UNIVERSIDADE ESTADUAL DE LONDRINA

FADIR SALMEN

SISTEMA DE DETECÇÃO DE ANOMALIAS UTILIZANDO METAHEURÍSTICA *FIREFLY*

Londrina
2017

FADIR SALMEN

SISTEMA DE DETECÇÃO DE ANOMALIAS UTILIZANDO METAHEURÍSTICA *FIREFLY*

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Mario Lemes Proença Junior.

Londrina
2017

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UEL

Salmen, Fadir.
SISTEMA DE DETECÇÃO DE ANOMALIAS UTILIZANDO METAHEURÍSTICA FIREFLY / Fadir Salmen. - Londrina, 2017.
125 f. : il.

Orientador: Mario 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, 2017.
Inclui bibliografia.

1. Caracterização de tráfego - Tese. 2. Detecção de Anomalia - Tese. 3. Algoritmo Firefly - Tese. 4. Fluxos IP - Tese. I. Lemes Proença Junior, Mario. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. III. Título.

FADIR SALMEN

SISTEMA DE DETECÇÃO DE ANOMALIAS UTILIZANDO METAHEURÍSTICA *FIREFLY*

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

BANCA EXAMINADORA

Prof. Dr. Mario Lemes Proença Junior
Universidade Estadual de Londrina
Orientador

Prof. Dr. Lucas Dias Hiera Sampaio
Universidade Tecnológica Federal do Paraná, campus Cornélio Procópio

---

Prof. Dr. Bruno Bogaz Zarpelão
Universidade Estadual de Londrina

Prof. Dr. Wesley Attrot
Universidade Estadual de Londrina

Londrina–PR, 03 de abril de 2017

À minha esposa Cilene, pelo apoio e dedicação. Aos meus filhos, Daniele e Rafael, que me fizeram conhecer o que é amor incondicional.

AGRADECIMENTOS

À minha querida esposa, Cilene, pela compreensão, paciência, apoio, incentivo, cumplicidade, alegria e amor a mim dedicados durante todos os anos de união.

Aos meus amados filhos, Daniele e Rafael, por entenderam as minhas faltas e momentos de afastamento.

Aos meus saudosos pais, Fauez e Dirce, os quais me proporcionaram o alicerce para a construção do meu caráter e moral.

Ao professor Dr. Mario Lemes Proença Jr, expresso o meu profundo agradecimento pela orientação, sugestões, conselhos e apoio incondicional disponibilizados na realização deste trabalho e que muito contribuíram no enriquecimento dos meus conhecimentos científicos.

Ao professor Dr. Bruno Bogaz Zarpelão, atencioso e prestativo, com quem tive o privilégio de ser aluno e que não mediu esforços em partilhar seus conhecimentos.

Ao Luiz F. Carvalho, pelo auxílio no desenvolvimento das pesquisas de mestrado e desse trabalho. Uma pessoa com grande capacidade técnica e competência e que hoje tenho o prazer de tê-lo como amigo.

Ao amigo Paulo Galego, pelo apoio e companheirismo nesta jornada de mestrado. Além, dos momentos prazerosos compartilhados nas viagens até a UEL.

Ao colega Anderson Hamamoto, pela contribuição nesse trabalho e a todos que o fizeram indiretamente.

Ao Departamento de Computação, em especial ao Anderson, pela atenção às minhas solicitações.

A todos os funcionários da UEL, em seu efetivo administrativo e operacional.

ALLAHU AKBAR

"Não confunda derrotas com fracasso nem vitórias com sucesso. Na vida de um campeão sempre haverá algumas derrotas, assim como na vida de um perdedor sempre haverá vitórias. A diferença é que, enquanto os campeões crescem nas derrotas, os perdedores se acomodam nas vitórias."

Roberto Shinyashiki

SALMEN, FADIR. **Sistema de Detecção de Anomalias utilizando Metaheurística *Firefly***. 125 p. Dissertação de Mestrado (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina–PR, 2017.

RESUMO

O monitoramento do tráfego de redes de computadores é uma tarefa desafiadora. Diversas técnicas são utilizadas para garantir o funcionamento adequado dessas redes. Dentre elas, podemos destacar as responsáveis por identificar os comportamentos anômalos no tráfego. Essas anomalias, que podem ser caracterizadas por ataques, invasões ou falhas, acarretam prejuízos significativos no funcionamento correto das redes, bem como afetam a disponibilidade e confiabilidade de serviços prestados. Neste trabalho, apresentamos o modelo *Firefly Algorithm for Digital Signature* (FADS). Esse modelo, utilizando a análise de fluxos IP, caracteriza o tráfego aplicando o algoritmo meta-heurístico *Firefly* e identifica os comportamentos anômalos da rede baseado no conceito de redes neurais artificiais. Para avaliar o modelo proposto foram coletados fluxos IP da Universidade Estatual de Londrina e da Universidade Tecnológica Federal do Paraná, campus Toledo. Desses fluxos, foram utilizados seis atributos, bits/s, pacotes/s, endereço IP de Origem, endereço IP de destino, porta de origem e porta de destino, com o objetivo de detectar as anomalias no tráfego de rede. Essas universidades proporcionaram testar o FADS em dois ambientes com características distintas. Os resultados alcançados evidenciam a eficiência do FADS na caracterização do tráfego e detecção de anomalias em um segmento de rede, auxiliando os administradores nas tarefas rotineiras de monitoramento e controle.

Palavras-chave: Caracterização de tráfego. Detecção de Anomalia. Algoritmo *Firefly*. Fluxos IP.

SALMEN, FADIR. **System of Detection of Anomalies using Firefly Metaheuristics**. 125 p. Master's Thesis (Master in Science in Computer Science) – State University of Londrina, Londrina–PR, 2017.

ABSTRACT

Monitoring the traffic of computer networks is a challenging task. Several techniques are used to unsure the adequate operation of these networks. Among them, we are able to highlight those responsible for identifying the anomalous behaviors in the traffic. These anomalies, that may be characterized by attacks, intrusions or faults, result in significant damages in the correct functioning of the networks, as well as affect the availability and reliability of performed services. On this work, we present the Firefly Algorithm for Digital Signature (FADS) model. This model, using the analysis of IP flows, characterizes the traffic applying Firefly metaheuristic algorithm and identifies the anomalous behaviors of the network based on the concept of artificial neural networks. To evaluate the proposed model, IP flows from the State University of Londrina and the Federal Technological University of Paraná, Toledo campus, were collected. From these flows, six attributes have been used, bits/s, packets/s, source IP address, destination IP address, source port and destination port, in order to detect anomalies in network traffic. These universities have provided tests on the FADS in two environments with different characteristics. The obtained results have shown FADS efficiency in the characterization of the traffic and the detection of anomalies in a segment of network, helping the administrators in the frequent tasks of monitoring and control.

Keywords: Traffic characterization. Anomaly Detection. Firefly Algorithm. IP Flows.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACO	*Ant Colony Optimization*
ADTW	*Adaptive Dynamic Time Warping*
AMON	*All−Packet Monitor*
ARIMA	*Autoregressive Integrated Moving Average*
CC	Coeficiente de Correlação
CoS	*Class of Service*
CVAP	*Validation for Cluster Analyses Platform*
DoS	*Denial of Service*
DDoS	*Distributed Denial of Service*
DSNS	*Digital Signature of Network Segment*
DSNSF	*Digital Signature of Network Segment using Flow Analysis*
EWMA	*Exponentially Weighted Moving Average*
E(IPDt)	Entropia IP de Destino
E(IPOr)	Entropia IP de Origem
E(PoDt)	Entropia Porta de Destino
E(IPOr)	Entropia Porta de Origem
FA	*Firefly Agorithm*
FADS	*Firefly Agorithm for Digital Signature*
FCAPS	*Fault, Configuration, Account, Performance, Security*
FN	Falso Negativo
FP	Falso Positivo
GA	*Genetic Agorithm*
GBA	*Gerenciamento de Backbone Automatizado*
HTTPS	*Hyper Text Transfer Protocol Secure*

HWDS	*Holt-Winters for Digital Signature*
ICMP	*Internet Control Message Protocol*
IETF	*Internet Engineering Task Force*
IP	*Internet Protocol*
IPFIX	*Internet Protocol Flow Informatio Export*
ISO	*Internation al Organization for Standardization*
MLP	*Multi Layer Perceptron*
NC	Número de *Clusters*
NIDS	*Network Intrusion Detection Systems*
NMSE	*Normalized Mean Square Error*
PR-SCTP	*Partial Reliability Stream Control Transmission Protocool*
PSO	*Particle Swarm Optimization*
RFC	*Request for Comments*
RNA	Redes Neurais Artificiais
RST	*Reset*
SCTP	*Stream Control Transmission Protoco*
sFlow	*sampling Flow*
SNMP	*Simple Management Protocol*
tanh	Tangente Hiperbólica
TCP	*Transmission Control Protocol*
TOS	*Type of Service*
UDP	*User Datagram Protocol*
UEL	Universidade Estadual de Londrina
UTFPR	Universidade Tecnológica Federal do Paraná
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

LISTA DE SÍMBOLOS

d	Dimensão
D	Distância Euclidiana
f_{obj}	Função objetivo
$tanh$	Tangente hiperbólica
I_r	Intensidade da luz a uma distância r
I_s	Intensidade da luz da fonte
K	Número de *clusters*
L	Ponto de corte adotado pelo EWMA
L_{ic_k}	Representatividade luminosa de um centróide
L_{rc_k}	Luminosidade residente do centróide do *cluster* k
mp_i	valor da média ponderada para o instante i;
N	Quantidade total de vagalumes por iteração
nc_k	Quantidade de vagalumes no *cluster* k
Q	Quantidade de dados que serão clusterizados
r_{ij}	Distância entre quaisquer dois vagalumes i e j
S_{en}	Saída esperada para o instante n
w_i	Peso sináptico aplicado à entrada neural i
x	Vetor representando a quantidade de dados de cada atributo
x_i	Dados da entrada neural i
β	Atratividade de um vagalume
β_0	Capacidade de atração em r=0
γ	Coeficiente absorção de luz
σ^2	variância dos dados utilizada pelo EWMA

SUMÁRIO

1 INTRODUÇÃO

A gestão de uma rede de computadores é um trabalho complexo e requer o apoio de uma série de ferramentas e técnicas que auxiliem na execução dessa tarefa de forma eficiente. A disponibilidade é um dos pilares da gestão de redes e a maioria dos serviços disponibilizados pelas redes de computadores precisam que suas operações sejam garantidas, pois quaisquer interrupções ou degradações no desempenho podem refletir na qualidade dos serviços prestados e em prejuízos financeiros. Assim, os administradores de redes devem estar atentos ao uso inteligente dos recursos da rede, identificando tráfegos anômalos que podem ou não estar relacionados com ações maliciosas.

Um tráfego anômalo pode ser entendido, neste contexto, como uma não conformidade a um padrão de dados normais do tráfego de rede. Esse tráfego anômalo pode ser, por exemplo, decorrente de um ataque de negação de serviço *Denial of Service* (DoS) [1], capaz de deixar uma rede ou serviço indisponível. De uma forma geral, esse tipo de ataque é realizado por um invasor que envia uma grande quantidade de pacotes para um endereço IP e porta específicos, simulando um acesso legítimo. Esse ataque, tem como objetivo interromper um servidor ou serviço oferecido pela rede. A negação de serviço distribuída *Distributed Denial of Service* (DDoS) [2] utiliza vários sistemas comprometidos para lançar vários ataques DoS coordenados contra uma ou mais vítimas. Na verdade, um ataque DDoS acrescenta a dimensão muitos-para-um para o problema de negação de serviço. Outro evento anômalo não necessariamente gerado de forma hostil é o *Flash Crowd* [3], cuja designação está relacionada aos eventos que geram um aumento acentuado no volume do tráfego de rede, causado por algum fator externo, por exemplo, o acesso ao portal de uma universidade em uma data de matrícula. Essas anomalias provocadas por um DoS, DDoS e *Flash Crowd* são as que mais interessam neste trabalho, pois conseguem gerar um grande volume de tráfego de rede. Essa alteração no volume de tráfego tende a comprometer o funcionamento da rede, causando, por exemplo, lentidão e interrupções dos serviços.

Para mitigar os efeitos nocivos dos tráfegos anômalos às redes de computadores, os administradores, durante muito tempo, buscaram soluções capazes de fornecer informações desse tráfego que os permitissem tomar decisões referentes à integridade, disponibilidade e confiabilidade dos dados trafegados de uma forma mais eficaz. Com o surgimento dos protocolos de gerência que disponibilizam amostras de fluxos IP [4], os administradores de redes puderam obter muito mais informações sobre seus ambientes de rede. Os fluxos IP foram definidos por Cisco Systems [4] como sendo um conjunto de pacotes passando por um ponto de observação na rede, durante um certo intervalo,

compartilhando um conjunto comum de propriedades. Essas características passaram a fornecer outras informações sobre o tráfego, tais como: contadores para bytes, pacotes e fluxos; identificar endereços IP de origem e destino; identificar portas de origem e destino; protocolos como TCP, UDP, ICMP; verificar as *flags* utilizadas na comunicação; período e a duração dos fluxos. Ainda segundo Cisco, um fluxo é identificado pela combinação de seis campos chave, que identificam um fluxo único de dados, que são: endereço IP de origem e de destino, porta TCP/UDP de origem e de destino, tipo de protocolo (TCP ou UDP) e o tipo de serviço (TOS). Essas informações mais refinadas do tráfego de rede possibilitaram o desenvolvimento de várias ferramentas direcionadas ao gerenciamento mais efetivo, possibilitando entre outros serviços, o de detectar intrusões na rede.

Existem algumas ferramentas utilizadas pelos gestores de rede para identificar ataques em seus ambientes. De acordo com Garcia-Teodoro et al. [5] existem sistemas à base de assinatura, cujo processo de detecção é geralmente rápido e viável por causa do procedimento habitual de correspondência de padrões considerado na fase de detecção. No entanto, o banco de dados de assinatura tem de ser atualizado a cada momento e um sistema baseado nesse princípio é incapaz de detectar ataques anteriormente não observados ou até mesmo desconhecidos. Para superar essa limitação, existem sistemas de detecção de anomalias baseados na caracterização do tráfego, que são capazes de aprender com o comportamento normal de um ambiente, e com base na sua história, detectar qualquer mudança na rotina da rede.

O objetivo deste trabalho é apresentar um modelo de detecção de anomalias no tráfego de rede, denominado de *Firefly Algorithm for Digital Signature* (FADS). O FADS cria um perfil de comportamento da rede também chamado de *Digital Signature of Network Segment using Flow Analysis* (DSNSF). A criação desse perfil é baseada nos conceitos apresentados por Proença at al. [6], o qual desenvolveu uma ferramenta para gerar perfis de comportamento normal do segmento de rede, denominada gerenciamento de *backbone* automático (GBA). Nesse trabalho, para se conseguir um perfil de tráfego dentro do conceito proposto por Proença, foram utilizados o algoritmo *Firefly* (FA) e o K-*means*. O K-*means* foi usado na decomposição das amostras em grupos menores, baseados em suas similaridades. Já o FA, foi escolhido baseado no fato de que ele não é dependente do gradiente da função a ser otimizada, mas simplesmente da avaliação da função objetivo. Essa característica proporciona ao FA alta convergência com um tempo de processamento baixo. Após a geração do DSNSF, a identificação das anomalias será realizada pelo FADS utilizando o conceito de redes neurais artificiais (RNA), que é o de adquirir conhecimento através da experiência.

Para a criação, testes e validação do modelo proposto, foram utilizados fluxos coletados da rede de computadores da Universidade Estadual de Londrina (UEL) e da

Universidade Tecnológica Federal do Paraná (UTFPR) - Campus de Toledo. Foram computados os DSNSFs do tráfego normal da rede para seis atributos: bits/s, pacotes/s, atributos quantitativos, e IP de origem, IP de destino, porta de origem e porta de destino, atributos qualitativos. Para mensurar os atributos qualitativos, foi utilizada a entropia de Shannon [7], a qual possibilitou medir o grau de concentração e dispersão do conjunto de amostras. Após a geração dos DSNSFs, os mesmos foram analisados e comparados aos movimentos reais do tráfego das redes, com o objetivo de avaliar se os perfis foram capazes de descreverem os comportamentos para cada um dos atributos analisados. Na etapa seguinte, o FADS realizou a análise e identificação do tráfego anômalo. Para isso, o modelo confrontou o DSNSF com o tráfego real na identificação de desvios de comportamento. Nessa análise, o tráfego real foi submetido à injeção de tráfego anômalo com a finalidade de comprovar sua eficiência na detecção de anomalias.

Este trabalho tem como objetivo o desenvolvimento de um sistema de detecção de anomalias utilizando caracterização do tráfego normal de rede baseado na análise de fluxo, apresentando as seguintes contribuições:

- Adaptação do Algoritmo Metaheurístico *Firefly*: Esse modelo propicia a criação do *Digital Signature of Network Segment using Flow Analysis* (DSNSF), capaz de caracterizar o perfil de comportamento normal da rede;
- Utilização dos conceitos de Redes Neurais Artificiais: Proporciona ao modelo FADS eficácia na detecção de anomalias presentes no tráfego de rede analisado;
- Emprego de diferentes atributos de fluxos IP: Seis atributos de fluxos IP são analisados fornecendo informações referentes à distribuição e volume do tráfego;
- Detecção e notificação de anomalias: O modelo FADS é capaz de reconhecer tráfegos anômalos em redes de computadores, além de notificar o administrador mediante alarmes;

O trabalho está organizado da seguinte forma: no capítulo 2 são apresentados os trabalhos relacionados ao tema proposto, abordando caracterização do tráfego e detecção de anomalias em redes de computadores. O capítulo 3 aborda alguns padrões e recursos da gerência de redes com a utilização de Fluxos IP. O capítulo 4 apresenta os conceitos referentes ao algoritmo *Firefly* e às Redes Neurais Artificias. No capítulo 5 é detalhado o sistema de detecção de anomalias FADS. O capítulo 6 apresenta os testes realizados nos cenários propostos, bem como os resultados das validações que comprovam a eficácia do modelo FADS. Por fim, o capítulo 7 apresenta as conclusões e projetos futuros.

2 TRABALHOS RELACIONADOS

Neste capítulo, apresenta-se uma revisão bibliográfica detalhada do estado da arte do assunto discutido neste trabalho. Esses artigos também evidenciam o uso de fluxos IP para caracterização do tráfego de rede e modelos de detecção de anomalias.

2.1 Detecção de anomalias em redes de computadores

O avanço tecnológico e a expansão da Internet propiciaram a difusão de ataques maliciosos às redes de computadores, tornando imprescindível o uso de técnicas e sistemas capazes de garantir a sua segurança [8]. Dessa forma, a busca por soluções eficazes e factíveis para a identificação e caracterização de anomalias em redes de computadores, com a finalidade de detectar problemas relacionados ao desempenho e intrusões na rede, é o objeto de estudo de muitos pesquisadores hoje.

Para Prabha e Sree [9], a segurança em sistemas de computadores tornou-se uma grande preocupação ao longo dos últimos anos, fazendo com que os sistemas de detecção de intrusão ou *Network Intrusion Detection Systems* (NIDS) tornassem-se recursos essenciais no gerenciamento de segurança em redes de computadores, cabendo ao NIDS alertar o administrador de rede quando detectar atividades anômalas [10].

De acordo com Patcha e Park [8], sistemas de detecção de intrusão baseados em anomalias, utilizando um modelo de comportamento normal da rede, são eficazes em encontrar e frustrar ataques conhecidos ou não em comparação com os sistemas de detecção de intrusão baseados em assinaturas.

Os modelos NIDS baseados em assinaturas utilizam padrões pré-definidos de ataques e de outras atividades maliciosas para identificar eventos que correspondam a esses padrões. Isso permite apenas o reconhecimento de ataques contidos em sua base de dados, que para uma maior eficiência do sistema deve ser constantemente atualizada [11].

Na pesquisa de Lima et al. [12], os autores desenvolveram um sistema baseado na combinação das técnicas *Particle Swarm Optimization* (PSO) e K-means, denominado PSO-Cls, capaz de analisar os dados de tráfego coletados da *Management Information Base* (MIB) através do protocolo *Simple Management Protocol* (SNMP) no intuito de caracterizar o comportamento normal de tráfego de rede.

Hall et al. [13] dizem que os NIDS baseados em caracterização usam técnicas capazes de predizerem um comportamento do tráfego da rede tido como normal e depois usá-lo como referência, onde qualquer resultado da análise do tráfego diferente desse pode caracterizar uma anomalia. A fidelidade dessa caracterização está diretamente relacionada

com a quantidade de informação obtida do tráfego de rede.

A utilização dos fluxos IP trouxe grandes benefícios no que tange a obtenção de informações do tráfego de rede, servindo como ferramenta fundamental no processo de caracterização [14]. O fluxo IP é definido pela Cisco [4] como uma sequência unidirecional de pacotes entre máquinas de origem e destino, com propriedades comuns. Esses pacotes compartilham os valores de endereços IP de origem e de destino, o protocolo IP, as portas de origem e de destino e o tipo de serviço IP. Com esses atributos, os fluxos IP são capazes de fornecerem aos administradores não somente o volume total, mas a composição do tráfego de rede, bem como os dispositivos e serviços envolvidos na comunicação.

O emprego dos fluxos IP na caracterização do tráfego foi utilizado no trabalho de Velan et al. [15] para diferenciar os vários tipos de padrões existentes, possibilitando não só observar as alterações nas propriedades da rede, mas também quantificá-las. Isso permitiu diferenciar e descrever as redes analisadas em sua pesquisa.

No trabalho de Pena et al. [16], um *Digital Signature of Network Segment using Flow Analysis* (DSNSF) é criado com base no modelo *Autoregressive Integrated Moving Average* (ARIMA), apresentando uma abordagem na detecção de anomalias utilizando análise de fluxos IP. Além disso, um algoritmo baseado na Lógica Paraconsistente é proposto para evitar altas taxas de alarmes falsos.

A análise de fluxos IP também foi utilizada no trabalho de Kallitsis et al. [17]. Os autores desenvolveram uma arquitetura de código aberto, *All−Packet Monitor* (AMON), para monitorar e analisar os fluxos de redes *multigigabit online*. Essa arquitetura aproveita o alto desempenho do monitoramento de pacotes PF_*RING* [18] e é facilmente aplicável em *commodity hardware*. AMON examina todos os pacotes, a partição do tráfego em sub−fluxos usando *hashing* rápida e calcula certos produtos de dados em tempo real. As estruturas de dados resultantes oferecem vistas sobre a intensidade e a conectividade do tráfego de rede na escala de tempo de encaminhamento.

No trabalho de Zacaron et al. [19], a análise de fluxos IP, utilizando o protocolo de exportação *Netflow*, juntamente com o modelo de clusterização k-*means*, possibilitaram aos autores criarem uma assinatura digital de segmento de rede (DSNSF-KM) capaz de caracterizar o tráfego de um segmento de rede. Esse DSNSF-FM tem como objetivo auxiliar os administradores na detecção de anomalias na rede.

Nessa mesma linha seguiu o trabalho de Garcia et al. [20]. Os autores, utilizando-se dos fluxos IP, fizeram a caracterização de um tráfego de um provedor de serviço Internet, fornecendo medições precisas e quantitativas do uso de aplicativos, da utilização da largura de banda e das preferências dos usuários.

A análise de fluxos IP com intuito de detectar ataques, faz parte de inúmeras pesquisas. Entre elas podemos mencionar o trabalho de Steeg et al. [21]. O autor apresenta

um protótipo de detecção de ataque DDoS em tempo real implantado em *switch*, com apenas 20% de comprometimento no rendimento do equipamento na análise do tráfego.

No trabalho de Qin et al. [22], foi proposta uma abordagem de detecção de ataque DDoS baseada em vetores de entropia de diferentes fluxos do tráfego e do uso de um algoritmo de agrupamento. Isso possibilitou a construção de um perfil normal de rede, viabilizando a detecção de desvios nesses modelos criados. Os resultados experimentais demonstram que o método proposto pode não só superar os métodos tradicionais em termos de precisão de detecção, como também pode ser aplicado em ambientes ou cenários práticos.

Uma nova abordagem a esse respeito foi apresentada por Du et al. [23]. O trabalho propõe a implementação de um sistema de computação distribuída de *streaming* que visa realizar a detecção de anomalias em tempo real, aproveitando o Apache, alcançando alta precisão no resultado geral.

Vários algoritmos têm sido usados no auxílio à otimização dos dados obtidos pelos fluxos IP. Dentre eles destacam-se os metaheurísticos. De acordo com Boussaíd et al. [24], o algoritmo meta-heurístico foi concebido para resolver uma ampla gama de problemas de difíceis otimizações sem ter que se adaptar profundamente a cada problema. Eles são amplamente utilizados para resolver problemas complexos na indústria e serviços, e em áreas que vão de finanças a gestão da produção e engenharia.

O algoritmo *Firefly* (FA) [25] é um algoritmo metaheurítico baseado no comportamento dos vagalumes e das suas características de bioluminescência. No estudo apresentado por Talatahari et al. [26], foi utilizado o FA para resolver, de forma eficiente, vários problemas de otimização em engenharia estrutural. O FA foi utilizado a fim de diminuir o custo de produção: características físicas de vigas, vaso de pressão cilíndrica, *design* helicoidal de mola de compressão e um *design* de viga de concreto armado, além de ajudar o desenvolvimento de uma proteção contra impactos laterais automotivos.

No trabalho de Adaniya et al.[27], o FA foi utilizado pelo algoritmo de agrupamento híbrido *Firefly Harmonic Clustering Algorithm* (FHCA) na geração de uma assinatura digital de segmento de rede (DSNS). As informações obtidas pelo DSNS possibilitaram detectar altas taxas de anomalias no segmento de rede analisado.

Em seus estudos, Hassanzadeh et al. [28] aplicaram FA na otimização do método de Otsu. O método de Otsu é usado para executar automaticamente a redução de uma imagem de nível de cinza para uma imagem binária. Os resultados da investigação mostraram a eficiência e a precisão do método na segmentação de imagens.

Na pesquisa de Shona e Senthilkumar [29], foi utilizado o FA no desenvolvimento de um IDS eficiente em *Mobile Ad hoc Networks* (MANETs) para eliminar um conjunto de

registros redundantes. Os resultados experimentais mostram que depois do processamento, há uma melhoria na precisão do algoritmo de aprendizagem durante a classificação de pacotes normais e anormais.

Outro algoritmo metaheurítico amplamente conhecido é o algorítmo genético (GA)[30]. Esse algoritmo simula a evolução natural das espécies baseando-se na seleção, cruzamento e mutação de cromossomos que representam possíveis soluções para um problema.

Na pesquisa de Hermandes et al. [31], o autor utiliza-se do GA na determinação dos agrupamentos que são utilizados na construção do perfil normal de rede e que serve de referência na análise do tráfego real e na detecção de anomalias.

Essa caracterização do tráfego de rede utilizando fluxos IP, denominada DSNSF, tem sido utilizada pelo Grupo de Pesquisa em Redes de Computadores e Comunicação de Dados do Departamento de Computação da Universidade Estadual de Londrina (UEL) [32] em muitas pesquisas. Dentre esses trabalhos estão o de Salmen et al. [33]. Os autores, utilizando-se do FA e GA, criaram perfis normais de rede baseado na análise de fluxos IP, denominados DSNSF-FA e DSNSF-GA, que em conjunto com o modelo *Adaptive Dynamic Time Warping* (ADTW) [34] puderam realizar testes de detecção de anomalias de rede em ambientes reais, provando a eficiência de ambos modelos.

Na pesquisa de Carvalho et al. [35], foi desenvolvido o modelo *Ant Colony Optimization for Digital Signature* (ACODS), baseado na otimização por colônias de formigas, para a criação de um DSNSF que, comparado ao tráfego real da rede e fazendo uso do ADTW, pôde identificar, com precisão, eventos anômalos de diversas intensidades.

No trabalho de Assis et al. [36] foi utilizada a análise do fluxo em sete dimensões. Para tanto, melhorias foram realizadas ao método de previsão *Holt-Winters*, culminando no modelo denominado *Holt-Winters for Digital Signature* (HWDS). Esse utilizou da análise heptadimensional do tráfego para caracterização do comportamento normal, detecção e geração de alarmes, bem como criação de relatórios destinados à auditoria do sistema e das anomalias detectadas.

Uma DSNSF derivada do algoritmo genético (GA) é proposta por Hamamoto et al. [37] e é comparada com uma modificação do *Ant Colony Optimization* (ACO) usadas para o mesmo objetivo. Ambos os métodos são modelos bio-inspirados e são amplamente aplicados a problemas de otimização. As assinaturas digitais resultantes foram comparadas ao tráfego real e o desempenho dos algoritmos foram analisados com o uso do Coeficiente de Correlação e do Erro Quadrado Médio Normalizado.

Na pesquisa realizada por Fernandes et al. [38] foram analisados e comparados três modelos utilizados para a detecção de anomalias utilizando DSNSFs: o modelo de

procedimento estatístico *Principal Component Analysis for Digital Signature* (PCADS), o metaheurístico *Ant Colony Optimization for Digital Signature* (ACODS) e o de previsão AutoRegressive *Integrated Moving Average for Digital Signature* (ARIMADS). Os resultados obtidos, apesar de identificar o ARIMADS como o modelo mais promissor no reconhecimento de anormalidades, demonstram que todos os modelos analisados possuem alta precisão na detecção de anomalias.

Outra abordagem na busca de modelos com melhores resultados na detecção de anomalias e intrusão é o uso das redes neurais artificias (RNA).

No trabalho de Lima et al. [39], foi mostrada a construção de um protótipo para um sistema de detecção de intrusão de rede, que utiliza uma rede neural artificial como um mecanismo de detecção. Nas fases de treinamento e aprendizagem da rede, que são processos adaptativos, foi aplicada a base de conhecimento do IDS *Snort*. Os IDSs construídos permitem a detecção de uma proporção aceitável de variantes de intrusão, para além das formas de intrusão já conhecidas. Essa última característica apresenta vantagens expressivas comparando-se a sistemas de detecção de intrusão puramente baseados em regras, pois descarta o uso de uma ampla base de conhecimentos e resolve os problemas de falsos negativos e falsos positivos, através do ajuste fino dos pesos, dado pela variação da aceitação na saída da rede, quando ela é treinada.

No trabalho de Mukkamala et al. [40], foram descritas duas abordagens para detecção de intrusão usando redes neurais e máquinas de vetores de suporte. A finalidade foi descobrir padrões úteis ou características que descrevem o comportamento do usuário em um sistema e usar o conjunto de recursos relevantes para construir classificadores que pudessem reconhecer anomalias e intrusões conhecidas, em tempo real. Usando um conjunto de dados de *benchmark* de uma competição *Knowledge Discovery and Data mining* (KDD) projetada pela DARPA foi demonstrado que classificadores eficientes e precisos podem ser construídos para detectar intrusões.

Na pesquisa de Moradi e Zulkernine [41], foi apresenta uma abordagem de rede neural para detecção de intrusão. A *Multi Layer Perceptron* (MLP) é usada para a detecção de intrusão com base em uma análise com abordagem *off-line*. Embora a maioria dos estudos anteriores tenham se concentrado na classificação de registros em uma das duas classes gerais, normal e ataque, essa pesquisa teve como objetivo resolver um problema de classe múltipla em que o tipo de ataque também é detectado pela rede neural. Diferentes estruturas de redes neurais foram analisadas para encontrar a rede neural ideal em relação ao número de camadas ocultas. Os resultados mostraram que o sistema projetado é capaz de classificar registros com cerca de 91% de precisão com duas camadas ocultas de neurônios na rede neural e 87% de precisão com uma camada oculta.

2.2 Trabalhos do Grupo de Pesquisa em Redes de Computadores e Comunicação de Dados DC - UEL

O Grupo de Pesquisa em Redes de Computadores e Comunicação de Dados do Departamento de Computação da UEL tem focado seus estudos no desenvolvimento de ferramentas capazes de auxiliarem os administradores na caracterização do tráfego e na detecção de anomalias no segmento de rede. Em todos as pesquisas, a criação de uma assinatura digital de segmento de rede (DSNSF), proposta inicialmente por Proença [6], mostrou-se eficiente como um método capaz de caracterizar o tráfego de rede, possibilitando o estudo e o uso de várias técnicas no intuito de obtê-la. Todos esses trabalhos foram abordados resumidamente na seção anterior. Assim, deixar-se-á registrado neste trabalho, utilizando a Tabela 1, todas as pesquisas realizadas por este grupo até o momento. Na tabela, são apresentados os anos das publicações, seus autores, a fonte de exportação de dados, os nomes dos modelos utilizados, a maneira como foi feita a detecção de anomalias, quantos atributos obtidos por meio dos fluxos de dados ou objetos obtidos por meio do SNMP foram analisados e o tempo usado para o treinamento do algoritmo para gerar o DSNSF.

Tabela 1 – Resumo dos trabalhos do grupo de pesquisa em redes da UEL

Ano	**Autores**	**Fonte de Dados**	**Modelo**	**Detecção de Anomalias**	**Atributos ou Objetos**	**Tempo de Treinamento**
2010	Lima et al. [12]	SNMP	PSO-CIs	Thresholds	4	4 semanas
2012	Adaniya et al. [27]	SNMP	FHCA	Thresholds	4	4 semanas
2012	Carvalho et al. [35]	FLUXOS	ACODS	ADTW	7	4 semanas
2012	Zacaron et al. [19]	FLUXOS	DSNSF-KM	Thresholds	3	7 semanas
2014	Assis et al. [36]	FLUXOS	HWDS	Thresholds	7	1 dia
2014	Pena et al.[16]	FLUXOS	ARIMA	Lógica Paraconcistente	6	3 semanas
2014	Hernandes et al. [31]	FLUXOS	GADS	ADTW Thresholds Lógica Paraconcistente	6	3 semanas
2015	Fernandes et al [38]	FLUXOS	PCADS	Thresholds	7	5 semanas
2015	Salmen et al. [33]	FLUXOS	FADS	ADTW Thresholds	6	3 semanas
2015	Hamamoto et al. [37]	FLUXOS	Fuzzy-GADS	Lógica Fuzzy	6	4 semanas

No próximo capítulo, são apresentados alguns conceitos relacionados aos padrões e recursos da gerência de redes de computadores com a utilização de fluxos IP.

