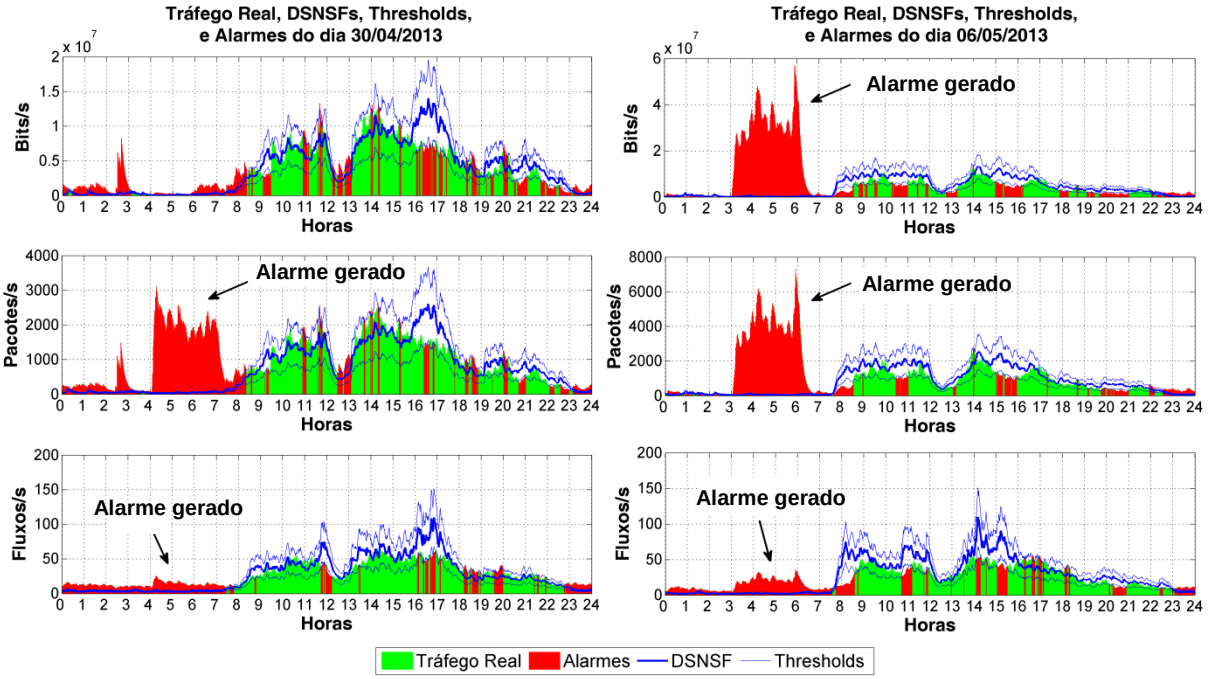

Figura 5.12 - Exemplo de alarmes gerados no tráfego de dois dias na rede da Universidade Tecnológica Federal do Paraná - Campus Toledo para os três atributos quantitativos estudados

Ao analisar a curva ROC na Figura 5.13, calculada para o mesmo conjunto de dados da seção anterior, podemos observar uma taxa de acertos alta, porém, com uma taxa de alarmes falsos relativamente elevada. Com relação ao teste de acurácia, presente na Figura 5.14, o sistema obteve resultados medianos, com uma acurácia média de 70%, ficando abaixo dos resultados obtidos no cenário 1. Nos dois testes, é possível observar que não há grandes ganhos ou perdas ao aumentar ou diminuir o valor do EL calculado pelo módulo de detecção.

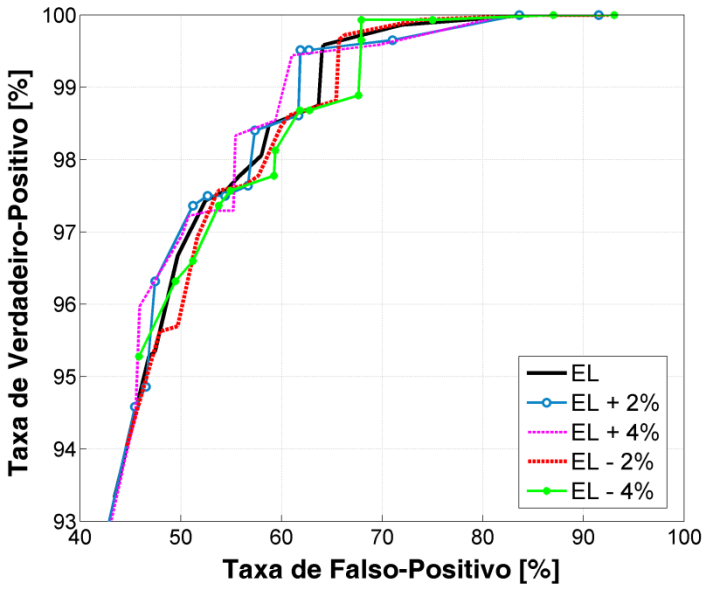

Figura 5.13 - Gráfico ROC mostrando os *trade-offs* entre os valores de TPR e FPR

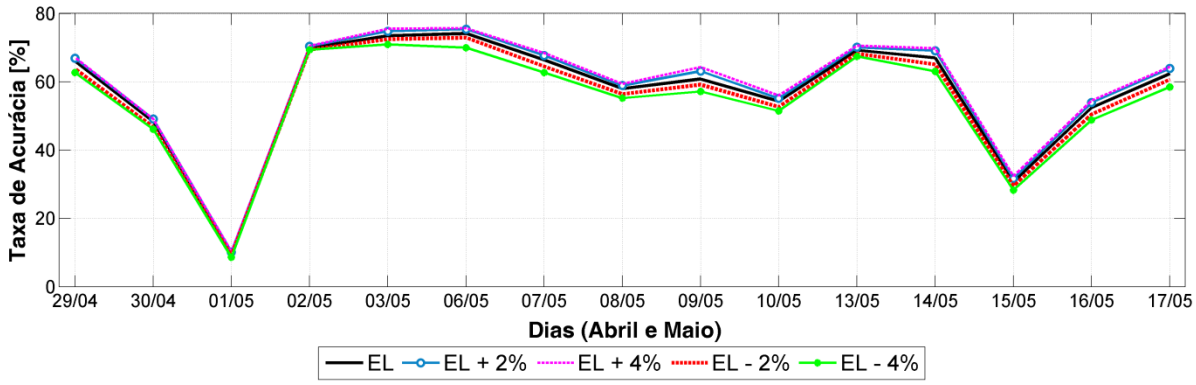

Figura 5.14 - Taxa de acurácia para as três semanas de testes

Por fim, para exemplificar o funcionamento do Módulo de Informação para este cenário, foi selecionado um dos dias dentre o conjunto de dados com anomalias injetadas artificialmente com a ferramenta *Scorpius*. No dia 30 de abril de 2013 foi realizada uma simulação de DoS, e as informações a respeito do ataque se encontram na Tabela 5.3. Os IPs e as Portas são atributos fictícios criados apenas para exemplificar o ataque.

