Universidade Estadual de Londrina

ANDERSON HIROSHI HAMAMOTO

DETECÇÃO DE ANOMALIAS EM TRÁFEGO DE REDE USANDO ALGORITMOS GENÉTICOS E LÓGICA FUZZY COM ANÁLISE DE FLUXOS IP

Londrina
2017

ANDERSON HIROSHI HAMAMOTO

DETECÇÃO DE ANOMALIAS EM TRÁFEGO DE REDE USANDO ALGORITMOS GENÉTICOS E LÓGICA FUZZY COM ANÁLISE DE FLUXOS IP

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 Jr.

Londrina
2017

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

Hamamoto, Anderson Hiroshi.
Detecção de Anomalias em Tráfego de Rede usando Algoritmos Genéticos e Lógica Fuzzy com Análise de Fluxos IP / Anderson Hiroshi Hamamoto. - Londrina, 2017.
105 f. : il.

Orientador: Mario Lemes Proença Jr.
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. Redes de Computadores - Tese. 2. Detecção de Anomalias - Tese. 3. Algoritmos Genéticos - Tese. 4. Lógica Fuzzy - Tese. I. Proença Jr, Mario Lemes. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. III. Título.

ANDERSON HIROSHI HAMAMOTO

DETECÇÃO DE ANOMALIAS EM TRÁFEGO DE REDE USANDO ALGORITMOS GENÉTICOS E LÓGICA FUZZY COM ANÁLISE DE FLUXOS IP

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

Orientador: Dr. Mario Lemes Proença Jr.
Universidade Estadual de Londrina - UEL

Prof. Dr. Elieser Botelho Manhas Jr.
Universidade Estadual de Londrina - UEL

Prof. Dr. Lucas Dias Hiera Sampaio
Universidade Tecnológica Federal do Paraná – UTFPR

Prof. Dr. Alexandre de Aguiar Amaral
Instituto Federal Catarinense – IFC

Londrina, 14 de Março de 2017.

Este trabalho é dedicado aos meus pais e a todos que, assim como eu, estão em pé sobre ombros de gigantes.

AGRADECIMENTOS

Eu agradeço aos meus pais, Célio e Jacira, pelo valor dado à educação e pelo apoio e carinho durante a realização deste trabalho.

Agradeço também ao Prof. Dr. Mario Lemes Proença Jr., pela paciência e orientação ao longo deste trabalho. Aos integrantes do grupo de pesquisa de redes do Departamento de Computação da Universidade Estadual de Londrina, em especial ao Luiz, que me auxiliou nos vários projetos.

Aos meus colegas de computação, que compartilharam da minha jornada de mestrado.

A CAPES, pela bolsa concedida.

"People think that computer science is the art of geniuses but the actual reality is the opposite, just many people doing things that build on each other, like a wall of mini stones."
(Donald Knuth)

HAMAMOTO, A. H.. **Detecção de Anomalias em Tráfego de Rede usando Algoritmos Genéticos e Lógica Fuzzy com Análise de Fluxos IP**. 79 p. Dissertação de Mestrado (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina–PR, 2017.

RESUMO

Devido ao crescimento de usuários e serviços prestados pelas redes de computadores, o seu gerenciamento se torna imprescindível. Por consequência, existe a necessidade de monitorar os eventos que ocorrem com os ativos de rede e os comportamentos dos usuários, a fim de garantir a integridade e disponibilidade dos serviços prestados. Porém, efetuar essa tarefa manualmente é impraticável, uma vez que o volume do tráfego de dados é substancial. Assim, métodos sofisticados que realizam a gerência e mantêm a segurança dos serviços oferecidos pelas redes de computadores são essenciais. Com isso em vista, este trabalho apresenta um sistema para detecção de anomalias utilizando informações extraídas de fluxos IP. Isso é feito analisando o comportamento normal do tráfego de rede, gerando um perfil de caracterização, denominado DSNSF (*Digital Signature of Network Segment using Flow Analysis*), usando um Algoritmo Genético. Com base nesse perfil, no tráfego real e nos limiares calculados com o EWMA (*Exponentially Weighted Moving Average*), um sistema usando Lógica *Fuzzy* determina se existe uma anomalia. As informações utilizadas para análise foram extraídas de coletas em ambientes reais com protocolos baseados em fluxos IP, sendo elas: bits por segundo, pacotes por segundo, entropia de IPs de origem e destino e entropia de portas de origem e destino, totalizando seis dimensões. Os experimentos conduzidos indicam que o sistema proposto obteve bons resultados, atingindo altas taxas de acurácia e precisão, com poucos falsos alarmes.

Palavras-chave: Redes de Computadores, Detecção de Anomalias, Algoritmos Genéticos, Lógica Fuzzy, Fluxos IP

HAMAMOTO, A. H.. **Network Traffic Anomaly Detection using Genetic Algorithms and Fuzzy Logic with IP Flow Analysis**. 79 p. Master's Thesis (Master in Science in Computer Science) – State University of Londrina, Londrina–PR, 2017.

ABSTRACT

Due to the growth of users and services provided through computer networks, their management becomes indispensable. Consequently, it is necessary to monitor events that are occurring with the network assets and the users behaviors, aiming to guarantee their integrity and availability in the provided services. However, realizing this task manually is impractical, given that network traffic volume is substantial. Thus, sophisticated methods that realize the management and maintain the security of services offered through computer networks are essential. Bearing this in mind, this work presents a system for anomaly detection using information extracted from IP flows. This is accomplished by analysing the standard behavior of the network traffic, generating a characterization profile, namely DSNSF (Digital Signature of Network Segment using Flow Analysis), using a Genetic Algorithm. Based on this profile, the real traffic and thresholds calculated with EWMA (Exponentially Weighted Moving Average), a Fuzzy Logic system determines if an anomaly exists. The information used for analysis were extracted from a collection in real environments using IP flows based protocols, which are: bits per second, packets per second, IP source and destination entropy and port source and destination entropy, totalizing six dimensions. The experiments conducted indicate that the proposed system obtained good results, achieving high accuracy and precision, with few false alarms.

Keywords: Computer Networks, Anomaly Detection, Genetic Algorithms, Fuzzy Logic, IP Flows

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACK	*Acknowledgment*
ACO	*Ant Colony Optimization*
ACODS	*Ant Colony Optimization for Digital Signature*
ADTW	*Adaptive Dynamic Time Warping*
ARIMA	*Autoregressive Integrated Moving Average*
DoS	*Denial of Service*
DDoS	*Distributed Denial of Service*
DTW	*Dynamic Time Warping*
DSNS	*Digital Signature of Network Segment*
DSNSF	*Digital Signature of Network Segment using Flow Analysis*
EWMA	*Exponentially Weighted Moving Average*
FA	*Firefly Algorithm*
FADS	*Firefly Algorithm for Digital Signature*
FHCA	*Firefly Harmonic Clustering Algorithm*
GADS	*Genetic Algorithm for Digital Signature*
GFADS	*GSA-based Flow Anomaly Detection System*
GSA	*Gravitational Search Algorithm*
HIDS	*Host-based Intrusion Detection System*
HTTP	*Hypertext Transfer Protocol*
HTTPS	*Hypertext Transfer Protocol Secure*
HW	*Holt-Winters*
ICMP	*Internet Control Message Protocol*
IDS	*Instrusion Detection System*
IETF	*Internet Engineering Task Force*

IP	*Internet Protocol*
IPFIX	*Internet Protocol Flow Information Export*
KHM	*K-Harmonic Means*
KPCA	*Kernel-based Principal Component Analysis*
MLP	*Multi-Layer Perceptron*
NADS	*Network Anomaly Detection Systems*
NIDS	*Network-based Intrusion Detection Systems*
NMSE	*Normalized Square Mean Error*
N-RBF	*Improved Radial Basis Function*
PCA	*Principal Component Analysis*
PSO	*Particle Swarm Optimization*
RFC	*Request For Comments*
ROC	*Receiver Operating Characteristics*
SNMP	*Simple Network Management Protocol*
SOFM	*Self-Organized Feature Map*
SVM	*Support Vector Machine*
TCP	*Transmission Control Protocol*
TFP	*Taxa de Falsos Positivos*
TI	Tecnologia da Informação
UDP	*User Datagram Protocol*
UEL	Universidade Estadual de Londrina
UTFPR	Universidade Tecnológica Federal do Paraná
VoIP	*Voice over Internet Protocol*