3 GERÊNCIA DE REDES

O crescimento significativo de usuários, aplicações e serviços fizeram com que as redes se tornassem mais complexas, onde falha ou degradação no desempenho da rede impactam negativamente nas relações de negócios e finanças de empresas e de usuários comuns. Considerando esse quadro, o gerenciamento dos ambientes de redes de computadores se tornou necessário para a manutenção do correto funcionamento dessas redes [42].

A gestão de redes está relacionada com a execução de atividades numa rede e às tecnologias necessárias para apoiá-las. De acordo com Kurose et al. [43], essa gerência inclui a disponibilidade, a integridade e a coordenação de elementos de hardware e software capazes de monitorar, analisar, avaliar e controlar os elementos e recursos da rede. Esses recursos propiciam satisfazer as necessidades de desempenho e de qualidade de serviço oferecidos pela rede. De acordo com Leinwand et al. [44], uma rede de computadores pode ser gerenciada de três formas:

- Gerência centralizada: existe apenas um gerente responsável pela gestão da rede. Nesse modelo, todos as informações e aplicações de gerenciamento estão centralizadas.
- Gerência hierárquica: nessa arquitetura, há diversos gerentes, sendo um atuando como central e os outros como clientes. Esse gerenciamento possibilita uma divisão e distribuição de tarefas, agilizando os processos.
- Gerência distribuída: é uma combinação das anteriores. Sendo independente de um sistema único de gestão, possuindo monitoração e tarefas distribuídas. Pode utilizar vários gerentes sem centralização de dados nem hierarquia entre eles, porém, os bancos de dados de gerência são replicados entre si para neutralizar a redundância.

Segundo Pinheiro [45], o gerenciamento de rede de computadores pode ser resumido em três etapas. A primeira, é a coleta de dados do processo de monitoração sobre os recursos gerenciados. A etapa seguinte está relacionada ao diagnóstico, que após a análise dos dados coletados tem o intuito de indicar a causa do problema apresentado no recurso gerenciado. Finalmente, um controle ou ação sobre o recurso gerenciado é efetuada na última etapa.

Para além dessa caracterização resumida do gerenciamento de rede, a ISO *(Internation al Organization for Standardization)* apresentou uma separação funcional no processo de gerenciamento, conhecido como FCAPS *(Fault, Configuration, Account,*

Performance, Security), que foi adotada por uma grande maioria dos fornecedores de sistemas de gerenciamento de redes. Essa separação funcional tem como objetivo garantir o bom funcionamento de uma rede. Segundo Clemm [46], esse modelo contém as seguintes características:

- Gerenciamento de falhas: descreve as ações rápidas necessárias para encontrar e isolar o componente responsável pela falha, reconfigurando e isolando a rede do problema até a sua reparação;
- Gerenciamento de configuração: engloba tarefas de registros de hardware e software, incluindo a manutenção e monitoração das mudanças realizadas às estruturas lógica e física da rede.
- Gerenciamento de contabilidade: responsável por verificar e contabilizar os limites do acesso de usuários e dispositivos aos recursos da rede. Além disso, fazem parte dessa gestão a cobrança pela utilização e alocação desses recursos.
- Gerenciamento de performance: consiste em monitorar, analisar, controlar e mensurar as atividades da rede com o intuito de obter informações sobre seu desempenho. Esses dados possibilitam garantir a operação da rede em conformidade com a qualidade de serviço acordados com seus usuários.
- Gerenciamento de segurança: tem a finalidade de garantir a proteção dos recursos da rede e das informações dos usuários de acordo com uma política de segurança definida.

Segundo Thottan el al. [47], além das áreas mencionadas anteriormente, a gerência de rede pode ser agrupada em duas categorias: monitoramento e controle de rede. À monitoração cabe a tarefa de avaliar e observar as condições e as configurações dos componentes da rede, enquanto que a finalidade do controle é a de ajustar parâmetros e executar ações em prol do desempenho da rede.

Com base nas cinco áreas de gerenciamento proposta pela ISO, o gerente consegue obter informações relevantes sobre os eventos da rede, permitindo um melhor monitoramento do tráfego. No entanto, com o crescimento das redes e volume de dados, esse gerenciamento manual se torna inviável, e soluções e processos automatizados de gerencia se fazem necessários [48].

O monitoramento da rede pode ser realizado de forma ativa e passiva [47]. Na abordagem ativa, pacotes de teste são inseridos na rede. Nesse procedimento, há uma preocupação com o volume de pacotes de teste inseridos, pois podem causar prejuízos à rede. No monitoramento passivo, o conteúdo do tráfego de rede é analisado sem

interferência no fluxo de dados. Tal abordagem analisa uma grande quantidade de informações sem comprometer o desempenho da rede.

Um sistema eficaz de detecção de anomalias possui uma fonte de dados capaz de fornecer informações relevantes e detalhadas para o monitoramento das atividades da rede auxiliando no reconhecimento de comportamentos anormais no tráfego de rede. Dessa forma, foi desenvolvida a gerência baseada em análise de fluxos IP. Esses fluxos permitem a construção de uma base de dados com informações mais específicas e indispensáveis do tráfego, possibilitando a avaliação mais abrangente e precisa do funcionamento de uma rede [49].

Desse modo, com o objetivo de prover ao gerente informações mais detalhadas do tráfego da rede, a Cisco *Systems* apresentou o protocolo NetFlow [50] baseado totalmente em fluxos IP. Servindo para o mesmo propósito, surgiram outros protocolos tais como o IPFIX (*Internet Protocol Flow Informatio Export*) [51], da IETF (*Internet Engineering Taask Force*) e o sFlow (*sampling Flow*) [52], da InMon.

3.1 Fluxos IP

Os fluxos IP foram definidos por Claise [53] como sendo um conjunto de pacotes passando por um ponto de observação na rede durante certo intervalo compartilhando um conjunto comum de propriedades. Basicamente, um fluxo IP é uma sequência unidirecional de diversos pacotes IP compartilhando os seguintes atributos:

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

Os endereços IP de origem e destino possibilitam identificar o ponto inicial e final do tráfego gerado. As aplicações responsáveis pela geração e recebimento dos pacotes são reveladas pelas portas de origem e de destino, respectivamente. Já o tipo de serviço caracteriza a prioridade do tráfego. A interface do roteador/switch permite quantificar a utilização dos dispositivos pelo tráfego de rede.

Com essas características, os fluxos podem ser relacionados à uma aplicação, usuário ou equipamento de rede. Aproveitando-se disso, as técnicas baseadas na detecção de anomalias em rede de computadores vêm empregando com muita mais frequência os protocolos de gerenciamento de rede e estatísticas baseadas em análise de fluxos IP.

A gerência de rede baseada na análise de fluxos trouxe melhoramentos significativos. Segundo Claise e Wolter [54] e Todinov [55], as análises de fluxos IP permitiram apurar a utilização de cada aplicação na rede, possibilitando uma melhor alocação dos recursos aos usuários finais; o volume do tráfego pôde ser analisado de uma forma individual a cada dispositivo; uma base histórica com informações mais detalhadas do tráfego da rede passou a ser possível.

Nas seções seguintes são abordados s principais protocolos de exportação de fluxos IP usados no gerenciamento de redes.

3.2 NetFlow

O protocolo NetFlow foi desenvolvido pela empresa CISCO em 1966 por Darren e Bruins, como uma tecnologia de monitoramento de tráfego de rede [56]. Esse protocolo gera informações e estatísticas sobre o tráfego, possibilitando uma completa análise da rede. Dentre essas análises, podemos destacar a viabilidade de mensurar os recursos da rede, detectar anomalias no tráfego e, medir o uso de aplicativos e da rede.

De acordo com a Figura 1, é possível verificar como são gerados os fluxos IP por um roteador/*switch* configurado com o protocolo de exportação NetFlow, à medida que ele recebe pacotes do tráfego da rede. No recebimento de um pacote, seus atributos são analisados pelo equipamento e as informações são reunidas em um fluxo com essas características já existentes. Caso contrário, se algum pacote possuir valores diferentes de atributos com relação aos fluxos ativos, um novo fluxo é criado. Na sequência, o pacote segue até o seu destino.

Para tornar esse gerenciamento escalável, algumas regras foram estabelecidas:

- fluxos inativos são descartados após um determinado tempo. O tempo padrão para essa remoção é de 15 segundos podendo ser ajustado de 10 até 600 segundos;

- fluxos ativos são terminados e removidos da *cache* quando atingirem 30 minutos. Esses dados são contabilizados se a conexão que gerou o fluxo ainda estiver ligada;

- o emprego de políticas de exclusão e substituição de fluxos após o preenchimento da *cache* ou quando uma conexão TCP com *flags* FIN ou RST expirar.

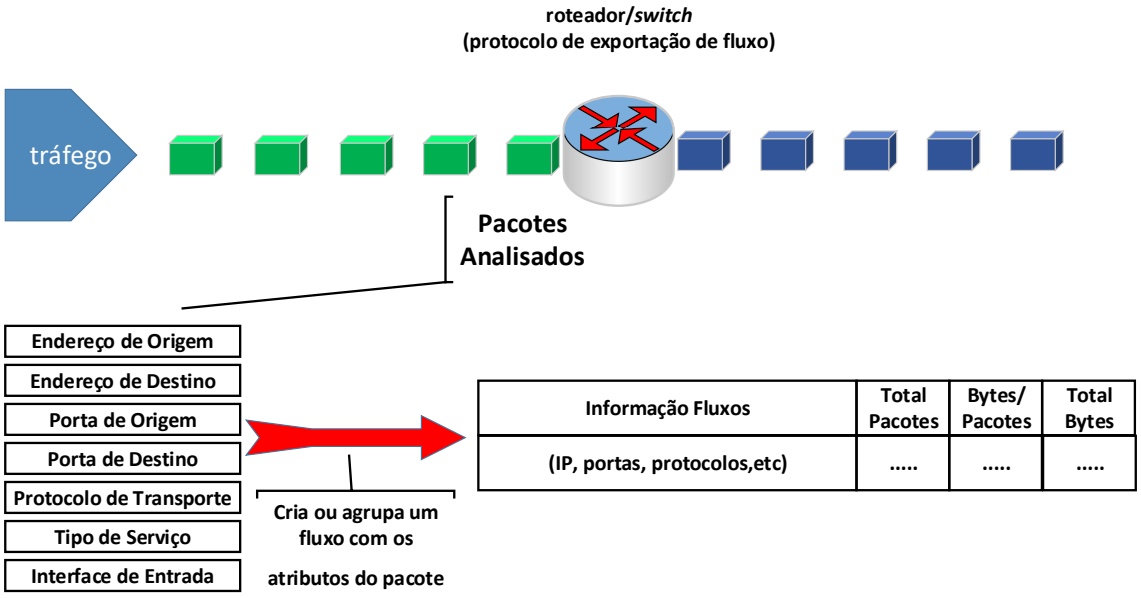


Figura 1 – Geração de um fluxo IP utilizando o protocolo NetFlow

O tamanho da cache do NetFlow pode ser definido por meio de software, variando de 1024 a 524 mil entradas de registros de fluxos. O consumo mínimo de memória de cada fluxo é de aproximadamente 64 bytes sendo o número máximo de registros suportado pela cache diretamente relacionado à quantidade de memória física contida no sensor de gerenciamento.

Após a criação dos fluxos, o Netflow acrescenta as seguintes informações:

- *timestamp*: caracteriza o tempo de atividade de um fluxo, sendo útil no cálculo do total de bytes e pacotes transmitidos em um determinado tempo;
- *flags* do protocolo TCP: utilizado na sinalização do processo *handshake*;
- máscara de sub-rede: usadas para o cálculo de prefixos.

Os fluxos expirados são agrupados juntos no datagrama NetFlow *Export* [57] para serem enviados/exportados. Um NetFlow *Export* pode consistir de até 30 registros de fluxo. Para configurar o NetFlow, basta o usuário especificar o endereço IP e o número da porta do aplicativo coletor.

3.3 IPFIX

O IPFIX foi padronizado pelo IETF como uma alternativa aos protocolos proprietários de exportação de fluxos utilizados no monitoramento de redes de computadores. Seu desenvolvimento foi baseado no Netflow versão 9 [58] sendo descrito em várias RFCs. A Tabela 2 detalha resumidamente as RFCs que descrevem o protocolo IPFIX, identificando o número da RFC, sua descrição e o ano da publicação.

Tabela 2 – RFCs do IPFIX

Número	**Descrição**	**Data**
RFC 3917	Requisitos para IPFIX	outubro, 2004
RFC 3955	Avaliação dos protocolos candidatos para IPFIX	outubro, 2004
RFC 5101	Especificações do IPFIX, protocolo para troca de informações de tráfego de fluxo	janeiro, 2008
RFC 5102	Modelo de informações para IPFIX	janeiro, 2008
RFC 5103	Exportação de fluxo bidirecional usando IPFIX	janeiro, 2008
RFC 5153	Orientações para implementação	abril, 2008
RFC 5470	Arquitetura para monitoramento seletivo de fluxos IP e exportação de um dispositivo IPFIX para um coletor	março, 2009
RFC 5471	Orientações e diretrizes para IPFIX (Lista de testes)	março, 2009
RFC 5472	Aplicabilidade IPFIX para uma variedade de aplicações	março, 2009
RFC 5473	Método para economizar largura de banda, as considerações aplicam-se para PSAMP	março, 2009
RFC 5610	Descreve uma extensão do IPFIX para exportação de tipos de informações estendidas	julho, 2009
RFC 5655	Descreve o formato do arquivo para armazenar os fluxos	outubro, 2009
RFC 5815	Definição de objetos gerenciados para IPFIX	abril, 2010
RFC 5982	Descreve alguns problemas de medições baseadas em fluxo	agosto, 2010
RFC 6183	Framework estende o modelo de referência especificado na RFC 5470	abril, 2011
RFC 6235	Técnicas para anonimizar endereços IP, esquemas comuns e definições de parâmetros	maio, 2011
RFC 6313	Extensão do IPFIX, permite estrutura de dados complexas como listas de tamanho variável e especificações referente às hierarquias na relação entre template	julho, 2011
RFC 6526	Utilização do protocolo SCTP, uma das vantagens é reduzir a probabilidade na perda de dados	março, 2012
RFC 6615	Definição de objetos gerenciados para IPFIX	junho, 2012
RFC 6727	Definições de objetos gerenciados para pacotes amostrados (PSAMP) módulo IPFIX-SELECTOR-MIB-MODULE	outubro, 2012
RFC 6728	Especifica o modelo de dados para IPFIX e PSAMP, na configuração e monitoramento	outubro, 2012

Entre suas características, estão a de um modelo de dados mais flexível, atuando independentemente do protocolo de transporte ou dos formatos das mensagens usadas [59] e de um controle de congestionamento e segurança. Essas peculiaridades foram definidas e elaboradas para não comprometerem a performance ou funcionalidade da rede.

O IPFIX é constituído basicamente por três tipos de processos [60]:

- processo de medição: nesta etapa, são gerados os fluxos a partir dos cabeçalhos e características dos pacotes coletados;
- processo de exportação: nesse momento, o IPFIX exporta o fluxo ao coletor;
- processo coletor: nesse passo, os fluxos coletados são analisados e armazenados em um banco de dados.

A arquitetura do IPFIX permite que os fluxos sejam exportados de forma não segura, utilizando o protocolo UDP (*User Datagram Protocol*); parcialmente segura, fazendo-se uso do PR-SCTP (*Partial Reliability Stream Control Transmission Protocool*) ou totalmente segura, a partir do protocolo SCTP (*Stream Control Transmission Protocool*) e TCP (*Transmission Control Protocol*).

No IPFIX, a unidade utilizada para o transporte de dados é denominada de *Mensagem* que é constituída de um cabeçalho de 32 bits e um vasto conjunto de registros (*Sets*). Na Figura 2, pode-se visualizar o formato básico de uma mensagem IPFIX.

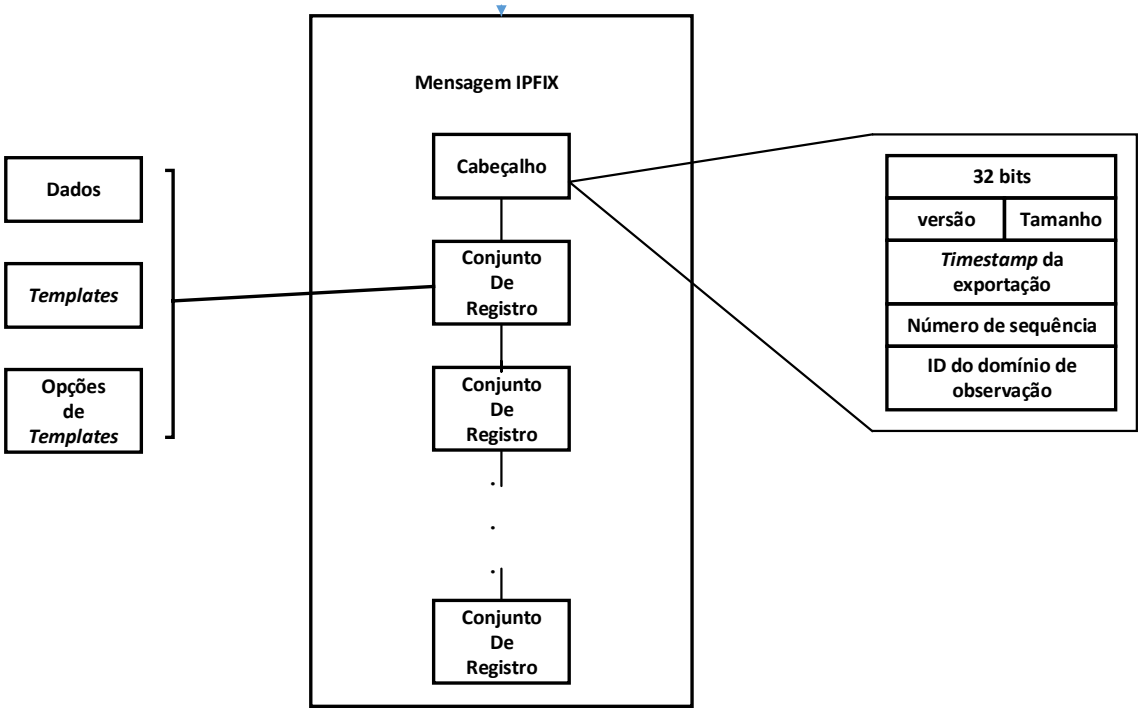


Figura 2 – Formato básico de uma mensagem IPFIX

O cabeçalho da mensagem IPFIX, traz as seguintes informações:

- versão: traz o número da versão do registro de fluxo utilizado;
- tamanho: registra o tamanho total da mensagem;
- *timestamp* da exportação: registra o instante em que a mensagem deixou o processo de exportação;
- número de sequência: atribui um número de sequência individual para cada mensagem. Utilizado para a detecção de mensagem perdida;
- ID do domínio de observação: indica onde o fluxo foi coletado.

Os conjuntos de registros, de acordo com Trammell e Boschi [60] e Claise [53], podem ser de três tipos: conjunto de Modelos (*Template Sets*), formado de *Template Records* cuja propriedade é a de definir a estrutura e a forma de como cada campo dos conjuntos de dados (*Data Records*) são analisados; conjunto de Opções de Modelo (*Options Template sets*), são constituídos por conjuntos de *Options Template Records* que podem ser considerados como uma extensão do *Templates*. Esse *set* define o modo de se interpretar cada campo de um registro de dados e sua estrutura, bem como, a aplicabilidade dos registros de dados. Por fim, temos o conjunto de dados (*Data Sets*). Esse conjunto possui os valores dos parâmetros definidos pelo *template* correlato.

3.4 sFlow

O nome sFlow é um acrônimo para *sampling flow*, ou seja, amostragem de fluxos. Esse protocolo de montagem e exportação de fluxos foi desenvolvido pela InMon e padronizado pelo IETF por meio da RFC 3176 [61]. O sFlow, segundo [61], é uma tecnologia para monitoramento de tráfego de redes de computadores inserida em *switchs* e roteadores e que utiliza de mecanismos de amostragem para efetuar as estatísticas do fluxo coletado da rede. Essa característica lhe permite ser aplicado em redes de alta velocidade.

Dentre as principais características do sFlow, pode-se enfatizar: sua escalabilidade tanto em relação à taxa de transmissão, sendo capaz de efetuar um monitoramento em redes de 10Gbps a 100Gbs, quanto relacionado ao tamanho da rede, onde vários dispositivos podem ser monitorados por apenas um coletor; sua acurácia, que por possuir uma amostragem simples e diretamente realizada em *hardware* não interfere no desempenho da rede; possui implementação simples e de custo baixo na execução do agente; e o tempo de amostragem, onde o protocolo possibilita a exportação

de fluxos a cada minuto, proporcionando uma análise mais rápida das informações coletadas da rede.

Um sistema de monitoramento utilizando o sFlow é constituído de dois módulos. O primeiro é o agente, que pode estar implementado em um *switch*, roteador ou sonda. Esse agente, utiliza técnicas de amostragem e de processamento das informações para gerar estatísticas do tráfego da rede e enviá-las ao(s) coletor(es). O segundo módulo é constituído de um analisador ou coletor de dados sFlow. Esse coletor é uma aplicação que analisa os dados recebidos e produz métricas úteis para o gerenciamento da rede. O funcionamento do protocolo sFlow e seus componentes podem ser vistos na Figura 3.

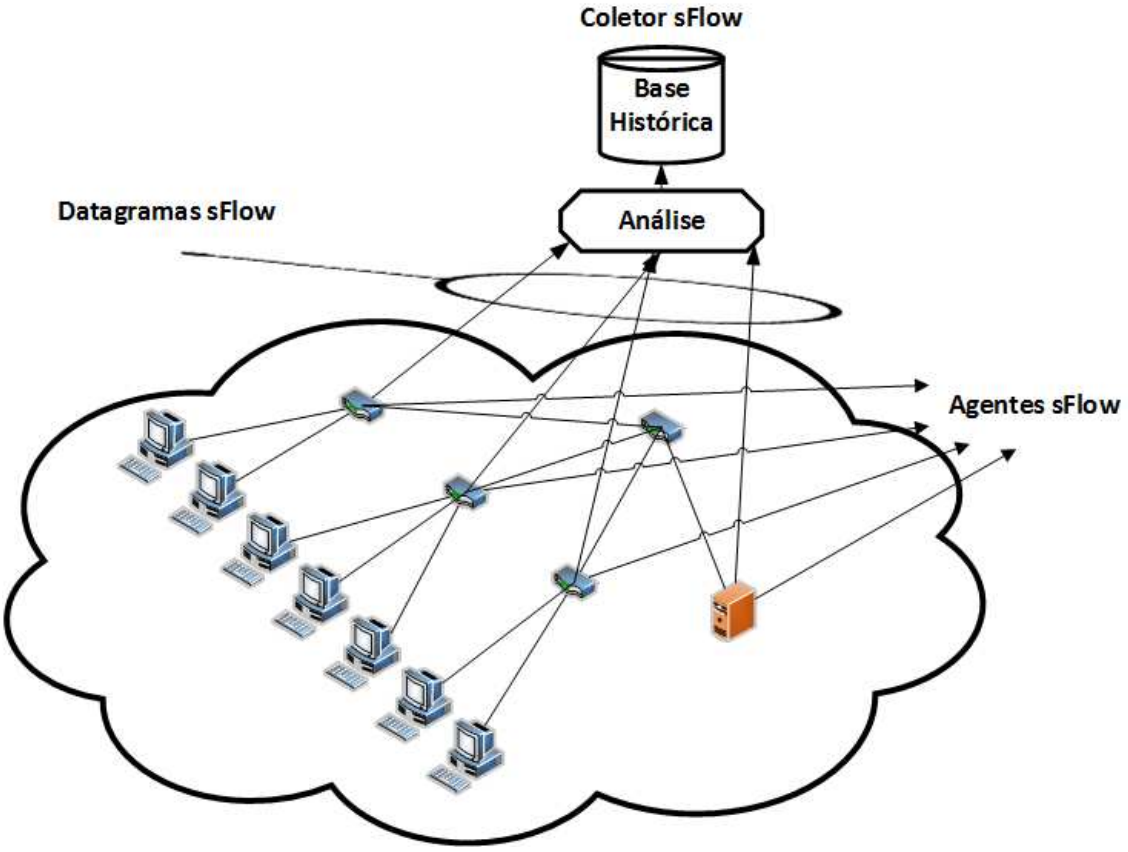


Figura 3 – Funcionamento do protocolo sFlow e seus componentes

Para uma diminuição na quantidade de dados analisados, o sFlow disponibiliza duas técnicas de amostragem [62] para o tráfego de rede, descritas a seguir:

- *Statistical packet-based sampling of switched flows*: Nessa técnica, o agente sFlow aproveita de um contador que é decrementado a cada pacote recebido pelo equipamento. No término dessa contagem, ou seja, quando o contador chegar no zero, uma amostra é realizada e adicionada à montagem de fluxos. Essa amostra então, representa a quantidade de pacotes pré-definidos no contador.

- *Time-based sampling of network interface statistics*: Essa abordagem utiliza contadores baseados no tempo para a realização periódica de sondagem (*polling*) nos agentes sflow. Um intervalo de sondagem máximo é estipulado, porém, para garantir uma eficiência do protocolo, cada agente pode ajustar livremente este valor.

Tabela 3 – Comparação entre NetFlow, IPFIX e sFlow

	NetFlow	**IPFIX**	**sFlow**
desenvolvedor	CISCO	baseado no NetFlow v9	InMon
ano de implementação	1996	2004	2001
RFC	3954	3917, 3955, 5101, 5102, 5103, 5153, 5470, 5471, 5472, 5473, 5610, 5655, 5815, 5982, 6183, 6235, 6313, 6526, 6615, 6727, 6728	3176
dados exportados	fluxos	fluxos	amostragem de pacotes
protocolo de exportação	UDP SCTP	UDP TCP PR-SCTP SCTP	UDP
tempo para a análise dos dados	término do fluxo	término do fluxo	instantâneo
tempo do fluxo ativo	30 min	30 min	—
Descarte dos fluxos inativos	ajustável entre 10 a 600s	ajustável entre 10 a 600s	—
tráfego gerado	alto	alto	baixo
desempenho em redes de alta velocidade	baixo	baixo	alto

Um resumo das diferenças entre os protocolos de exportação de fluxos NetFlow, IPFIX e sFlow, pode ser observado pela Tabela 3.

No capítulo seguinte são apresentados os conceitos referentes ao algoritmo *Firefly*, bem como definições sobre Redes Neurais Artificiais (RNA). Tais abordagens são utilizadas pelo FADS durante as etapas de caracterização do perfil normal do tráfego e detecção de anomalias, respectivamente.

4 ALGORITMO FIREFLY E REDES NEURAIS ARTIFICIAIS

Neste capítulo, são apresentados os conceitos referentes ao algoritmo *Firefly* e às Redes Neurais Artificias, utilizados pelo FADS nos processos de caracterização de tráfego e detecção de anomalias no tráfego de rede de computadores, respectivamente.

4.1 Algoritmo *Firefly* (FA)

O algoritmo *Firefly* é uma metaheurística bioinspirada baseada no comportamento social dos vagalumes e de sua emissão de luz. Essa peculiaridade na emissão de luz é também conhecida como luminosidade ou intensidade luminosa. A frequência e intensidade luminosa, produzidos por um processo de bioluminescência, fazem parte de um sistema de atração entre os vagalumes, no qual as fêmeas respondem a um padrão único de piscar de um macho da mesma espécie.

A fim de desenvolver um algoritmo baseado no comportamento dos vagalumes, Yang [63] idealizou três regras:

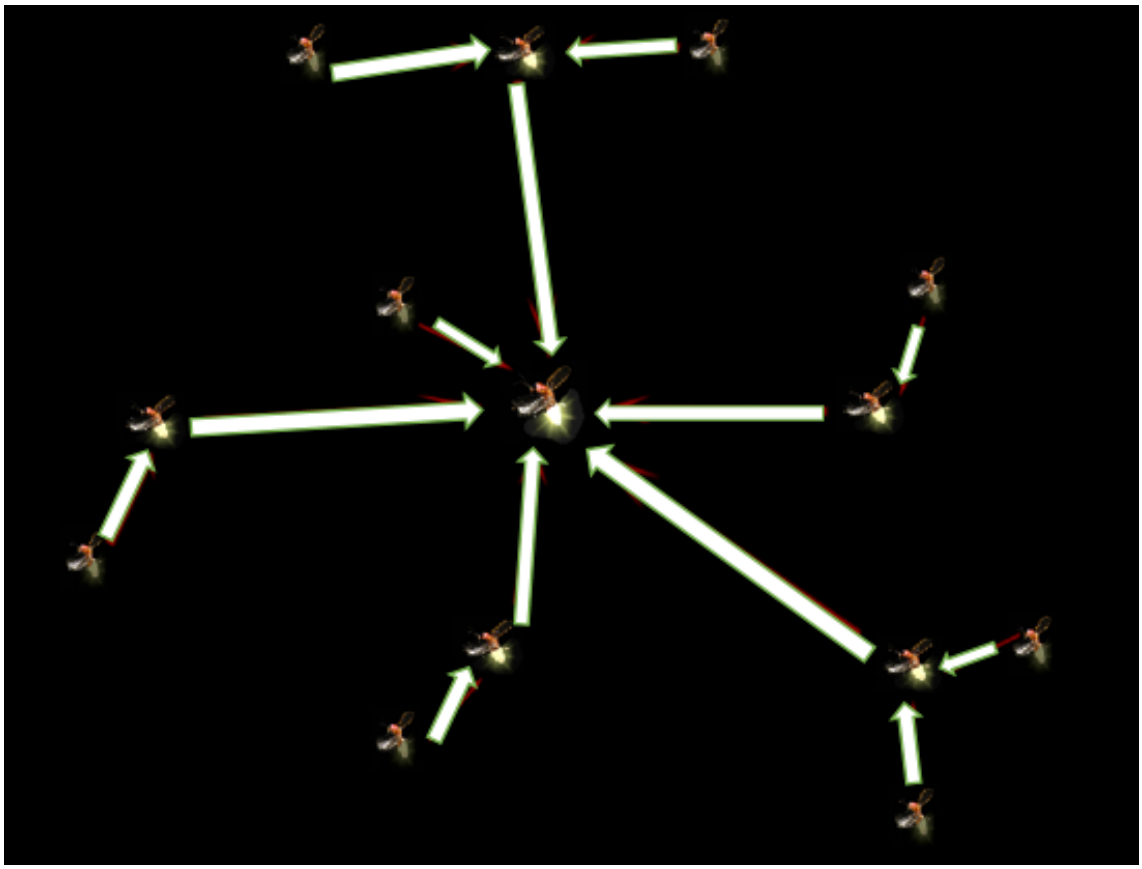


Figura 4 – Comportamento de atração dos vagalumes

- Os vagalumes são atraídos um pelos outros, independentemente do seu sexo;
- Essa atratividade é proporcional ao seu brilho e diminui à medida que a distância entre eles aumenta;
- A função objetivo a ser otimizada determina o brilho de um vagalume.

Pode-se retratar, segundo Yang, o comportamento de atração entre os vagalumes pela Figura 4. Nota-se, que vagalumes com maior intensidade luminosa atraem os de menor intensidade. Esse processo leva a população de vagalumes a convergir para a solução ótima.

Esse algoritmo é aplicado nas resoluções de problemas de optimização não linear, porém, de acordo com Nadhir [64], no FA há questões importantes a serem esclarecidas, tais como a atratividade e intensidade de luz, a distância e o movimento. Essas questões são explicadas na sequência.

4.1.1 Atratividade e intensidade de luz

A intensidade da luz varia com o quadrado da distância de acordo com a Equação 4.1 [25].

$$I_r = \frac{I_s}{r^2} \tag{4.1}$$

onde, I_{r} é a intensidade da luz a uma distância r e I_{s} é a intensidade da fonte.

Sabe-se também que a luz é absorvida pelo meio, de modo que deve-se permitir que a atratividade varie de acordo com o grau dessa absorção. Para um determinado meio com um coeficiente fixo de absorção de luz γ, a intensidade da luz pode ser determinada pela equação 4.2 [25]:

$$I_r = I_s.e^{-\gamma r} \tag{4.2}$$

Para evitar a singularidade em $r = 0$ em 4.1, a equação pode ser aproximada da seguinte fórmula Gaussiana, de acordo com a equação 4.3 [25]:

$$I_r = I_s.e^{-\gamma r^2} \tag{4.3}$$

Sabe-se, que a atratividade dos vagalumes é proporcional à intensidade da luz vistos por vagalumes adjacentes e assim, a atratividade β de um vagalume é determinada pela equação 4.4 onde β_0 é a capacidade de atração em $r = 0$ [25]:

$$\beta = \beta_0.e^{-\gamma r^m} \tag{4.4}$$

,sendo m $\geq$1

4.1.2 Distância

A distância entre quaisquer dois vagalumes i e j em x_i e x_j respectivamente, é determinada pela distância Euclidiana seguindo a equação 4.5 [25].

$$r_{ij} = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})} \tag{4.5}$$

onde x_{ik} é o k-ésimo componente da coordenada espacial x_i do i-ésimo vagalume, x_{jk} é o k-ésimo componente da coordenada espacial x_j do j-ésimo vagalume e d é o número de dimensões.

4.1.3 Movimento

O movimento de um vagalume i de ser atraído por outro vagalume j mais atraente, com mais luminosidade, é determinado pela equação 4.6 [25]:

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha\varepsilon \tag{4.6}$$

onde x_i é o vagalume menos brilhante movendo-se em direção ao vagalume x_j de brilho superior.

Da equação 4.6, sempre o segundo termo é devido à atração, enquanto o terceiro termo é aleatório com α sendo um parâmetro controlador do *step size* e ε é o vetor de números aleatórios extraídos de uma distribuição Gaussiana [65].

Nessa pesquisa, foram adotados os valores de β_0=1, γ=1 e α=0,2, baseados no trabalho realizado por Yang [25].