Como podemos observar na Figura 5.15, novamente, os atributos do ataque afetaram uma grande proporção dos atributos qualitativos do tráfego, ocorrendo com mais frequência do que os atributos comuns da rede.

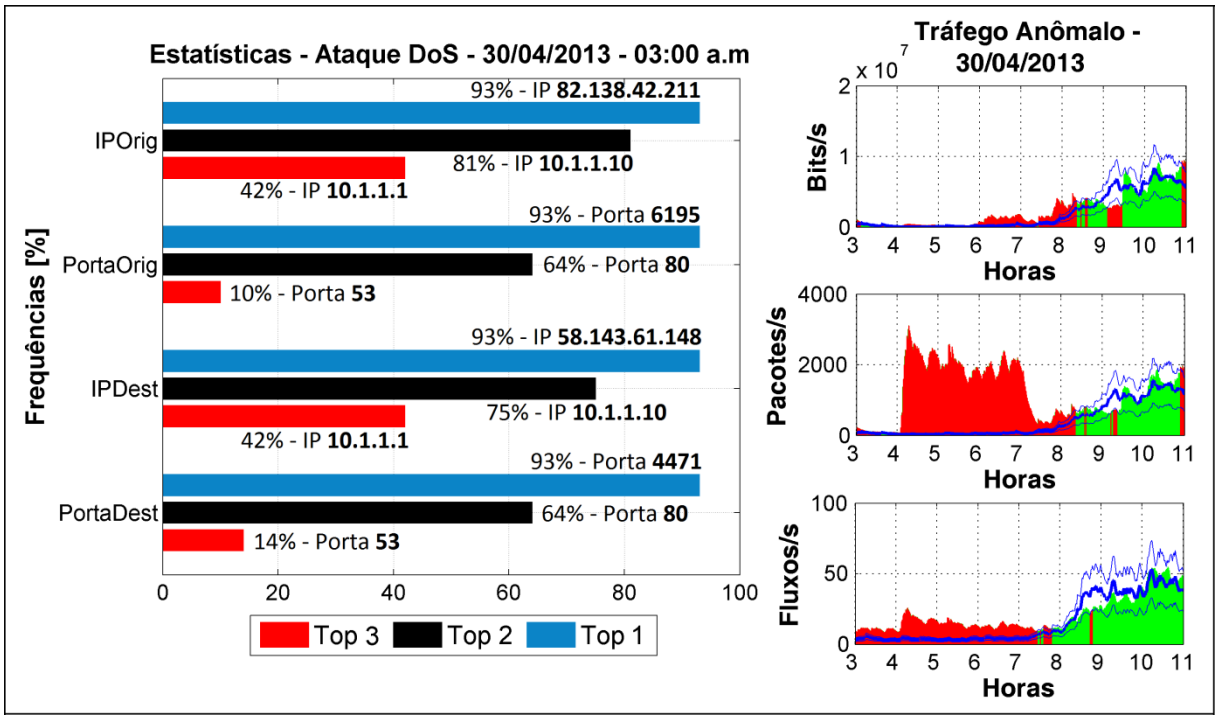

Figura 5.15 - Módulo de informação em uma simulação de DoS

Ao analisar o comportamento do tráfego (em verde) da Figura 5.9, podemos observar que, por se tratar de uma rede de menor porte, atividades isoladas podem afetar de forma significativa o comportamento do tráfego, o que gera a ocorrência de inúmeros *outliers*, comprometendo a precisão do algoritmo de detecção. Portanto, ao analisar todos os testes apresentados nesta seção, chegou-se à conclusão de que o sistema proposto neste trabalho se mostrou menos eficiente do que quando utilizado em redes com grande quantidade de equipamentos e volume de tráfego, contudo como só são gerados alarmes referentes à variações de volume, ou seja, *flash crowds*, o sistema pode ser utilizando, porem novas pesquisas e modificações deverão ser implementados com a finalidade de melhorar a eficiência e acurácia em cenários desse tipo.

5.2.3 Cenário 3 – Análise Comparativa Entre os Modelos PCADS e KMDS

Neste cenário, a caracterização de tráfego produzida pelo PCADS é comparada com o modelo KMDS (*K-Means for Digital Signature*) definido em [68], uma abordagem para caracterização de trafego que utiliza o algoritmo de clusterização *K-means*.

5.2.3.1 KMDS: *K-Means for Digital Signature*

O K-Means (KM) é um processo que divide uma população n-dimensional em *K* grupos com base numa amostra. O KM particiona os pontos de uma matriz de dados em *K* grupos. As linhas da matriz correspondem aos pontos, e as colunas correspondem às variáveis [69].

Com o auxílio da clusterização, uma técnica de mineração de dados, busca-se quantificar os dados similares em determinados grupos. Este processo visa minimizar a distância entre os pontos de um determinado grupo, e aumentar a distância entre os grupos. A equação que mede a similaridade entre os dados é chamada função objetivo, descrita na equação (5).

$$J = \sum_{k=1}^{K} \sum_{s=1}^{S} \sqrt{\left| P_S^k - c^k \right|^2}, \tag{20}$$

sendo K o número de clusters, S o número de pontos, P_S^k o valor dos pontos pertencentes ao cluster k, e c^k o centro do cluster k.

A finalidade do uso da clusterização consiste em criar um modelo a partir do qual seja possível extrair um padrão de informação.

Algoritmo 2 – Algoritmo do KMDS utilizado para a criação do DSNSF

Entrada: X – Conjunto de bits/s, pacotes/s ou fluxos/s coletados de um banco de dados históricos organizado em uma matriz $n \times p$.
Saída: μ – Um vetor representando a estimativa do tráfego de bits/s, pacotes/s ou fluxos/s de um dia, ou seja, o DSNSF.

1: Posicionar os k pontos no espaço que representam os pontos a serem agrupados.
2: Atribuir cada ponto ao grupo mais próximo do centroide.
3: Quando todos os pontos já tiverem sido alocados, recalcular a posição dos k centroides.
4: Repetir os passos 2 e 3 até que os centroides não se movam mais ou o número limite de iterações seja excedido.
5: **for** i = 1 to *total_clusters* do
6: **if** número de pontos no cluster *k* (*i*) < γ
7: ignore o cluster *k* (*i*)
8: **end if**
9: **end for**
10: $\mu \leftarrow média\ ponderada\ entre\ os\ centros$
11: **return** μ

O funcionamento do KMDS é descrito no Algoritmo 2. O KMDS utiliza um total de quatro clusters para gerar um DSNSF, além de definir uma variável γ correspondente ao número mínimo de pontos pertencentes a um cluster, sendo γ = 5%, ou seja, cada cluster deve ter no mínimo 5% do número total de elementos a serem clusterizados. Isso para

prevenir que possíveis *outliers* ou anomalias comprometam a caracterização de tráfego. O resultado do KMDS consiste na média ponderada dos centroides dos clusters mais representativos [68].