LISTA DE SÍMBOLOS

a_s	Abertura da função de pertinência do sino generalizada
a_{tra}	Limite inferior da função de pertinência Trapezoidal
a_{tri}	Limite inferior da função de pertinência Triangular
A	Valor de acurácia
b_s	Inclinação da função de pertinência do sino generalizada
b_{tra}	Limite de suporte inferior da função de pertinência Trapezoidal
b_{tri}	Centro da função de pertinência Triangular
c_g	Centro da função de pertinência Gaussiana
c_s	Centro da função de pertinência do sino generalizada
c_{tra}	Limite de suporte superior da função de pertinência Trapezoidal
c_{tri}	Limite superior da função de pertinência Triangular
d_{tra}	Limite superior da função de pertinência Trapezoidal
$\overline{d}$	Média da diferença entre o DSNSF e o tráfego
f	Distância Euclidiana
$F1$	Valor de *F-measure*
F_n	Número de falsos negativos
F_p	Número de falsos positivos
H	Entropia de Shannon
$l_{inferior}$	Limiar inferior dos dados de entrada
$l_{superior}$	Limiar superior dos dados de entrada
L	Constante usada para definir os limiares usando EWMA
$limiares$	Limiares aceitáveis entre o tráfego real e o previsto
P	Valor de precisão
P_l	Valor de precisão para o limiar l

r	Valor de correlação
R	Valor de revocação
R_l	Valor de revocação para um limiar l
S	Soma total de ocorrências das observações
T_n	Número de verdadeiros negativos
T_p	Número de verdadeiros positivos
x_k	Tráfego real observado para o atributo k
$\hat{x_k}$	Tráfego previsto para o atributo k
z_i	*Exponentially Weighted Moving Average* do instante i
Γ	Valor de corte usado para a detecção de anomalias com Lógica *Fuzzy*
λ	Constante *lambda* entre 0 e 1 usado no EWMA
θ_k	Limiar calculado utilizando o EWMA para o atributo k
σ^2	Variância dos valores de entrada
σ_s	Desvio padrão da função de pertinência Gaussiana
$\sigma^2_{z_i}$	Variância dos valores calculados com o EWMA
ζ_g	Grau de pertinência calculado com função Gaussiana
ζ_k	Grau de pertinência para o atributo k do tráfego da rede
ζ_s	Grau de pertinência da função do sino generalizada
ζ_t	Grau de pertinência para o exemplo da temparatura
ζ_{tra}	Grau de pertinência calculado com a função trapezoidal
ζ_{tri}	Grau de pertinência calculado com a função triangular

SUMÁRIO

1 INTRODUÇÃO

As redes de computadores tornaram-se um meio de comunicação essencial para a sociedade. Por meio delas, vários serviços são acessados, tais como *e-mail*, *Internet Banking*, *Cloud Computing*, VoIP (*Voice over IP*) e *streaming* de áudio e vídeo. Além disso, diariamente usuários buscam acesso à informação de forma rápida e prática na *web* e comunicam-se através de mensageiros instantâneos. Em adição a esses serviços básicos acessados pelos usuários por meio de aplicativos e navegadores de Internet, novos conceitos e paradigmas surgem em redes de computadores. Entre eles, podemos citar a Internet das Coisas, as Redes Definidas por *Software*, as Redes Veiculares, Plataforma como Serviço, Infraestrutura como Serviço, entre outros. Nesses serviços prestados, informações pessoais e financeiras sobre usuários e organizações são armazenados, exigindo o monitoramento e o controle de acesso.

Por essas razões, a gerência se faz necessária para assegurar a integridade, disponibilidade e resiliência das redes de computadores. Dentre as atividades desempenhadas pela gerência, encontra-se o monitoramento, cuja finalidade é prover informações relevantes para a manutenção e controle dos mecanismos de comunicação. Tais informações podem ser adquiridas de várias formas, como a captura de pacotes ou utilizando protocolos padronizados como SNMP (*Simple Network Management Protocol*) [1], sFlow [2], NetFlow [3] e IPFIX (*IP Flows Information Export*) [4]. Porém, devido ao grande volume de dados que trafega na rede, a análise manual do tráfego é onerosa e complexa, em um cenário onde os problemas devem ser detectados e mitigados rapidamente. Portanto, torna-se necessário empregar ferramentas que façam essa análise automaticamente, providenciando ao gerente de redes um relatório sobre o estado da rede.

Sistemas de Detecção de Intrusões (IDS, do inglês *Intrusion Detection Systems*) são utilizados para analisar o tráfego a fim de prover uma rede mais confiável e segura. Com base em informações obtidas sobre o fluxo de dados transmitidos, algoritmos de aprendizado de máquina, mineração de dados, metaheurísticas e algoritmos evolutivos podem ser aplicados para facilitar a gerência de redes. A aplicação de tais sistemas auxilia o gerente providenciando um relatório sobre o estado da rede, facilitando a identificação e mitigação de problemas. Com isso, caso algum problema esteja ocorrendo, o responsável pode tomar as devidas providências, retornando a rede as suas operações normais.

A detecção de anomalias é definida como o problema de encontrar padrões que não condizem com o comportamento esperado [5]. No contexto de redes de computadores, anomalias podem ser causadas por usuários, aplicações e equipamentos. Em alguns casos, as anomalias são provenientes de usuários mal intencionados, que deliberadamente afetam o tráfego causando problemas na operação dos serviços. Uma categoria de ataque que é

explorada é a de negação de serviços, em especial a sua versão distribuída.

A Verisign, uma empresa de serviços de domínios e segurança, publicou um relatório sobre ataques de Negação de Serviços Distribuído – *Distributed Denial of Service* (DDoS) – lançado no segundo trimestre de 2016 [6], que indicou um aumento de 75% desse tipo de ataque ao ano. Foram observados instâncias de ataques que alcançaram 256 Gigabits e 64 milhões de pacotes por segundo. Além disso, foram constatados que 35% dos DDoS utilizaram três ou mais tipos de ataques simultaneamente, o que aumenta a complexidade para detectar e mitigar tais ameaças. A Panda Security, empresa que desenvolve soluções de segurança de TI, publicou um relatório sobre as predições de segurança cibernética para 2017 [7]. Nesse relatório, os ataques de DDoS são considerados uma das maiores ameaças, com um relato de que grandes empresas de telecomunicações já foram comprometidas por esse ataque, ficando inoperantes durante um período de até onze horas. As previsões indicam um cenário ainda mais desafiador para a segurança nos próximos anos.

Como pode ser observado nos relatórios citados, há um aumento no número e na complexidade dos ataques, o que torna-se necessário desenvolver IDSs que sejam adaptativos e dinâmicos. Para tanto, necessita-se avaliar o tráfego de rede recente, que reflete o cenário atual, acompanhando as adaptações e a dinamicidade das redes de computadores. Assim sendo, um sistema que consiga observar o tráfego recente da rede e providenciar uma análise precisa sobre o seu estado é explorado neste trabalho.

O sistema proposto, destinado à detecção de anomalias do tráfego, utiliza como base fluxos IP. No total são extraídos seis atributos em intervalos de um minuto para a análise: bits por segundo, pacotes por segundo, entropia de IP de origem, entropia de IP de destino, entropia de porta de origem e entropia de porta de destino. Com os dados históricos do tráfego, são geradas caracterizações que representam o comportamento normal para os seis atributos. Essa metodologia é denominada DSNSF (*Digital Signature of Network Segment using Flow Analysis*). O sistema proposto possui duas etapas distintas: geração da caracterização com Algoritmo Genético e cálculo dos limiares usando EWMA (*Exponentially Weighted Moving Average*) e detecção de anomalias com a Lógica *Fuzzy*.

A caracterização é feita usando um Algoritmo Genético, que encontra o valor esperado do comportamento da rede. Algoritmo Genético é uma metaheurística amplamente aplicada em problemas de otimização e tem o funcionamento baseado na evolução das espécies. As soluções são evoluídas iterativamente, até que um critério de parada seja atingido. O histórico do tráfego é utilizado para calcular os limiares usando o EWMA, que definem flutuações normais entre o real e o esperado.

Na segunda etapa, o DSNSF e os limiares são usados para a detecção de anomalias com a Lógica *Fuzzy*. Através de uma função de pertinência, calcula-se o nível de anomalia para cada atributo analisado. Em conjuntos que seguem as definições da Lógica *Fuzzy*,

os valores variam entre 0 e 1, enquanto na Lógica Booleana os valores são 0 ou 1, sendo a principal diferença entre elas. A sua aplicação é vantajosa em ambientes que possuem informações imprecisas e verdades parciais, o que são características de medições sobre o tráfego de rede, pois utilizam estatística e trazem um nível de incerteza. Ainda mais, na segurança de redes não existe uma separação clara entre o que não é o que é uma anomalia [8]. Com isso, a Lógica *Fuzzy* é propícia para ser aplicada em detecção de anomalias em redes.

A avaliação do sistema proposto é feita através de experimentos utilizando o tráfego coletado em ambientes reais. Os locais de coleta foram a Universidade Estadual de Londrina (UEL) e a Universidade Tecnológica Federal do Paraná (UTFPR) - Campus Toledo. Na UEL foi utilizado o protocolo sFlow para a coleta nos anos de 2012, 2015 e 2016, enquanto o tráfego da UTFPR foi coletado em 2013 com o protocolo IPFIX. Assim, a eficácia do sistema poderá ser avaliado em redes com diferentes configurações e protocolos de coleta.

O restante do trabalho está dividido nos seguintes capítulos: no Capítulo 2 são apresentados os trabalhos relacionados. No Capítulo 3 é descrito o funcionamento de Algoritmos Genéticos e Lógica *Fuzzy*. O Capítulo 4 apresenta o sistema desenvolvido ao longo deste trabalho. O Capítulo 5 mostra os resultados obtidos e, por fim, o Capítulo 6 conclui o trabalho apresentado.

2 TRABALHOS RELACIONADOS

Este capítulo apresenta alguns trabalhos relacionados ao tema de detecção de anomalias em redes de computadores. Uma abordagem muito explorada nessa área é a aplicação de Sistemas de Detecção de Intrusões. Esse campo de pesquisa foi iniciado com os trabalhos de Anderson[9] e Denning[10], e desde então vem sendo aprimorado. Com as aplicações das redes de computadores nos mais variados aspectos da sociedade, a sua importância deve ser enfatizada, visando manter os seus serviços operantes.