Adotando-se todas as considerações e afirmações citadas anteriormente, um fluxograma geral do algoritmo FA, pode ser apresentado pela Figura 5.

Na figura verifica-se que para que seja possível a aplicação do FA na resolução de um problema, deve-se necessariamente seguir alguns preceitos:

- Criar uma população inicial, onde cada vagalume representa um candidato à solução;
- Definir uma função objetivo que determina a luminosidade de um vagalume e consequentemente sua atratividade;
- Iniciar a análise com o sorteio de um vagalume de referência e assim os outros, com luminosidades inferiores, são atraídos por esse;

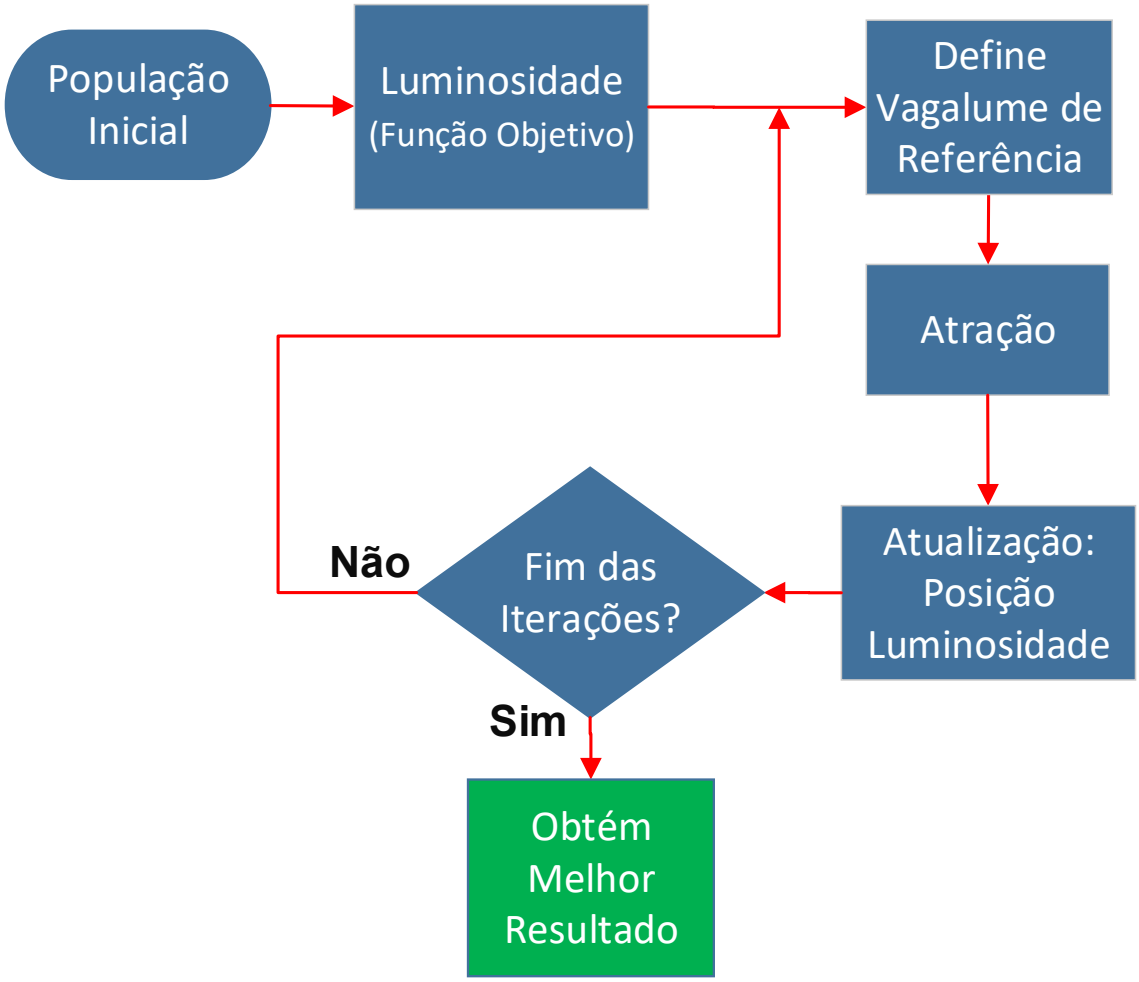


Figura 5 – Fluxograma geral do FA

- Após a atração, atualiza a posição e a luminosidade de cada vagalume;
- A cada iteração os vagalumes são reagrupados ao enxame em busca das melhores soluções;

4.2 Redes Neurais Artificiais (RNA)

Uma rede neural artificial (RNA) pode ser definida, segundo Bishop [66], como um método capaz de solucionar problemas utilizando inteligência artificial. Simulando o comportamento do cérebro humano de tal forma a ser capaz de aprender através das experiências.

As redes neurais artificiais são formadas por módulos que recebem e encaminham informações. Esses módulos são formados por neurônios, responsáveis pelo processamento das informações; pelas conexões, onde os sinais são propagados e pelos pesos sinápticos, cuja função é a de ponderar os dados transmitidos [66].

Em uma RNA, cada neurônio é constituido por três funções, sendo elas, a função

soma, a função ativação e a função de saída. Essas três funções, quando associadas a um neurônio, recebem o nome de função identidade. Um modelo de neurônio utilizando esse conceito [67], é mostrado na Figura 6.

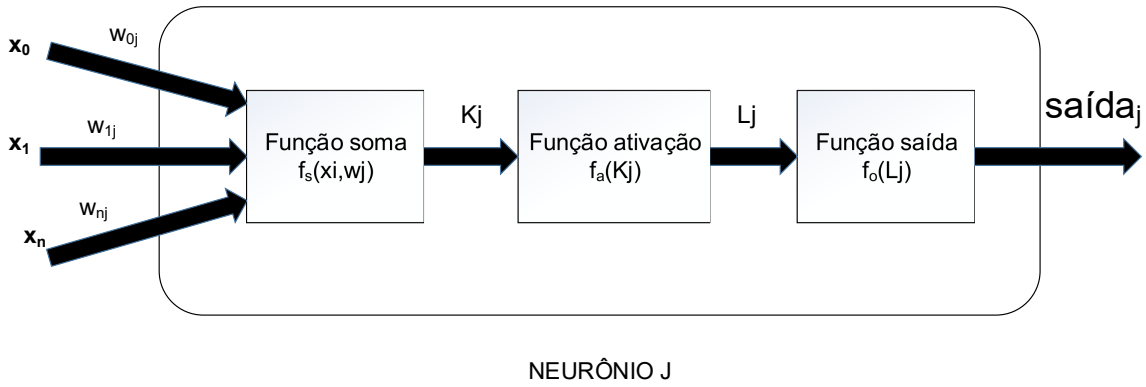


Figura 6 – Modelo de neurônio

Pela Figura 6, pode-se observar que a função soma combina as entradas x_i e seus respectivos pesos w_{ij}. A função de ativação calcula o nível de atividade do neurônio e a função de saída, associa a ativação de um neurônio a um valor de saída.

4.2.1 Arquitetura

Existem três tipos principais de arquitetura em RNA:

- Rede *feedforward* com uma única camada: essa rede é constituída de uma camada de entrada e uma camada de saída [68]. Nessa rede, o sinal se propaga apenas da entrada para a saída, em um sentido denominado de positivo. Essa rede pode ser observada na Figura 7;

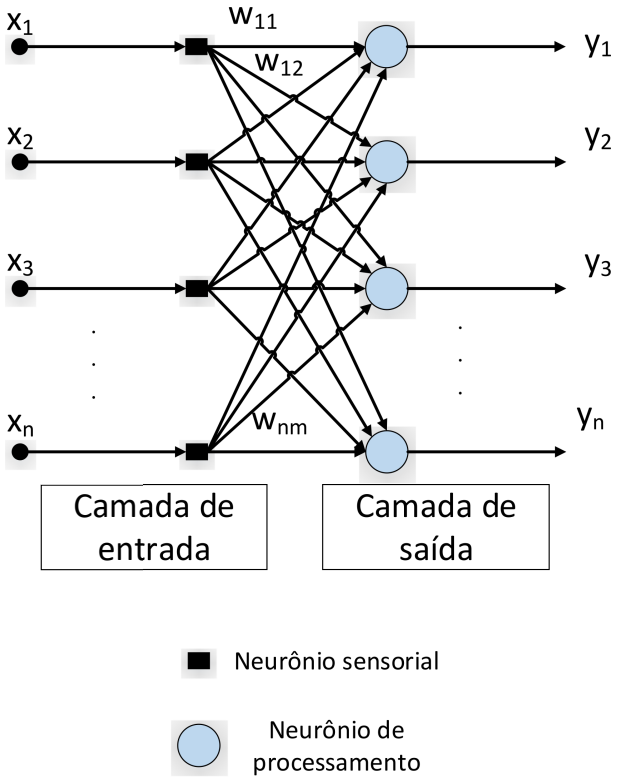


Figura 7 – Rede *feedforward* com uma única camada

- Rede *feedforward* de múltiplas camadas: nessa rede, como mostrado pela Figura 8, existem uma ou mais camadas intermediárias ou escondidas, cujas saídas são utilizadas como entrada para a próxima camada [69];

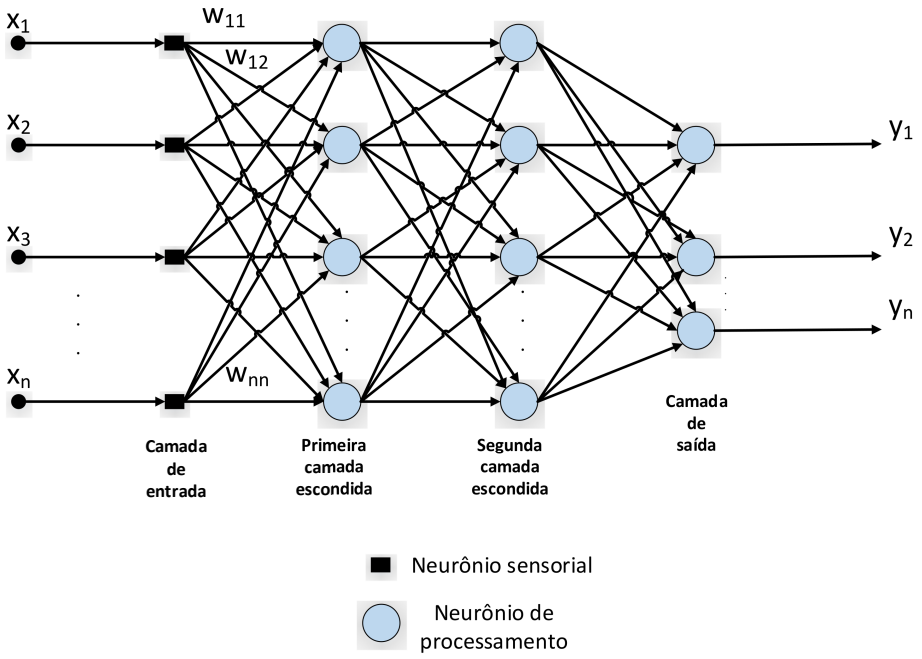


Figura 8 – Rede *feedforward* de múltiplas camadas

- Redes recorrentes: nessa arquitetura, as redes possuem no mínimo um elo

realimentando a saída de neurônios para a entrada de outros neurônios da rede [70]. Um exemplo dessa rede é mostrado na Figura 9.

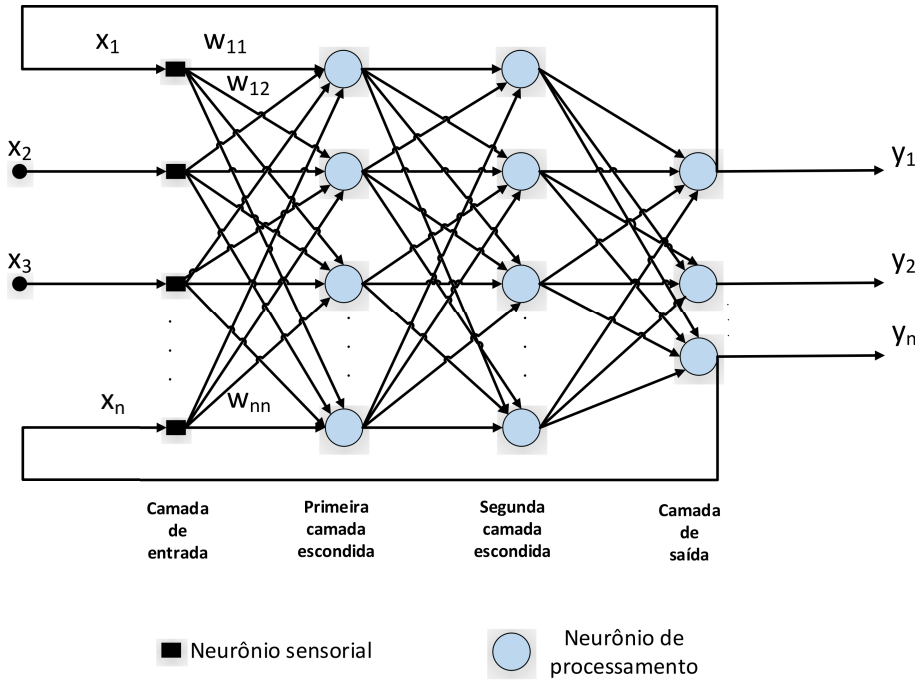


Figura 9 – Redes recorrentes

Para o desenvolvimento de aplicações usando RNA, se faz necessário realizar três passos [71]: coleta de dados, treinamento e teste. Esses passos são descritos a seguir.

4.2.2 Coleta de Dados

A coleta de dados se resume em separar os dados relativos ao problema em dois conjuntos, o de treinamento e o de teste.

4.2.3 Treinamento

No treinamento, a habilidade de aprender é a característica mais importante das redes neurais artificiais, onde seus pesos são ajustados em busca de uma solução generalizada. Esse processo também é chamado aprendizado e é baseado em informações conhecidas de entrada e saída, onde o modelo adquire a sistemática necessária para realizar apropriadamente o processo pretendido dos dados fornecidos [72].

De acordo com Demuth et al. [71], o processo de treinamento de uma RNA pode ser divido em dois tipos:

- Supervisionado: utiliza um agente externo que informa à rede qual é sua correspondência de saída diante de uma combinação ou padrão de entrada;

- Não supervisionado: não há a utilização de uma agente externo indicando a resposta desejada, nesse caso, a rede se auto organiza por correlações existentes entre os dados de entrada.

4.2.4 Teste

O teste é o último passo no desenvolvimento de uma RNA. Nessa etapa, são utilizados novos dados de entrada não previamente utilizados e a performance da rede, medida nessa fase, é uma boa indicação de sua performance real.

No capítulo seguinte, é apresentado e discutido o modelo FADS. Esse modelo foi desenvolvido na busca de melhores soluções na detecção de anomalias no tráfego de rede.

5 SISTEMA DE DETECÇÃO DE ANOMALIAS - FADS

O sistema *Firefly Algorithm for Digital Signature* (FADS) foi desenvolvido como uma ferramenta de auxílio à gestão de rede de computadores. Sua finalidade é a de detectar comportamentos anômalos no tráfego de rede. O FADS é composto de duas etapas. A primeira, é a de construir um perfil de comportamento normal da rede, denominado de *Digital Signature of Network Segment using Flow Analysis* (DSNSF), baseado no reconhecimento de padrões do tráfego de rede. Esse perfil, descreve o comportamento esperado da rede e serve de referência ao estabelecimento de um padrão entre um tráfego normal ou anômalo do segmento analisado. Na segunda etapa, o FADS identifica o tráfego anômalo. Nessa abordagem, o tráfego real é comparado ao DSNSF em busca de desvios de comportamento.

Nas seções seguintes é descrito o funcionamento do FADS, suas fundamentações e as etapas necessárias à construção do DSNSF e detecção de anomalias.

5.1 Caracterização do tráfego de rede com FADS

Para a caracterização do tráfego de rede, o FADS utiliza dois algoritmos, o K-*means* [73] e o *Firefly* [25]. O K-*means* foi o método utilizado para decompor uma amostra de dados analisados em grupos menores, baseados em suas similaridades, denominados de *clusters*. Já o Algoritmo *Firefly* (FA) [25], foi utilizado na mineração de dados. Mineração de dados é um processo de busca de um padrão sobre uma grande quantidade de dados analisados. O FA foi escolhido baseado no fato de que ele não é dependente do gradiente da função a ser otimizada, mas simplesmente da avaliação da função objetivo a ser otimizada. Essa característica proporciona ao FA alta convergência com um tempo de processamento baixo.

Na subseção seguinte, é descrita a arquitetura do sistema do FADS, assim como os argumentos que são usados pelo FADS para otimizar o processo de agrupamento na obtenção do DSNSF, que é utilizado na detecção de anomalias no segmento de rede analisado.

5.1.1 Arquitetura do FADS

O FADS trabalha com a base histórica de dados do tráfego da rede, disposta em janelas temporais de 5 minutos, dos mesmos três dias de semanas anteriores ao dia que é analisado. Isto é, se for analisado uma segunda-feira, a base histórica de dados a ser utilizada é referente às três segundas-feiras anteriores à essa. A técnica utilizada

na determinação da quantidade ideal de semanas anteriores utilizadas pelo FADS, será esclarecida na subseção seguinte.

5.1.1.1 Aprendizado

As quantidades de semanas utilizadas para o aprendizado foram definidas após os resultados dos testes obtidos utilizando o erro quadrático médio normalizado (NMSE) e o Coeficiente de Correlação (CC), com base nas relações dos DSNSFs gerados e do tráfego real do mesmo dia, para os atributos bits/s e pacotes/s. Para isso, foram usados os dados de 1 a 6 semanas anteriores à data analisada. A data escolhida para este teste foi o dia 28 de agosto de 2015. Os dados que serviram de base ao teste realizado foram obtidos do tráfego da rede da Universidade Estadual de Londrina.

Os resultados dessa análise estão presentes na Tabela 4, onde verifica-se que os valores mais satisfatórios foram obtidos utilizando 3 semanas anteriores à data analisada, sendo 0,003684 e 0,771544, respectivamente para NMSE e CC, os melhores resultados conseguidos para o atributo bit/s e 0,000442 e 0,961283, relativamente para NMSE e CC, os valores mais adequados conseguidos na análise do atributo pacotes/s.

Tabela 4 – Semanas de Aprendizado

	bits/s		pacotes/s	
semanas anteriores	NMSE	CC	NMSE	CC
1	0,007059	0,849049	0,000733	0,961741
2	0,014984	0,930540	0,000663	0,966337
3	**0,003684**	**0,771544**	**0,000442**	**0,961283**
4	0,004406	0,930540	0,000501	0,965878
5	0,003930	0,918163	0,000466	0,969683
6	0,004380	0,927484	0,000499	0,958611

5.1.1.2 Agrupamento

Após a coleta dos dados, de acordo com o atributo desejado, referentes aos 5 minutos de cada uma das três semanas anteriores, esses são distribuídos em três *clusters*, de acordo com as similaridades definida pelo K-*means*.

Para encontrar e validar a quantidade ideal de *clusters* a serem utilizados pelo K-*means*, foi utilizada a ferramenta *Validation for Cluster Analyses Platform* (CVAP)[74]. A validação é uma etapa importante e necessária na análise de *clusters*. O CVAP fornece ferramentas importantes e ambiente de análise conveniente para avaliação de validade de soluções de *clustering*, estimativa do número de clusters e comparação de desempenho entre algoritmos de agrupamento. Estão incluídos nesta ferramenta 4 índices de validade

externa, 14 índices de validade interna e 5 algoritmos de agrupamento, além de suportar as métricas de similaridade de distância Euclidiana e coeficiente de correlação de Pearson [75]. Baseado nessas características, o CVAP foi escolhido para análise e avaliação dos agrupamentos utilizados pelo FADS.

Para índices internos, foi avaliado os resultados utilizando quantidades e características inerentes ao conjunto de dados. O número de *clusters* (NC) ideal geralmente é determinado com base em um índice de validade interna. Os princípios de alguns índices amplamente utilizados para estimar o NC e avaliar a qualidade de agrupamento são introduzidos a seguir:

- Silhouette: É um índice composto refletindo uma compactação e separação de *clusters*; um índice de silhueta maior que a média indica uma melhor qualidade global do resultado de agrupamento [76].

- Davies-Bouldin: De acordo com Dimitriadou[77], é uma medida da similaridade média entre cada *cluster* e seu mais semelhante; pequenos valores correspondem aos aglomerados que são compactos e têm centros que estão longe um do outro; por conseguinte, o seu valor mínimo determina um ótimo NC.

- Calinski-Harabasz: Dudoit [78] definem como a medida de isolamento entre os *cluster* e a coerência dentro do *cluster*; o seu valor máximo determina o melhor NC.

- Dunn: Definido por Halkid [79] como sendo uma medida que maximiza as distâncias inter-*cluster* enquanto minimiza as distâncias dentro do *cluster*; seus maiores valores indicam a presença de aglomerados compactos e bem separados, de modo que o NC que maximiza o índice é tomado como o ideal.

Para alimentar a base de dados do CVAP, foram utilizadas 5 amostras de 900 pontos de dados, referentes a cada 5 minutos de cada semana. O CVAP retornou o melhor valor para cada um dos 4 índices utilizados na determinação da quantidade de *clusters* a serem usados para os 900 pontos analisados de cada uma das 5 amostras. Após os resultados, foi utilizado a função estatística moda, que indica o valor que mais se repete numa amostra, sobre os valores dos 5 resultados de cada um dos índices e a esses novamente a moda a fim de encontrar a quantidade de *clusters* que o K-*means* usará. Os resultados destas análises são apresentados pela Tabela 5, onde chegou-se ao valor de 3 *clusters* como o melhor resultado.

Tabela 5 – CVAP

Índice	Avaliação					
	1ª	2ª	3ª	4ª	5ª	moda
Silhouette	2	3	3	2	3	**3**
Davies-Bouldin	4	3	3	3	4	**3**
Calinski-Harabasz	3	3	3	2	3	**3**
Dunn	2	3	3	2	3	**3**
	moda final					**3**

Para cada um dos 3 *clusters*, o FADS determina o seu melhor representante, ou seja, o centróide. Encontrar o centróide é um problema de otimização que busca a distância Euclidiana entre os pontos do *cluster* e o centróide. A distância Euclidiana é apresentada pela equação 5.1.

$$D_{ij} = \sum_{i=1}^{Q} \sum_{j=1}^{K} \sqrt{\sum_{n=1}^{d} (x_{i_n} - c_{j_n})^2} \tag{5.1}$$

onde Q é a quantidade de dados que são clusterizados, K é o total de *clusters*, d a dimensão, x_{i_n} indica o valor do dado i em n e c_{j_n} é o valor do centro do *cluster* j na dimensão n.

No final das iterações, existem três centróides, um para cada *cluster* definido pelo K-*means*.

5.1.1.3 Critério de parada

De acordo com Yang [63], a convergência da função objetivo utilizando FA ocorre entre 50 e 100 iterações. Para definir o melhor número de iterações para o FADS, foram realizados testes de NMSE e CC variando-se os números de iterações entre 10 e 100 para todos os seis atributos usados. Para essa análise, também foram utilizados os dados obtidos do tráfego da rede da Universidade Estadual de Londrina referentes ao dia 28 de agosto de 2015. Os resultados dessas análises são apresentados pelas Tabelas 6 e 7.

Tabela 6 – Resultados do CC por Iterações

Iterações	BITS	PACOTES	EIPOr	EIPDt	EPTOr	EPTDt
10	0,913176	0,965812	**0,910075**	0,842698	0,844782	0,973623
20	0,913292	0,965491	0,906753	0,844603	0,843796	0,972270
30	0,913466	0,966216	0,906243	0,844889	0,844771	0,973608
40	0,914586	0,965285	0,909274	0,850299	0,844951	0,973239
50	0,912855	0,965642	0,909834	**0,850480**	0,844387	0.972872
60	0,914284	0,965736	0,909360	0,846566	0,845136	0,972979
70	0,913786	0,965981	0,905746	0,846779	0,843373	0,972324
80	**0,914764**	**0,966480**	0,909026	0,841683	**0,847968**	**0,973771**
90	0,913800	0,965748	0,909854	0,842738	0,847000	0,973149
100	0,913985	0,965878	0,906302	0,847132	0,846963	0,973737

Pelos valores obtidos pode-se observar que os melhores resultados, tanto para a CC quanto para o NMSE, ocorreram em 80 iterações. Diante disso e buscando também o menor custo computacional, foi definido o valor de 80 iterações a serem utilizadas pelo FADS.

Tabela 7 – Resultados do NMSE por Iterações

Iterações	BITS	PACOTES	EIPOr	EIPDt	EPTOr	EPTDt
10	0,004478	0,000506	**0,000862**	0,004585	0,024865	0,000803
20	0,004468	0,000512	0,000898	0,004536	0,024995	0,000842
30	0,004467	0,000505	0,000904	0,004530	0,024758	0,000809
40	0,004429	0,000512	0,000876	0,004397	0,024840	0,000817
50	0,004485	0,000511	0,000865	**0,004395**	0,024893	0,000824
60	0,004434	0,000509	0,000882	0,004493	0,024490	0,000819
70	0,004451	0,000502	0,000906	0,004486	0,024691	0,000842
80	**0,004406**	**0,000496**	0,000881	0,004606	**0,024387**	**0,000798**
90	0,004467	0,000504	0,000868	0,004579	0,024415	0,000817
100	0,004450	0,000501	0,000904	0,004485	0,024422	0,000807

O motivo do NMSE e do CC terem seus valores aumentados a partir de 80 iterações se deve ao fato do NMSE estar relacionado ao tráfego real enquanto o CC e a função objetivo, às semanas de treinamento.

5.1.1.4 Representatividade dos Centróides

Nessa etapa, duas técnicas podem ser usadas: A atribuição de pesos e a média.

5.1.1.4.1 Atribuição de pesos

Para cada um dos centróides resultantes do processamento do FA em cada *cluster*, é atribuído um peso à sua luminosidade. Esse peso é aplicado de acordo com a quantidade de dados que cada um representa, definido pela equação 5.2. Em seguida, é processado o FA sobre esses três centróides com valores de luminosidade normalizados.

$$Lic_k = Lrc_k(\frac{nc_k}{N}) \tag{5.2}$$

onde, Lrc_k corresponde à luminosidade residente do centróide do *cluster* k; N à quantidade total de vagalumes por iteração e nc_k refere-se à quantidade de vagalumes do *cluster* k.

5.1.1.4.2 Média

Nessa operação, o representante dos centróides é obtido utilizando a média aritmética simples. Isso é obtido dividindo-se a soma dos valores de cada centróide pelo número deles, definido pela equação 5.3:

$$\bar{x} = \frac{1}{3}\sum_{i=1}^{3} x_i \tag{5.3}$$

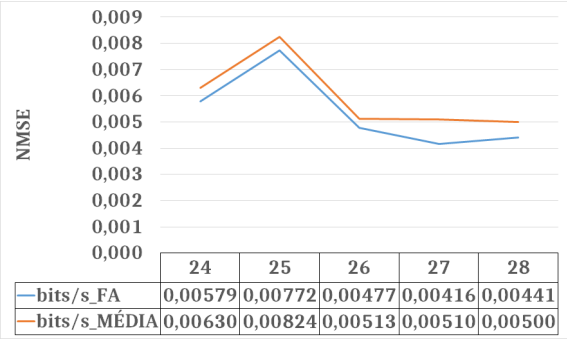
onde, x_i corresponde aos valores dos centróides resultantes do processamento do FA em cada *cluster*.

O uso de qualquer uma das operações citadas anteriormente resulta no centróide representante dos dados inicialmente selecionados, ou seja, dos 900 pontos de dados iniciais. Assim, é gerado o primeiro ponto do DSNSF. Essa abordagem será realizada até que toda a base de dados histórica seja processada e que os pontos que geram o DSNSF esperado sejam conhecidos.

Com a finalidade de verificar qual o melhor método para essa operação, foram realizados testes de desempenho com os DSNSFs gerados pelos dois métodos para cada atributo, ou seja, aplicando o FA e usando a média. Esses resultados foram aferidos utilizando o NMSE e são mostrados pelas Figuras 10 a 15. Para uma melhor visualização das informações contidas nos gráficos e figuras apresentados nesse trabalho, foram utilizadas, em alguns casos, as seguintes abreviações para os atributos qualitativos:

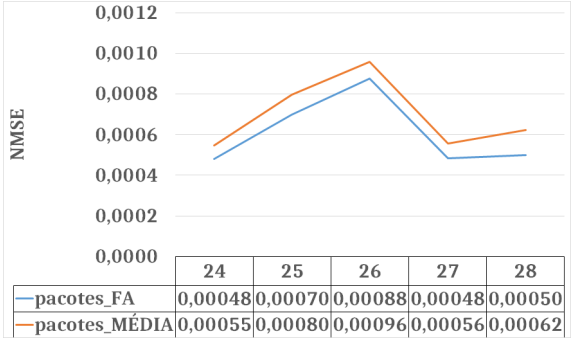
- entropia de Porta de Destino, abreviado em E(PtDt);

- entropia de Porta de Origem, abreviado em E(PtOr);
- entropia de endereço IP de Destino, abreviado em E(IPDt);
- entropia de endereço IP de Origem, abreviado em E(IPOr).



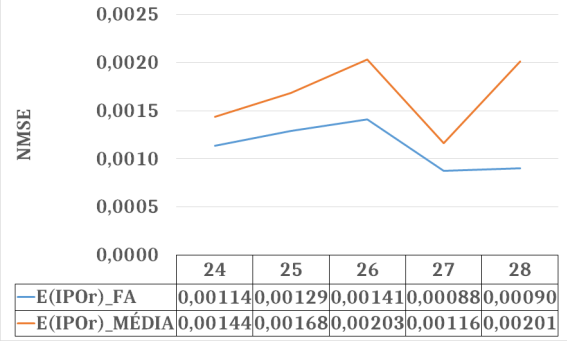
	24	25	26	27	28
bits/s_FA	0,00579	0,00772	0,00477	0,00416	0,00441
bits/s_MÉDIA	0,00630	0,00824	0,00513	0,00510	0,00500

Figura 10 – bits/s - FA x Média



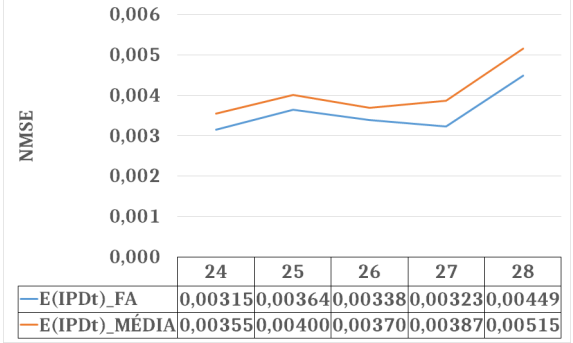
	24	25	26	27	28
pacotes_FA	0,00048	0,00070	0,00088	0,00048	0,00050
pacotes_MÉDIA	0,00055	0,00080	0,00096	0,00056	0,00062

Figura 11 – pacotes/s - FA x Média



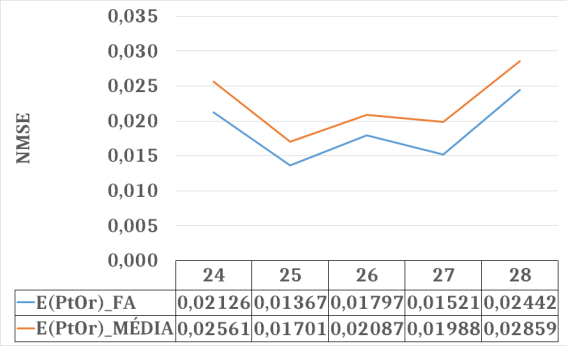
	24	25	26	27	28
E(IPOr)_FA	0,00114	0,00129	0,00141	0,00088	0,00090
E(IPOr)_MÉDIA	0,00144	0,00168	0,00203	0,00116	0,00201

Figura 12 – E(IPOr) - FA x Média



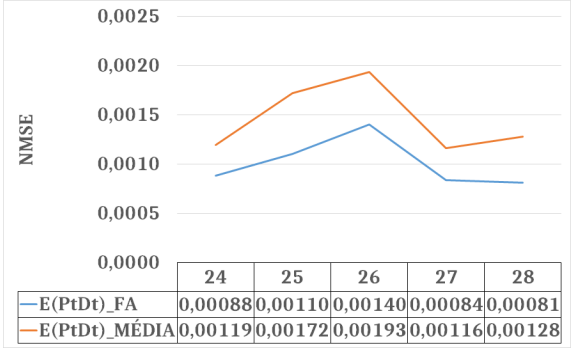
	24	25	26	27	28
E(IPDt)_FA	0,00315	0,00364	0,00338	0,00323	0,00449
E(IPDt)_MÉDIA	0,00355	0,00400	0,00370	0,00387	0,00515

Figura 13 – E(IPDt) - FA x Média



	24	25	26	27	28
E(PtOr)_FA	0,02126	0,01367	0,01797	0,01521	0,02442
E(PtOr)_MÉDIA	0,02561	0,01701	0,02087	0,01988	0,02859

Figura 14 – E(PtOr) - FA x Média



	24	25	26	27	28
E(PtDt)_FA	0,00088	0,00110	0,00140	0,00084	0,00081
E(PtDt)_MÉDIA	0,00119	0,00172	0,00193	0,00116	0,00128

Figura 15 – E(PtDt) - FA x Média

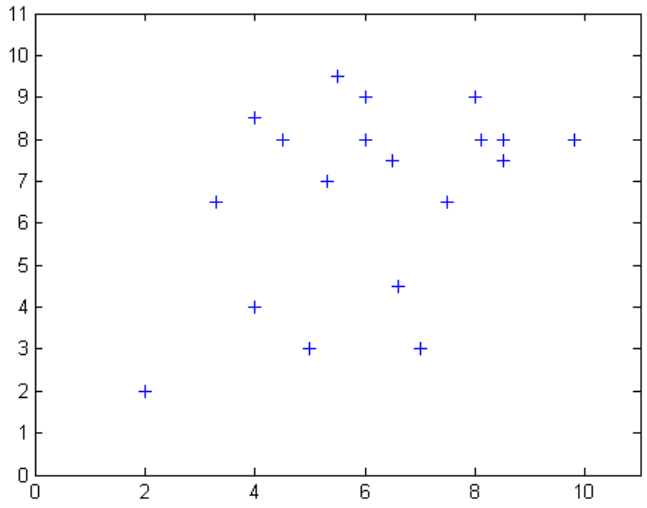
Pelas figuras, verifica-se que os melhores resultados foram obtidos utilizando o método de atribuição de pesos sobre os três centróides resultantes, pois, os valores obtidos pelo NMSE para esse método, foram menores do que para a média.