5.2.3.2 Análise comparativa

Com a finalidade de comparar os métodos PCADS e KMDS, foi utilizado o tráfego extraído da Universidade Estadual de Londrina no mesmo período utilizado nos testes do cenário 1. No entanto, como será comparada apenas a caracterização de tráfego, os atributos utilizados foram apenas os quantitativos (bits, pacotes e número de fluxos transmitidos por segundo).

Os DSNSFs do KMDS foram gerados conforme os parâmetros presentes em [68], com um total de clusters igual a 4, e utilizando 7 semanas anteriores para treinamento.

A Figura 5.16 exibe um comparativo das assinaturas digitais produzidas pelos dois métodos para os três atributos quantitativos estudados durante o período de uma semana (de 5 a 9 de Novembro de 2012). As linhas azuis são os DSNSFs gerados pelo PCADS, enquanto que as linhas vermelhas, são os DSNSFs do KMDS. Como é possível observar, as duas assinaturas se assemelham, porém, há certos *outliers* presentes nos DSNSFs do KMDS que diferem do comportamento real do tráfego. Isso indica a sensibilidade do KMDS em incorporar comportamentos anormais presentes nos dados de treinamento na assinatura digital, como é evidente nas assinaturas dos dias 8 e 9 de Novembro. Já o PCADS não é tão sensível a tais casos, visto que apresenta uma previsão mais uniforme.

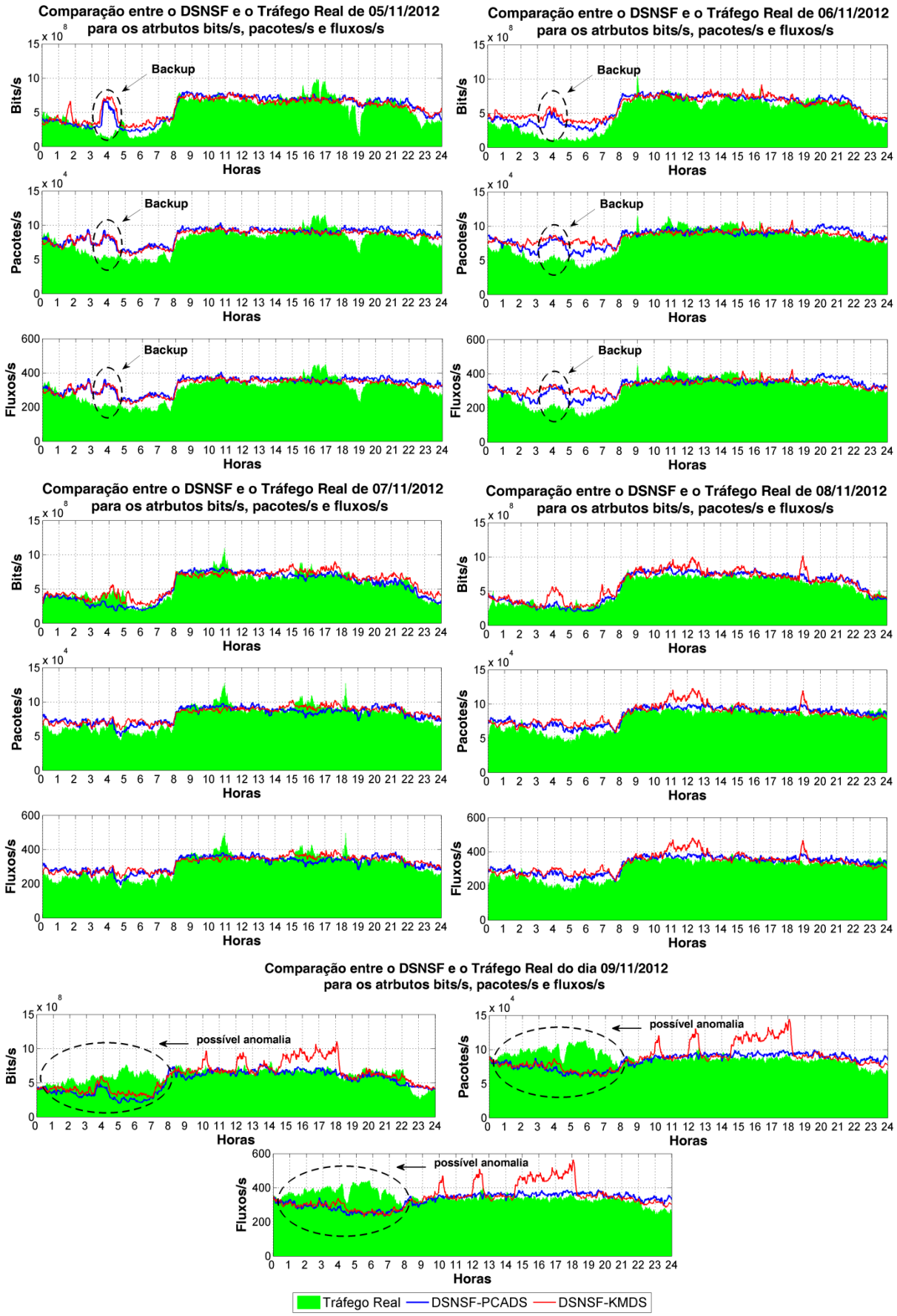

Figura 5.16 - Comparativo entre os DSNSFs produzidos pelos métodos PCADS e KMDS para uma semana

Foi feita uma análise numérica durante 4 semanas com a finalidade de comparar a eficácia dos dois métodos na caracterização de tráfego dos três atributos quantitativos estudados.

A Figura 5.17 exibe os resultados do NMSE, e a Figura 5.18, os resultados do teste de correlação. Ao analisar os resultados destes dois testes, podemos observar que no geral os métodos apresentaram resultados similares, porém, o PCADS obteve melhores resultados para os três atributos, com erros menores para o NMSE e correção forte entre os DSNSFs e o tráfego real.