Gerência de redes por meio de IDS é um tópico que não possui soluções definitivas, e portanto, é alvo de muitas pesquisas. Há dois tipos de IDS [11, 12]:

- ***Host-based Intrusion Detection Systems*** (HIDS): protegem um *host* ou sistema em particular, monitorando e analisando as suas configurações e as atividades das suas aplicações;
- ***Network-based Intrusion Detection Systems*** (NIDS): protegem uma rede composta por vários *hosts* ou sistemas. Em sua maioria analisam o tráfego de rede para verificar as ocorrências de eventos anômalos.

Este trabalho trata de detecção de anomalias em uma rede de *hosts*, portanto é um NIDS. Há três abordagens para NIDS:

- **Baseado em assinaturas**: requer o conhecimento dos padrões de cada tipo de anomalia. Caso o comportamento da rede se enquadre em um dos padrões conhecidos, um alarme identificando a anomalia é gerado [13]. No entanto, se houver uma anomalia, mas o seu padrão não for conhecido, um alarme não será emitido e o gerente de redes não saberá que um problema está ocorrendo;
- **Baseado em detecção de anomalias**: avalia o comportamento normal da rede e cria perfis do comportamento esperado. Com isso, o tráfego real é comparado com o previsto e, caso existam desvios que o sistema indique serem anômalos, um alarme é disparado [14]. Porém, esse tipo de sistema não possui a capacidade de identificar as anomalias, ficando a cargo do gerente em fazê-lo;
- **Híbrido**: NIDS híbrido combina as duas abordagens anteriores para criar um sistema mais completo, o qual possui a capacidade de detectar anomalias desconhecidas, e caso a assinatura de uma anomalia esteja registrada em sua base, também é capaz de identificá-la [15].

Bhuyan, Bhattacharyya e Kalita[15] classificam NIDS para detecção de anomalias em 6 categorias:

- **Estatística**: usa um modelo estatístico para verificar se uma nova observação se adequa ao modelo criado;
- **Baseado em Classificação**: identifica uma nova observação usando um classificador treinado com uma base de dados rotulada;
- **Baseado em *Clustering* e *Outlier***: separa as observações de acordo com semelhanças de características e classifica cada um dos conjuntos formados;
- ***Soft Computing***: usa fatores de imprecisão, incerteza, verdades parciais e aproximações, tais como Algoritmos Genéticos e Lógica *Fuzzy*;
- **Baseado em Conhecimento**: compara o comportamento da rede com um conjunto de padrões ou regras pré-definido;
- **Aprendizado de Combinações**: combina várias técnicas para gerar um detector de anomalias, geralmente um classificador.

O foco deste trabalho encontra-se na abordagem de *Network Anomaly Detection Systems* (NADS), que são NIDSs baseados em detecção de anomalias. Essa metodologia analisa o comportamento recorrente da rede a fim de determinar se uma anomalia está ocorrendo em um dado instante. Há duas principais abordagens para esse tipo de sistemas: supervisionado e não-supervisionado. Sistemas supervisionados requerem conhecimento providenciado por um agente externo, conjunto da dados rotulados e não possuem a capacidade de detectar *zero-day vulnerabilities* [12]. O sistema proposto neste trabalho aplica técnicas não-supervisionadas, *i.e.*, Algoritmos Genéticos e Lógica *Fuzzy*, que não necessitam de informações além das disponibilizadas por coletas de fluxos IP, o que é uma vantagem em relação à técnicas supervisionadas. Ainda mais, ambas técnicas pertencem ao conjunto de *Soft Computing*, que são conhecidos pela sua performance adequada em ambientes com informações imprecisas [16], como é o caso de detecção de anomalias em redes [8].

Anil e Remya[17] aplicam Algoritmos Genéticos, *Self-Organized Feature Map* (SOFM) e *Support Vector Machine* (SVM) para detectar anomalias no conjunto de dados *KDD99* [18]. Os autores utilizaram o SOFM para reduzir o conjunto de dados a ser processado, um Algoritmo Genético para selecionar as características mais relevantes e, por fim, o SVM é aplicado para classificar uma observação em anômala ou não. Essa técnica híbrida produziu um alto índice de detecção de anomalias, baixa taxa de falsos positivos e um baixo custo computacional.

No trabalho de Jongsuebsuk, Wattanapongsakorn e Charnsripinyo[19] são utilizados Algoritmos Genéticos e Lógica *Fuzzy* para a detecção de ataques desconhecidos. Os autores aplicam um Algoritmo Genético para otimizar regras presentes em um Sistema de Lógica *Fuzzy* a fim de classificar observações na etapa de testes. O conjunto de dados utilizado é coletado por meio de um *sniffer* de pacotes, formando uma observação a cada dois segundos, extraindo informações dos cabeçalhos de IP, TCP, UDP e ICMP. Cada regra do algoritmo possui 12 genes, tais como número de pacotes TCP, número de pacotes UDP, entre outros. O último campo de cada regra é o tipo de ataque a qual o conjunto pertence. Nos resultados, o trabalho atingiu uma alta precisão e uma taxa de falsos positivos menor que 1%, com a capacidade de detectar tanto ataques conhecidos como desconhecidos.

Kuang, Xu e Zhang[20] combinam *Kernel-based Principal Component Analysis* (KPCA), *Support Vector Machine* (SVM) e Algoritmos Genéticos para detecção de intrusões usando a base de dados KDD99. No trabalho apresentado, os autores aplicam o KPCA para selecionar as características mais relevantes do tráfego, o SVM é aplicado para classificar se uma observação representa um ataque e o Algoritmo Genético para otimizar os parâmetros do SVM. Os autores aplicam o *Improved Radial basis Function* (N-RBF) como a função do *kernel* do SVM para reduzir o tempo de treinamento. Com base nos resultados, o método apresentado é comparado com vários outros em diversas instâncias de testes, atingindo os melhores valores nas métricas utilizadas para avaliação.

Um problema existente em IDS é a taxa de falsos positivos que são gerados. No trabalho de Narsingyani e Kale[21], esse problema é estudado usando a base de dados do KDD99, aplicando um Algoritmo Genético para a detecção de ataques do tipo *Denial of Service* (DoS). Nesses trabalhos são usadas sete características presentes na base de dados: *Duration*, *Protocol*, *Service*, *flag*, *Source_byte*, *Dst_byte* e *Attack-name*. Com base nas características usadas, regras que definem ataques de DoS são geradas. Os autores usaram 40, 60 e 90 regras para testes, atingindo uma taxa de 96,39% de verdadeiros positivos e 1,20% de falsos positivos com 90 regras.

Jadidi *et al.*[22] utilizam fluxos IP para a detecção de anomalias em redes aplicando Redes Neurais Artificais e *Gravitational Search Algorithm* (GSA). GSA é um algoritmo de busca heurística baseado na lei da gravidade e nas leis de Newton. Os autores apresentam um sistema denominado GFADS (*GSA-based Flow Anomaly Detection System*), que usa o GSA para ajustar os pesos em uma *Multi-Layer Perceptron* (MLP) como alternativa ao *Backpropagation*, que é o método mais comum para treinamento de uma Rede Neural Artificial. O método apresentado atingiu uma acurácia de 99,43% e uma taxa de falsos positivos de 0,64%.

Mkuzangwe, McDonald e Nelwamondo[23] utilizam dois algoritmos para detectar ataques do tipo *TCP Syn flood*, que enviam pacotes TCP SYN para um servidor sem

respondê-lo com uma mensagem ACK para o estabelecimento da conexão. O objetivo do ataque é ocupar recursos do servidor a fim de impedi-lo de fornecer serviços aos clientes. Os algoritmos utilizados são o *Adaptive Threshold Algorithm* e o *Cumulative Sum*. Além dos dois algoritmos, o trabalho apresenta um sistema que combina os dois métodos usando o operador lógico OU. Os resultados obtidos mostram que a combinação dos dois métodos apresenta um melhor desempenho para detectar os ataques, porém também possui uma maior taxa de falsos alarmes.

Redes móveis *ad hoc* são vulneráveis a ataques de perdas de pacotes, em que nós maliciosos descartam os pacotes, ao invés de encaminhá-los. Para detectar esse tipo de ataques Chaudhary, Tiwari e Kumar[24] desenvolveram um Sistema de Detecção de Intrusões que aplica um *Sugeno-type Fuzzy Inference System*. Para analisar o método apresentado foi utilizado o Qualnet Simulator 6.1, cuja função é simular ataques de perda de pacotes. Os autores obtiveram diferentes valores para as taxas de verdadeiros positivos e falsos positivos baseado na mobilidade da rede *ad hoc*, alcançando taxas de verdadeiros positivos de 89,1% e falsos positivos de 1,6%.

Aziz *et al.*[25] combinam Algoritmos Genéticos e *Deterministic Crowding Niching*, um conceito utilizado em Sistemas Imuno-Artificiais, baseado no sistema imunológico humano para criar um IDS usando a base de dados do NSL-KDD. O método proposto pelos autores foi comparado com outros classificadores presentes no projeto WEKA, alcançando uma acurácia de 81,76%, taxa menor somente quando comparado ao *NBTree*.