5.1.2 Caracterizando o tráfego

Para gerar o DSNSF, foi utilizado um histórico dos dados de fluxos IP, da Universidade Estadual de Londrina (UEL), do segundo semestre de 2015. Estes dados são coletados e armazenados em uma base histórica para futuras consultas e quando solicitados, são entregues em arquivos. Foram utilizados os arquivos contendo os atributos quantitativos bits/s e pacotes/s e qualitativos, entropia de IP de Origem, entropia de IP de Destino, entropia de Porta de Origem e entropia de Porta de Destino, coletados por segundo, dos dias 17 a 21 e de 24 a 28 de agosto de 2015 referentes a 1^a e a 2^a semana respectivamente analisadas, que serviram de aprendizagem e criação do DSNSF pelo FADS. A geração do DSNSF pode ser dividida em 4 etapas. Essas etapas são esclarecidas e retratadas utilizando um conjunto menor de dados exemplificados pelas Figuras 16 a 20, a seguir.

5.1.2.1 1^a Etapa na geração do DSNSF

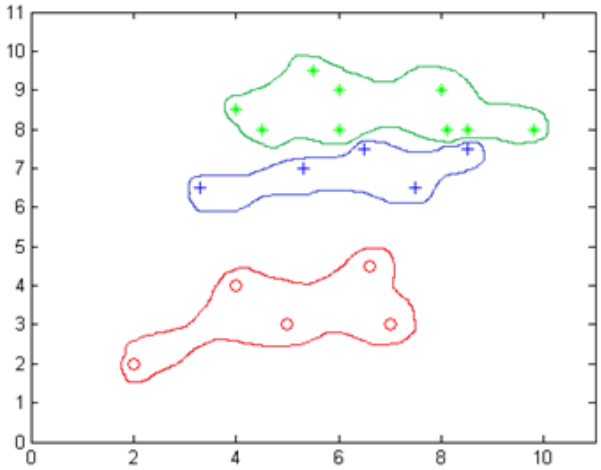
Inicialmente, as informações contidas nos arquivos dos dias 17 a 21 e de 24 a 28 de agosto de 2015 são preparadas para fornecerem dados a cada 5 minutos. Como um dia possui 24 horas, são gerados 288 pontos de dados representando cada um dos 5 minutos desse dia. Essa etapa é mostrada pela Figura 16.



XN	YN	LUMINOSIDADE
2,0	2,0	0,160
3,3	6,5	0,355
4,0	8,5	0,361
4,0	4,0	0,291
4,5	8,0	0,485
5,0	3,0	0,258
5,3	7,0	1,162
5,5	9,5	0,350
6,0	8,0	0,773
6,0	9,0	0,436
6,5	7,5	1,136
6,6	4,5	0,442
7,0	3,0	0,262
7,5	6,5	0,712
8,0	9,0	0,337
8,1	8,0	0,422
8,5	8,0	0,368
8,5	7,5	0,397
9,8	8,0	0,256

Figura 16 – 1ª etapa na geração do DSNSF

5.1.2.2 2^a Etapa na geração do DSNSF



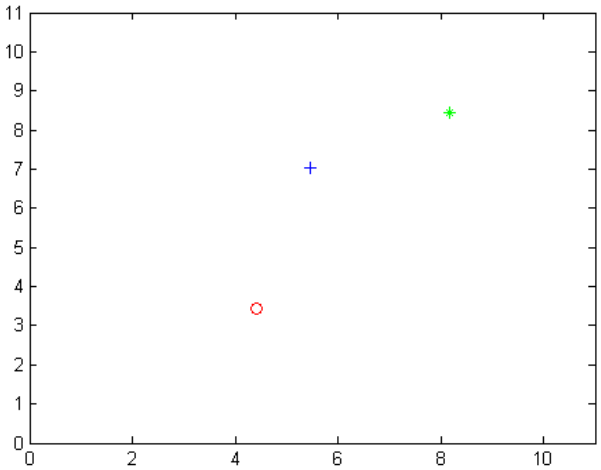
CLUSTER VERDE		
X	Y	LUMINOSIDADE
4,0	8,5	0,361
8,5	8,0	0,368
8,0	9,0	0,337
4,5	8,0	0,485
5,5	9,5	0,350
6,0	8,0	0,773
8,1	8,0	0,422
6,0	9,0	0,436
9,8	8,0	0,256

CLUSTER AZUL		
X	Y	LUMINOSIDADE
3,3	6,5	0,355
6,5	7,5	1,136
7,5	6,5	0,712
8,5	7,5	0,397
5,3	7	1,162

CLUSTER VERMELHO		
X	Y	LUMINOSIDADE
5,0	3,0	0,258
4,0	4,0	0,291
2,0	2,0	0,160
7,0	3,0	0,262
6,6	4,5	0,442

23

Figura 17 – 2ª etapa-A na geração do DSNSF - Agrupamento



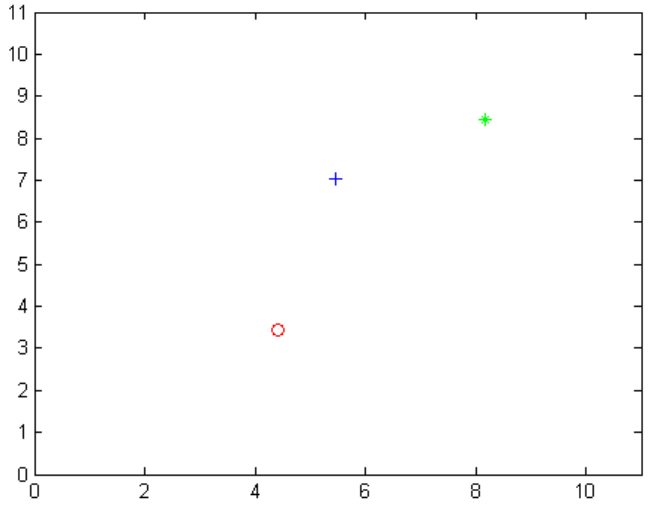
	X	Y	LUMINOSIDADE RESIDUAL
CENTRÓIDE Azul	5,5	7,0	50,248
CENTRÓIDE Vermelho	4,4	3,4	77,191
CENTRÓIDE Verde	8,2	8,5	2845,498

Figura 18 – 2ª etapa-B na geração do DSNSF - Definição dos Centróides

A segunda etapa é dividida em duas partes, *A* e *B*. Na primeira, mostrada pela Figura 17, os dados são processados pelo algoritmo K-*means* que os distribuem em três *clusters*, de acordo com suas similaridades. Na segunda parte, em cada *cluster* é aplicado o algoritmo FA para encontrar seu respectivo centróide. Este processo se dá pela otimização da função objetivo utilizada, demonstrada pela equação 5.1, onde a luminosidade dos vagalumes se relaciona diretamente. Esse processo está representado pela Figura 18.

5.1.2.3 3ª Etapa na geração do DSNSF

Nessa terceira etapa, após a obtenção dos três centróides, exemplificado pela Figura 18, é atribuído a cada um deles um peso. Esse peso está relacionado diretamente à quantidade de dados de cada *cluster* que eles representam e é aplicado sobre sua luminosidade residual. A equação de atribuição de peso é mostrada pela equação 5.2. Essa etapa pode ser visualizada na Figura 19.



	X	Y	LUMINOSIDADE PESO
CENTRÓIDE Azul	5,5	7,0	13,223
CENTRÓIDE Vermelho	4,4	3,4	20,313
CENTRÓIDE Verde	8,2	8,5	1347,868

Figura 19 – 3ª etapa na geração do DSNSF - normalização ponderada da luminosidade dos centróides

5.1.2.4 4ª Etapa na geração do DSNSF

Nessa última etapa, é utilizado o algoritmo FA sobre os três centróides a fim de encontrar o resultante da primeira amostra de 5 minutos analisada. Esse centróide global é o responsável pelo primeiro ponto de dados do DSNSF. Essa etapa é mostrada pela Figura 20.

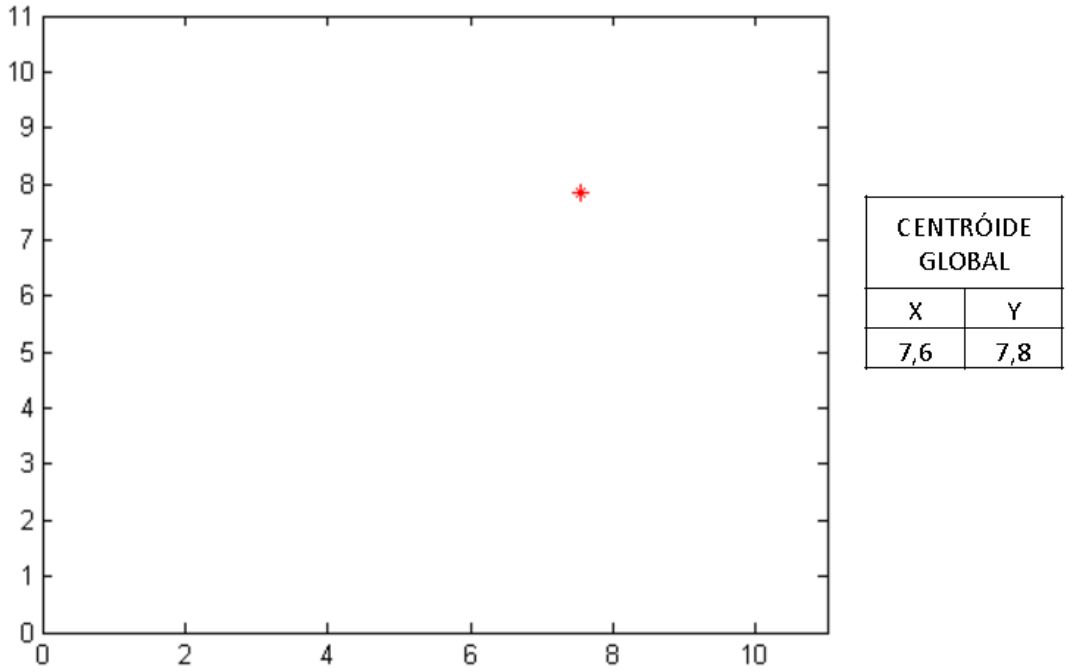


Figura 20 – 4ª etapa na geração do DSNSF

Na sequência, inicia-se a análise das outras 287 amostras. No final, tem-se 288 pontos de dados, que possibilitam construir o DSNSF desejado para as 24 horas do dia analisado.

O Algoritmo (1) expressa o pseudocódigo da criação do DSNSF utilizando o FA.

Algoritmo 1 – DSNSF

Entrada: atributos coletados da base histórica
Saída: X: Vector representante do comportamento normal do atributo do dia (DSNSF), organizado em 288 intervalos de 5 minutos

para: $i = 1$ to 288 **faça:**
  Aplica K-means, K=3
  para: $t = 1$ até o número de iterações **faça:**
    Aplica FA em cada *cluster*
    Calcule o centro de cada *cluster* da melhor solução - Equação (5.1)
  fim-para:
  Em cada *cluster*, aplica a função peso - Equação (5.2)
  para: $t = 1$ até o número de iterações **faça:**
    Aplica o FA nos três *clusters* individualmente, K=1
    Calcule o centro de cada cluster da melhor solução - Equação (5.1)
  fim-para:
  $X_i \leftarrow$ recebe centróide resultante
fim-para:
Retornar: X

Após as definições anteriores, pode-se retratar pela Figura 21 a operação global do algoritmo pelo Diagrama do FADS, aplicado na rede da UEL.

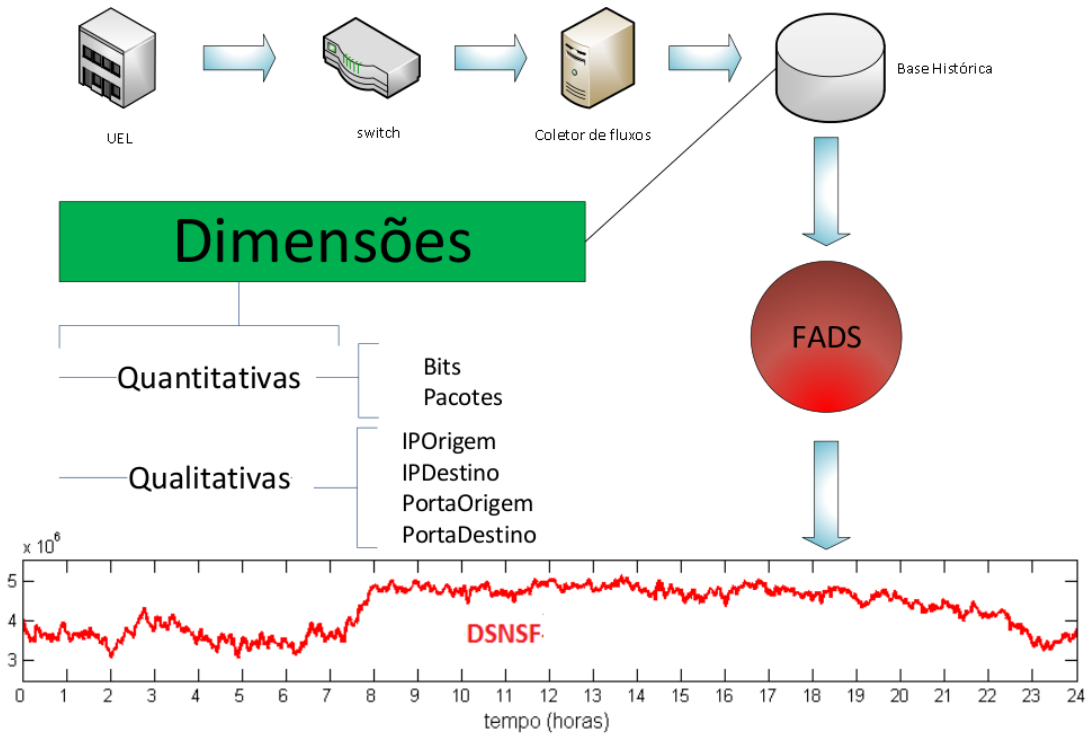


Figura 21 – Diagrama do FADS

Pela Figura 21, verifica-se que os fluxos IP da rede da UEL são coletados por um *switch* central e exportados no formato *sflow* para um coletor de fluxo, onde são armazenados num banco histórico. Durante a caracterização do tráfego da rede, o FADS requisita desse banco as informações referentes às três semanas anteriores do atributo a ser analisado. Esses atributos, também denominados de dimensões, podem ser quantitativos, bits/s e pacotes/s, quanto qualitativos, IP de origem e de destino e porta de origem e de destino. Esses dados então, são processados pelo FADS gerando o DSNSF do tráfego da rede, de acordo com o atributo analisado.

Na próxima seção, são descritos os mecanismos utilizados pelo FADS na detecção de anomalias em redes de computadores.

5.2 Detecção de anomalias com FADS

Para a detecção de anomalias no tráfego de rede, o FADS utilizou em seu algoritmo conceitos de redes neurais artificias (RNA) [80]. Nesse método, foi utilizado o FA para o aprendizado de aproximação da função objetivo e na classificação dos dados anômalos. Nessa etapa, o FADS confronta o DSNSF gerado anteriormente com o tráfego real da rede em busca do reconhecimento de comportamentos anômalos.

Neste trabalho, foi utilizada a estrutura de uma RNA com duas entradas x_1 e x_2 e um neurônio. A entrada x_1 recebe os dados reais do tráfego e a entrada x_2, o DSNSF gerado para o dia analisado. Esse modelo foi escolhido por proporcionar a solução de problemas linearmente separáveis, pela facilidade de uso e por ter gerado resultados eficientes. O modelo de aprendizado foi o supervisionado, onde os ajustes dos pesos dos vetores de entrada são definidos em função de um resultado esperado ($S_{\mathrm{e_n}}$) para cada uma das combinações dos valores de entrada.

Na sequência, são descritas as etapas de treinamento e de teste da RNA utilizada pelo FADS na detecção de anomalias no tráfego de rede.

5.2.1 Treinamento do FADS

Para o treinamento e processo de aprendizagem, o FADS utilizou o algoritmo *Firefly*. Nesse processo, a função objetivo recebeu a fórmula expressa pela Equação 5.4.

$$f_{obj} = \sum_{n=1}^{j}(tanh(\sum_{i=1}^{2}(x_i.w_i)) - S_{e_n}) \tag{5.4}$$

onde:

$tanh$= tangente hiperbólica;

n= define o instante de análise dos dados de entrada;

x_{i}= dados de entrada;

w_{i}= pesos sinápticos;

$S_{\mathrm{e_n}}$= saída esperada para o instante n;

Da Equação 5.4, pode-se ainda observar a função soma, identificada pela equação 5.5; a função de ativação, dada pela $tanh$; a função de saída mostrada pela equação 5.6 e a parcela referente ao treinamento supervisionado, dada por $\mathrm{S_{e_n}}$. Essas equações estão relacionadas ao modelo proposto na Figura 6.

$$\sum_{i=1}^{2}(x_i.w_i) \tag{5.5}$$

$$\sum_{n=1}^{j}(tanh(\sum_{i=1}^{2}(x_i.w_i))) \tag{5.6}$$

Para a entrada x_1, foram utilizados os dados reais do tráfego da UEL, referentes ao atributo pacotes/s do dia 27 de julho de 2015. Nesses dados, foram injetados pelo *Scorpius* anomalias simulando um DoS das 9h às 10h, um DDoS das 11h às 12h e das 16h

às 17h, um *flash crowd*. A característica dessa informação pode ser visualizada na Figura 22.

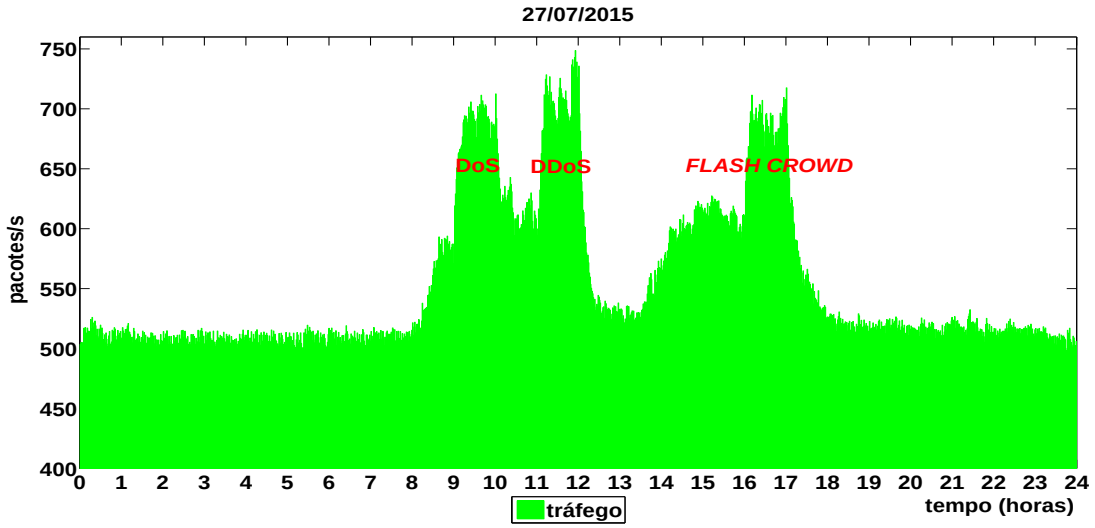


Figura 22 – Dados da entrada x_1, com anomalias injetadas pelo *Scorpius*

Na entrada x_2, os dados foram referentes ao DSNSF do dia 27 de julho de 2015, apresentado na Figura 23.

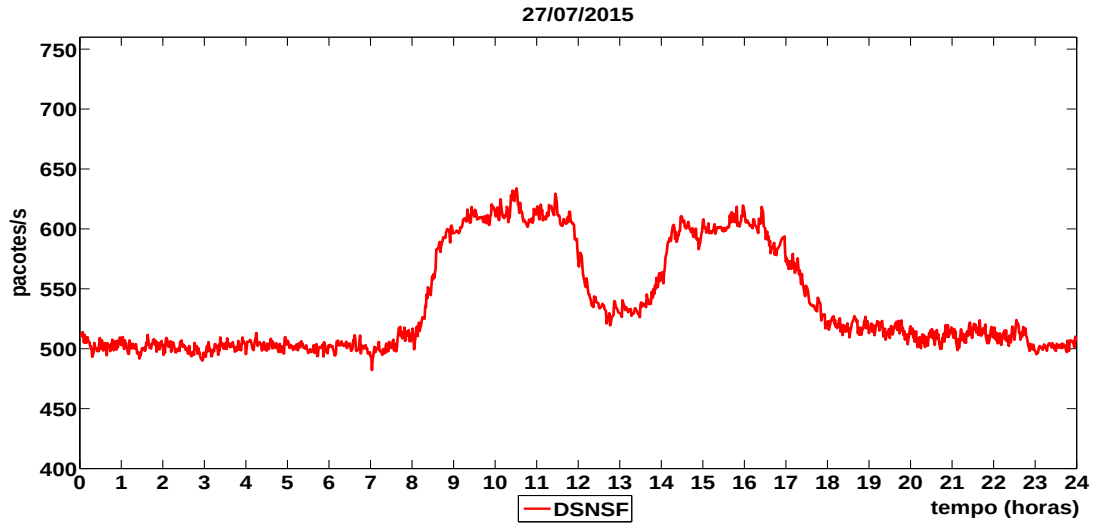


Figura 23 – Dados da entrada x_2

Para o aprendizado supervisionado, S_{en} recebeu as informações do resultado esperado para cada combinação dos valores de entrada x_1 e x_2, no instante analisado, representado pela Figura 24.

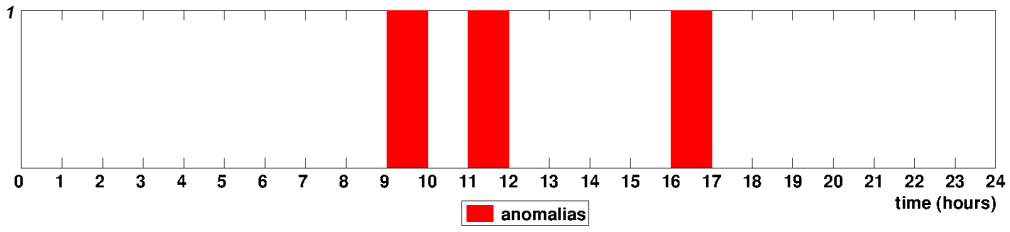


Figura 24 – Valores esperados de saída

Após a definição dos valores de entrada e de saída, o FADS processou essas informações até a minimização da função objetivo, definida pela Equação 5.4. Esse processo é apresentado na Figura 25.

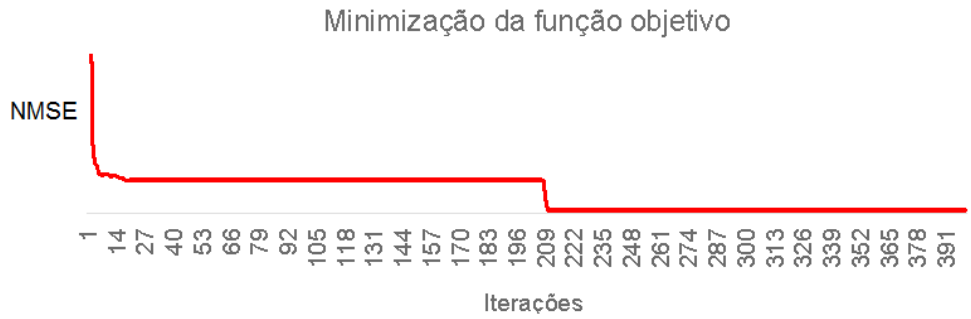


Figura 25 – Minimização da função objetivo

No final dessa etapa, o FADS já ajustou os pesos das conexões. Na próxima etapa, descrita a seguir, é realizado o teste da RNA, onde sua performance é determinada.

5.2.2 Teste do FADS

Nesse passo, foi realizado o teste da rede. Para esse teste, foram utilizados novos dados de entrada não previamente utilizados. A performance da rede, medida nessa fase, é uma boa indicação de sua performance real. Para isso, foram utilizados os dados do atributo pacotes/s do dia 29 de julho de 2015 da UEL. Foram injetados anomalidades referentes a DoS das 10h às 11h, para o DDoS, da 13h às 14h e *flash crowd* das 17h às 18h. Os resultados desse teste é apresentado na Figura 26.

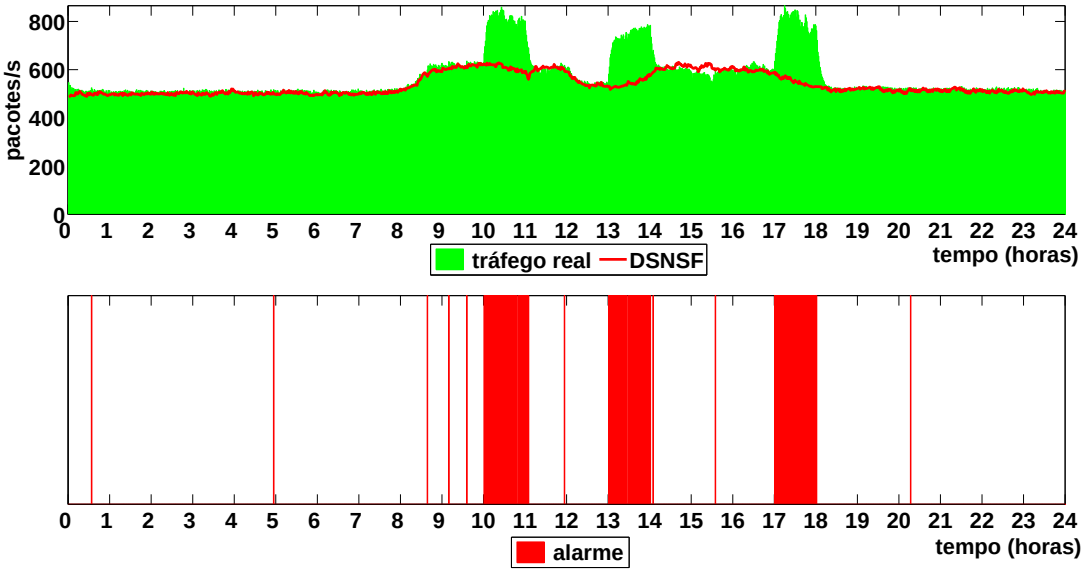


Figura 26 – Teste do FADS para 29 de julho de 2015

Como pôde ser observado, o FADS conseguiu detectar anomalias no tráfego da rede em todos os horários em que foram injetadas as anomalias pelo *Scorpius*, com 14 alarmes falsos positivos e 2 falsos negativos. Para melhor avaliar esse teste, foram utilizadas algumas métricas que são comentadas na sequência.

De acordo com a Tabela 8, o FADS conseguiu detectar 98,3% do ataque DoS e DDoS, com 100% no *flash crowd*. Seu valor da acurácia foi de 98,9%. Esses resultados mostram que o FADS conseguiu um ótimo aprendizado em sua rede neural, detectando padrões de ataque que não foram disponibilizados anteriormente à rede, na sua etapa de aprendizagem, porém, mais testes devem ser realizados com o objetivo de corroborar se estas medidas de avaliação se repetem, confirmando a eficiência do FADS.

Tabela 8 – Métricas de avaliação do FADS para 17 de março de 2015

	DoS	DDoS	*flash crowd*	acurácia
FADS	**98,3%**	**98,3%**	**100%**	**98,9%**

Com a rede treinada e avaliada, o FADS pode ser utilizado como uma ferramenta capaz de caracterizar e detectar padrões de ataques no tráfego de rede.

No próximo capítulo, o FADS é testado em vários ambientes, com seus resultados analisados e comentados.

6 TESTES E RESULTADOS

Neste capítulo são descritos os testes que avaliaram a eficiência do sistema FADS na previsão do comportamento normal do tráfego e na detecção de anomalias em redes de computadores. A metodologia usada está fundamentada na criação de uma assinatura digital (DSNSF), que é uma caracterização do perfil normal da rede. Esse perfil foi gerado pelo modelo FADS para cada dia analisado, obtida de um histórico do comportamento do tráfego de semanas anteriores. Além disso, fluxos de dados artificiais/sintéticos foram inseridos ao tráfego da rede analisada, que serviu como base para o reconhecimento de eventos anômalos que possam causar prejuízos à rede.

Os dados que serviram de base aos testes realizados foram obtidos dos tráfegos das redes da Universidade Estadual de Londrina (UEL) e da Universidade Tecnológica Federal do Paraná (UTFPR) - Campus de Toledo. A rede da UEL é constituída por um número maior de usuários do que a da UTFPR, sendo sua taxa de transmissão para o acesso à Internet da ordem de 1Gb/s com a utilização do formato *sFlow* [81] na amostragem de pacotes na relação de 1:256, enquanto que na UTFPR o acesso à Internet é realizado a 20Mb/s e os dados são coletados no formato *NetFlow.v9* [53] com amostragem de pacotes na razão de 1:1. Essas universidades, por apresentarem características e configurações distintas, permitiram avaliar a eficiência do sistema FADS em diferentes perspectivas.

Na sequência, são apresentados os conceitos das métricas, descritas por Liu [82], utilizadas para a mensuração dos resultados bem como os detalhes dos cenários propostos e a discussão dos resultados por eles apresentados.

6.1 Métricas de Avaliação

Métricas são processos de mensuração que estimam uma tendência ou comportamento. Isso permite aferir e analisar o desempenho de qualquer sistema.

Nas subseções seguintes, são abordadas as métricas de avaliação utilizadas nesse trabalho na análise dos resultados obtidos na caracterização e detecção de anomalias no tráfego de rede.

6.1.1 Caracterização de Tráfego

Uma etapa essencial para a detecção de sinais anômalos utilizando o sistema FADS é a realização da caracterização do tráfego de rede pelo modelo FADS. Nessa técnica, o comportamento normal da rede é identificado e um perfil, denominado de *Digital Signature of Network Segment using Flow analysis* (DSNSF), é gerado para descrevê-lo. Se esse perfil normal da rede não for bem definido, todo o processo referente à detecção

de anomalias na rede pode ser prejudicado. Assim, para avaliar a eficácia dos DSNSFs gerados foram utilizadas duas técnicas, o erro quadrático médio normalizado (NMSE) [83] e o coeficiente de correlação (CC) [84]. Essas técnicas são descritas a seguir.

6.1.1.1 Erro Quadrático Médio Normalizado (NMSE)

O erro quadrático médio normalizado, do inglês *Normalized Mean Square Error* (NMSE), tem como objetivo avaliar a diferença absoluta entre o DSNSF gerado e o tráfego real obtido. Essa métrica obtém resultados de zero à infinito, onde valores próximos à zero constituem excelente caracterização enquanto que valores elevados indicam divergências entre o DSNSF e o movimento do tráfego real. Para o cálculo do NMSE é utilizada a Equação 6.1 [83].

$$NMSE = \frac{1}{n}\sum_{i=1}^{n}\frac{(D_i - R_i)^2}{\overline{DR}} \tag{6.1}$$

Da Equação 6.1 tem-se D e R que são duas variáveis discretas representando os vetores com os valores do DSNSF e do tráfego real respectivamente, com o índice $i = 1, 2, 3, \ldots, n$.

6.1.1.2 Coeficiente de Correlação (CC)

A métrica coeficiente de correlação é utilizada para indicar quão similar são duas variáveis. Nessa pesquisa, ela indica a relação entre as tendências do DSNSF com o movimento do tráfego real analisado, ou seja, o quanto o DSNSF gerado acompanha as tendências de decrescimento e crescimento do movimento real analisado. Para o cálculo do CC entre duas variáveis quaisquer, utiliza-se a Equação 6.2:

$$CC = \frac{\sum_{i=1}^{n}((D_i - \bar{D}).(R_i - \bar{R}))}{\sqrt{\sum_{i=1}^{n}(D_i - \bar{D})^2 . \sum_{i=1}^{n}(R_i - \bar{R})^2}} \tag{6.2}$$

onde, D e R representam respectivamente o DSNSF e o tráfego real.

O CC tem como resultado de sua equação valores entre -1 e 1. Esses valores indicam a relação diferenciada entre as variáveis analisadas. Os valores do CC e seus respectivos significados podem ser melhor interpretados fazendo-se uso da Tabela 9.

Tabela 9 – Significado dos resultados esperados para o CC

CC	**Significado**
1	Variáveis completamente iguais (há correlação)
0	Variáveis completamente diferentes (não há correlação)
-1	Variáveis são opostas (há correlação, porém inversa)

Pode-se observar que os resultados dos CC esperados para essa pesquisa devem estar mais próximos de 1, o que indica um melhor ajuste entre o DSNSF gerado e o tráfego real da rede.

6.1.2 Detecção de Anomalias

Após a análise da eficácia da caracterização do tráfego da rede utilizando o NMSE e CC, deve-se adotar novas métricas que possam avaliar as taxas de detecção de anomalias no sistema apresentado. Essas técnicas devem levar em consideração algumas características importantes que possam garantir a sua objetividade, tais como as taxas de alarmes verdadeiros positivos (VP), verdadeiros negativos (VN), falsos positivos (FP) e falsos negativos (FN).

Diante dessas necessidades, foi produzida uma tabela de contingência descrevendo todas as quatro possíveis relações entre a sinalização de saída do sistema com o que realmente aconteceu. Essas informações podem ser observadas na Tabela **??**.