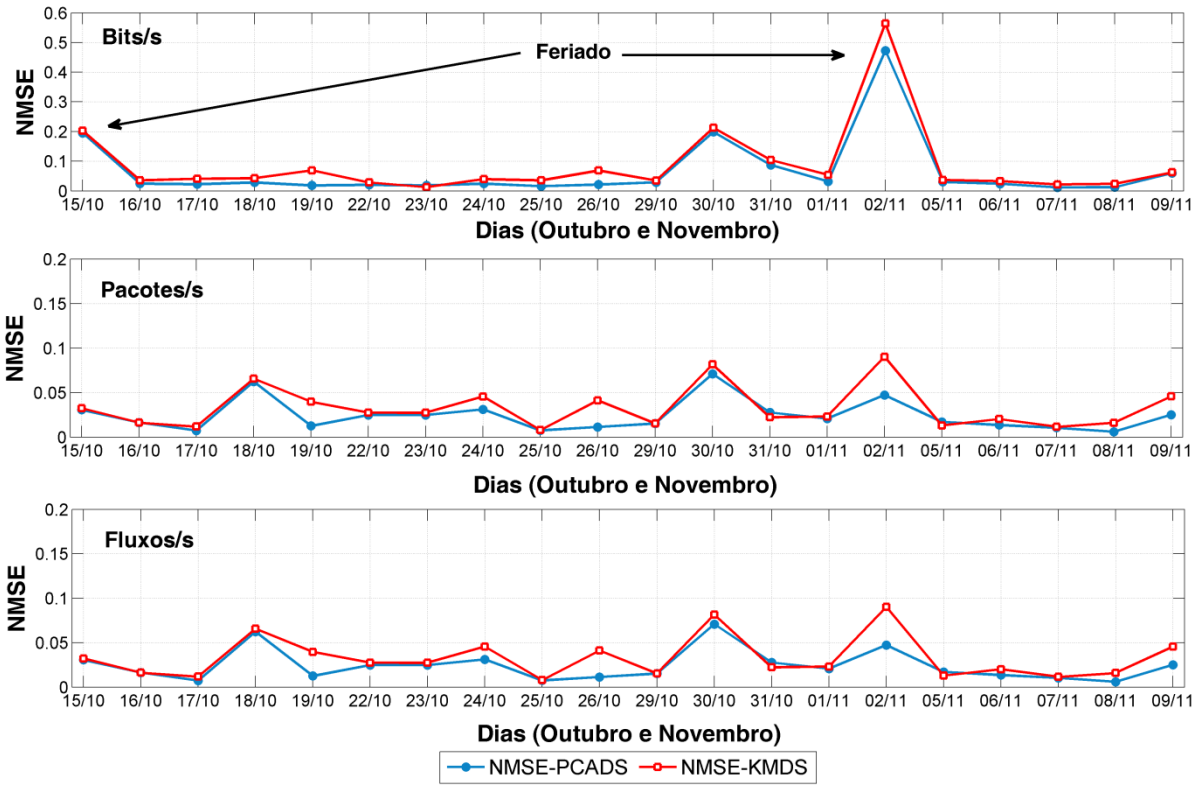

Figura 5.17 - Comparativo entre os métodos PCADS e KMDS utilizando o NMSE

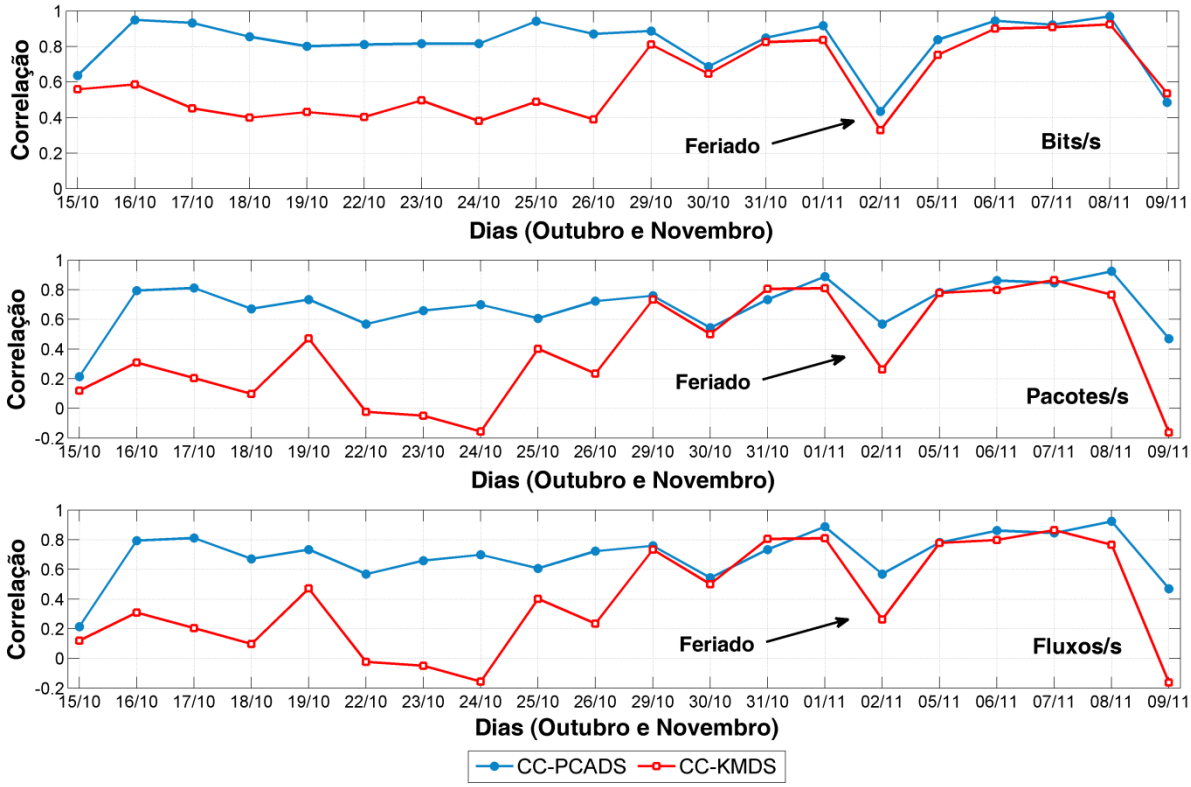

Figura 5.18 - Comparativo entre os métodos PCADS e KMDS utilizando o teste de Correlação

5.3 COMPLEXIDADE COMPUTACIONAL

Dentre todos os passos da Análise de Componentes Principais, o mais custoso se resume ao cálculo dos autovetores e autovalores. Subtrair a média dos dados ou calcular a matriz de covariância não exigem muito poder computacional, pois são operações de ordem linear. Segundo apresentado no trabalho de Lakhina et al. [29], o cálculo das componentes principais de uma matriz X de dimensões n x p é equivalente a resolver o problema do autovalor simétrico para uma matriz de covariância X^T.