Nos trabalhos de Fernandes Jr., Rodrigues e Proença Jr.[26] e Fernandes Jr. *et al.*[27] os autores extraem 7 dimensões a partir de fluxos IP para a criação de uma caracterização do tráfego de rede denominada *Digital Signature of Network Segment using Flow Analysis* (DSNSF). O DSNSF é uma predição do comportamento do tráfego da rede para um dado intervalo de tempo do dia. Nessa metodologia, as informações extraídas dos fluxos IP são bits, pacotes e fluxos por segundo e as entropias de IPs e Portas de origem e destino. Com base no histórico do tráfego de fluxos IP, uma previsão do comportamento da rede é gerada, com o objetivo de ser comparada com o tráfego real. Caso haja uma discrepância significativa, o sistema gera alertas ou alarmes baseado na severidade do possível problema detectado. Além disso, os sistemas propostos nos trabalhos são híbridos, com a capacidade de detectar anomalias desconhecidas e identificar as que possuem uma assinatura registrada.

A Tabela 1 apresenta o conjunto de dados utilizado e o desempenho em acurácia e taxa de falsos positivos (TFP) alcançados pelos trabalhos apresentados neste capítulo. Jongsuebsuk, Wattanapongsakorn e Charnsripinyo[19] isolam a taxa de detecção para cada tipo de ataque, em que a média se encontra no valor de 98,38%, os autores não calcularam a taxa de falsos positivos. Mkuzangwe, McDonald e Nelwamondo[23] analisam a probabilidade de detecção que atinge 100% para ataques do tipo *SYN flood*, em

Tabela 1 – Comparativo dos trabalhos relacionados.

Trabalho	**Dados**	**Acurácia**	**TFP**
Anil e Remya[17]	KDD99	88,28%	9,17%
Jongsuebsuk, Wattanapongsakorn e Charnsripinyo[19]	pacotes	-	-
Kuang, Xu e Zhang[20]	KDD99	94,28	0,975
Narsingyani e Kale[21]	KDD99	96,39%	1,2%
Jadidi *et al.*[22]	fluxos IP	99,43%	0,64%
Mkuzangwe, McDonald e Nelwamondo[23]	DARPA Intrusion Detection Dataset (1999)	-	-
Chaudhary, Tiwari e Kumar[24]	Simulador QualNet	-	-
Aziz *et al.*[25]	KDD99	81,76%	-
Fernandes Jr., Rodrigues e Proença Jr.[26]	fluxos IP	85%	23%
Fernandes Jr. *et al.*[27]	fluxos IP	96%	21%

contrapartida, a taxa de falsos positivos também é elevada, chegando a 100%. Chaudhary, Tiwari e Kumar[24] alcançam taxas de 89,1% de verdadeiros positivos e 1,6% de falsos positivos para a simulação de redes de alta mobilidade. No trabalho de Aziz *et al.*[25], os autores não analisam a taxa de positivos. Essa comparação dos resultados indica que as métricas mais utilizadas na literatura para análise do desempenho em IDS são a acurácia e a taxa de falsos positivos. Além disso, analisando os valores obtidos para essas métricas, pode-se considerar o ideal atingir a acurácia acima de 95% e taxas de falsos positivos abaixo de 1%, o que indica altas taxas de classificação e poucos alarmes falsos.

A diferença principal entre o trabalho apresentado nesta dissertação e os citados neste capítulo, são os dados utilizados para análise. A proposta apresentada realiza a detecção de anomalias usando o tráfego de rede obtido com fluxos IP coletados em um ambiente real. Com isso, as análises dos resultados apresentados no Capítulo 5 utilizam dados coletados em 2012, 2013, 2015 e 2016. Em alguns dos trabalhos relacionados que foram citados, a base de dados KDD99 é utilizada, em que os pacotes foram coletados em 1998. Dado a dinamicidade das redes de computadores, essa base pode não refletir o cenário atual do tráfego de redes. Além disso, o KDD99 é uma base de dados rotulados,

em que cada instância está classificada em anômalo ou não, e ele é muito utilizado para a implementação de um IDS baseado em assinaturas. Isso implica na necessidade de manter essa base atualizada, o que pode ser uma tarefa árdua, uma vez que novas técnicas de ataques e vulnerabilidades são descobertas e exploradas constantemente [28]. Outros trabalhos usam análise de pacotes, que são computacionalmente mais custosos, uma vez que todos os pacotes devem ser analisados [22]. Os trabalhos de Fernandes Jr., Rodrigues e Proença Jr.[26] e Fernandes Jr. *et al.*[27] apresentam sistemas semelhantes ao proposto, pois utilizam dados de fluxos IP e algoritmos de treinamento não supervisionados, o que não requerem uma base de dados rotulados, como na abordagem proposta por Jadidi *et al.*[22].

O sistema proposto nesta dissertação utiliza seis dimensões extraídas de coletas realizadas com protocolos baseados em fluxos IP. Essa análise multidimensional proporciona diagnosticar o estado do tráfego em relação ao volume (bits e pacotes), aplicações (portas) e *hosts* (IPs). A caracterização dessas dimensões é realizada com um Algoritmo Genético, que trabalha de forma não supervisionada e sem a intervenção humana. A detecção de anomalias é efetuada usando Lógica *Fuzzy*, com o objetivo de averiguar a existência de anomalias nas seis dimensões de forma individualizada. Posteriormente, o resultado obtido utilizando a Lógica *Fuzzy* é agregado em um único valor para ser comparado com o valor de corte Γ, como descrito no Capítulo 4.

3 ALGORITMOS GENÉTICOS E LÓGICA FUZZY

Neste capítulo serão apresentados os conceitos básicos de Algoritmos Genéticos e Lógica Fuzzy. Esses conceitos servirão de base para a compreensão do sistema proposto para a detecção de anomalias em tráfego de rede usando análise de fluxos IP.

3.1 Algoritmos Genéticos

O conceito de Algoritmos Genéticos pertence ao conjunto de métodos computacionais bio-inspirados. Esses métodos são baseados em comportamentos biológicos observados na natureza e adaptados para várias aplicações. Em sua maioria, são utilizados em problemas que necessitam percorrer um espaço de busca. Alguns exemplos de problemas em que algoritmos bio-inspirados são amplamente aplicados são de otimização, tais como os problemas do Caixeiro Viajante [29] e do Escalonamento de Tarefas [30].

Existem vários métodos baseados em modelos biológicos além de Algoritmos Genéticos. Alguns exemplos de tais métodos são *Ant Colony Optimization* [31], *Firefly Algorithm* [32], *Particle Swarm Optimization* [33] e *Shuffled Frog Leaping Algorithm* [34]. Os algoritmos supracitados são inspirados no comportamento de formigas, vagalumes, pássaros e sapos, respectivamente.

Além dos Algoritmos Genéticos, há outros modelos computacionais que utilizam o conceito de evolução e seleção natural. Dois exemplos desses modelos são Programação Genética [35] e Programação Evolutiva [36]. Diferentemente dos Algoritmos Genéticos, Programação Genética evolui os programas de computadores, e geralmente são representados por meio de uma estrutura em árvore. A Programação Evolutiva, por sua vez, é aplicada para a otimização de parâmetros.

A definição de Algoritmos Genéticos foi criada por John Holland [37] em 1972. O seu funcionamento foi inspirado na Teoria da Evolução observada por Charles Darwin [38]. A seleção natural e a sobrevivência do mais apto são conceitos presentes nos Algoritmos Genéticos. A partir de um conjunto inicial de soluções, as operações genéticas são executadas nesse conjunto de forma iterativa, otimizando as soluções a cada passo até que um resultado aceitável ou condição de parada é atingida.

Para compreender o conceito de Algoritmos Genéticos, é necessário entender a terminologia utilizada para descrever os seus componentes, que remetem ao processo de evolução:

- **Codificação**: forma em que as soluções em um Algoritmo Genético são representadas; tradicionalmente, essa representação é feita por um vetor de caracteres binários;

- **Genes**: características que compõem uma solução, por exemplo, para uma solução representada por um vetor de caracteres binários, um gene poderia ser um dos bits do cromossomo;
- **Cromossomo**: uma solução em potencial, que pode ser composta por vários genes;
- **População**: conjunto de soluções ou cromossomos presentes em uma dada iteração ou geração;
- **Função *fitness***: função ou método utilizado para avaliar o quão apto é um cromossomo para o problema em questão;
- **Geração**: uma iteração do algoritmo. O número total de gerações é o número de iterações do algoritmo;
- **Seleção**: operador genético responsável por selecionar cromossomos da população que perpetuarão os seus genes para a próxima geração;
- ***Crossover***: operador genético que combina cromossomos selecionados a fim de produzir cromossomos mais aptos;
- **Mutação**: operador genético que modifica cromossomos de uma forma aleatória, aumentando a variedade de soluções, que por sua vez, pode aumentar a aptidão dos cromossomos.

O Algoritmo 1 representa o funcionamento de um Algoritmo Genético genérico. Primeiramente, uma população inicial é gerada, geralmente de forma aleatória, baseada em parâmetros relevantes ao problema a ser solucionado. A população inicial é então otimizada usando os valores de *fitness* para analisar a adaptação das soluções para o problema tratado. Os operadores genéticos otimizam as soluções presentes em cada geração até que uma condição de parada seja atingida. Por fim, o *fitness* da população é calculado uma última vez, retornando o cromossomo com o maior *fitness* como a solução final. O diagrama de fluxo da execução de um Algoritmo Genético é apresentado na Figura 1.