Tabela 10 – Tabela de Contingência

		Previsão	
		Anomalidade	Normalidade
Tráfego Real	Anomalidade	**VP**	**FN**
	Normalidade	**FP**	**VN**

Na tabela de contingência apresentada, verifica-se que se uma instância com tráfego real com anomalidade e previsão indicando também uma anomalidade, o intervalo é definido como VP; caso contrário, se o sistema indicar normalidade, o intervalo é definido como FN. No caso de um tráfego real com normalidade for sinalizado pelo sistema como sendo anomalidade, esse é classificado como FP; caso contrário, se o sistema sinalizar normalidade, o intervalo é definido como VN.

Fazendo-se uso da Tabela **??**, novas métricas podem ser utilizadas para mensurar a precisão do sistema proposto em detectar as anomalias presentes nos tráfegos das redes analisadas. Essas métricas são melhores esclarecidas na sequência.

6.1.2.1 Acurácia

A acurácia avalia a porcentagem de amostras classificadas corretamente tanto positivas (VP) quanto negativas (VN), em relação à soma de todas as amostras. Esse total de amostras consiste nas corretamente classificadas (VP e VN) quanto as erroneamente classificadas, denominadas de falso positivo (FP) e falso negativo (FN). Essa métrica tem a finalidade de calcular a probabilidade do sistema em gerar resultados corretos. A acurácia é definida de acordo com a Equação 6.3:

$$Acurácia = \frac{(VP + VN)}{(VP + VN + FP + FN)} \tag{6.3}$$

6.1.2.2 Especificidade

Especificidade é utilizada para calcular a porcentagem de amostras verdadeiramente negativas (VN) em relação ao total de amostras negativas (VN + FP). Ela é gerada pela Equação 6.4:

$$Especificidade = \frac{VN}{(VN + FP)} \tag{6.4}$$

6.1.2.3 Precisão

A precisão calcula a porcentagem de amostras verdadeiramente positivas (VP) em relação ao total de amostras classificadas como positivas (VP + FP). Ela é gerada pela Equação 6.5.

$$Precisão = \frac{VP}{(VP + FP)} \tag{6.5}$$

6.1.2.4 *Recall*

O *recall* calcula a porcentagem de amostras positivas (VP) classificadas corretamente em relação ao total de amostras classificadas como realmente positivas (VP + FN). Ela é gerada pela Equação 6.6.

$$Recall = \frac{VP}{(VP + FN)} \tag{6.6}$$

6.1.2.5 *F-measure*

A F-*measure* reflete a média entre a precisão e o *recall* e é dada pela Equação 6.7.

$$F = 2.\frac{(Precisão.Recall)}{(Precisão + Recall)} \tag{6.7}$$

Neste trabalho foram usadas duas outras técnicas de detecção de anomalias já apresentadas em outras pesquisas afins, que tiveram seus resultados comparados ao do modelo FADS proposto. Essas técnicas são o EWMA e o *Fuzzy*-GADS, descritas a seguir.

6.1.2.6 *EWMA*

O *Exponentially Weighted Moving Average* (EWMA) [85] [86] é uma estatística utilizada para monitorar um processo, calculando a média dos dados. A técnica utilizada pelo EWMA é a de aplicar menos pesos aos dados à medida que são removidos no tempo.

Neste trabalho, o EWMA é utilizado no estabelecimento dos limiares inferior e superior, também denominados de *tresholds*. É através desses limiares que é realizada a detecção de desvios do comportamento normal da rede em relação ao DSNSF gerado.

O EWMA utiliza pesos exponenciais de acordo com a distância temporal entre os elementos analisados, de acordo com a Equação 6.8.

$$mp_i = \lambda x_i + (1 - \lambda) mp_{(i-1)} \tag{6.8}$$

onde:

$\mathrm{mp_i}$ = valor da média ponderada para o instante i;

λ = constante entre 0 e 1;

$\mathrm{x_i}$ = valor observado no instante de tempo i;

i = está relacionado à quantidade de tráfego utilizado para a geração do DSNSF, sendo $i = 1, 2, 3, \ldots, n$.

n = número de semanas utilizadas para a geração do DSNSF.

Pela Equação 6.9, pode-se observar a fórmula da variância de mp_{i}, em que σ^2 é a variância dos dados usados para a geração do DSNSF. Pode ser verificado que quanto maior o valor de i, mais o termo [1 - (1 - $\lambda)^{2\mathrm{i}}$] converge para 1.

$$\sigma^2_{mp_1} = \sigma^2 (\frac{\lambda}{2 - \lambda}) \left[1 - (1 - \lambda)^{2i}\right] \tag{6.9}$$

Para o cálculo dos limiares superior e inferior para um instante i, onde L representa a amplitude dos limiares, o EWMA usa a Equação 6.10.

$$limiares = DSNSF \pm L\sigma \sqrt{\frac{\lambda}{(2 - \lambda)}} \tag{6.10}$$

Os valores aplicados neste trabalho para λ e L, foram 0,25 e 3, respectivamente. Esses valores foram adotados seguindo o estudo de Montgomery [87].

6.1.2.7 *Fuzzy*-GADS

O sistema *Fuzzy*-GADS foi desenvolvido por Hamamoto [37] como uma ferramenta capaz de detectar anomalias em redes de computadores, baseado no Algoritmo Genético (GA), no EWMA e na lógica *Fuzzy*.

O Algoritmo Genético (GA) é uma metaheurística baseada na Teoria da Evolução, desenvolvido por John Holland [88]. Essa abordagem é amplamente aplicada em problemas de otimização. Um Algoritmo Genético inicia a execução partindo de um conjunto inicial de soluções, em que as operações genéticas de Seleção, Recombinação e Mutação são aplicadas com o objetivo de alcançar melhores soluções.

No sistema *Fuzzy*-GADS, o GA é aplicado para gerar a caracterização do tráfego de rede e o EWMA, para calcular um limiar de desvio admissível entre a caracterização e o tráfego real. Com isso, é possível utilizar uma abordagem aplicando Lógica *Fuzzy* [89] para decidir se um dado intervalo de tempo apresenta uma anomalia.

A Lógica *Fuzzy* é um método que possui um bom desempenho para problemas que incluem um certo grau de incertezas [90], assim como existem na detecção de anomalias em tráfego de redes. Ao invés de atribuir valores absolutos de verdade (sim ou não, 0 ou 1, falso ou verdadeiro), a Lógica *Fuzzy* atribui um grau de pertinência, representado por um valor entre 0 e 1. Para cada atributo analisado, é calculado um valor de pertinência usando uma função Gaussiana. Essa função, usa como parâmetros o tráfego real, a caracterização e o limiar. O grau de pertinência é obtido pela Equação 6.11.

$$\zeta_k = 1 - \mathrm{e}^{\frac{-(x_k - \hat{x_k})^2}{2\theta_k^2}} \tag{6.11}$$

onde, x_k é o valor do tráfego, $\hat{x_k}$ é o DSNSF e o θ_k^2 é o EWMA do intervalo de tempo.

Caso o somatório dos graus de pertinência esteja acima de um valor pré-definido, um alerta é gerado, indicando que uma anomalia está ocorrendo nesse intervalo de tempo. Essa abordagem faz uma quantificação dos desvios entre a caracterização e o tráfego real em todos os atributos analisados, obtendo uma análise mais geral.

Um alto grau de pertinência indica um grande desvio do tráfego esperado, indicando a possibilidade da presença de uma anomalia. Caso o valor total dos valores de pertinência seja acima de um limiar geral, definido por Γ, uma anomalia está presente. Os graus de pertinência de cada atributo são agregados em único valor por um somatório, representado pelas regras nas equações 6.12 e 6.13. Assim, os atributos

utilizados para análise são avaliados de uma forma geral, uma vez que uma anomalia afeta vários atributos da rede.

$$Regra1: \ Se \ \sum_{k=1}^{G} \zeta_k \geq \Gamma \ , \ então \, \text{é} \, \text{“}anomalia\text{”} \tag{6.12}$$

$$Regra2: \ Se \ \sum_{k=1}^{G} \zeta_k < \Gamma \ , \ então \, \text{é} \, \text{“}normal\text{”} \tag{6.13}$$

6.2 *Cenários de Teste*

Nessa seção, são apresentados os cenários, os testes realizados e as análises dos resultados obtidos.

O sistema FADS foi testado em quatro cenários. No primeiro, objetivou-se comprovar a eficiência do FADS em caracterizar o tráfego de cada dia útil de uma semana do tráfego da UEL. O segundo cenário utiliza anomalias geradas artificialmente nos dados da UEL com a finalidade de comparar os resultados obtidos pelos métodos FADS e EWMA em detectar essas anomalias. O cenário seguinte analisa os resultados do uso do FADS em detectar anomalias provenientes da rede da UTFPR. Finalmente, o quarto cenário, utilizando os dados da UEL, confronta a eficiência dos métodos FADS, EWMA [91] e GA-Fuzzy [37] na detecção de anomalias. Esses cenários de testes estão relatados na Tabela 11.

Tabela 11 – Cenários de teste

Cenário	Experimento	Rede	Modelo(s)	Período
1	Caracterização de Tráfego	UEL	FADS	Treinamento: 03/08/2015 a 21/08/2015
				Testes: 24/08/2015 a 28/08/2015
2	Detecção de Anomalias	UEL	EWMA	Treinamento: não foi necessário
				Testes: 23/09/2015
			FADS	Treinamento: 27/07/2015
				Testes: 23/09/2015
3	Detecção de Anomalias	UTFPR	FADS	Treinamento: 27/07/2015 - UEL
				Testes: 23/04/2013
4	Detecção de Anomalias	UEL	EWMA	Treinamento: não foi necessário
				Testes: 28/08/2015
			Fuzzy-GADS	Treinamento: 31/07/2015 a 21/08/2015
				Testes: 28/08/2015
			FADS	Treinamento: 27/07/2015
				Testes: 28/08/2015

6.2.1 Cenário 1

Para esse cenário, procurou-se avaliar a eficiência do FADS em construir os perfis de seis atributos da rede analisada, sendo eles: bits/s, pacotes/s, entropia de endereço IP de Origem, entropia de endereço IP de Destino, entropia de Porta de Origem e entropia de Porta de Destino.

O FADS analisou os fluxos de 3 segundas, terças, quartas, quintas e sextas-feiras, objetivando gerar um perfil para cada dia útil da semana. Com esses registros, o FADS retornou para cada intervalo de um minuto, um valor que é tido como o comportamento normal daquele intervalo para o tráfego do seguimento de rede analisado.

Para análise desses atributos foram utilizados os dados históricos dos dias 03/08/2015 a 21/08/2015, coletados da Rede da UEL (Universidade Estadual de Londrina). Esses dados, estruturados no formato *sFlow* [81], foram coletados diretamente do *switch Extreme* BD8801, onde se concentra todo o tráfego da universidade, localizado na administração central da rede da UEL. Para a exportação dos dados no formato *sFlow*, foi utilizada a ferramenta *nfdump* [92].

Os seis atributos extraídos nos fluxos de dados usados, são representados nas figuras como:

- número de bits trafegados por segundo;

- número de pacotes trafegados por segundo;
- entropia de Porta de Destino, representada nas figuras como E(PtDt);
- entropia de Porta de Origem, representada nas figuras como E(PtOr);
- entropia de endereço IP de Destino, representada nas figuras como E(IPDt);
- entropia de endereço IP de Origem, representada nas figuras como E(IPOr).

As abreviações utilizadas para representar as entropias nos gráficos foram necessárias para um melhor dimensionamento de todos os atributos nas figuras.

6.2.1.1 DSNSFs

A caracterização do tráfego das 24h de cada dia no período de uma semana referentes aos dias 24/08/2015 a 28/08/2015 estão representados pelas Figuras 28, 29 e 27, em que a linha vermelha sinaliza os DSNSFs de cada atributo e a área em verde, o tráfego real de cada dia.

Para a geração dos DSNSFs, o FADS utilizou os dados referentes aos mesmos dias das três semanas anteriores à data analisada, ou seja, para gerar os DSNSFs dos atributos do dia 26 de agosto de 2015, foram usados para treinamento os dados referentes aos dias 5, 12 e 19 de agosto de 2015. Para o dia 27, utilizou-se as datas de 6, 13 e 20 e assim sucessivamente para todos os dias analisados.

Na Figura 28 tem-se os resultados dos perfis normais referentes aos dias 24 e 25 de agosto de 2015, respectivamente segunda e terça-feira. A Figura 29 ilustra os DSNSFs referentes a quarta-feira do dia 26 e da quinta-feira do dia 27 de agosto. Já a Figura 27 apresenta a aplicação do FADS sobre o movimento do dia 28. Pode-se observar nessas figuras que as assinaturas geradas pelo FADS conseguiram descrever, para todos os atributos, o comportamento do tráfego para os segmentos de rede analisados. Esses perfis, independente dos atributos e da característica do tráfego, acompanharam o movimento do dia analisado tanto nos horários de menor uso da rede, que se encontram entre 23h e 8h, quanto naqueles em que há uma maior exigência da rede, compreendido entre 8h e 23h.

28 de agosto de 2015

sexta-feira

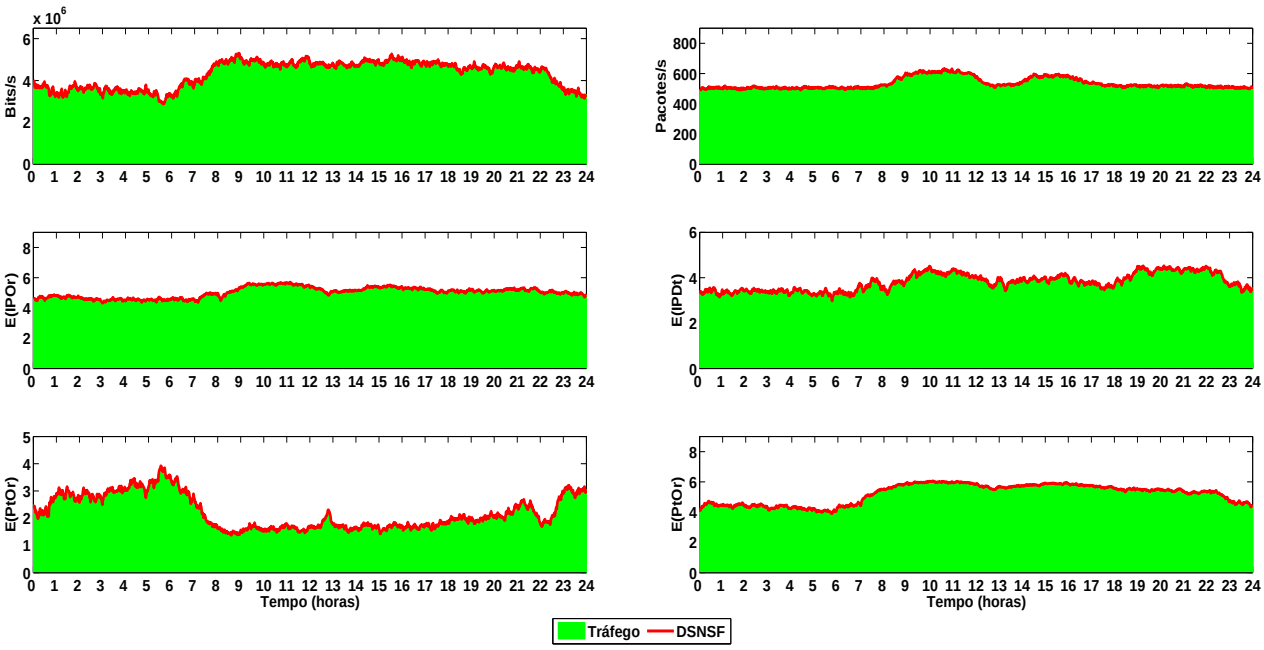


Figura 27 – DSNSFs para o dia 28 de agosto de 2015 - FADS

a) 24 de agosto de 2015 segunda-feira

b) 25 de agosto de 2015 terça-feira

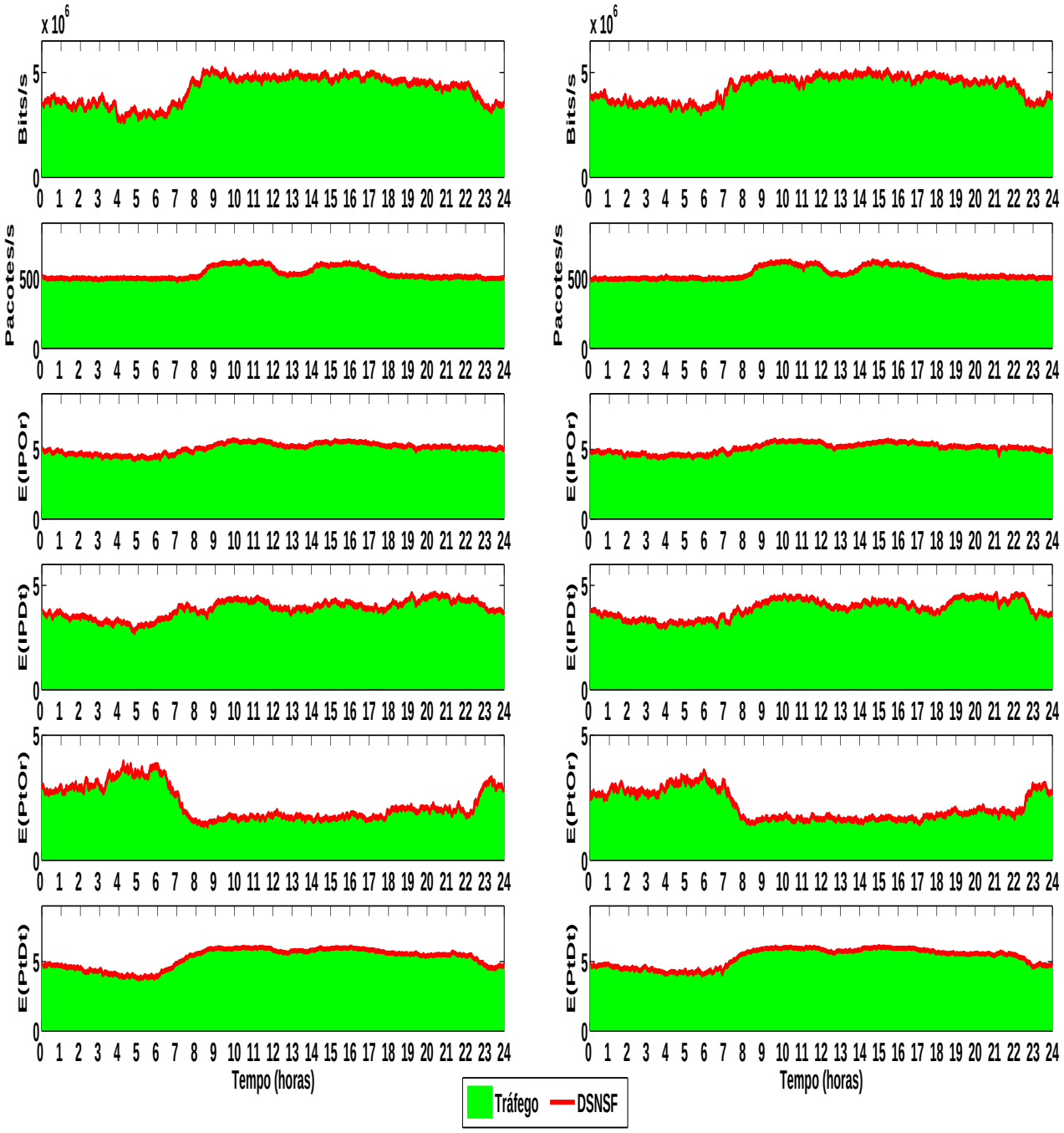


Figura 28 – DSNSFs para os dias 24 e 25 de agosto de 2015 - FADS

a) 26 de agosto de 2015

quarta-feira

b) 27 de agosto de 2015

quinta-feira

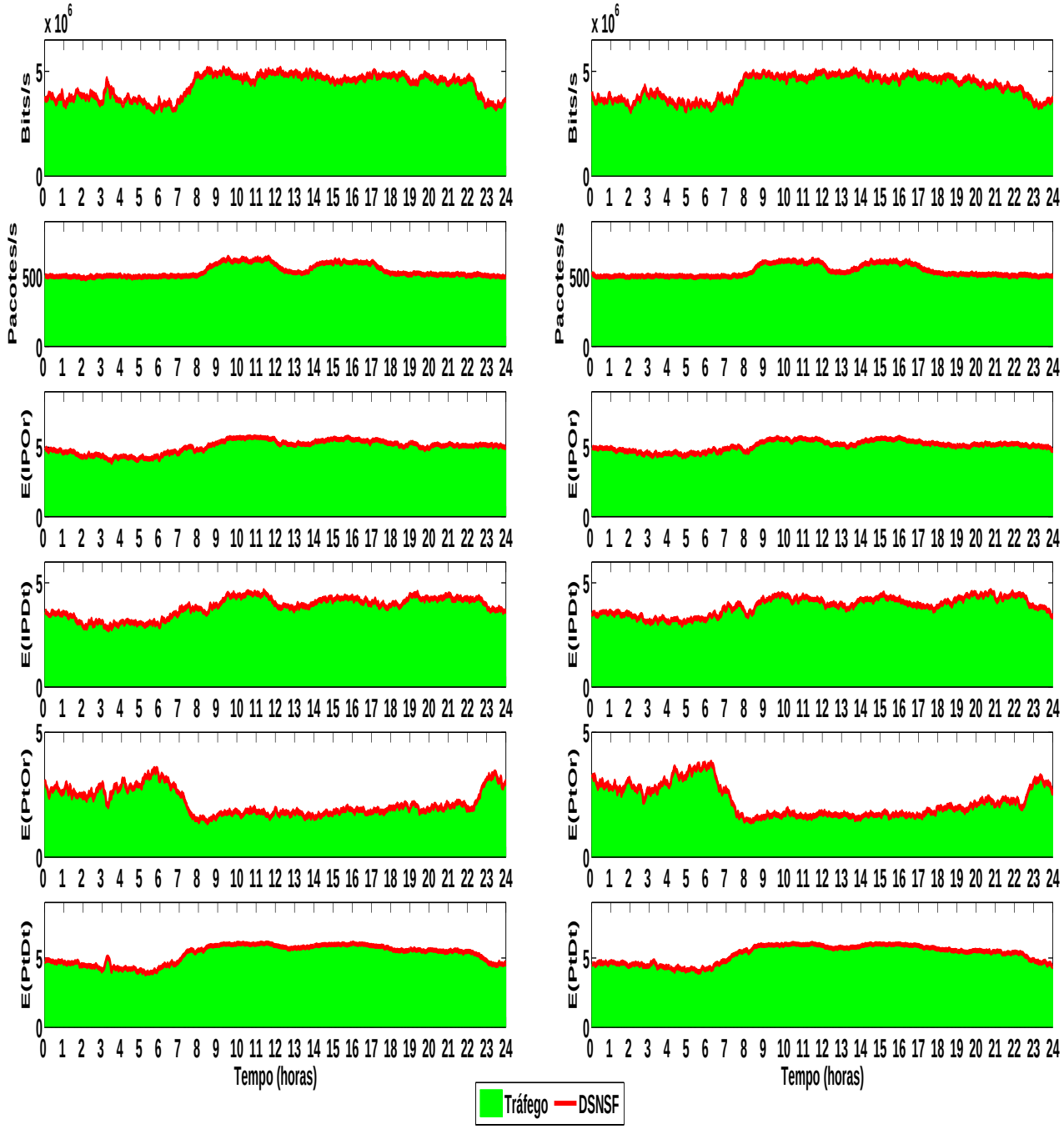


Figura 29 – DSNSFs para os dias 26 e 27 de agosto de 2015 - FADS

6.2.1.2 Métricas de Avaliação

Para corroborar a precisão do modelo FADS na caracterização do tráfego da semana analisada, foram utilizados 2 métricas: o Coeficiente de Correlação (CC) e o Erro Médio Quadrático Normalizado (NMSE).

A função do CC é indicar a direção e a força da relação entre duas variáveis que

para essa proposta são os DSNSFs e os dados reais do dia. Caso as mudanças sofridas por uma variável sejam acompanhadas por outra, existe uma correlação entre eles. O CC tem seu valor entre -1 e 1, onde 1 indica forte correlação positiva, -1 forte correlação negativa e 0 corresponde a nenhuma correlação.

Com relação ao NMSE, este quantifica a diferença entre valores estimados e o valor real obtido. Este método tem como objetivo medir o quão distante o DSNSF gerado se encontra, na média normalizada, do movimento real de tráfego observado. O modelo está mais próximo do movimento real, quanto mais próximo do zero for o valor obtido.

Os resultados obtidos pelas métricas NMSE e CC são apresentadas pelos gráficos das Figuras 30 e 31, respectivamente.

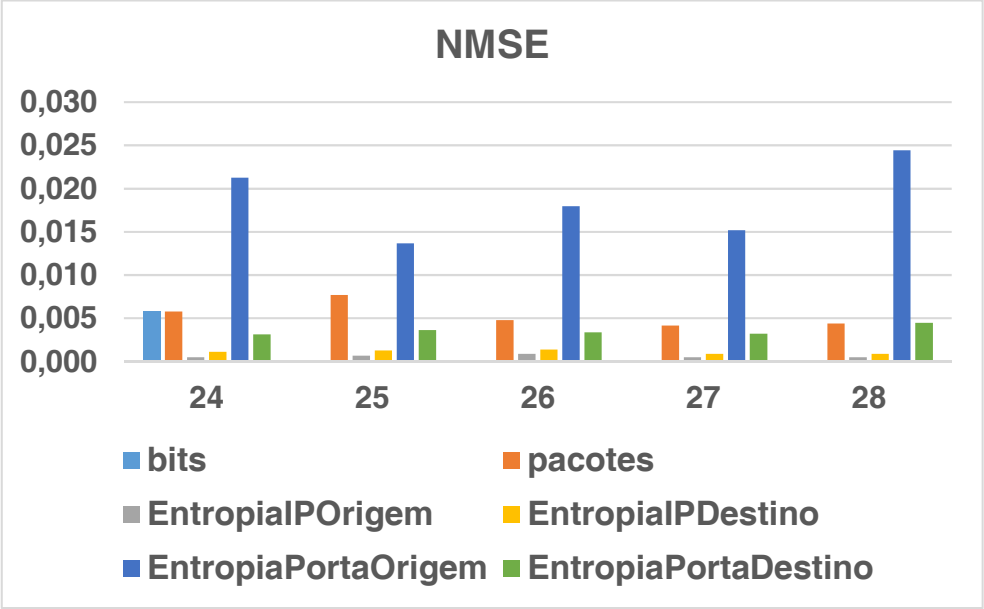


Figura 30 – NMSE - Todos Atributos - 24 a 28 de agosto de 2015 - FADS

Como pode ser observado na Figura 30, todos os erros se mantiveram abaixo de 0,010 para todos os dias referentes à semana analisada, exceto para a entropiaPortaOrigem que chegou próximo a 0,025. Esses dados indicam uma ótima caracterização do tráfego.

Analisando os valores das correlações apresentadas pela Figura 31, pode-se observar que os valores mensurados estão próximos de 1. Esses resultados indicam que houve ótima correlação entre o DSNSF gerado pelo FADS e os dados reais de cada dia.

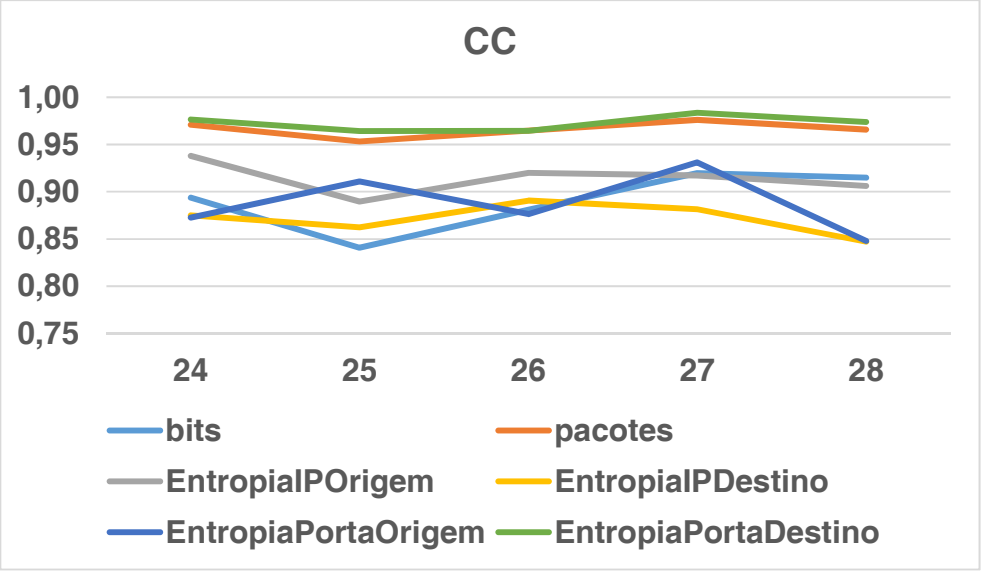


Figura 31 – CC - Todos Atributos - 24 a 28 de agosto de 2015 - FADS

Fazendo-se uso da Tabela 12, nota-se que o valor médio do NMSE para os atributos analisados foi de 0,005032. Esse resultado se encontra próximo do ideal que é zero, garantindo que houve boa previsão e aproximação do tráfego esperado. A mesma análise pode ser feita com o valor do CC médio obtido. Com uma medida superior a 0,91, o valor do CC médio indicou que o FADS conseguiu capturar as tendências de crescimento e decrescimento do tráfego em todos os atributos.

Após uma pesquisa com a finalidade de se explicar as variações nos resultados obtidos pelo NMSE e CC com relação aos atributos quantitativos e qualitativos foi o de que em 2015 os números de portas diferentes diminuíram drasticamente, fazendo com que o valor da entropia de porta de origem ficasse fora do padrão esperado, tendo como referências pesquisas realizadas em períodos anteriores a essa. Verificou-se também que existem muitos fluxos usando a porta 443, que é o HTTPS, o que levou a essa nova caracterização do atributo. Um outro fator foi a amostragem. Atualmente a amostragem é feita com 1:512, anteriormente a esta data era de 1:256.

Na Tabela 12 pode-se verificar que o dia com melhores resultados foi o 27 com valor de NMSE em 0,004132 e correlação em 0,934815.

Tabela 12 – Média dos valores de NMSE e CC de 24 a 28 de agosto de 2015 - FADS

	Média	
dia/08	NMSE	CC
24	**0,005451**	**0,921049**
25	**0,004688**	**0,903465**
26	**0,004968**	**0,916250**
27	**0,004132**	**0,934815**
28	**0,005921**	**0,909297**

Média Geral	**0,005032**	**0,916975**

Comparando os valores individuas do NMSE e do CC de cada atributo com a média calculada, verifica-se pouca variação entre os DSNSFs em relação ao tráfego real. Isso indica uma excelente performance do FADS pois manteve uma consistência na caracterização do tráfego de rede analisado nos períodos de maior e menor atividade. Assim, com esse cenário, conclui-se que o FADS foi capaz de gerar com alta precisão o comportamento do segmento estudado para todos os atributos analisados.

6.2.2 Cenário 2

O segundo cenário utiliza novos dados, ainda oriundos da Universidade Estadual de Londrina (UEL), com a finalidade de comparar o desempenho do FADS e do EWMA na detecção de anomalias em redes de computadores. Esse novo intervalo de dados está compreendido entre os dias 2 a 16 de setembro de 2015. Essa data é utilizada pelo FADS para gerar os DSNSFs do dia 23 de setembro de 2015, que são utilizados na detecção de anomalias.

Fazendo-se uso da ferramenta *Scorpius* [93], que faz a simulação de anomalias no tráfego de rede, foram injetados diretamente no arquivo de dados analisado do dia 23, fluxos de pacotes correspondentes à anomalias provocadas por DoS, DDoS e *Flash Crowd*. As informações referentes às anomalias utilizadas por ambos os modelos, são descritas pela Tabela 13.

Tabela 13 – Descrição das Anomalias utilizadas no cenário 2

Ataque	**Horário**	**IP de Origem**	**Porta de Origem**	**IP de Destino**	**Porta de Destino**
DoS	das 9h às 10h	13.122.10.147	1433	43.233.244.117	2210
DDoS	das 13h às 14h	5 IPs		207.170.215.37	1209
Flash Crowd	das 15h às 16h	10 IPs		20.181.181.90	3523

Na sequência, apresentaremos os DSNSFs gerados pelo FADS e as métricas utilizadas para sua avaliação.

6.2.2.1 DSNSFs

Para a geração dos DSNSFs do dia 23, foram utilizados os dados referentes ao mesmo dia das três semanas anteriores a essa data, ou seja, os dias 2, 9 e 16 de setembro de 2015.

A Figura 32 apresenta por meio da área verde, os movimentos do tráfego para todos os atributos do dia 23 de setembro de 2015 e pela linha vermelha, a caracterização efetuada pelo FADS.