Para resolver este problema, é necessário calcular a Decomposição em Valores Singulares (SVD, do inglês *Single Value Decomposition*). Como apresentado por Biblieri e Yao em [70], o SVD é um método utilizado para computar todos os autovetores e autovalores de uma matriz e sua complexidade é de $O(np^2)$. Assim, podemos concluir que a complexidade do PCADS é limitada por $O(np^2)$.

Ao analisar o desempenho do sistema em termos práticos, foram realizados testes extensivos em uma máquina operando com um processador de 3.0 GHz. Nesse

ambiente, o modelo foi capaz de gerar os DSNSFs de um dia para os três atributos em menos de 3 segundos. Além disso, o aumento do número de atributos de fluxos processados ou o aumento na quantidade de segmentos analisados não afeta expressivamente o sistema, já que sua complexidade é limitada apenas pelo número de dias de treinamento (p) e pelo número de instâncias de cada dia (n). Esse desempenho atingido pelo PCADS indica a escalabilidade e viabilidade do modelo em operar, em tempo real, com uso reduzido de recursos computacionais.

6 CONCLUSÃO

Neste trabalho, foi apresentado um sistema de detecção de anomalias baseado em perfil. A principal contribuição consiste na aplicação e contextualização do método PCA a um ambiente de detecção de anomalias. Deste modo, o sistema utiliza o PCADS (*Principal Component Analysis for Digital Signature*), um algoritmo desenvolvido para caracterização de tráfego, o qual é capaz de criar uma assinatura digital de segmento de rede, o DSNSF, baseado no método estatístico PCA (*Principal Component Analysis*), onde são exploradas as características dos autovetores em obter a variância de cada variável do conjunto de dados de entrada, garantindo que tais assinaturas criadas pelo sistema representem as principais características e padrões dos dados de treinamento utilizados. A fim de identificar eventos anômalos, o tráfego real é comparado com o DSNSF de acordo com limites aceitáveis do mesmo (Limites de Autovalor) criados por meio dos autovalores do PCA obtidos durante a fase de caracterização do tráfego.

Alguns tipos de ataques e anomalias de volume como DoS, DDoS e *Flash Crowds* ocasionam variações de volume tráfego em atributos distintos de fluxos IP. Destaca-se também como contribuição obtida, a detecção de anomalias de volume de tráfego mediante a análise de sete atributos de fluxos IP, onde: i) três atributos quantitativos – bits, pacotes e número de fluxos – são utilizados com o objetivo de caracterizar o tráfego de rede por meio do DSNSF, um passo fundamental para identificar de modo eficaz diferentes comportamentos anômalos e ii) quatro atributos descritivos – IP de origem, IP de destino, Porta TCP/UDP de origem e Porta TCP/UDP de destino – que são usados pelo sistema por meio do Módulo de Informação a fim de fornecer ao gerente de rede informações necessárias para identificar o problema e tomar medidas especificas contra ele.

Para validação do sistema, este foi avaliado em três cenários distintos, utilizando dados reais de duas universidades, por meio de métricas estatísticas de avaliação. Em relação à caracterização de tráfego para a criação DSNSF, o sistema proposto obteve bons resultados. Os resultados apresentaram índices de erro NMSE geralmente abaixo de 0,1 para rede da UEL, e abaixo de 0,5 para a rede da UTFPR. Ainda, os DSNSFs gerados pelo PCADS apresentaram correlação forte ao compará-los com o tráfego real em ambos os cenários. No Cenário 3, onde é comparada a caracterização de tráfego do PCADS e KMDS, apesar da semelhança das assinaturas produzidas pelos dois métodos, o PCADS apresentou melhores

resultados nos testes, além do fato de não ser muito sensível à *outliers* e anomalias que possam existir nos dados de treinamento durante a caracterização do tráfego.

Com relação à detecção de anomalias, os resultados referentes às taxas de alarmes falsos e a taxa de acurácia do Cenário 1 são promissoras. Porém, no Cenário 2, o módulo de detecção do sistema obteve resultados abaixo da média. Por se tratar de uma rede de pequeno porte, o tráfego da rede da UTFPR, estudada no Cenário 2, possui inúmeros *outliers*, uma vez que um único usuário tem a capacidade de alterar o comportamento normal da rede através de uma utilização mais intensa, comprometendo a precisão do sistema. Deste modo, na rede do Cenário 2, com pequeno número de equipamentos e baixo volume de tráfego, os resultados se mostraram inferiores ao Cenário 1, que não inviabiliza o processo, contudo dada essa característica, novas pesquisas e testes devem ser realizados com o objetivo de aprimorar a eficácia do sistema. Em uma rede de grande porte, como a do Cenário 1, este problema não é verificado, visto que por suportar um tráfego mais elevado, esta é menos sensível a uma quantidade pequena de dados que diferem do padrão.

Os resultados obtidos utilizando dados reais, juntamente com a baixa complexidade computacional do PCADS, implicam que esta abordagem utilizando a Análise de Componentes Principais pode ter alta aplicabilidade para a identificação automática de anomalias de volume através de uma assinatura digital, constituindo um passo promissor rumo a um sistema mais amplo para o diagnóstico de anomalias em redes de grande porte.

Como trabalhos futuros, pretende-se aperfeiçoar os métodos de caracterização de tráfego e de detecção de anomalias do sistema, principalmente com relação à sua aplicabilidade em redes de pequeno porte, almejando taxas de alarmes falsos mais factíveis. Outro passo consiste em realizar comparações com outros sistemas de detecção, bem como utilizar mais cenários e diferentes tipos de ataques para a validação do sistema. Por fim, mediante o uso da variedade de atributos e informações provenientes dos fluxos IP, o próximo passo é, ao identificar um evento anômalo, classificar o tipo de anomalia ocorrido, a fim de que medidas preventivas possam ser tomadas para cada tipo específico de problema sem intervenção humana, a contribuir para um sistema proativo, ágil e eficiente.

REFERÊNCIAS

[1] THATTE, G.; MITRA, U.; HEIDEMANN, J. Parametric Methods for Anomaly Detection in Aggregate Traffic. **Networking, IEEE/ACM Transactions on,** v. 19, n. 2, p. 512-525, 2011. ISSN 1063-6692.

[2] GARCÍA-TEODORO, P.; DÍAZ-VERDEJO, J.; MACIÁ-FERNÁNDEZ, G.; VÁZQUEZ, E. Anomaly-based network intrusion detection: Techniques, systems and challenges. **Computers & Security,** v. 28, n. 1–2, p. 18-28, 2// 2009. ISSN 0167-4048. Disponível em: < http://www.sciencedirect.com/science/article/pii/S0167404808000692 >. Acesso em: 2009/3//.