Algoritmo 1 Exemplo de Algoritmo Genético genérico.

```
1: Gerar a população inicial
2: while condição de parada não for atingida do
3:     Calcular o valor de fitness da população
4:     Seleção
5:     Crossover
6:     Mutação
7:     Substituir a população antiga pela nova população
8: Calcular o valor de fitness da população
9: return O cromossomo com o maior valor de fitness
```

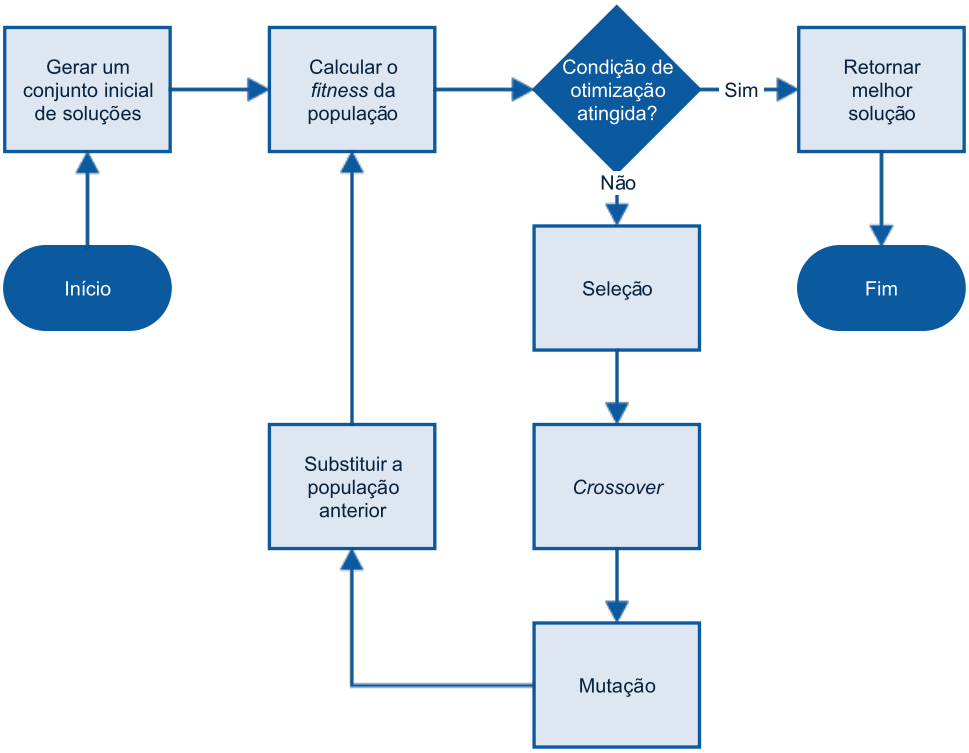


Figura 1 – Diagrama de fluxo de um Algoritmo Genético genérico.

Algoritmos Genéticos é um método generalista que providencia um modelo para a resolução vários problemas. A implementação de um Algoritmo Genético é voltado para a resolução de um problema em particular utilizando os operadores genéticos. Para cada problema, os parâmetros e os operadores genéticos utilizados devem ser definidos de tal forma que o seu desempenho seja o melhor possível [39]. Portanto, aplicar um Algoritmo Genético para resolver um problema implica em definir como cada operador genético é executado. Além disso, outros aspectos pertinentes devem ser definidos, como a codificação dos cromossomos, função de *fitness* e as condições de parada do algoritmo. Por exemplo, problemas de combinação e otimização numérica geralmente são codificados de formas diferentes, o que gera a necessidade de diferentes implementações dos operadores genéticos.

3.1.1 Codificação de Cromossomos

Tradicionalmente, a codificação dos cromossomos é feita usando vetores de caracteres binários (0 e 1). Com essa codificação é simples representar a aplicação dos operadores genéticos. Porém, há casos em que essa representação não é a mais adequada para o problema. Um exemplo seria um problema de otimização numérica, em que representações usando números são mais adequadas. Nesse caso, os operadores genéticos também devem ser ajustados para aplicá-los na codificação usada.

As codificações dos cromossomos em Algoritmos Genéticos mais utilizadas são [40]:

- **Binário**: representação usando vetores de caracteres de 0 e 1;
- **Numérico**: representação usando um valor numérico;
- **Permutação**: representação usando uma cadeia de caracteres indicando combinações e/ou ordens para as soluções.

Os cromossomos ainda podem ser divididos em genes, em que cada gene representa uma característica presente na codificação. A Figura 2 apresenta uma representação binária de um cromossomo. Cada célula é um gene com possíveis valores de 0 ou 1.

Figura 2 – Exemplo de cromossomo com representação binária.

3.1.2 Função *Fitness*

A função *fitness* é usada para avaliar o nível de adaptação de cada elemento da população, *i.e.*, a qualidade que cada cromossomo apresenta como solução para o problema. A operação de seleção é baseada nos valores de *fitness* calculados, direcionando as futuras gerações para soluções mais promisoras. Devido à importância dessa operação, ela é considerada um aspecto central dos Algoritmos Genéticos [41], sendo o objetivo minimizar ou maximizar a função *fitness* utilizada. Funções de similaridade e distâncias são exemplos que pode ser aplicada para determinar o *fitness* dos cromossomos. Para o sistema apresentado, funções de distâncias podem utilizadas, medindo a diferença entre a caracterização gerada e o tráfego real. Na Seção 4.3 a função *fitness* aplicado no sistema apresentado é detalhada.

3.1.3 Geração da População Inicial

A população inicial geralmente é gerada aleatoriamente usando parâmetros que asseguram a aplicabilidade da solução ao dado problema. No caso de representações binárias, vetores gerados de forma aleatória com posições fixas podem ser utilizados. Em problemas combinatórios como o do Caixeiro Viajante, combinações aleatórias das soluções factíveis podem ser geradas. No caso de representação numérica, valores aleatórios dentro de limites possíveis como soluções do problema podem ser aplicados.

3.1.4 Seleção

O operador de seleção é responsável por eleger as soluções que prevalecerão para a próxima geração e participarão do *crossover*. Com base nos valores de *fitness*, os cromossomos mais aptos possuem maiores chances de serem selecionados. Os métodos mais aplicados de seleção são [42][43]:

- **Roleta**: Usa o valor de *fitness* de cada cromossomo em relação à soma do *fitness* da população como a probabilidade de seleção;

- **Torneio**: Seleciona n cromossomos aleatórios e escolhe o que possuir o maior valor de *fitness*;

- **Elitismo**: Seleciona os n melhores indivíduos da população e completa a nova população usando algum outro método de seleção;

- **Classificação**: Ordena os cromossomos de acordo com o seu valor de *fitness* e utiliza a sua posição como a probabilidade de seleção.

Os métodos de seleção mencionados possuem a mesma simplicidade para implementação, porém a complexidade computacional de cada um é diferente, o que pode tornar um método mais viável do que outro.

3.1.5 *Crossover*

Com os cromossomos selecionados, a operação de *crossover* é executada. Nessa etapa, os cromossomos selecionados são combinados de alguma forma a fim de produzirem cromossomos mais aptos. Dependendo da representação dos cromossomos, diferentes métodos de *crossover* podem ser aplicados.

No caso da representação binária, geralmente o ponto ou pontos de *crossover* são determinados, e os genes são trocados a partir daquele ponto. Outra abordagem que pode ser utilizada é trocar vários genes aleatórios entre os cromossomos selecionados. O *crossover* em cromossomos representados em forma de permutação pode ser feito igual à representação binária, porém com restrições por se tratarem de arranjos combinatórios. Em representações numéricas, a forma mais comum de se realizar *crossover* é por meio de operações aritméticas, como a média entre os cromossomos selecionados.

A Figura 3 é um exemplo de *crossover* em um único ponto entre cromossomos de representação binária. O cromossomo resultante é composto pelos 3 primeiros genes do cromossomo A e os 3 últimos genes do cromossomo B.

É possível observar que caso uma operação de *crossover* entre 2 cromossomos resulte em um único cromossomo, o número de seleções que devem ser feitas para que se mantenha o número de cromossomos na população é o mesmo número de elementos contidos na mesma. Porém, pode-se usar abordagens em que isso não seja necessariamente verdade. A Figura 4 é um exemplo em que a operação de *crossover* resulta em dois cromossomos distintos que compõem a próxima geração da população.

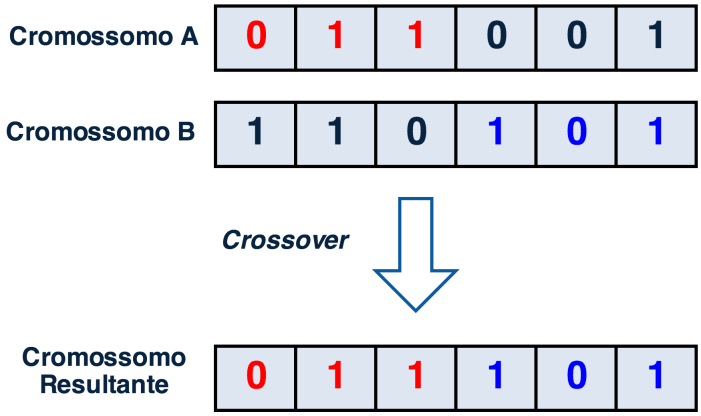


Figura 3 – Exemplo de *crossover* entre cromossomos com representações binárias.