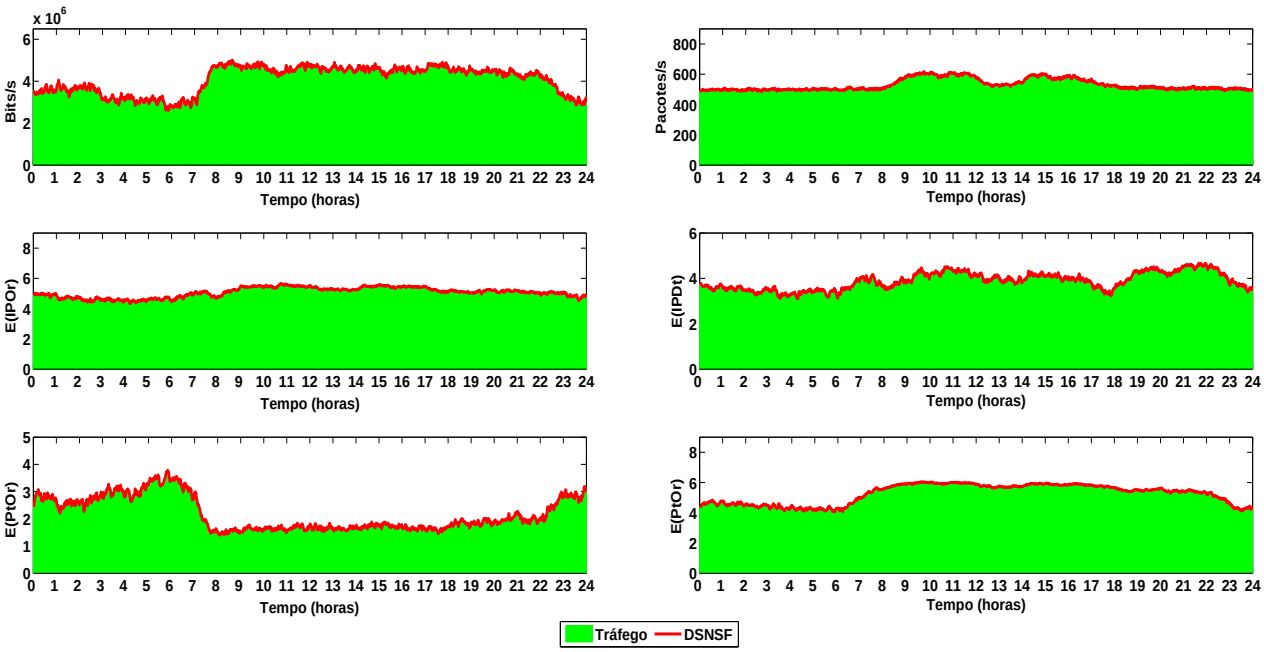


Figura 32 – DSNSFs para o dia 23 de setembro de 2015 - FADS

Na Figura 32 é possível verificar a eficiência do sistema FADS na criação de perfis da rede, pois a figura mostra coerência entre os DSNSFs e o tráfego real. Para mensurar a eficiência dos DSNSFs gerados, foram utilizados os testes de NMSE e CC, tendo seus resultados ilustrados pelas Figuras 33 e 34, respectivamente.

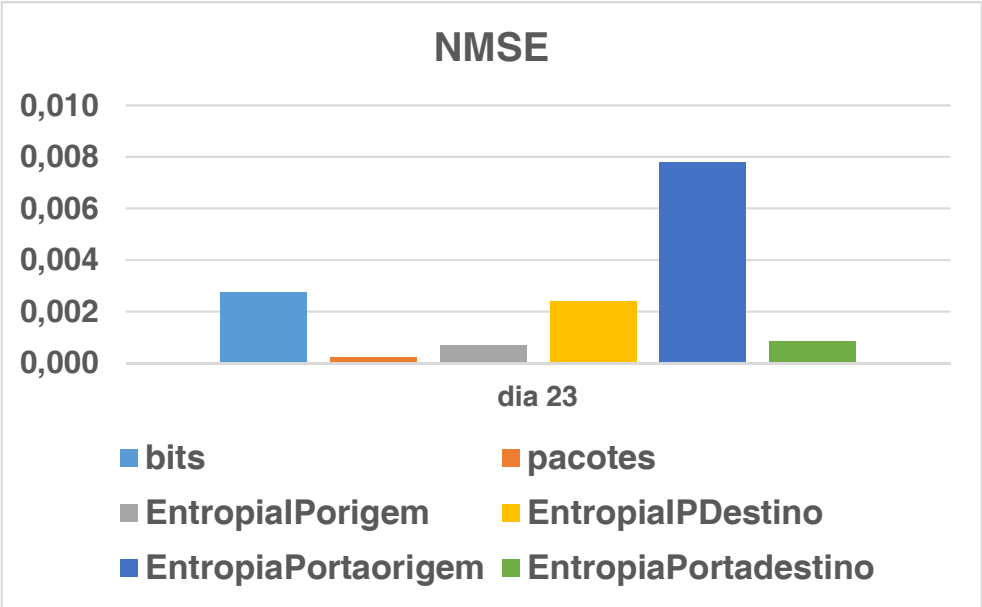


Figura 33 – NMSE para o dia 23 de setembro de 2015 - FADS

A Figura 33 apresenta os valores de NMSE para cada um dos atributos do dia analisado. O NMSE estima a habilidade de previsão do perfil da rede pelo sistema FADS. Nota-se que os valores estão próximos de zero para quase todos os atributos, indicando uma excelente caracterização do tráfego de rede.

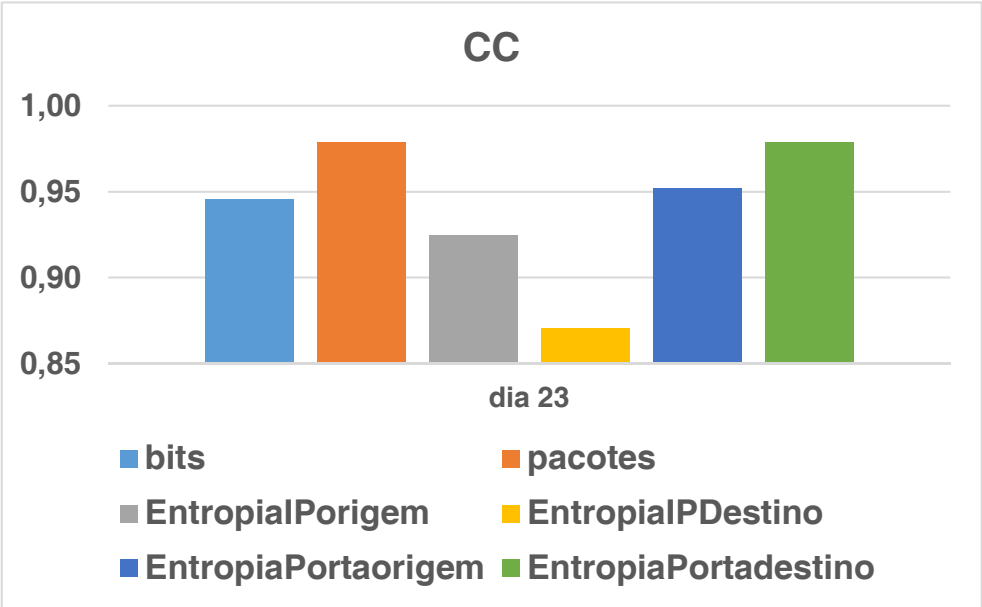


Figura 34 – CC para o dia 23 de setembro de 2015 - FADS

O CC mensura a correlação entre variáveis, o que possibilita utilizá-lo para verificar se o tráfego previsto pelo FADS corresponde ao tráfego real. A Figura 34 apresenta os valores do CC para os atributos do dia analisado. Observa-se que os valores

estão próximos de 1 para todos os atributos, indicando uma ótima correlação entre os DSNSFs gerados pelo FADS e o tráfego real do dia.

Na sequência, foi analisado as anomalias do dia 23, as respostas do FADS e EWMA frente a essas anomalias e os resultados das métricas utilizadas para mensurar a eficiência de cada um dos modelos.

6.2.2.2 Detecção de Anomalias do FADS x EWMA

Como descrito anteriormente, esse cenário avalia a eficiência dos modelos FADS e EWMA em detectar as anomalias, geradas artificialmente, nos fluxos referentes ao dia 23 de setembro de 2015.

Para a classificação desejada, o modelo FADS usou os dados da base histórica da UEL referentes ao dia 27 de julho de 2015 para ajustar os pesos e os *thresholds* de sua unidade.

As métricas utilizadas para medir as eficiências dos modelos nesse cenário são a acurácia, a sensibilidade, a especificidade, o *recall*, a precisão e o f-*measure*.

6.2.2.2.1 Alarmes gerados pelo FADS

A partir das anomalias inseridos pelo *Scorpius*, podemos avaliar a capacidade do FADS em detectar e gerar alarmes corretamente. O modelo só dispara um alarme ao administrador da rede se ao menos três atributos forem considerados anômalos pelo modelo num mesmo intervalo de tempo. A Figura 35(a) mostra por meio da área verde, todos os atributos atacados com DoS, DDoS e *flash crowd* e pela linha azul, os DSNSFs gerados pelo FADS. Na Figura 35(b) tem-se representados pela área em vermelho, os alarmes gerados pelo FADS para cada um dos atributos analisados.

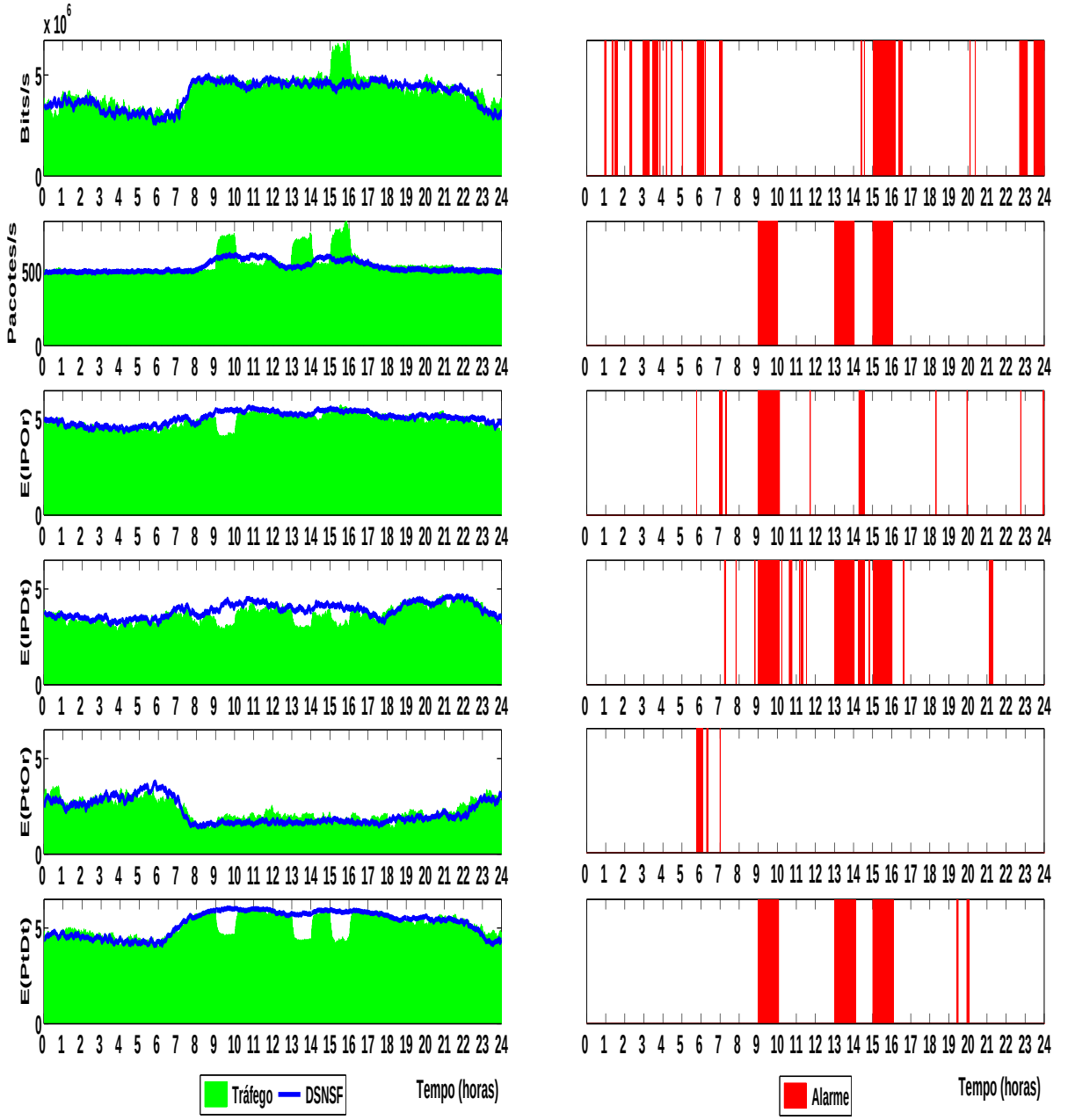


Figura 35 – Alarmes gerados pelo FADS para cada atributo de 23 de setembro de 2015

Pela Figura 36, observa-se o alarme geral fornecido pelo modelo FADS. Esses alarmes só são gerados quando se tem, num mesmo intervalo de tempo, três ou mais atributos gerando alarmes.

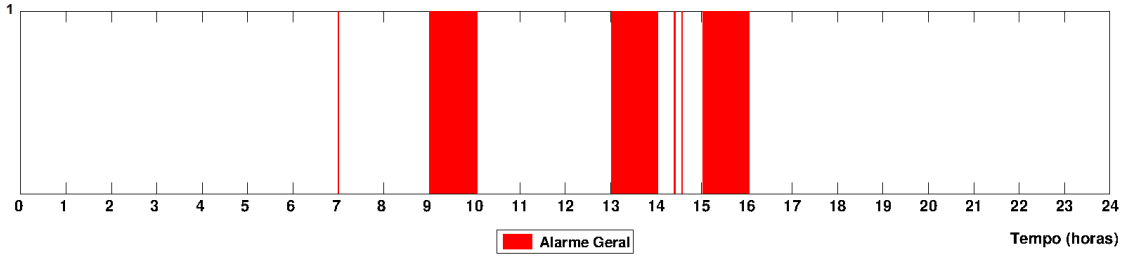


Figura 36 – Alarme geral do FADS para 23 de setembro de 2015

Pode-se observar que o modelo FADS foi capaz de identificar os ataques nos momentos em que eles ocorreram, com poucos falsos positivos.

Na sequência, verificar-se-á a resposta do EWMA para o mesmo dia atacado.

6.2.2.2.2 Alarmes gerados pelo EWMA

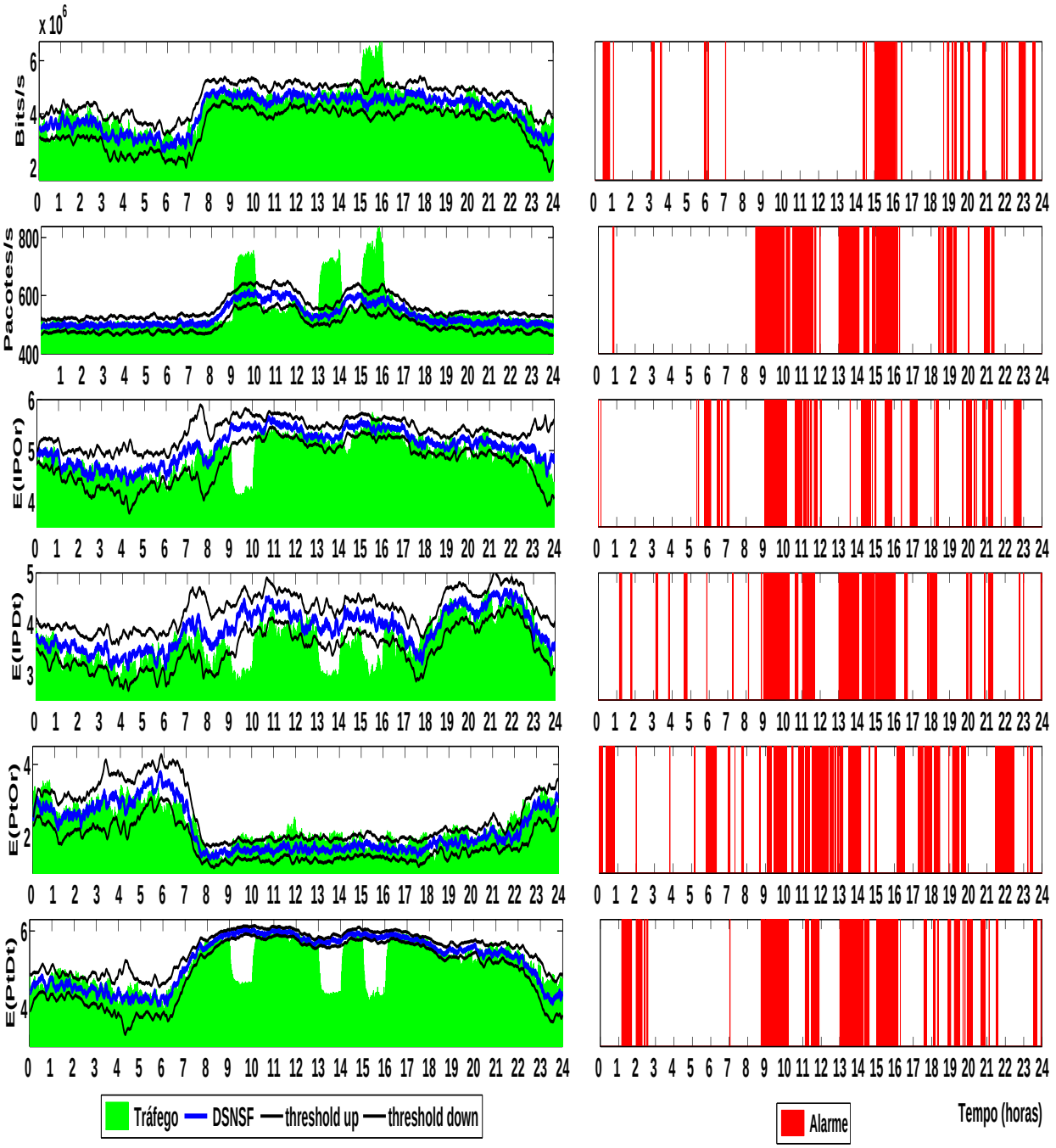


Figura 37 – Alarmes gerados pelo EWMA para cada atributo de 23 de setembro de 2015

Na Figura 37 pode ser observado na área verde, o tráfego com as anomalias; pela linha azul, o perfil do tráfego e em preto, os *tresholds* superior e inferior. O tráfego que

ultrapassou o *tresholds* é interpretado pelo EWMA como anomalia e um alarme é gerado. Esses alarmes podem ser vistos pela área em vermelho.

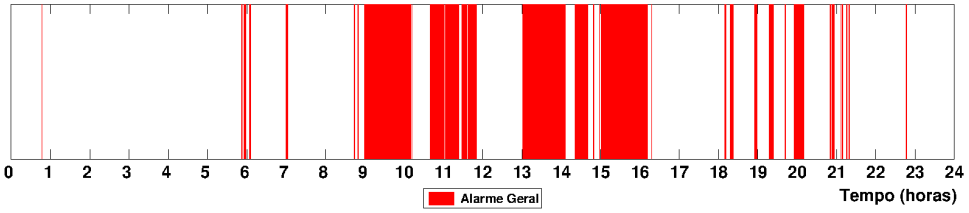


Figura 38 – Alarme geral do EWMA para 23 de setembro de 2015

Na Figura 38, observa-se os alarmes que forma disponibilizados ao administrador em resposta às anomalias detectadas. Esses alarmes, assim como no modelo FADS, só são gerados quando se tem, num mesmo intervalo de tempo, três ou mais atributos gerando alarmes.

Pode ser visualizado que o EWMA gerou muitos outros alarmes além dos esperados em comparação com a resposta do FADS frente aos mesmos ataques. Os resultados das métricas utilizadas para comparar o desempenho de cada um dos modelos analisados são apresentados a seguir.

6.2.2.2.3 Comparação dos modelos FADS e EWMA

Para analisar o desempenho dos modelos usados nesse cenário, fez-se uso de várias métricas. Uma delas, representada pela Tabela 14, é a taxa de acerto por tipo de ataque. Essa tabela informa a quantidade de acertos que cada modelo conseguiu detectar por tipo de anomalia. Verifica-se que o EWMA se saiu melhor neste quesito, detectando o período total de cada ataque gerado pelo *Scorpius*. Já o FADS, obteve 100% de acertos na detecção do DoS e DDoS, ficando com 98,3% de precisão para o *flash crowd*.

Tabela 14 – Taxa de acerto por tipo de ataque - EWMA x FADS - cenário 2

	DoS	DDoS	Flash Crowd
FADS	**100,0%**	**100,0%**	**98,3%**
EWMA	**100,0%**	**100,0%**	**100,0%**

A Tabela 15 apresenta as taxas de alarmes geradas pelos modelos. Observa-se que o EWMA gerou 100% de alarmes verdadeiros positivos contra 99,44% do FADS. Com relação aos alertas para verdadeiros negativo, o FADS foi melhor, com 98,97% contra 87,14% do EWMA. Na taxa de falsos positivos o FADS conseguiu melhores resultados com apenas 0,95% contra 12,86%. Na geração de alarmes falsos negativos teve-se o EWMA com 0% e o FADS com 0,56%.

Tabela 15 – Taxa de alarmes - EWMA x FADS - cenário 2

	VP	VN	FP	FN
FADS	**99,44%**	**98,97%**	**0,95%**	**0,56%**
EWMA	**100,00%**	**87,14%**	**12,86%**	**0,00%**

Com relação ao desempenho dos modelos, a Figura 39 apresenta os resultados das várias métricas empregadas nesse cenário. Podemos observar que ambos os modelos obtiveram altas taxas de acurácia, acima de 88%, sendo que o FADS atingiu 99,1%. Analisando os outros índices, pode-se observar que as taxas foram equilibradas para a sensibilidade e *recall*. No quesito especificidade, o FADS superou o EWMA com 99% contra 87,1%. A precisão indicou uma grande vantagem do FADS com 93,7% e EWMA 52,6%. Essa vantagem foi mensurada pelo *f-measure* em 96,5% para o FADS contra 69% do EWMA.

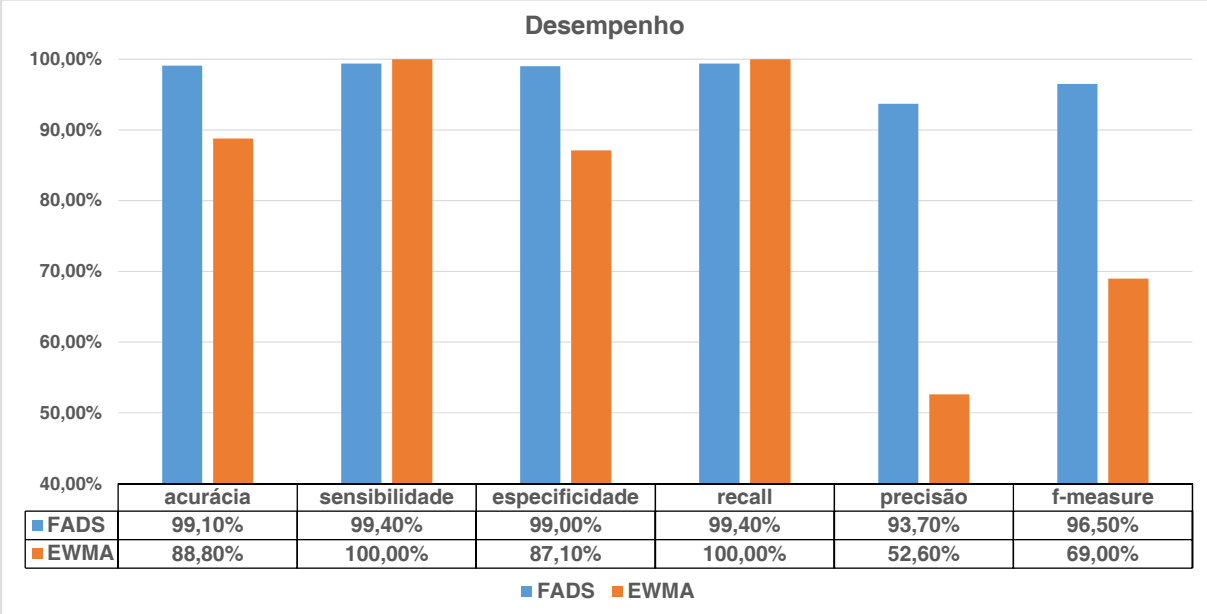


Figura 39 – Desempenho do FADS x EWMA - cenário 2

6.2.3 Cenário 3

Nesse cenário, testou-se a eficiência do modelo FADS em detectar anomalias no tráfego de rede, em um ambiente diferente do cenário anterior. Para isso, escolheu-se o ambiente de rede da UTFPR, pois essa rede apresenta diferentes características com relação à da UEL. As diferenças se dão pelo fato da rede da UTFPR possuir um número menor de equipamentos e de acessos à rede, bem como o protocolo de exportação de fluxos e a amostragem de pacotes possuírem características distintas dos usados pela UEL.

Os testes iniciaram com a caracterização do perfil normal do tráfego, que se corretamente elaborado permite ao FADS uma detecção eficaz do tráfego anômalo da

rede.

Na geração dos DSNSFs, o FADS utiliza a mesma técnica abordada nos cenários anteriores, que é a de utilizar os dados dos mesmos dias das três semanas anteriores à data analisada. Nesse cenário, a data escolhida e que serviu de base para as análises propostas foi o dia 23 de abril de 2013. Essa data foi escolhida por conter menos erros em seu banco de dados histórico. Para a caracterização dos atributos do tráfego do dia 23, o FADS usou para treinamento os dados referentes aos dias 2, 9 e 16 de abril de 2013. Essa caracterização foi avaliada pelas métricas NMSE e CC.

Para a detecção de anomalias foram injetados fluxos de pacotes, utilizando o *Scorpius*, nos dados da UTFPR referentes ao dia 23. Esses fluxos produzem os tráfegos anômalos nessa rede. Após esse procedimento, os dados são submetidos à análise de acurácia, especificidade, *f-measure*, precisão e *recall*.

6.2.3.1 DSNSFs

A Figura 40 apresenta o DSNSF gerado para cada um dos atributos analisados. O tráfego real está representado pela área em verde e o DSNSF, gerado para cada atributo do fluxo, tem sua identificação pela linha em vermelho. É possível observar que o FADS conseguiu caracterizar o tráfego da rede mesmo apresentando variações significativas de volume em seu tráfego, como pode ser observado pelos atributos bit/s e pacotes/s. Na caracterização dos atributos qualitativos, o modelo também apresentou resultados coerentes ao tráfego real.

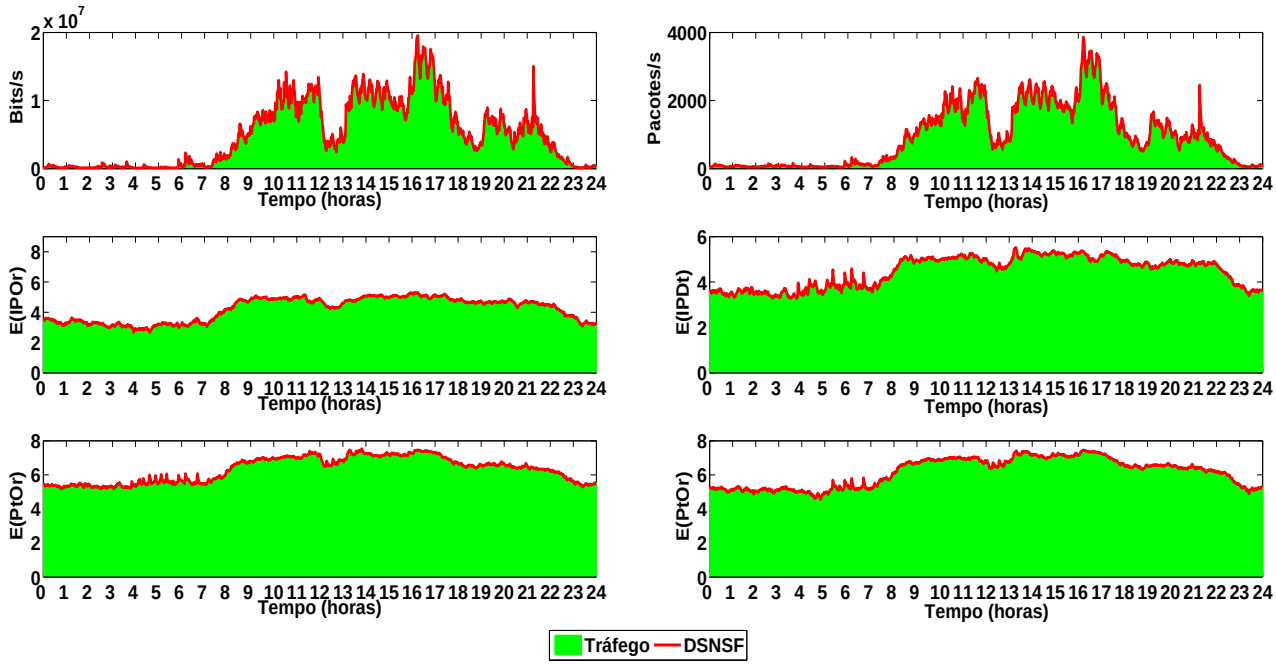


Figura 40 – DSNSFs de cada atributo do dia 23 de abril de 2013 - FADS - UFTPR

Para mensurar a capacidade do FADS em reconhecer as características do tráfego da UTFPR, utilizou-se o NMSE e o CC. As Figuras 41 e 42, ilustram os resultados obtidos para cada um dos atributos analisados.

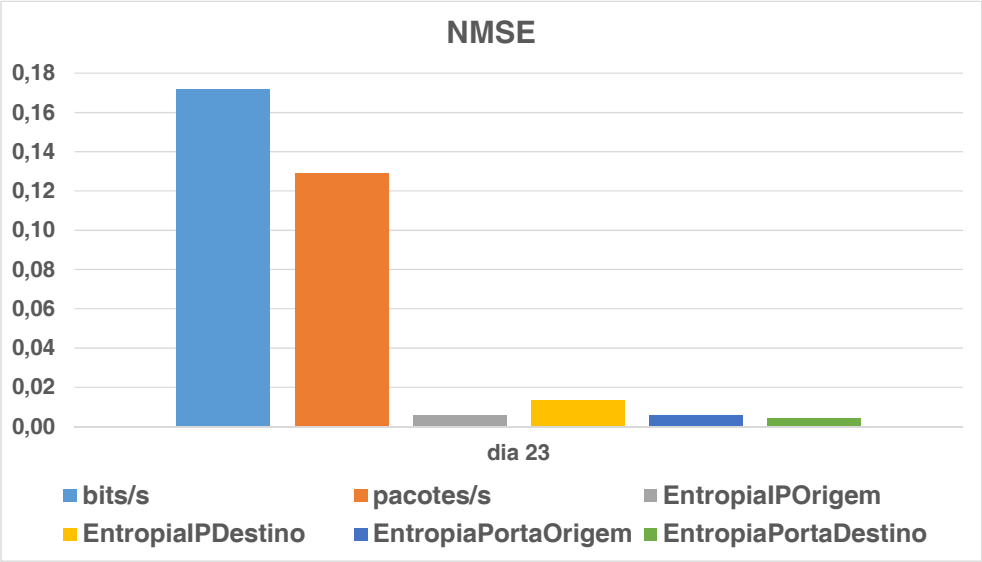


Figura 41 – NMSE para o dia 23 de abril de 2013 - FADS - UTFPR

Os resultados mostrados pela Figura 41 indicam que os erros apresentados pelo NMSE, relacionando o DSNSF de cada atributo com o seu tráfego real, ficou abaixo de 0,18, sendo que para as entropias os valores se encontram abaixo de 0,02. Esses erros maiores refletidos pelos atributos bit/s e pacotes/s se deram pelas características do tráfego desse dia, que tiveram grandes oscilações.

A Figura 42 ilustra os resultados das correlações. Observa-se que os valores se encontram acima de 0,9 para todos os atributos, bem próximo do ideal que é 1, indicando uma excelente correlação entre o DSNSF e o tráfego real.

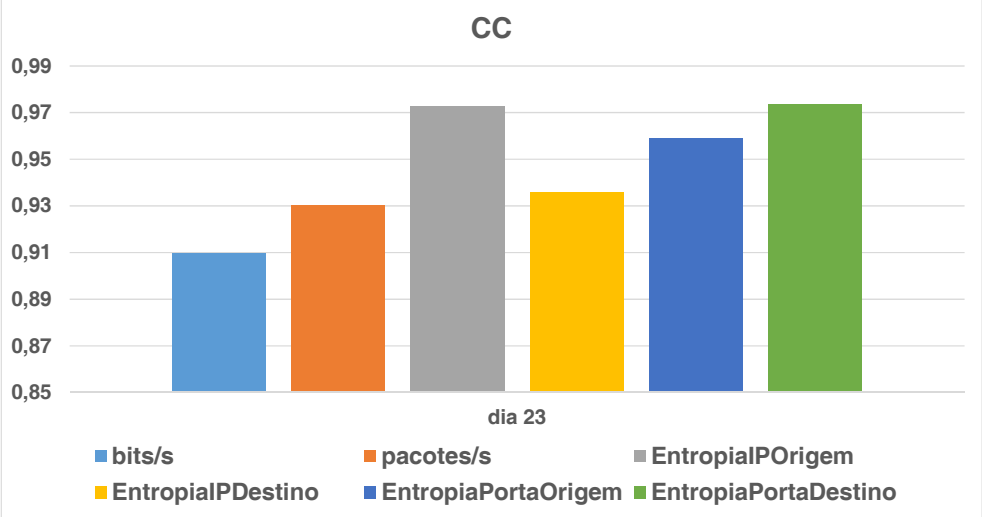


Figura 42 – CC para o dia 23 de abril de 2013 - FADS - UTFPR

As informações resultantes do NMSE e CC leva à conclusão de que o FADS conseguiu caracterizar o tráfego da UTFPR com eficiência. Essa caracterização correta do tráfego permite obter bons resultados na detecção de anomalias no tráfego dessa rede.

6.2.3.2 Detecção de anomalias usando o FADS

Objetivando testar a eficiência do modelo FADS, foram usadas anomalias sintéticas com a finalidade de simular comportamentos anômalos na rede. Foram geradas três anomalidades com características, amplitudes e horários diferentes, como mostra a Tabela 16.

Tabela 16 – Descrição das Anomalias utilizadas no cenário 3

Ataque	Horário	IP de Origem	Porta de Origem	IP de Destino	Porta de Destino
DoS	das 10h às 11h	179.58.43.56	3397	120.122.44.22	2096
DDoS	das 13h às 14h	5 IPs		42.194.55.105	8839
Flash Crowd	das 16h às 17h	10 IPs		190.225.135.730	2968

Esses três eventos anômalos são combinados ao tráfego coletado da UTFPR com a finalidade de testar se o FADS é capaz de reconhecer essas anomalias. Os alarmes gerados pelo modelo só acontecem se ao menos três atributos forem considerados anômalos, num mesmo intervalo de tempo. Para a geração dos alarmes, o modelo continuou a usar os valores dos pesos e limites de *thresholds*, ajustados durante seu aprendizado com os dados do tráfego da UEL do dia 27 de julho de 2015.