[3] LU, K.; WU, D.; FAN, J.; TODOROVIC, S.; NUCCI, A. Robust and efficient detection of DDoS attacks for large-scale internet. **Comput. Netw.,** v. 51, n. 18, p. 5036-5056, 2007. ISSN 1389-1286.

[4] QING, S.; WEN, W. A survey and trends on Internet worms. **Computers & Security,** v. 24, n. 4, p. 334-346, 6// 2005. ISSN 0167-4048. Disponível em: < http://www.sciencedirect.com/science/article/pii/S0167404804002585 >.

[5] YAO, G.; BI, J.; XIAO, P. VASE: Filtering IP spoofing traffic with agility. **Computer Networks,** v. 57, n. 1, p. 243-257, 1/16/ 2013. ISSN 1389-1286. Disponível em: < http://www.sciencedirect.com/science/article/pii/S1389128612003349 >.

[6] VOAS, J. M.; PAYNE, J. E.; COHEN, F. B. A model for detecting the existence of software corruption in real time. **Computers & Security,** v. 12, n. 3, p. 275-283, 5// 1993. ISSN 0167-4048. Disponível em: < http://www.sciencedirect.com/science/article/pii/016740489390117N >.

[7] PING, D.; ABE, S.; YUSHENG, J.; SATO, S.; ISHIGURO, M. **Detecting and Tracing Traffic Volume Anomalies in SINET3 Backbone Network**. Communications, 2008. ICC '08. IEEE International Conference on. 19-23 May 2008, 2008. 5833-5837 p.

[8] CASAS, P.; FILLATRE, L.; VATON, S.; NIKIFOROV, I. Volume Anomaly Detection in Data Networks: An Optimal Detection Algorithm vs. the PCA Approach. In: VALADAS, R. e SALVADOR, P. (Ed.). **Traffic Management and Traffic Engineering for the Future Internet**: Springer Berlin Heidelberg, v.5464, 2009. cap. 7, p.96-113. (Lecture Notes in Computer Science). ISBN 978-3-642-04575-2.

[9] PATCHA, A.; PARK, J.-M. An overview of anomaly detection techniques: Existing solutions and latest technological trends. **Comput. Netw.,** v. 51, n. 12, p. 3448-3470, 2007. ISSN 1389-1286.

[10] KRISHNAMURTHY, B.; SEN, S.; ZHANG, Y.; CHEN, Y. **Sketch-based change detection: methods, evaluation, and applications**. Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement. Miami Beach, FL, USA: ACM**:** 234-247 p. 2003.

[11] PRIETO, A. G.; STADLER, R. **Adaptive real-time monitoring for large-scale networked systems**. Integrated Network Management, 2009. IM '09. IFIP/IEEE International Symposium on. 1-5 June 2009, 2009. 790-795 p.

[12] PROENÇA, M., JR.; COPPELMANS, C.; BOTTOLI, M.; SOUZA MENDES, L. Baseline to help with network management. In: ASCENSO, J.;VASIU, L., *et al* (Ed.). **e-Business and Telecommunication Networks**: Springer Netherlands, 2006. cap. 12, p.158-166. ISBN 978-1-4020-4760-2.

[13] ZHOU, C. V.; LECKIE, C.; KARUNASEKERA, S. Decentralized multi-dimensional alert correlation for collaborative intrusion detection. **J. Netw. Comput. Appl.,** v. 32, n. 5, p. 1106-1123, 2009. ISSN 1084-8045.

[14] XU, X. Sequential anomaly detection based on temporal-difference learning: Principles, models and case studies. **Applied Soft Computing,** v. 10, n. 3, p. 859-867, 6// 2010. ISSN 1568-4946. Disponível em: < http://www.sciencedirect.com/science/article/pii/S1568494609001756 >.

[15] LIN, S.-W.; YING, K.-C.; LEE, C.-Y.; LEE, Z.-J. An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. **Applied Soft Computing,** v. 12, n. 10, p. 3285-3290, 10// 2012. ISSN 1568-4946. Disponível em: < http://www.sciencedirect.com/science/article/pii/S1568494612002402 >.

[16] ANDROULIDAKIS, G.; PAPAVASSILIOU, S. **Intelligent Flow-Based Sampling for Effective Network Anomaly Detection**. Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE. 26-30 Nov. 2007, 2007. 1948-1953 p.

[17] BARFORD, P.; KLINE, J.; PLONKA, D.; RON, A. **A signal analysis of network traffic anomalies**. Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment. Marseille, France: ACM**:** 71-82 p. 2002.

[18] LI, H. **Research of K-MEANS Algorithm Based on Information Entropy in Anomaly Detection**. Multimedia Information Networking and Security (MINES), 2012 Fourth International Conference on. 2-4 Nov. 2012, 2012. 71-74 p.

[19] YASAMI, Y.; MOZAFFARI, S. P.; KHORSANDI, S. **Stochastic learning automata-based time series analysis for network anomaly detection**. Telecommunications, 2008. ICT 2008. International Conference on. 16-19 June 2008, 2008. 1-6 p.

[20] KIND, A.; STOECKLIN, M. P.; DIMITROPOULOS, X. Histogram-based traffic anomaly detection. **Network and Service Management, IEEE Transactions on,** v. 6, n. 2, p. 110-121, 2009. ISSN 1932-4537.

[21] TAJBAKHSH, A.; RAHMATI, M.; MIRZAEI, A. Intrusion detection using fuzzy association rules. **Applied Soft Computing,** v. 9, n. 2, p. 462-469, 3// 2009. ISSN 1568-4946. Disponível em: < http://www.sciencedirect.com/science/article/pii/S1568494608000975 >.

[22] LIMA, M. F.; SAMPAIO, L. D. H.; ZARPELÃO, B. B.; RODRIGUES, J. J. P. C.; ABRÃO, T.; JR, M. L. P. **Networking Anomaly Detection using DSNS and Particle Swarm Optimization with Re-Clustering**. IEEE GLOBAL COMMUNICATIONS CONFERENCE (IEEE GLOBECOM 2010). Miami, USA: IEEE, 2010. p.

[23] RAWAT, S.; GULATI, V. P.; PUJARI, A. K. **Frequency- and ordering-based similarity measure for host-based intrusion detection**. Bingley, ROYAUME-UNI: Emerald, 2004. 11

[24] SODIYA, A. S.; LONGE, H. O. D.; AKINWALE, A. T. A new two-tiered strategy to intrusion detection. **Information Management & Computer Security,** v. 12, n. 1, p. 18, 2004.

[25] MAZEL, J.; CASAS, P.; LABIT, Y.; OWEZARSKI, P. **Sub-Space clustering, Inter-Clustering Results Association & anomaly correlation for unsupervised network anomaly detection**. Network and Service Management (CNSM), 2011 7th International Conference on. 24-28 Oct. 2011, 2011. 1-8 p.