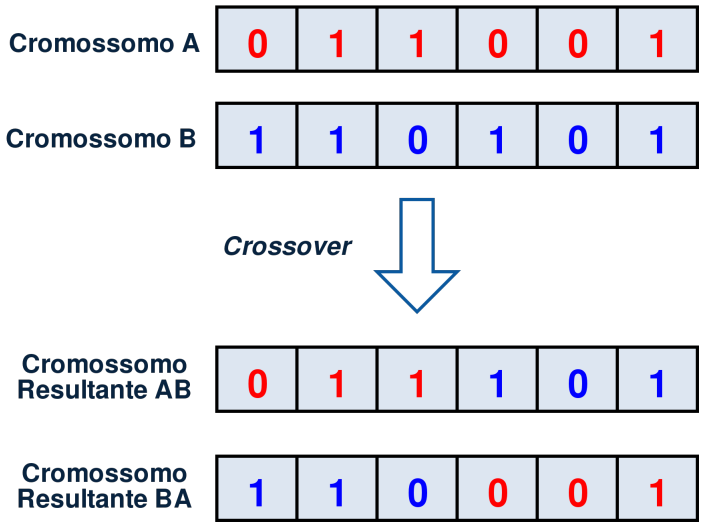


Figura 4 – Exemplo de *crossover* que resulta em dois cromossomos.

3.1.6 Mutação

A mutação é uma operação que modifica os cromossomos que passaram pela seleção e o *Crossover* com o propósito de melhorar os cromossomos além de prover um método para variá-los. Geralmente os cromossomos são modificados em pequenas proporções e, caso isso melhore o seu *fitness*, terá maiores chances de transmitir essa mutação aos seus descendentes, de forma que as futuras gerações sejam beneficiadas. Assim, se a mutação for prejudicial à adaptação, há pequenas chances de incorporá-la nas próximas iterações.

Métodos comuns de mutação são inversão de bits, troca de posição entre dois genes e operação aritmética respectivamente para codificação binária, permutação e numérica. A Figura 5 representa a mutação por inversão de bits em um cromossomo representado de forma binária, dado que os genes foram selecionados para sofrerem mutação de acordo com o parâmetro de probabilidade utilizado no Algoritmo Genético.

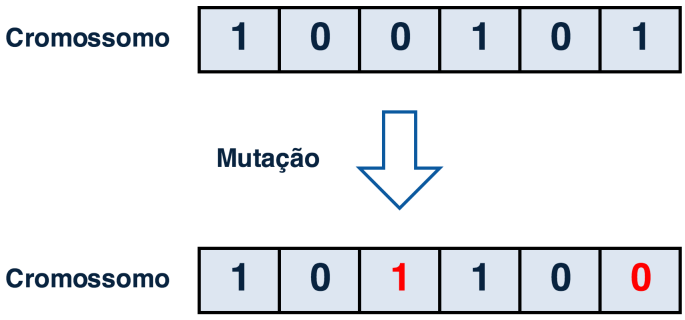


Figura 5 – Exemplo de mutação entre cromossomos com representação binária.

3.2 Lógica Fuzzy

O conceito de Lógica Fuzzy ou Lógica Nebulosa foi criado por Lotfi Zadeh [44]. Ela se diferencia da Lógica Booleana por utilizar um conjunto *rough* ao invés de *crisp* [45]. Conjuntos *crisp* são utilizados na Lógica Booleana, em que os valores possíveis são 0 ou 1, enquanto na Lógica Fuzzy são usados conjuntos *rough*, onde esses valores variam entre 0 e 1 [46]. Dessa forma, é possível atribuir graus de pertinência usando Lógica *Fuzzy* em que o resultado não é necessariamente verdadeiro ou falso, o que não é possível na Lógica Booleana.

Zadeh [47] afirma que a Lógica Fuzzy é uma lógica que tem a capacidade de lidar com ambientes imprecisos utilizando racionalização aproximada. O autor ainda afirma que esse método é propício para tratar incertezas, informações incompletas, conflitantes e verdades parciais. Wu e Banzhaf[8] justificam a sua aplicação no contexto de detecção de anomalias em redes por dois motivos: primeiro, a detecção de anomalias envolve vários atributos numéricos que são coletados e medidos estatisticamente, o que pode levar a um elevado nível de erros de detecção; segundo, não existe uma fronteira clara e bem definida separando o que são anomalias e o que é um comportamento normal na segurança de computadores. Devido a esses fatores, a Lógica *Fuzzy* foi aplicada neste trabalho para a detecção de anomalias em conjunto com um Algoritmo Genético.

3.2.1 Sistema de Inferência Fuzzy

Um Sistema de Inferência Fuzzy mapeia um conjunto de entrada para uma saída aplicando racionalização Fuzzy no processo. Isso ocorre por meio de uma função de pertinência[1], que é aplicada usando os parâmetros de entrada, resultando em uma saída entre 0 e 1. Existem várias funções de pertinência que podem ser aplicadas nesse processo, alguns exemplos amplamente aplicados são as funções triangular, trapezoidal, sino generalizada e Gaussiana [48]. Esse processo inicial é a "fuzzyficação", que resulta em um conjunto "fuzzy". O conjunto resultante dessa primeira etapa é utilizado por um conjunto de regras que é responsável por providenciar uma conclusão útil. Esse processo é a "defuzzyficação",

[1] Função que define o quanto um elemento pertence a um conjunto.

que resulta na saída final. Por exemplo, para um problema de classificação, esse valor seria o rótulo a qual o conjunto de entrada pertence.

A função triangular é dada por:

$$\zeta_{tri} = max\left(min\left(\frac{x - a_{tri}}{b_{tri} - a_{tri}}, \frac{c_{tri} - x}{c_{tri} - b_{tri}}\right), 0\right), \tag{3.1}$$

em que x é o valor a ser analisado, e a_{tri} é o limite inferior; b_{tri} indica o centro (valor em que o grau de pertinência é 1) e c_{tri} é o limite superior tal que $a_{tri} < b_{tri} < c_{tri}$.

O cálculo do grau de pertinência com a função trapezoidal é obtido por:

$$\zeta_{tra} = max\left(min\left(\frac{x - a_{tra}}{b_{tra} - a_{tra}}, 1, \frac{d_{tra} - x}{d_{tra} - c_{tra}}\right), 0\right), \tag{3.2}$$

em que x é o valor a ser avaliado, a_{tra} é o limite inferior, b_{tra} é o limite de suporte inferior, c_{tra} é o limite de suporte superior e d_{tra} é o limite superior tal que $a_{tra} < b_{tra} \leq c_{tra} < d_{tra}$.

A função do sino generalizada é calculada por:

$$\zeta_s = \frac{1}{1 + \left|\frac{x - c_s}{a_s}\right|^{2b_s}}, \tag{3.3}$$

em que x é o valor a ser analisado, c_s é o centro, a_s determina a abertura e b_s geralmente é um valor positivo que determina a inclinação da função.

A função Gaussiana é dada por:

$$\zeta_g = e^{\frac{-(x - c_g)^2}{2\sigma^2}} \tag{3.4}$$

em que x é o valor a ser avaliado, c_g é o centro e σ é o desvio padrão.

Segue um exemplo para compreender o funcionamento da Lógica Fuzzy. Dada a necessidade de decidir se a temperatura da água está ideal, um Sistema de Inferência Fuzzy pode ser aplicado. Supondo que a temperatura ideal da água é em torno de 50°C, na Lógica Booleana a temperatura seria ideal (50°C) ou não (valores que não sejam 50°C). Na lógica Fuzzy uma função de pertinência é aplicada para decidir o quão ideal a temperatura da água se encontra. Para tanto, os graus de pertinência dos valores $x \in [0, 100]$ são analisados com as funções triangular, trapezoidal, Gaussiana e sino generalizada.

A Figura 6 apresenta os graus de pertinência obtidos ao aplicar a função triangular para valores de x com os parâmetros $a_{tri} = 20, b_{tri} = 50$ e $c_{tri} = 70$. Como pode ser observado, o único valor que possui o grau de pertinência 1 é 50, que se encontra no centro da função. Conforme os valores se afastam do centro os graus de pertinência vão decaindo, e para $x \leq 20$ ou $x \geq 70$ o grau de pertinência se encontra em 0.

A função trapezoidal funciona de uma forma diferente, pois ela atribui um intervalo de valores em que o grau de pertinência é 1. Para esse exemplo, os valores dos parâmetros

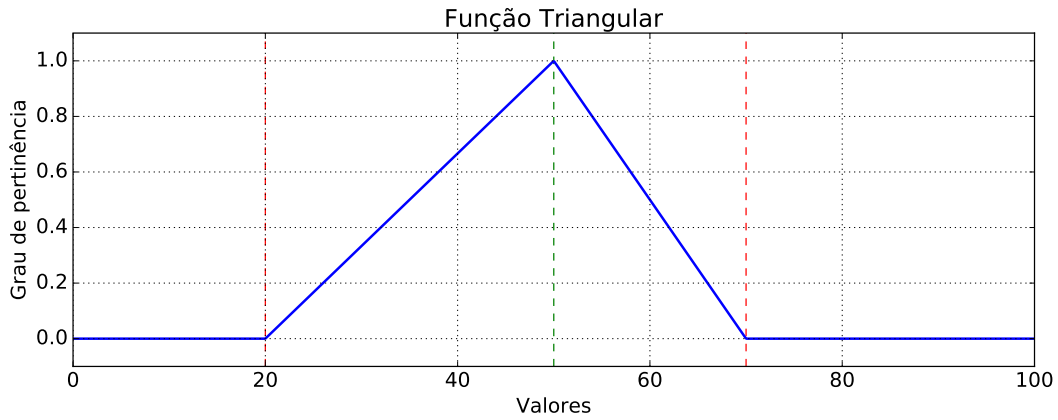


Figura 6 – Exemplo de graus de pertinência usando a função triangular.