Observa-se na Figura 43(a) todos os atributos alterados pelos DoS, DDoS e *flash crowd*, representados pela cor verde e os DSNSFs, pela linha azul. Em vermelho, na Figura 43(b), estão sinalizados os alarmes gerados pelo FADS para cada um dos atributos analisados.

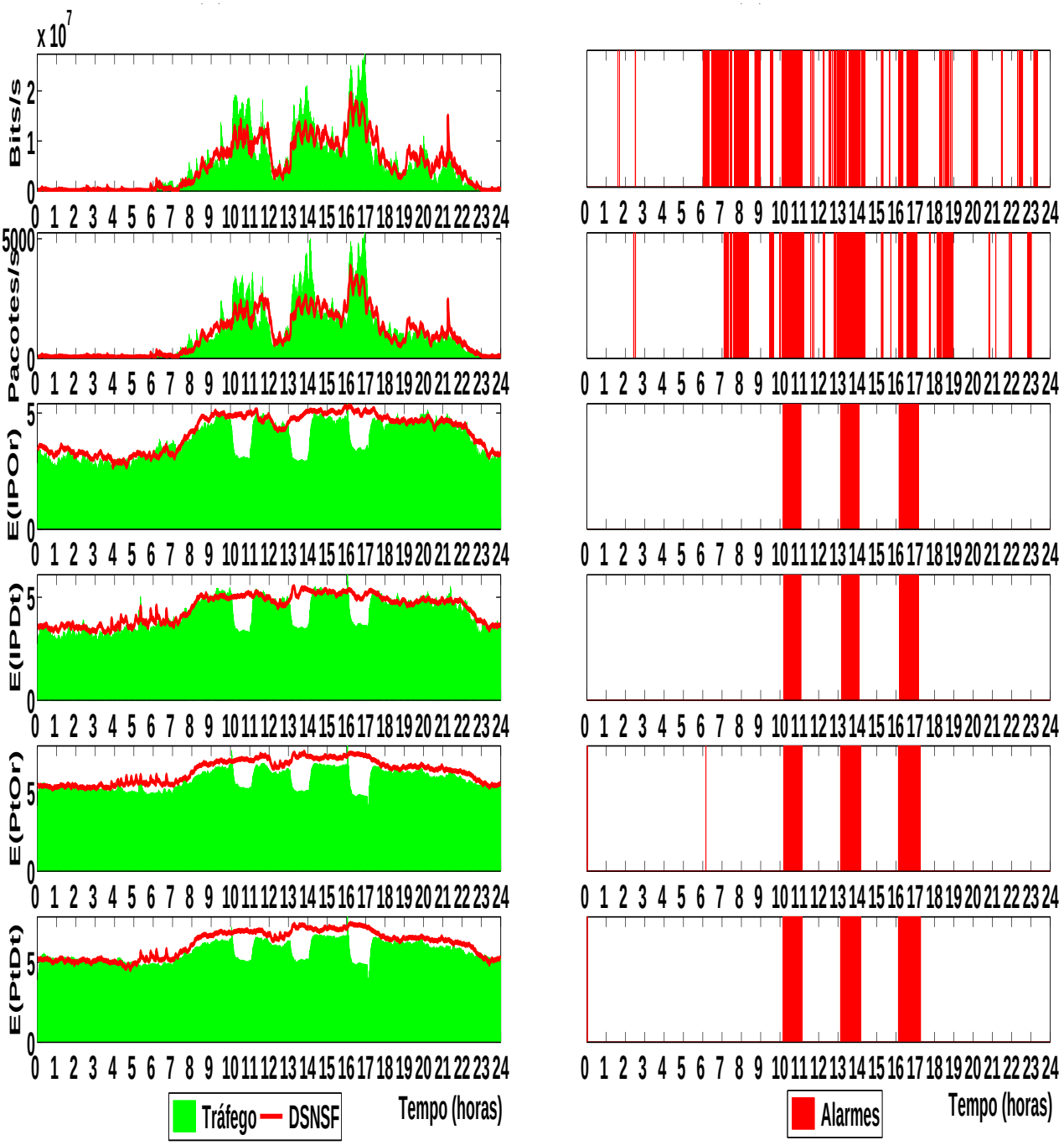


Figura 43 – Alarmes gerados pelo FADS para cada atributo de 23 de abril de 2015 - UTFPR

Devido às características dos dados da UTFPR que se diferenciam do da UEL pelo protocolo de exportação e amostragem, os resultados da inserção das anomalias pelo *Scorpius* produziram alterações não só nos horários definidos para o cenário, mas em todo

o segmento. Isso desencadeou a geração de vários alarmes falsos positivos pelo FADS, que podem ser observadas nos atributos bits/s e pacotes/s.

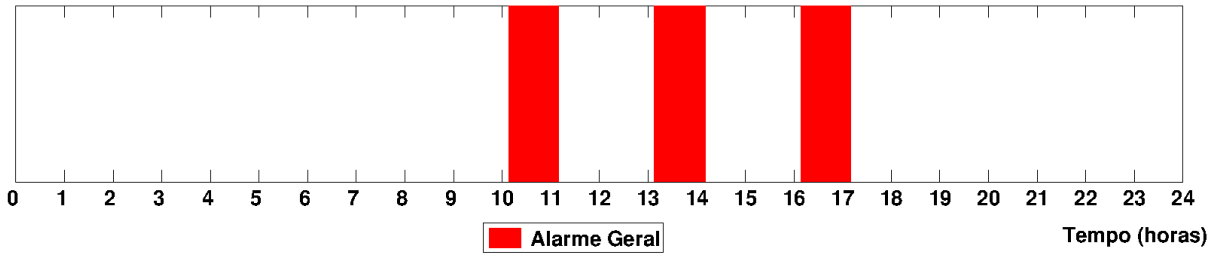


Figura 44 – Alarme geral do FADS para 23 de abril de 2015 - UTFPR

Pela Figura 44, pode ser observado o alarme geral fornecido pelo modelo FADS. Nota-se que o modelo foi capaz de identificar as anomalias nos momentos em que eles ocorreram sem nenhum falso positivo.

A Figura 45 mostra a quantidade de acertos do modelo por tipo de anomalia. Observa-se que o FADS foi capaz de identificar mais de 86% dos dados anômalos, tendo seu maior êxito na identificação do DDoS com 88,3%.

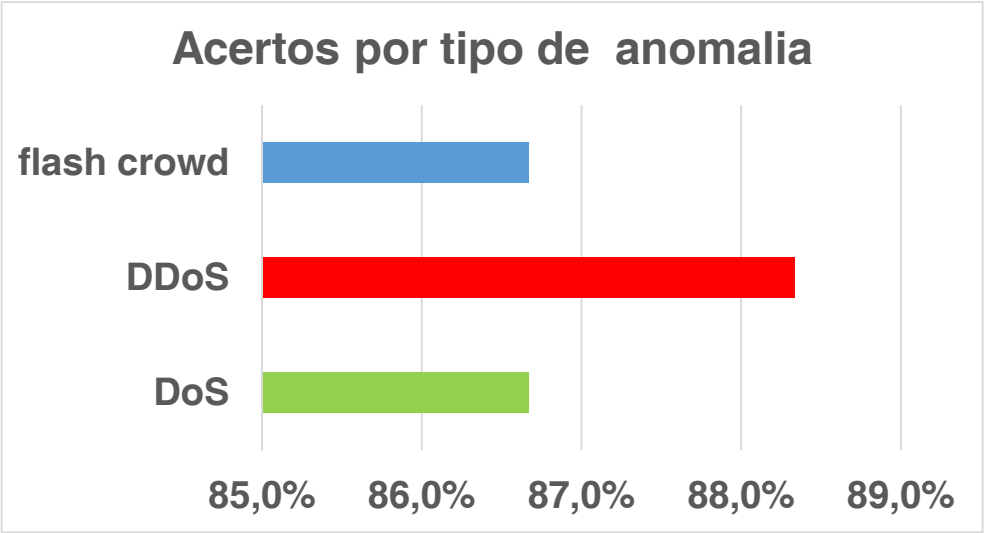


Figura 45 – Acertos por tipo de anomalia - FADS - UFTPR

Pode-se observar na Tabela 17 todas as seis métricas utilizadas na avaliação do FADS.

Tabela 17 – Eficiência do FADS - UFTPR

acurácia	96,3%
sensibilidade	87,2%
especificidade	97,6%
recall	87,2%
precisão	84,4%
f-measure	85,7%

Nota-se um valor para a acurácia acima de 96%, o que indica um ótimo desempenho em gerar resultados corretos. Para as taxas de especificidade e precisão foram obtidos 97,6% e 84,4% respectivamente. A taxa do *recall* foi de mais de 87%, indicando um alto nível de acerto de anomalias. No quesito *f-measure*, o modelo também obteve bons resultados, de 85,7%.

Assim como no cenário anterior, o FADS mostrou-se eficaz em identificar o tráfego anômalo, com valores de verdadeiro positivo (VP) em 86,7% e uma taxa de falso positivo (FP) de apenas 2,3%.

6.2.4 Cenário 4

O quarto cenário tem o propósito de avaliar o desempenho dos modelos EWMA, *Fuzzy*-GADS e FADS na detecção de anomalias em redes de computadores. Para isso, são utilizados dados obtidos da Universidade Estadual de Londrina, referentes ao mês de agosto de 2015.

As caracterizações do tráfego da rede geradas pelo FADS, que segue a mesma metodologia usada nos cenários anteriores, serve de auxílio à detecção de anomalias pelo modelo EWMA. Esses modelos também usaram as mesmas configurações, valores de pesos, *thresholds* e decisões, na geração de alarmes adotadas anteriormente nos ambientes de testes.

As inserções sintéticas dos fluxos de pacotes utilizaram a mesma ferramenta *Scorpius*, sendo usado o dia 28 de agosto de 2015 para recebê-los.

6.2.4.1 DSNSFs

Para a geração dos perfis da rede do dia 28 de agosto de 2015, foram usados os dados do banco histórico referentes aos dias 7, 14 e 21 de agosto de 2015. Os resultados dessa operação podem ser visualizados na Figura 46.

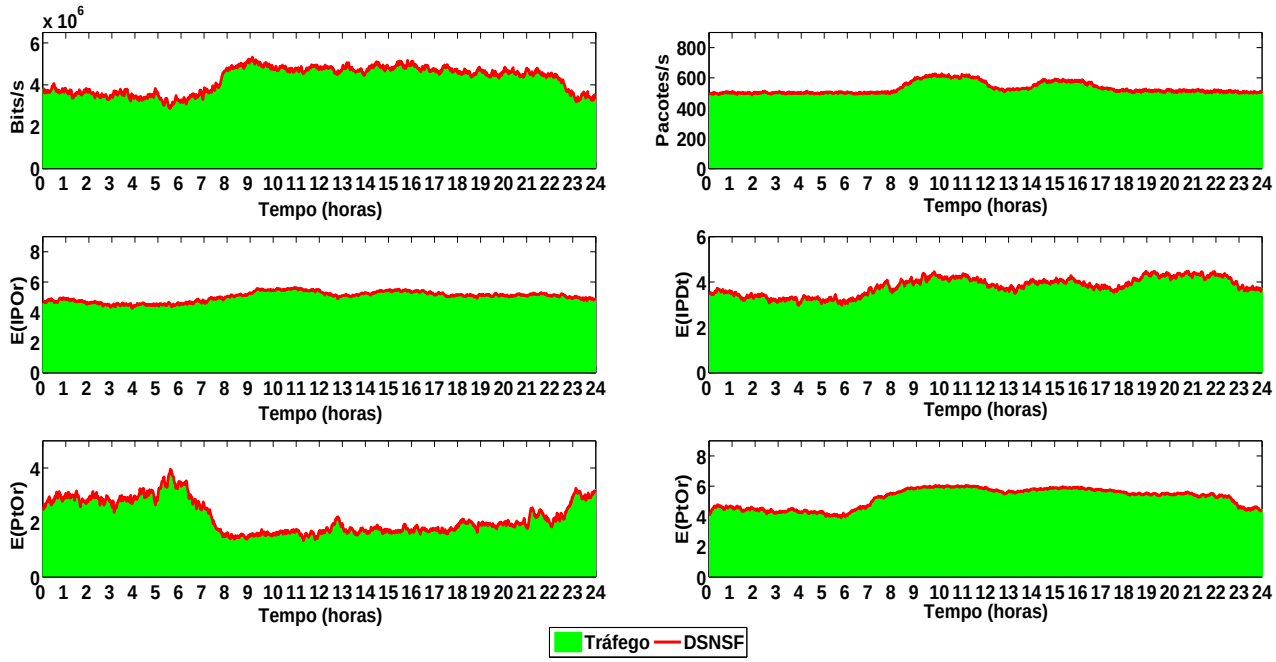


Figura 46 – DSNSFs para o dia 28 - FADS - UEL

Nota-se na Figura 46 o tráfego real em verde e os perfis gerados pelo FADS em vermelho. Observa-se também que todos os DSNSFs acompanham o movimento do tráfego real durante todo o dia, tanto nos horários em que o tráfego é maior, compreendido entre 8h e 22h, quanto nos horários de menor utilização da rede, das 22h às 8h.

A eficiência do FADS para o dia 28 pode ser vista pelas Figuras 47 e 48, que ilustram os resultados das métricas NMSE e CC utilizadas.

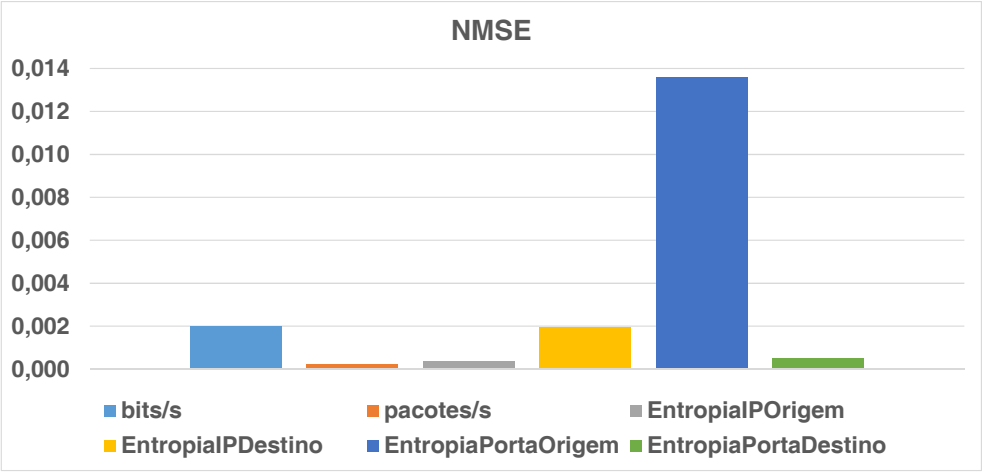


Figura 47 – NMSE paras o dia 28 de agosto de 2015 - FADS - UEL

Pelos resultados do NMSE, conclui-se que o FADS conseguiu predizer com uma ótima precisão o tráfego real do dia 28, obtendo taxas menores que 0,002 para quase todos os atributos, ficando somente a entropia de porta de origem próxima de 0,012.

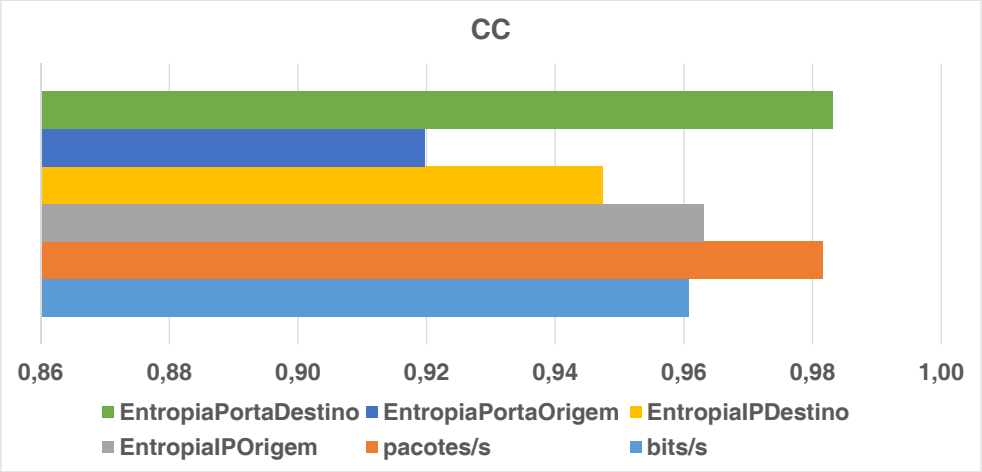


Figura 48 – CC para o dia 28 de agosto de 2015 - FADS - UEL

Os resultados para os coeficientes de correlação desse cenário foram melhores que os obtidos nos anteriores onde foram analisados também dados da UEL. Pode-se observar valores maiores que 0,92 para todos os atributos.

Com esses resultados de NMSE e CC, pode-se esperar bons resultados na detecção de anomalias pelos modelos EWMA e FADS, que usam esses perfis como um dos elementos decisórios nas gerações dos alarmes.

6.2.4.2 Detecção de anomalias

Assim como nos cenários anteriores, o tráfego anômalo foi injetado fazendo-se uso da ferramenta *Scorpius*. O dia que recebeu essas anomalias e que foi analisado pelos modelos propostos foi o 28 de agosto de 2015, uma sexta-feira. A Tabela 18 informa como foram administradas essas anomalias no segmento analisado.

Tabela 18 – Descrição das anomalias utilizadas no cenário 4

Ataque	Horário	IP de Origem	Porta de Origem	IP de Destino	Porta de Destino
DoS	das 8h às 9h	118.166.96.200	6103	94.102.236.112	8950
DDoS	das 13h às 14h	5 IPs		253.7.232.14	1964
Flash Crowd	das 16h às 17h	10 IPs		138.123.133.181	2756

A seguir, são analisadas as respostas dos modelos EWMA, FADS e *Fuzzy*-GADS frente à essas anomalias.

6.2.4.2.1 EWMA

Os resultados da detecção de anomalias usando o EWMA com o ponto de corte adotado em L=3,5 são apresentados na Figura 49, onde observa-se pela área em vermelho,

os alarmes gerados para cada um dos atributos analisados. Foram gerados vários alarmes fora dos horários em que realmente aconteceram as anomalias inseridas pelo *Scorpius*, porém, em função da regra adotada pelo modelo em que o alarme geral só é produzido quando ocorrem três ou mais atributos gerando alarmes num mesmo segmento da rede, esses falsos positivos puderam ser bem administrados, como é ilustrado pela Figura 50, onde foram constatados vários alarmes esperados e poucos indesejados.

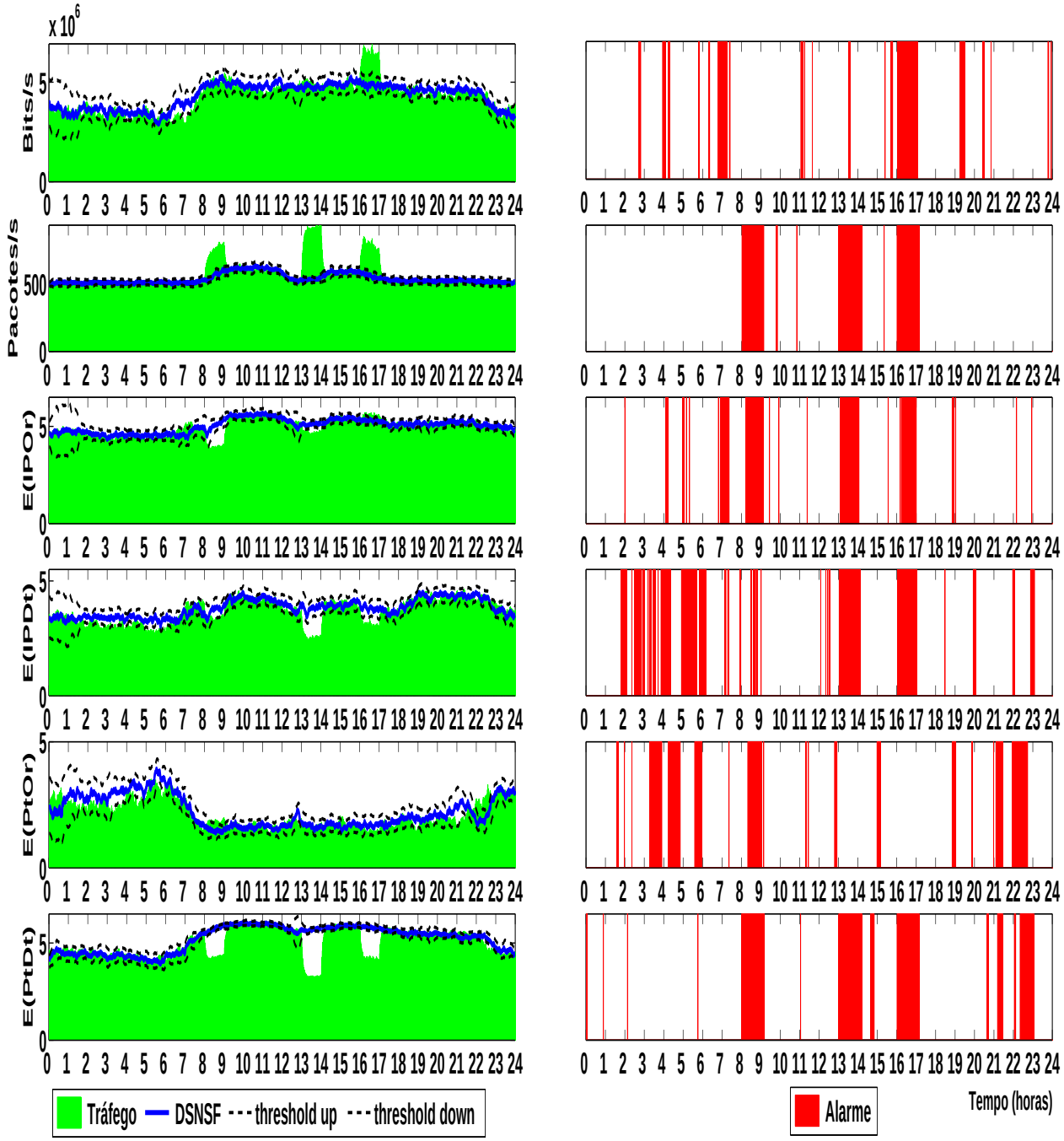


Figura 49 – Alarmes gerados pelo EWMA para cada atributo do 28 de agosto de 2015 - UEL

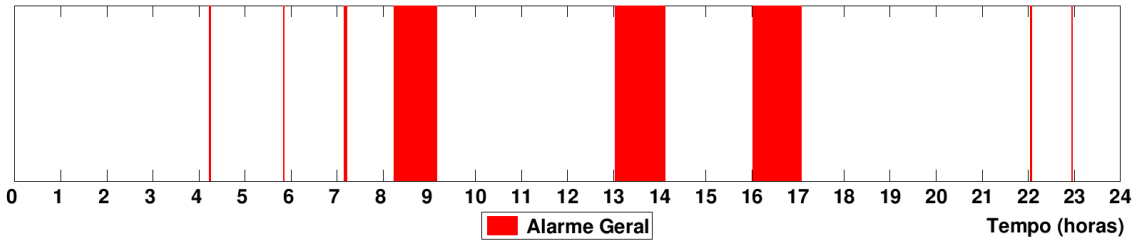


Figura 50 – Alarme geral EWMA para o dia 28 de agosto de 2015 - UEL

6.2.4.2.2 FADS

Na Figura 52 são apresentados os resultados por atributo dos alarmes gerados pelo FADS. Esses alarmes estão sinalizados em vermelho. Assim como o EWMA esse modelo acusou anomalias também fora dos segmentos alterados pelo *Scorpius*, porém, em número bem menor.

Pela Figura 51, observa-se o alarme geral fornecido pelo modelo. Esses alarmes só são gerados quando se tem, num mesmo intervalo de tempo, três ou mais atributos gerando alarmes. Essa técnica permitiu ao modelo isolar os alarmes falsos positivos dos atributos e produzir uma sinalização mais precisa na análise das anomalias nesse segmento analisado.

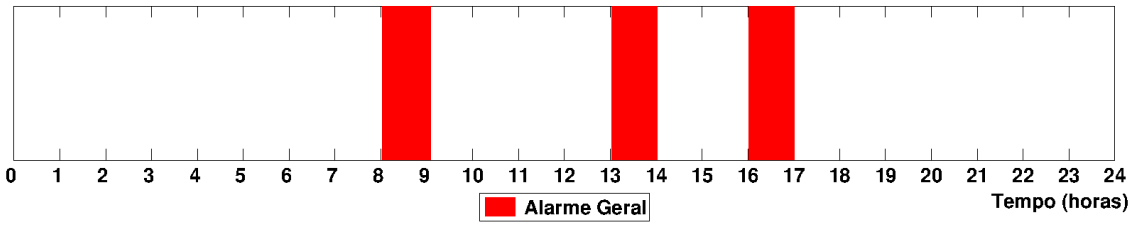


Figura 51 – Alarme geral do FADS para o dia 28 de agosto de 2015 - UEL

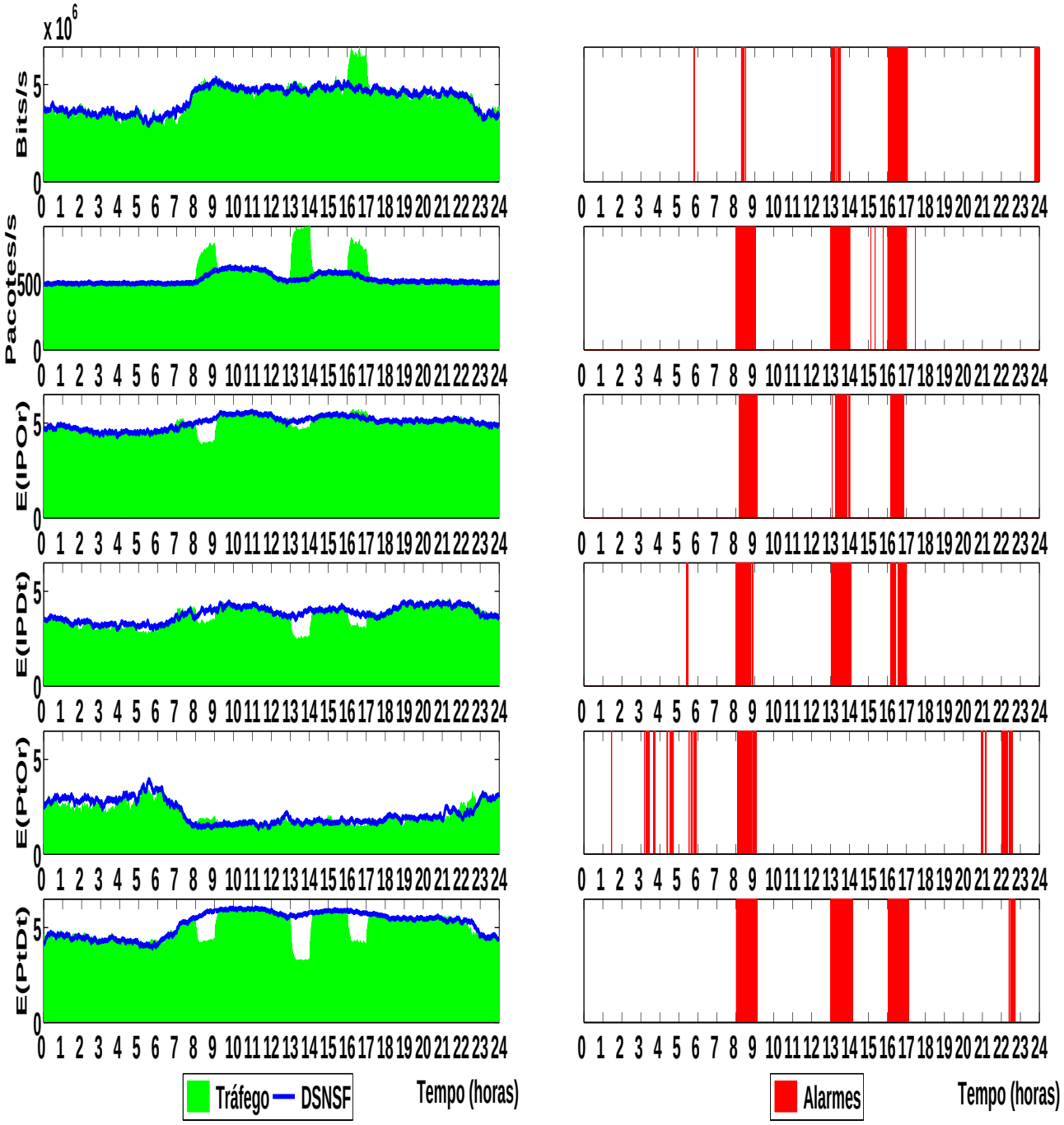


Figura 52 – Alarmes gerados pelo FADS para cada atributo do dia 28 de agosto de 2015 - UEL

6.2.4.2.3 *Fuzzy*-GADS

Para a detecção de anomalias utilizando o modelo *Fuzzy*-GADS, são necessários inicialmente gerar os perfis do dia 28 de agosto de 2015 de cada um dos atributos estudados. Para isso, o GA utilizou os dados dos mesmos dias das quatro semanas anteriores à data analisada. Esses dias são 31 de julho e 7, 14 e 21 de agosto, referentes ao ano de 2015. Esse modelo também utiliza o EWMA em seu processo de detecção de anomalias e seus valores foram obtidos utilizando o DSNSF de cada atributo e o tráfego

real correspondente. O parâmetro L que define a amplitude dos limiares do EWMA foi definido como 3. Os resultados dessa operação podem ser visualizados pela Figura 53, onde observa-se o tráfego anômalo pela área em verde, o DSNSF pela linha em vermelho e a área dos *thresholds* em azul.

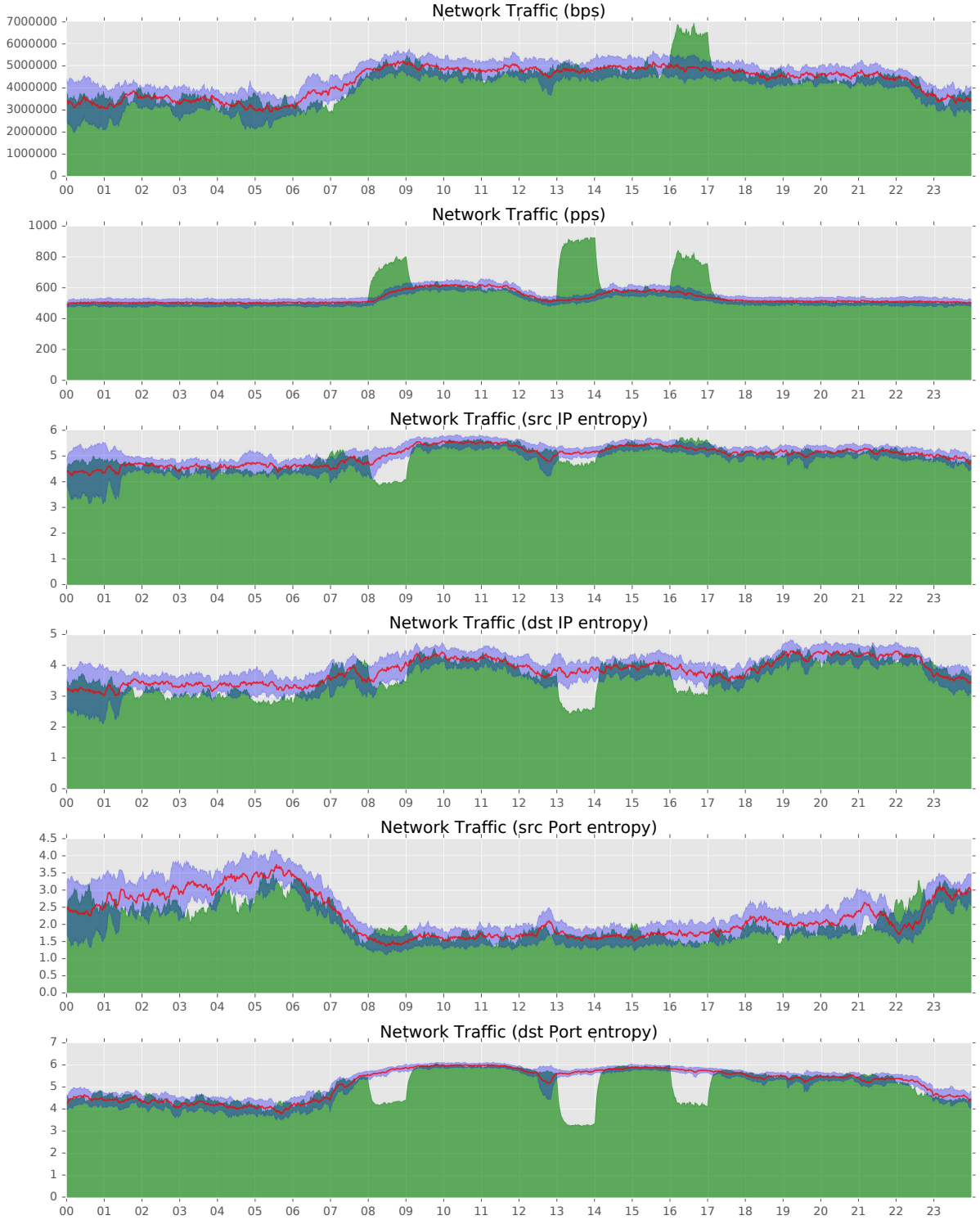


Figura 53 – Tráfego, DSNSF e *thresholds* do dia 28 de agosto de 2015 - *Fuzzy*-GADS [UEL]

Para calcular os graus de pertinência usando a função de pertinência Gaussiana, os valores do tráfego de rede, DSNSF e *thresholds* são utilizados como parâmetros. A Figura 54 mostra os graus de pertinência de todos os seis atributos do segmento analisado.