[26] DUFFIELD, N.; HAFFNER, P.; KRISHNAMURTHY, B.; RINGBERG, H. **Rule-Based Anomaly Detection on IP Flows**. INFOCOM 2009, IEEE. 19-25 April 2009, 2009. 424-432 p.

[27] FONTUGNE, R.; FUKUDA, K. A Hough-transform-based anomaly detector with an adaptive time interval. **SIGAPP Appl. Comput. Rev.,** v. 11, n. 3, p. 41-51, 2011. ISSN 1559-6915.

[28] ROSSI, D.; VALENTI, S. **Fine-grained traffic classification with netflow data**. Proceedings of the 6th International Wireless Communications and Mobile Computing Conference. Caen, France: ACM**:** 479-483 p. 2010.

[29] LAKHINA, A.; CROVELLA, M.; DIOT, C. Diagnosing network-wide traffic anomalies. **SIGCOMM Comput. Commun. Rev.,** v. 34, n. 4, p. 219-230, 2004. ISSN 0146-4833.

[30] _____. Mining anomalies using traffic feature distributions. **SIGCOMM Comput. Commun. Rev.,** v. 35, n. 4, p. 217-228, 2005. ISSN 0146-4833.

[31] PASCOAL, C.; ROSARIO DE OLIVEIRA, M.; VALADAS, R.; FILZMOSER, P.; SALVADOR, P.; PACHECO, A. **Robust feature selection and robust PCA for internet traffic anomaly detection**. INFOCOM, 2012 Proceedings IEEE. 25-30 March 2012, 2012. 1755-1763 p.

[32] RINGBERG, H.; SOULE, A.; REXFORD, J.; DIOT, C. **Sensitivity of PCA for traffic anomaly detection**. Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems. San Diego, California, USA: ACM**:** 109-120 p. 2007.

[33] KANDA, Y.; FONTUGNE, R.; FUKUDA, K.; SUGAWARA, T. ADMIRE: Anomaly detection method using entropy-based PCA with three-step sketches. **Computer Communications,** v. 36, n. 5, p. 575-588, 3/1/ 2013. ISSN 0140-3664. Disponível em: < http://www.sciencedirect.com/science/article/pii/S0140366412003994 >.

[34] KANDA, Y.; FUKUDA, K.; SUGAWARA, T. **Evaluation of Anomaly Detection Based on Sketch and PCA**. Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE. 6-10 Dec. 2010, 2010. 1-5 p.

[35] CALLEGARI, C.; GAZZARRINI, L.; GIORDANO, S.; PAGANO, M.; PEPE, T. **A Novel PCA-Based Network Anomaly Detection**. Communications (ICC), 2011 IEEE International Conference on. 5-9 June 2011, 2011. 1-5 p.

[36] ISO. **Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 4: Management framework**. ISO/IEC 7498, 1989, International Standards Organization.

[37] UDUPA, D. K. **Network Management Systems Essentials**. McGraw-Hill, Inc., 1995. 557 ISBN 0070657661.

[38] OLIVEIRA, D. T. **Gerência de Redes de Computadores: Uma Abordagem com o uso do SNMP**. Centro Universitário do Triângulo - Unit. Uberlândia, Dez, 2002.

[39] CISCO. Introduction to Cisco IOS NetFlow - A Technical Overview. 2012 Disponível em: < http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6555/ps6601/prod_white_paper0900aecd80406232.html >. Acesso em: 16/09/2013.

[40] CLAISE, B. RFC5101 - Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. 2008. Disponível em: < http://www.ietf.org/rfc/rfc5101.txt >.

[41] PHAAL, P.; PANCHEN, S.; MCKEE, N. RFC 3176 - InMon Corporation' s sFlow: A Method for Monitoring Traffic in Switched and Routed Networks. Internet Engineering Task Force (IETF), September 2001. Disponível em: < http://www.ietf.org/rfc/rfc3176.txt >.

[42] CLAISE, B. RFC3954 - Cisco Systems NetFlow Services Export Version 9. Internet Engeneering Task Force (IETF), 2004. Disponível em: < http://www.ietf.org/rfc/rfc3954.txt >.

[43] ROHMAD, M. S.; AZMAT, F.; MANAF, M.; MANAN, J. L. A. **Enhanced Netflow version 9 (e-Netflow v9) for network mediation: Structure, experiment and analysis**. Information Technology, 2008. ITSim 2008. International Symposium on. 26-28 Aug. 2008, 2008. 1-6 p.

[44] HANDELMAN, S.; STIBLER, S.; BROWNLEE, N.; RUTH, G. RFC2724 - RTFM: New Attributes for Traffic Flow Measurement. Internet Engeneering Task Force (IETF), 1999. Disponível em: < http://www.ietf.org/rfc/rfc2724.txt >.

[45] MOLINA, M.; TARTARELLI, S.; RASPALL, F.; NICCOLINI, S. **Implementation of an IPFIX compliant flow traffic meter: challenges and performance assessment**. IP Operations & Management, 2003. (IPOM 2003). 3rd IEEE Workshop on. 1-3 Oct. 2003, 2003. 61-67 p.

[46] QUITTEK, J.; ZSEBY, T.; CLAISE, B.; ZANDER, S. RFC3917 - Requirements for IP Flow Information Export (IPFIX). 2004. Disponível em: < http://www.ietf.org/rfc/rfc3917.txt >.

[47] TRAMMELL, B.; BOSCHI, E. An introduction to IP flow information export (IPFIX). **Communications Magazine, IEEE,** v. 49, n. 4, p. 89-95, 2011. ISSN 0163-6804.

[48] SADASIVAN, G.; BROWNLEE, N.; CLAISE, B.; QUITTEK, J. RFC5470 - Architecture for IP Flow Information Export. 2009. Disponível em: < http://www.ietf.org/rfc/rfc5470.txt >.

[49] KOBAYASHI, A.; CLAISE, B. RFC5982 - IP Flow Information Export (IPFIX) Mediation: Problem Statement. 2010. Disponível em: < http://www.ietf.org/rfc/rfc5982.txt >.

[50] KOBAYASHI, A.; CLAISE, B.; MUENZ, G.; ISHIBASHI, K. RFC6183 - IP Flow Information Export (IPFIX) Mediation: Framework. 2011. Disponível em: < http://www.ietf.org/rfc/rfc6183.txt >.

[51] IANA. IP Flow Information Export (IPFIX) Entities. Internet Assigned Numbers Authority (IANA), 2011. Disponível em: < http://www.iana.org/assignments/ipfix/ipfix.xhtml >. Acesso em: 16 de Set. 2013.