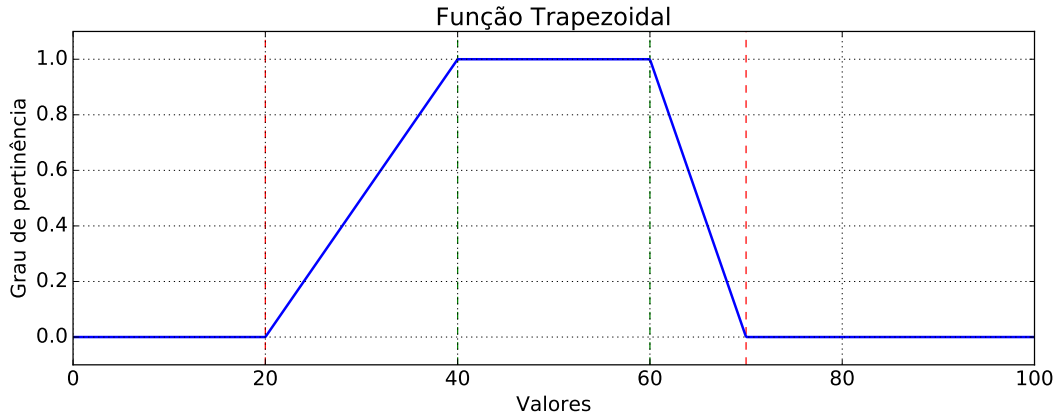


Figura 7 – Exemplo de graus de pertinência usando a função trapezoidal.

utilizados foram $a_{tra} = 20$, $b_{tra} = 40$, $c_{tra} = 60$, $d_{tra} = 70$. Conforme pode ser observado na Figura 7, os valores de $x \geq 40$ e $x \leq 70$ possuem o grau de pertinência a. Valores que não estão dentro desse intervalo apresentam graus de pertinência que diminuem linearmente de acordo com os parâmetros, e para $x \leq 20$ e $x \geq 70$ possuem o grau de pertinência 0.

As Figuras 9 e 8 apresentam os graus de pertinência para a função generalizada de sino e Gaussiana, respectivamente. Para a função de sino, os parâmetros utilizados foram: $a_s = 10$, $b_s = 2$ e $c_s = 50$. Para a função Gaussiana, os parâmetros foram $c_g = 10$ e $\sigma = 10$. Para essas duas funções, o importante é a diferença entre o centro e o valor analisado. Percebe-se que o grau de pertinência para 40 e 60 é o mesmo, uma vez que a diferença até o centro é 10 para ambos os valores.

Para "defuzzyficar" os resultados dessa entrada, um conjunto simples de regras podem ser aplicadas, como apresentado na Equação 3.5. Nesse conjunto de regras, 4 estados são possíveis: "não ideal", "praticamente não ideal", "quase ideal" e "ideal". Assim, a temperatura de entrada é analisada e, com base no grau de pertinência, uma classificação é atribuída. Nota-se que, para diferentes funções de pertinência, a saída pode ser diferente

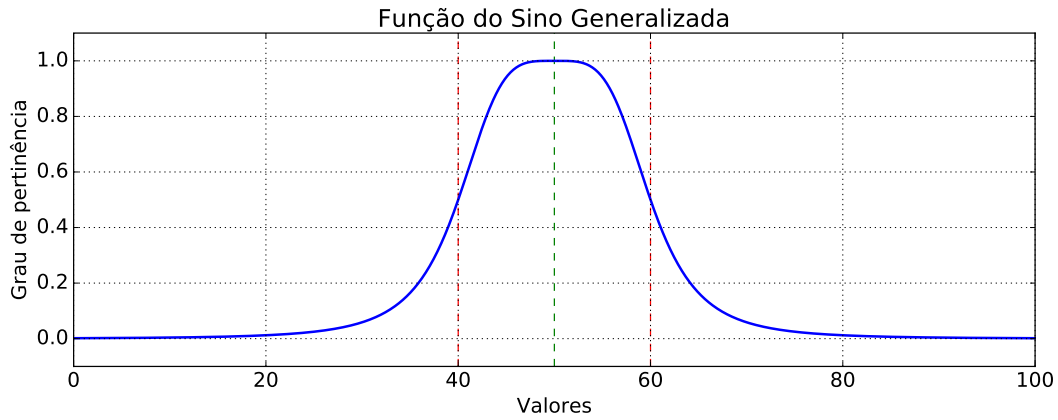


Figura 8 – Exemplo de graus de pertinência usando a função generalizada de sino.

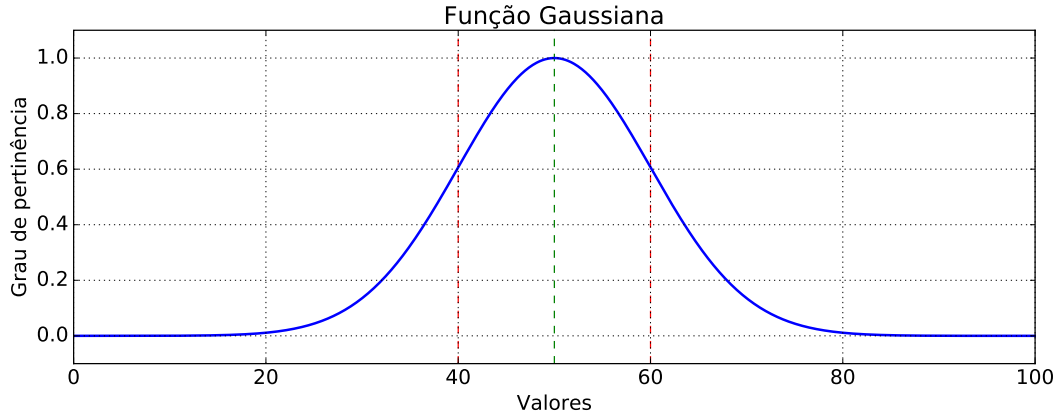


Figura 9 – Exemplo de graus de pertinência usando a função Gaussiana.

para uma mesma temperatura.

$$\begin{aligned}
&\textbf{Regra 1}: \textit{SE}\ \zeta_t < 0,1\ \textit{ENTÃO}\ \text{“}\textbf{não ideal}\text{”}\\
&\textbf{Regra 2}: \textit{SE}\ 0,1 \leq \zeta_t < 0,7\ \textit{ENTÃO}\ \text{“}\textbf{praticamente não ideal}\text{”}\\
&\textbf{Regra 3}: \textit{SE}\ 0,8 \geq \zeta_t < 0,9\ \textit{ENTÃO}\ \text{“}\textbf{quase ideal}\text{”}\\
&\textbf{Regra 4}: \textit{SE}\ \zeta_t \geq 0,9\ \textit{ENTÃO}\ \text{“}\textbf{ideal}\text{”}
\end{aligned} \tag{3.5}$$

Suponha duas temperaturas $t_1 = 50°C$ e $t_2 = 50.01°C$. Pela lógica booleana, essas duas temperaturas são significantemente diferentes, uma delas seria considerada “ideal” e a outra não. Analisando essa situação pela perspectiva humana, não há uma diferença perceptível entre as duas temperaturas. De forma semelhante, a Lógica Fuzzy consegue emular essa análise até um certo limite através dos graus de pertinência, o que pode auxiliar na resolução de problemas em que seria necessário um especialista humano. Assim, é necessário considerar qual das duas lógicas propiciam resultados mais promissores para um dado problema. Como citado anteriormente, a detecção de anomalias em redes

possui características que faz a aplicação da Lógica Fuzzy ser vantajosa.

Neste trabalho a Lógica Fuzzy é aplicada para decidir se uma anomalia está presente no tráfego de rede em um intervalo de tempo. Um problema comum na detecção de anomalias em redes á a taxa elevada de falsos positivos e negativos. Um sistema ideal para o propósito de detecção de anomalias possui a capacidade de corretamente detectá-los com a menor taxa de erros possível. Assim, um alarme é disparado quando existem fortes indicativos da presença de um problema.

4 DETECÇÃO DE ANOMALIAS USANDO ALGORITMOS GENÉTICOS E LÓGICA FUZZY

Neste capítulo é apresentado o Fuzzy-GADS, que aplica o GADS *(Genetic Algorithm for Digital Signature)* para gerar o DSNSF [49], e a Lógica *Fuzzy* para detectar as anomalias, usando informações obtidas a partir de fluxos IP.

4.1 Análise do Tráfego de Rede com DSNSF

O conceito de assinatura digital de segmento de rede foi primeiramente explorado no trabalho de Proença Jr.[50], que utilizava dados coletados por meio do protocolo SNMP para a criação do DSNS (*Digital Signature of Network Segment*), também referido como *baseline*. Essa análise criava uma previsão do comportamento do tráfego, alertando o gerente de redes caso um evento anômalo estivesse ocorrendo. Com base nesse estudo, outros trabalhos que utilizaram análises semelhantes foram realizados.