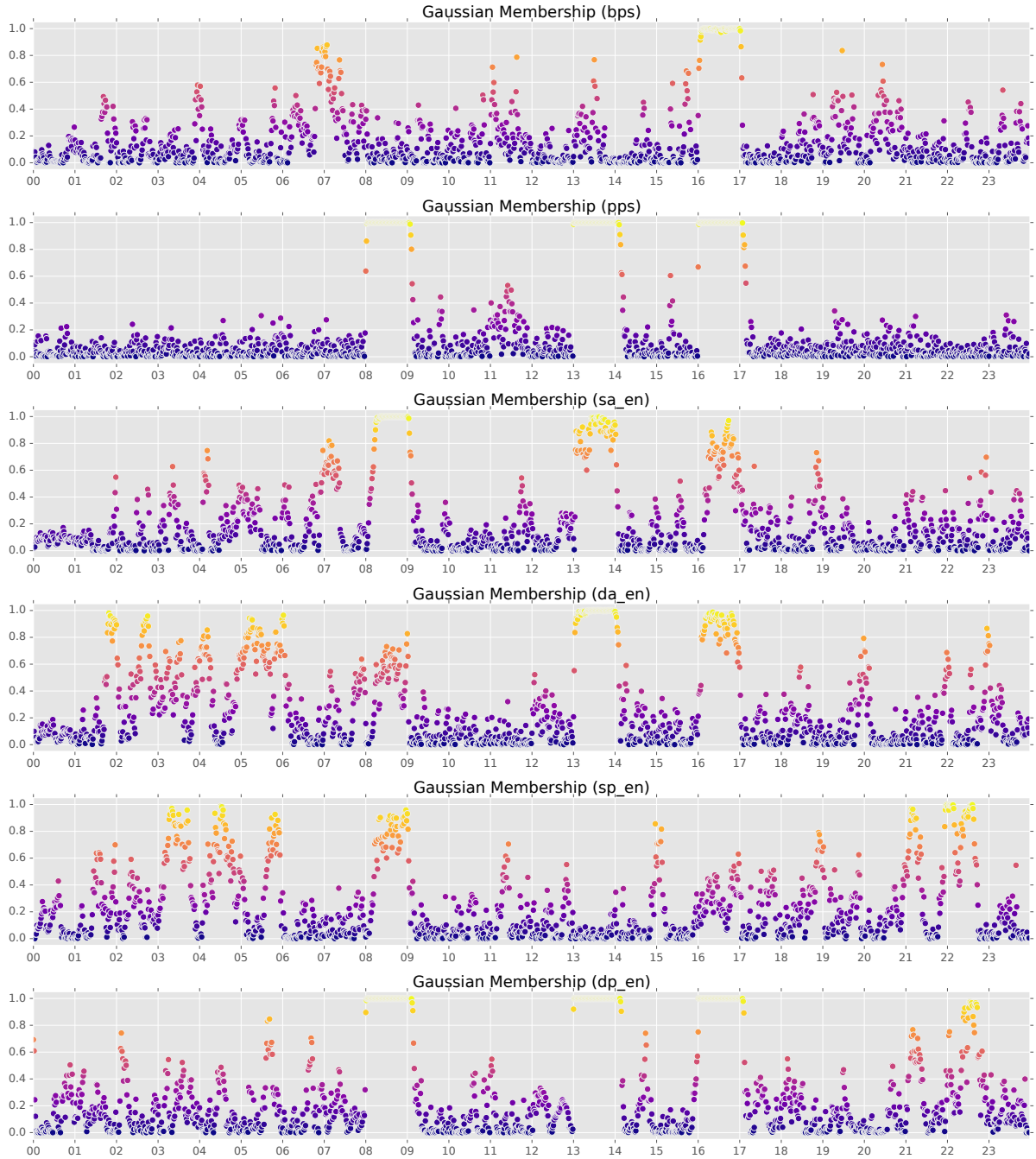


Figura 54 – Graus de pertinência dos atributos para o dia 28 de agosto de 2015. *Fuzzy*-GADS [UEL]

Para que os alarmes sejam gerados, a soma dos valores dos graus de pertinência de 5 atributos, para cada instante analisado, deve ser maior do que o valor de Γ=3,5. Essa definição do valor de atributos anômalos necessários para gerar um alarme foi obtida analisando o estudo de Assis et al. [36].

A resposta do modelo *Fuzzy*-GADS em detectar as anomalias inseridas no tráfego do dia 28 de gosto de 2015 está ilustrada pela Figura 55. Em verde estão representados os instantes em que o modelo não detectou anomalias e em vermelho, as anomalias detectadas.

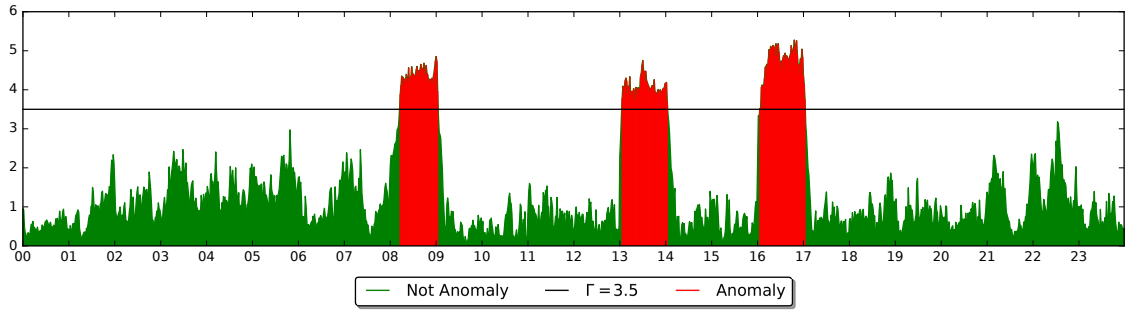


Figura 55 – Alarme geral *Fuzzy*-GADS para o dia 28 de agosto de 2015 - UEL

Com o objetivo de avaliar o FADS com outro modelo, foi solicitado que Hamamoto [37], que é o autor do *fuzzy*-GADS, realizasse testes e fornecesse dados conforme apresentado nas figuras 53, 54 e 55.

6.2.4.3 Eficiência dos modelos EWMA, FADS e *Fuzzy*-GADS

Os resultados gerados pelos modelos propostos nesse cenário, foram avaliados com a finalidade de mensurar a eficiência de cada um na detecção das anomalias apresentadas.

Na Figura 56 pode-se verificar pela taxa de acurácia obtida, a probabilidade de cada modelo em gerar resultados corretos. Todos os modelos conseguiram boas taxas, acima de 96%, sendo que o modelo com maior confiabilidade foi o FADS.

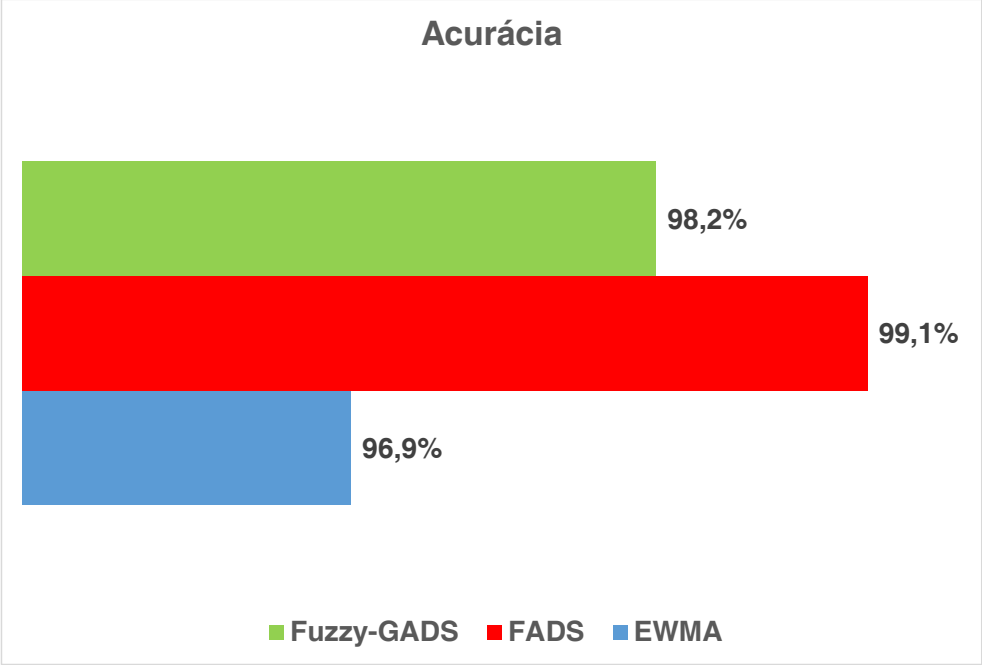


Figura 56 – Acurácia EWMA x FADS x *Fuzzy*-GADS para o dia 28 de agosto de 2015 - UEL

Os valores referentes ao *recall* e à precisão são mostrados na Tabela 19.

Tabela 19 – *Recall* e Precisão - cenário 4

	recall	precisão
EWMA	**91,7%**	**85,1%**
FADS	**94,4%**	**98,0%**
Fuzzy-GADS	**89,6%**	**96,4%**

Observa-se que o FADS teve uma melhor eficiência em detectar as anomalias no tráfego em relação às outras técnicas. Com relação ao *recall*, o melhor resultado foi o obtido pelo FADS em 94,4% e o *Fuzzy*-GADS, obteve o menor valor de 89,6%. Na precisão, o FADS também obteve o melhor resultado de 98%, ficando o EWMA em terceiro lugar com 85,1%. Todos os modelos obtiverem taxas excelentes em classificar corretamente as amostras verdadeiro positivo (VP) com uma pequena vantagem para o FADS.

7 CONCLUSÃO

Ainda cabe ao administrador de rede de computadores a responsabilidade de desenvolver soluções visando a qualidade na transmissão de dados, manutenção dos serviços providos e a segurança da rede de forma eficaz e eficiente. Para garantir essa gerência, juntamente com outras ferramentas, faz-se necessário o uso das que auxiliem de uma maneira não supervisionada essa administração como um todo. Um dos pontos cruciais nessa administração é a de poder identificar anomalias na rede, sejam elas oriundas de ataques maliciosos ou simplesmente uma alteração nos fluxos de pacotes que possam causar interrupções ou degradações no desempenho da rede podendo refletir na qualidade dos serviços prestados bem como em prejuízos financeiros.

Neste contexto, este trabalho apresentou a ferramenta *Firefly Algorithm for Digital Signature* (FADS), que fornece aos administradores a possibilidade de detectar anomalias na rede. O FADS trabalha em duas fases. Inicialmente, gera uma caracterização do perfil do tráfego da rede, que é uma previsão do comportamento do tráfego para um dia qualquer, mediante a análise dos dados históricos desta rede. Essa caracterização resulta numa assinatura digital de um segmento de rede, denominada de *Digital Signature of Network Segment using Flow Analysis* (DSNSF).

Para a geração do DSNSF o FADS utilizou o algoritmo firefly (FA). O FA é um algoritmo meta-heurístico baseado no comportamento dos vagalumes e de suas características de luz emitidas. Este comportamento, baseado na inteligência do enxame, permite ao algoritmo encontrar a melhor solução, ou seja, a otimização de uma função objetivo.

A segunda fase do FADS consiste em detectar anomalias no tráfego de rede. Para isso, o modelo utiliza uma técnica inspirada nas redes neurais artificiais (RNA). As RNA são fundamentadas num conjunto de procedimentos computacionais que apresentam um modelo matemático semelhante ao cérebro humano propiciando conhecimento diante da experiência.

Para avaliar a eficácia do FADS foram usados quatro cenários utilizando informações reais obtidas por técnicas de análises de fluxos de dados. Essas informações reais foram coletadas na Universidade Estadual de Londrina (UEL) e na Universidade Tecnológica Federal do Paraná (UTFPR), campus de Toledo. Desses dados, foram analisados seis atributos proporcionando uma análise abrangente do comportamento da rede e de seus recursos.

Nesses cenários, puderam ser testadas as habilidades do FADS em caracterizar o tráfego, gerando os DSNSFs e em detectar anomalias em redes de computadores, com o

ativamento de alarmes.

Para poder comparar a eficiência do FADS, outras três técnicas diferentes e já utilizadas pelo grupo de pesquisa em redes de computadores e comunicação de dados do departamento de computação da UEL foram empregadas. O EWMA foi uma delas, cujo resultado de uma amostra é baseado num percentual de peso das amostras anteriores. A técnica de uso de *thresholds* para delimitar e diferenciar um tráfego anômalo de um normal, também foi utilizada. E por último, foi empregado o *Fuzzy*-GADS, que utiliza o algoritmo genético (GA), inspirado na evolução natural das espécies juntamente com a lógica *fuzzy*, definida como um conjunto de princípios matemáticos para produzir um conhecimento baseado no grau de pertinência dos termos.

No primeiro cenário, analisou-se a propriedade do FADS em construir os DSNSFs dos seis atributos escolhidos, bits/s, pacotes/s, Entropia de IP de Origem, Entropia de IP de Destino, Entropia de Porta de Origem e Entropia de Porta de Destino. Esses atributos foram analisados para uma semana, tendo como fonte de dados a UEL. Os resultados obtidos mostraram um valor médio de NMSE em torno de 0,005 e do CC maior do que 0,91. As diferenças individuais de cada atributo com relação à média foram pequenas. Esses dados deixaram evidente que o FADS foi capaz de caracterizar o tráfego da rede analisada com muita eficiência.

Para o cenário dois, foram utilizadas novas datas com o intuito de testar a capacidade do FADS em detectar anomalias no tráfego da rede e comparar seus resultados com o obtido utilizando a técnica EWMA. Os dados utilizados pertencem à rede da UEL. As anomalias geradas no tráfego da UEL foram inseridas com a ferramenta *Scorpius*, que possui a capacidade de gerar fluxos de pacotes com características de um DoS, DDoS e de um *flash crowd*, em horários pré-definidos. Nesse teste, o FADS superou a técnica EWMA na detecção de anomalias. Seu índice de acurácia foi maior do que 99%, com precisão de 93,7%.

Para testar o FADS na caracterização do tráfego e na detecção de anomalias em redes de computadores num ambiente de rede diferente do usado nos cenários anteriores, foram utilizados os dados da UTFPR, campus de Toledo, comentado no cenário três. Na caracterização do tráfego, o FADS atingiu ótimos resultados, com NMSE menores do que 0,18 e de CC maiores do que 0,91. No quesito anomalias, o modelo atingiu mais de 86% na detecção individual das anomalias geradas pelo DoS, DDoS e *flash crowd* e com valor de acurácia de 96,3%, demonstrando alta taxa de acerto no geral.

No cenário quatro, comparou-se o desempenho de dois outros modelos ao do FADS, o EWMA e o *Fuzzy*-GADS. Foram analisados os resultados da detecção de anomalias utilizando novamente a UEL, porém, com outras datas. Todas as três técnicas conseguiram detectar com boa eficiência as anomalias apresentadas. Em termos de comparação, a acurácia mostrou que a técnica FADS conseguiu superar as outras duas,

obtendo 99,1% contra 98,2% e 96,9%, respectivamente ao *Fuzzy*-GADS e ao EWMA.

As contribuições desse trabalho estão direcionadas na aplicação do algoritmo metaheurístico *Firefly* na mineração de dados, gerando perfis normais do tráfego e em redes neurais, utilizado na detecção de anomalias em redes de computadores.

Assim, o modelo FADS alcançou seu objetivo, provando ser capaz de fornecer aos administradores de rede informações significativas do uso dos recursos da rede. Além de poder alertá-los, fazendo uso de alarmes, sobre anomalias detectadas na rede, provenientes ou não de ações maliciosas.

Como trabalhos futuros será testado o modelo FADS em outros cenários e com outros métodos, verificando a viabilidade de inclusão de novos atributos para serem analisados e ainda, tentar formular mecanismos para mitigar os efeitos das anomalias.

TRABALHOS PUBLICADOS PELO AUTOR

1. Fadir Salmen; Paulo R. Galego Hernandes Jr.; Luiz Fernando Carvalho; Mario Lemes Proença Jr. **Using Firefly and Genetic Metaheuristics for Anomaly Detection based on Network Flows**, The Eleventh Advanced International Conference on Telecommunications (AICT 2015), Junho/2015, p. 113-118, ISBN: 978-1-61208-411-4. Bruxelas, Bélgica. (Qualis CC 2012, B1)

REFERÊNCIAS

[1] CARL, G. et al. Denial-of-service attack-detection techniques. *IEEE Internet Computing*, v. 10, n. 1, p. 82–89, Jan 2006. ISSN 1089-7801.

[2] ZARGAR, S. T.; JOSHI, J.; TIPPER, D. A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. *IEEE Communications Surveys Tutorials*, v. 15, n. 4, p. 2046–2069, Fourth 2013. ISSN 1553-877X.

[3] KANDULA, S. et al. Botz-4-sale: Surviving organized ddos attacks that mimic flash crowds. In: USENIX ASSOCIATION. *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*. [S.l.], 2005. p. 287–300.

[4] CLAISE, B. RFC 5101, Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. *IETF RFC*, 2008. Disponível em: <http://tools.ietf.org/html/rfc5101>.

[5] GARCIA-TEODORO, P.; MUNOZ-FELDSTEDT, P.; RUETE-ZUNIGA, D. Automatic signature generation for network services through selective extraction of anomalous contents. In: *Telecommunications (AICT), 2010 Sixth Advanced International Conference on*. [S.l.: s.n.], 2010. p. 370–375.

[6] PROENCA, M. et al. Baseline to help with network management. In: *e-Business and Telecommunication Networks*. [S.l.]: Springer Netherlands, 2006. p. 158–166.

[7] SHANNON, C. E. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, ACM, v. 5, n. 1, p. 3–55, 2001.

[8] PATCHA, A.; PARK, J.-M. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, v. 51, n. 12, p. 3448 – 3470, 2007. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S138912860700062X>.

[9] PRABHA, K.; SREE, S. S. A survey on ips methods and techniques. *International Journal of Computer Science Issues (IJCSI)*, International Journal of Computer Science Issues (IJCSI), v. 13, n. 2, p. 38, 2016.

[10] YADAV, S. *Integrated network intrusion detection*. Google Patents, 2015. US Patent 9,143,525. Disponível em: <https://www.google.com/patents/US9143525>.

[11] YASSIN, W. et al. Signature-based anomaly intrusion detection using integrated data mining classifiers. In: *Biometrics and Security Technologies (ISBAST), 2014 International Symposium on*. [S.l.: s.n.], 2014. p. 232–237.

[12] LIMA, M. F. et al. Anomaly detection using baseline and k-means clustering. In: IEEE. *Software, Telecommunications and Computer Networks (SoftCOM), 2010 International Conference on*. [S.l.], 2010. p. 305–309.

[13] HALL, J.; BARBEAU, M.; KRANAKIS, E. Anomaly-based intrusion detection using mobility profiles of public transportation users. In: *WiMob'2005), IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, 2005.* [S.l.: s.n.], 2005. v. 2, p. 17–24 Vol. 2. ISSN 2160-4886.

[14] HOFSTEDE, R. et al. Flow monitoring explained: From packet capture to data analysis with netflow and ipfix. *IEEE Communications Surveys Tutorials*, v. 16, n. 4, p. 2037–2064, Fourthquarter 2014. ISSN 1553-877X.

[15] VELAN, P. et al. Network traffic characterisation using flow-based statistics. In: *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. [S.l.: s.n.], 2016. p. 907–912.

[16] PENA, E. H. et al. Anomaly detection using digital signature of network segment with adaptive arima model and paraconsistent logic. In: IEEE. *Computers and Communication (ISCC), 2014 IEEE Symposium on*. [S.l.], 2014. p. 1–6.

[17] KALLITSIS, M. et al. Amon: An open source architecture for online monitoring, statistical analysis, and forensics of multi-gigabit streams. *IEEE Journal on Selected Areas in Communications*, v. 34, n. 6, p. 1834–1848, June 2016. ISSN 0733-8716.

[18] DERI, L. et al. Improving passive packet capture: Beyond device polling. In: AMSTERDAM, NETHERLANDS. *Proceedings of SANE*. [S.l.], 2004. v. 2004, p. 85–93.

[19] ZACARON, A. M. et al. Digital signature of network segment using flow analysis. In: *DCNET/ICE-B/OPTICS'12*. [S.l.: s.n.], 2012. p. 35–40.

[20] GARCÍA-DORADO, J. L. et al. Characterization of isp traffic: Trends, user habits, and access technology impact. *IEEE Transactions on Network and Service Management*, IEEE, v. 9, n. 2, p. 142–155, 2012.

[21] STEEG, D. van der et al. Real-time ddos attack detection for cisco ios using netflow. In: *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. [S.l.: s.n.], 2015. p. 972–977. ISSN 1573-0077.

[22] QIN, X.; XU, T.; WANG, C. Ddos attack detection using flow entropy and clustering technique. In: *2015 11th International Conference on Computational Intelligence and Security (CIS)*. [S.l.: s.n.], 2015. p. 412–415.

[23] DU, Y. et al. A real-time anomalies detection system based on streaming technology. In: IEEE. *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2014 Sixth International Conference on*. [S.l.], 2014. v. 2, p. 275–279.

[24] BOUSSAïD, I.; LEPAGNOT, J.; SIARRY, P. A survey on optimization metaheuristics. *Information Sciences*, v. 237, p. 82 – 117, 2013. ISSN 0020-0255. Prediction, Control and Diagnosis using Advanced Neural Computations. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0020025513001588>.

[25] YANG, X.-S. *"Firefly algorithms for multimodal optimization" , in: Stochastic Algorithms: Foundations and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. v. 5792. (Lecture Notes in Computer Science, v. 5792). ISBN 978-3-642-04943-9.

[26] TALATAHARI, S.; GANDOMI, A. H.; YUN, G. J. Optimum design of tower structures using firefly algorithm. *The Structural Design of Tall and Special Buildings*, Wiley Online Library, v. 23, n. 5, p. 350–361, 2014.

[27] ADANIYA, M. H. et al. Anomaly detection using dsns and firefly harmonic clustering algorithm. In: IEEE. *Communications (ICC), 2012 IEEE International Conference on*. [S.l.], 2012. p. 1183–1187.

[28] HASSANZADEH, T.; VOJODI, H.; MOGHADAM, A. M. E. An image segmentation approach based on maximum variance intra-cluster method and firefly algorithm. In: IEEE. *Natural computation (icnc), 2011 seventh international conference on*. [S.l.], 2011. v. 3, p. 1817–1821.

[29] SHONA, M. D.; SENTHILKUMAR, M. An ensemble data preprocessing approach for intrusion detection system using variant firefly and bk-nn techniques. *International Journal of Applied Engineering Research*, v. 11, n. 6, p. 4161–4166, 2016.

[30] HOLLAND, J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992. ISBN 0262082136.

[31] HERNANDES JR., P. R. G. et al. Digital signature of network segment using genetic algorithm and ant colony optimization metaheuristics. In: *The Eighth International Conference on Emerging Security Information, Systems and Technologies*. [S.l.: s.n.], 2014. p. 62–67. ISBN 978-1-61208-376-6. ISSN 2162-2116.

[32] GRUPO de pesquisa em redes de computadores e comunicação de Dados da UEL. 2014. <http://redes.dc.uel.br/>.

[33] SALMEN, F. et al. Using Firefly and Genetic Metaheuristics for Anomaly Detection based on Network Flows. In: *AICT 2015, The Eleventh Advanced International Conference on Telecommunications*. [s.n.], 2015. p. 113–118. ISBN 978-1-61208-411-4. Disponível em: <http://www.thinkmind.org/index.php?view=article&articleid=aict_2015_6_20_10106>.

[34] TANG, Z.; GUO, Y.; CHEN, X. Self-adaptive step counting on smartphones under unrestricted stepping modes. In: *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*. [S.l.: s.n.], 2016. v. 1, p. 788–797.

[35] CARVALHO, L. F. et al. Ant colony optimization for creating digital signature of network segments using flow analysis. In: *SCCC*. [S.l.: s.n.], 2012. p. 171–180.

[36] ASSIS, M. V. de; RODRIGUES, J. J.; PROENÇA, M. L. A seven-dimensional flow analysis to help autonomous network management. *Information Sciences*, Elsevier, v. 278, p. 900–913, 2014.

[37] HAMAMOTO, A. H.; CARVALHO, L. F.; PROENCA, M. L. Aco and ga metaheuristics for anomaly detection. In: IEEE. *2015 34th International Conference of the Chilean Computer Science Society (SCCC)*. [S.l.], 2015. p. 1–6.

[38] FERNANDES, G. et al. Statistical, forecasting and metaheuristic techniques for network anomaly detection. In: *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2015. (SAC '15), p. 701–707. ISBN 978-1-4503-3196-8. Disponível em: <http://doi.acm.org/10.1145/2695664.2695852>.

[39] LIMA, I. V. M. de; DEGASPARI, J. A.; SOBRAL, J. B. M. Intrusion detection through artificial neural networks. In: *NOMS 2008 - 2008 IEEE Network Operations and Management Symposium*. [S.l.: s.n.], 2008. p. 867–870. ISSN 1542-1201.

[40] MUKKAMALA, S.; JANOSKI, G.; SUNG, A. Intrusion detection using neural networks and support vector machines. In: *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*. [S.l.: s.n.], 2002. v. 2, p. 1702–1707. ISSN 1098-7576.

[41] MORADI, M.; ZULKERNINE, M. A neural network based system for intrusion detection and classification of attacks. In: *Proceedings of the 2004 IEEE international conference on advances in intelligent systems-theory and applications*. [S.l.: s.n.], 2004.

[42] STALLINGS, W. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1998.

[43] KUROSE, J. F. et al. *Redes de Computadores ea Internet: uma abordagem top-down*. [S.l.]: Pearson, 2010.

[44] LEINWAND, A.; CONROY, K. F. Network management: a practical perspective. *Unix and Open Systems Series, Reading, MA: Addison-Wesley,/ c1996, 2nd ed.*, 1996.

[45] PINHEIRO, J. M. Gerenciamento de redes de computadores: Uma breve introdução. 2006. *http://www. projetoderedes. com. br/artigos/artigo_gerenciamento_de_redes_de_computadores. php. Acesso em*, v. 22, p. 01–10, 2012.

[46] CLEMM, A. *Network Management Fundamentals*. [S.l.]: Cisco Press, 2006. ISBN 1587201372.

[47] THOTTAN, M.; JI, C. Anomaly detection in ip networks. *IEEE Transactions on signal processing*, IEEE, v. 51, n. 8, p. 2191–2204, 2003.

[48] UDUPA, D. K. *Network management systems essentials*. [S.l.]: McGraw-Hill, Inc., 1995.

[49] DUFFIELD, N. et al. Rule-based anomaly detection on ip flows. In: IEEE. *INFOCOM 2009, IEEE*. [S.l.], 2009. p. 424–432.

[50] CISCO, I. *NetFlow, Introduction to Cisco IOS NetFlow–A Technical Overview, White Paper*. 2012.

[51] QUITTEK, J. et al. *Requirements for IP flow information export (IPFIX)*. [S.l.], 2004.

[52] PANCHEN, S.; PHAAL, P.; MCKEE, N. Inmon corporation's sflow: A method for monitoring traffic in switched and routed networks. 2001.

[53] CLAISE, B. Cisco systems netflow services export version 9. 2004.

[54] CLAISE, B.; WOLTER, R. *Network management: accounting and performance strategies*. [S.l.]: Cisco Press, 2006.

[55] TODINOV, M. T. *Flow Networks: Analysis and Optimization of Repairable Flow Networks, Networks with Disturbed Flows, Static Flow Networks and Reliability Networks*. [S.l.]: Newnes, 2013.

[56] CHOUDHARY, S.; SRINIVASAN, B. Usage of netflow in security and monitoring of computer networks. *International Journal of Computer Applications*, Foundation of Computer Science, v. 68, n. 24, 2013.

[57] LEINEN, S. Evaluation of candidate protocols for ip flow information export (ipfix). 2004.

[58] LIU, Y. et al. Next generation internet traffic monitoring system based on netflow. In: IEEE. *Intelligent System Design and Engineering Application (ISDEA), 2010 International Conference on*. [S.l.], 2010. v. 1, p. 1006–1009.

[59] MOLINA, A. et al. Implementation of an ipfix compliant flow traffic meter: challenges and performance assessment. In: IEEE. *IP Operations & Management, 2003.(IPOM 2003). 3rd IEEE Workshop on*. [S.l.], 2003. p. 61–67.

[60] TRAMMELL, B.; BOSCHI, E. An introduction to ip flow information export (ipfix). *IEEE Communications Magazine*, IEEE, v. 49, n. 4, 2011.

[61] PHAAL, P.; PANCHEN, S.; MCKEE, N. *Rfc 3176: sflow*. [S.l.]: September, 2001.

[62] LI, B. et al. A survey of network flow applications. *Journal of Network and Computer Applications*, Elsevier, v. 36, n. 2, p. 567–581, 2013.

[63] YANG, X.-S.; HE, X. Firefly algorithm: recent advances and applications. *International Journal of Swarm Intelligence*, Inderscience Publishers Ltd, v. 1, n. 1, p. 36–50, 2013.

[64] NADHIR, K.; CHABANE, D.; TAREK, B. Firefly algorithm for optimal allocation and sizing of distributed generation in radial distribution system for loss minimization. In: *Control, Decision and Information Technologies (CoDIT), 2013 International Conference on*. [S.l.: s.n.], 2013. p. 231–235.

[65] KLIEWER, A. et al. Comparison of gaussian, logarithmic transform and mixed gaussian–log-normal distribution based 1dvar microwave temperature–water-vapour mixing ratio retrievals. *Quarterly Journal of the Royal Meteorological Society*, Wiley Online Library, v. 142, n. 694, p. 274–286, 2016.

[66] BISHOP, C. M. *Neural networks for pattern recognition*. [S.l.]: Oxford university press, 1995.

[67] TSCHICHOLD, N. *RuleNet-A Knowledge-Based Neural Network Model with Application Examples in Mobile Robotics*. Tese (Doutorado) — Ph. D. Thesis 11356, ETH Zurich, 1995.

[68] SANGER, T. D. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural networks*, Elsevier, v. 2, n. 6, p. 459–473, 1989.

[69] HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. *Neural networks*, Elsevier, v. 2, n. 5, p. 359–366, 1989.

[70] MIKOLOV, T. et al. Recurrent neural network based language model. In: *Interspeech*. [S.l.: s.n.], 2010. v. 2, p. 3.

[71] DEMUTH, H. B. et al. *Neural network design*. [S.l.]: Martin Hagan, 2014.

[72] HAYKIN, S.; NETWORK, N. A comprehensive foundation. *Neural Networks*, v. 2, n. 2004, 2004.

[73] CHEN, H.; LIU, C. Research and application of cluster analysis algorithm. In: *Measurement, Information and Control (ICMIC), 2013 International Conference on*. [S.l.: s.n.], 2013. v. 01, p. 575–579.

[74] WANG, K.; WANG, B.; PENG, L. Cvap: Validation for cluster analyses. *Data Science Journal*, v. 8, p. 88–93, 2009.

[75] WINTER, J. C. de; GOSLING, S. D.; POTTER, J. Comparing the pearson and spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. *Psychological methods*, American Psychological Association, v. 21, n. 3, p. 273, 2016.

[76] KAUFMAN, L.; ROUSSEEUW, P. J. *Finding groups in data: an introduction to cluster analysis*. [S.l.]: John Wiley & Sons, 2009. v. 344.

[77] DIMITRIADOU, E.; DOLNIČAR, S.; WEINGESSEL, A. An examination of indexes for determining the number of clusters in binary data sets. *Psychometrika*, Springer, v. 67, n. 1, p. 137–159, 2002.

[78] DUDOIT, S.; FRIDLYAND, J. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome biology*, BioMed Central, v. 3, n. 7, p. 1, 2002.

[79] HALKIDI, M.; BATISTAKIS, Y.; VAZIRGIANNIS, M. On clustering validation techniques. *Journal of intelligent information systems*, Springer, v. 17, n. 2-3, p. 107–145, 2001.

[80] PRIDDY, K. L.; KELLER, P. E. *Artificial neural networks: an introduction*. [S.l.]: SPIE Press, 2005. v. 68.

[81] PHAAL, P.; PANCHEN, S.; MCKEE, N. *InMon corporation's sFlow: A method for monitoring traffic in switched and routed networks*. [S.l.], 2001.

[82] LIU, Y.; SHRIBERG, E. Comparing evaluation metrics for sentence boundary detection. In: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*. [S.l.: s.n.], 2007. v. 4, p. IV–185–IV–188. ISSN 1520-6149.

[83] 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*, Elsevier, v. 27, n. 15, p. 2427–2434, 1993.

[84] HINE, J.; WETHERILL, G. B. Coefficient of correlation. In: _____. *A Programmed Text in Statistics Book 4: Tests on Variance and Regression*. Dordrecht: Springer Netherlands, 1975. p. 32–41. ISBN 978-94-011-6509-9. Disponível em: <http://dx.doi.org/10.1007/978-94-011-6509-9_5>.

[85] CISAR, P.; CISAR, S. M. Ewma statistic in adaptive threshold algorithm. In: IEEE. *2007 11th International Conference on Intelligent Engineering Systems*. [S.l.], 2007. p. 51–54.

[86] MATIAS, R. et al. Comparison analysis of statistical control charts for quality monitoring of network traffic forecasts. In: IEEE. *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*. [S.l.], 2011. p. 404–409.

[87] MONTGOMERY, D. C. *Introduction to statistical quality control*. [S.l.]: John Wiley & Sons, 2007.

[88] HOLLAND, J. H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. [S.l.]: U Michigan Press, 1975.

[89] KLIR, G.; YUAN, B. *Fuzzy sets and fuzzy logic*. [S.l.]: Prentice hall New Jersey, 1995. v. 4.

[90] RALESCU, D. Applications of fuzzy sets to systems analysis. 1979.

[91] CISAR, P.; CISAR, S. M. Ewma statistic in adaptive threshold algorithm. In: *2007 11th International Conference on Intelligent Engineering Systems*. [S.l.: s.n.], 2007. p. 51–54. ISSN 1543-9259.

[92] HAAG, P. *NFDUMP*. 2015. <http://nfdump.sourceforge.net>.

[93] MARCOS, V.; JR, M. L. P. Scorpius: sflow network anomaly simulator. *Journal of Computer Science*, Science Publications, v. 11, n. 4, p. 662, 2015.