[52] Scrutinizer NetFlow & sFlow Analyzer. Plixer International, Disponível em: < http://www.plixer.com/Scrutinizer-Netflow-Sflow/scrutinizer-flow-analyzer.html >. Acesso em: 16 de Set. 2013.

[53] NfSen Documentation. Disponível em: < http://nfsen.sourceforge.net/ >. Acesso em: 19 de Set. 2013.

[54] PRAS, A. nTop – Network TOP: An Overview. University of Twenty, The Netherlands, Disponível em: < http://www.ntop.org/wp-content/uploads/2011/09/ntop-overview.pdf >. Acesso em: 19 de Set 2013.

[55] Traffic Monitoring Using sFlow. InMon Corp., sflow.org, 2003. Disponível em: < http://www.sFlow.org/sFlowOverview.pdf >. Acesso em: 19 de Set. 2013.

[56] PEARSON, K. On lines and planes of closest fit to systems of points in space. **Philosophical Magazine,** v. 2, n. 6, p. 559-572, 1901.

[57] HOTELLING, H. **Analysis of a complex of statistical variables into principal components**. Baltimore: Warwick & York, 1933. 48 p. Disponível em: < http://catalog.hathitrust.org/Record/006826924
http://hdl.handle.net/2027/wu.89097139406 >.

[58] SMITH, L. I. **A tutorial on principal components analysis**. 2002

[59] DONGARRA, J. J.; SORCNSEN, D. C. **A fast algorithm for the symmetric eigenvalue problem**. Computer Arithmetic (ARITH), 1985 IEEE 7th Symposium on. 4-6 June 1985, 1985. 337-342 p.

[60] PARLETT, B. **The Symmetric Eigenvalue Problem (Classics in Applied Mathematics)**. Society for Industrial Mathematics, 1987. ISBN 0898714028. Disponível em: < http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0898714028 >.

[61] JOLLIFFE, I. T. Principal Component Analysis. **Springer Verlag,** v. 2, p. 489, 2002.

[62] nfdump - Documentation. Disponível em: < http://nfdump.sourceforge.net/ >. Acesso em: 16 de Set. 2013.

[63] PROENCA, M. L.; ZARPELAO, B. B.; MENDES, L. S. **Anomaly detection for network servers using digital signature of network segment**. Telecommunications, 2005. advanced industrial conference on telecommunications/service assurance with partial and intermittent resources conference/e-learning on telecommunications workshop. aict/sapir/elete 2005. proceedings. 17-20 July 2005, 2005. 290-295 p.

[64] POLI, A. A.; CIRILLO, M. C. On the use of the normalized mean square error in evaluating dispersion model performance. **Atmospheric Environment. Part A. General Topics,** v. 27, n. 15, p. 2427-2434, 10// 1993. ISSN 0960-1686. Disponível em: < http://www.sciencedirect.com/science/article/pii/096016869390410Z >.

[65] PROENÇA, M. L. J.; ZARPELÃO, B. B.; MENDES, L. D. S. Anomaly Detection Using Digital Signature of Network Segment Aiming to Help Network Management **Journal of Communication and Information Systems (JCIS),** v. 23, n. 1, p. 11, 2008. ISSN 1980-6604.

[66] BENESTY, J.; CHEN, J.; HUANG, Y.; COHEN, I. Pearson Correlation Coefficient. In: (Ed.). **Noise Reduction in Speech Processing**: Springer Berlin Heidelberg, v.2, 2009. cap. 5, p.1-4. (Springer Topics in Signal Processing). ISBN 978-3-642-00295-3.

[67] FAWCETT, T. An introduction to ROC analysis. **Pattern Recogn. Lett.,** v. 27, n. 8, p. 861-874, 2006. ISSN 0167-8655.

[68] FERNANDES, G.; ZACARON, A. M.; RODRIGUES, J. J. P. C.; LEMES PROENCA, M. **Digital signature to help network management using principal component analysis and K-means clustering**. Communications (ICC), 2013 IEEE International Conference on. 9-13 June 2013, 2013. 2519-2523 p.

[69] MACQUEEN, J. **Some methods for classification and analysis of multivariate observations**. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics. Berkeley, Calif.: University of California Press. 1967, 1967. 281-297 p.

[70] BIGLIERI, E.; YAO, K. Some properties of singular value decomposition and their applications to digital signal processing. **Signal Processing,** v. 18, n. 3, p. 277-289, 11// 1989. ISSN 0165-1684. Disponível em: < http://www.sciencedirect.com/science/article/pii/016516848990039X >.

[71] Scorpius - Simulador de Anomalias utilizando Fluxos IP, Grupo de redes UEL, Disponível em: < http://redes.dc.uel.br/scorpius >. Acesso em 28 de Mar. 2014

Trabalhos publicados pelo Autor

Conferências:

1. FERNANDES JR, G.; ZACARON, ALEXANDRO, M.; RODRIGUES, J. J. P. C.; PROENÇA JR, Mario Lemes. Digital Signature to Help Network Management Using Principal Component Analysis and K-Means Clustering. In: IEEE International Conference on Communications (IEEE ICC 2013), 2013, Budapest. IEEE International Conference on Communications (IEEE ICC 2013). Budapest, 2013. (Qualis 2013, A2).

2. CARVALHO, L. F.; ASSIS, M. V. O.; FERNANDES JR, G.; RODRIGUES, J. J. P. C.; PROENÇA JR, Mario Lemes. Digital Signature of Network Segment using PCA, ACO and Holt-Winters for Network Management. In: 15th IEEE International Conference on e-Health Networking, Application & Services (IEEE Healthcom 2013), 2013, Lisbon. 15th IEEE International Conference on e-Health Networking, Application & Services (IEEE Healthcom 2013). Lisbon, 2013. (Qualis 2013, B4)

Periódicos:

3. PROENÇA JR, Mario Lemes; FERNANDES JR, G.; CARVALHO, L. F.; Marcos V. O. de Assis; RODRIGUES, J. J. P. C. Digital Signature to Help Network Management Using Flow Analysis. International Journal of Network Management (IJNM), ISSN: 1099-1190, 2013. (Qualis 2013, B3) (*Em avaliação*)

4. CARVALHO, L. F.; FERNANDES JR, G.; ASSIS, M. V. O.; RODRIGUES, J. J. P. C.; PROENÇA JR, Mario Lemes. Digital Signature of Network Segment to Help Healthcare Environments. Innovation and Research in BioMedical engineering (IRBM), ISSN: 1959-0318, 2013. (*Em avaliação*).

5. FERNANDES JR, G.; PROENÇA JR, Mario Lemes. Autonomous Profile-based Anomaly Detection System using Principal Component Analysis and Flow Analysis. Applied Soft Computing, ISSN: 1568-4946. (Qualis 2013, A2) (*Em avaliação*).