O trabalho de Lima *et al.*[51] utiliza clusterização baseado em PSO (*Particle Swarm Optimization*), para a criação do DSNS. Adaniya *et al.*[52] aplicam o FHCA (*Firefly Harmonic Clustering Algorithm*), que combina os algoritmos KHM (*K-Harmonic Means*) e FA (*Firefly Algorithm*) para a geração do *baseline*. Os algoritmos apresentados nesses dois trabalhos utilizam dados coletados com o protocolo SNMP para a análise. Entretanto, com o desenvolvimento das redes de computadores, novos protocolos destinados ao auxílio da gerência foram desenvolvidos. Entre eles, detacam-se protocolos baseados em fluxos IP.

A RFC (*Request For Comments*) 3917 define um fluxo IP como um conjunto de pacotes IP passando por um ponto de observação na rede durante um intervalo de tempo, tal que todos os pacotes possuam um conjunto de propriedades em comum [4]. A tupla de IPs e portas de origem e destino e o protocolo é o conjunto de propriedades que é utilizado em muitos trabalhos, incluindo o apresentado nesta dissertação. O protocolo padronizado pela IETF (*Internet Engineering Task Force*) é o IPFIX. Outros protocolos existentes são o sFlow (InMon) [2] e o NetFlow (Cisco) [3], que operam de formas semelhantes. A maior vantagem de se usar tais protocolos é a quantidade de informações do tráfego que eles fornecem, como IPs e portas de origem e destino.

Com isso, trabalhos que exploram o monitoramento com fluxos IP foram desenvolvidos. No trabalho de Assis *et al.*[53], os autores aplicam a metaheurística ACO (*Ant Colony Optimization*) e o algoritmo estatístico HW (*Holt-Winters*) para a caracterização do tráfego de redes com a análise de fluxos IP. O trabalho realizado por Fernandes *et al.*[54], aplica o PCA (*Principal Component Analysis*) e a clusterização *K-means* para

geração dos DSNSFs. Já no trabalho realizado por Pena *et al.*[55], dois DSNSFs gerados pelo ACO e ARIMA (*Autoregressive Integrated Moving Average*) são utilizados para a detecção de anomalias com uma máquina paraconsistente.

Desse modo, percebe-se que muitos trabalhos anteriores já utilizam o DSNSF com diferentes algoritmos. O sistema proposto neste trabalho também utiliza informações do tráfego coletados com um protocolo baseado em fluxos IP. A geração do DSNSF é realizada utilizando um Algoritmo Genético, e a detecção de anomalias é feita por um sistema de Lógica *Fuzzy*. A principal diferença entre este trabalho e os supracitados, encontra-se na detecção, em que é realizada a quantificação dos desvios de comportamento em relação a cada dimensão do tráfego analisada. Isso é feito utilizando graus de anomalia para as dimensões da análise, medindo o desvio entre o real e o observado.

4.2 Visão Geral do Sistema

O sistema apresentado possui duas etapas distintas:

1. Geração do DSNSF usando o GADS e cálculo dos limiares usando dados coletados dos ativos de rede através de protocolos baseados em fluxos IP;
2. Aplicação da Lógica Fuzzy usando o DSNSF e os limiares como parâmetros com o objetivo de determinar se há uma anomalia no tráfego de rede.

O GADS usa fluxos IP coletados de n semanas anteriores para gerar a caracterização, que é uma predição do comportamento do tráfego da rede esperado em um intervalo de tempo. Por exemplo, para gerar o DSNSF de uma segunda-feira, são usadas n segundas-feiras imediatamente anteriores. Assim, caso haja alguma operação que ocorra nesse dia da semana em específico, o padrão dessa operação é incorporado ao DSNSF.

A análise realizada pelo sistema proposto utiliza seis dimensões que são extraídas dos fluxos IP coletados:

- Média de bits por segundo;
- Média de pacotes por segundo;
- Entropia de IP de origem;
- Entropia de IP de destino;
- Entropia de porta de origem;
- Entropia de porta de destino.

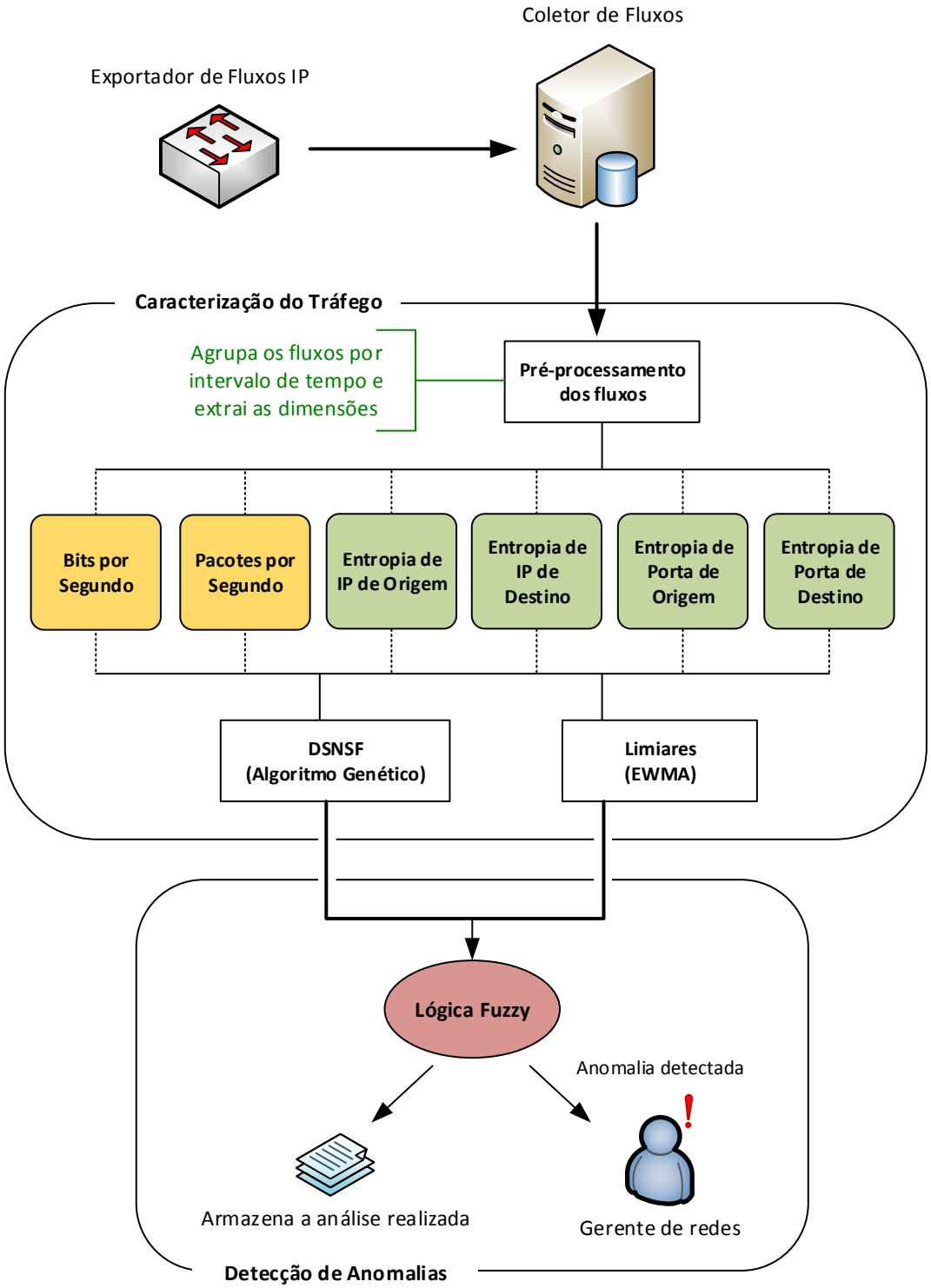


Figura 10 – Sistema de detecção de anomalias proposto.

A Figura 10 representa o processo de coleta dos fluxos até a geração do DSNSF. Primeiramente, os fluxos IP coletados são exportados para um coletor que possui a função de armazená-los. Os fluxos são então agrupados por intervalo de tempo, e as dimensões utilizadas são extraídas na etapa de pré-processamento. Com isso, o DSNSF e os limiares são calculados para que um sistema de Lógica *Fuzzy* possa detectar possíveis anomalias

no comportamento da rede. A análise gerada é armazenada e, caso uma anomalia esteja presente, o gerente da rede é notificado para que medidas possam ser tomadas.

A entropia de Shannon [56] indica o nível de dispersão ou concentração de informações de acordo com a distribuição do conjunto. Dado o conjunto $X = \{c_1, \ldots, c_i, \ldots, c_n\}$, em que c_i é a frequência do elemento i no intervalo de tempo, a Equação 4.1 representa o cálculo da entropia de Shannon para o conjunto X, e a Equação 4.2 representa a soma de todas as ocorrências dos atributos avaliados.

$$H(X) = -\sum_{i=1}^{n} \left(\frac{c_i}{S}\right) \log_2 \frac{c_i}{S} \tag{4.1}$$

onde,

$$S = \sum_{i=1}^{n} c_i \tag{4.2}$$

A entropia é utilizada porque ela sintetiza informações qualitativas (IP de origem, IP de destino, porta de origem e porta de destino) em um valor numérico. Além disso, a entropia é uma função côncava, sendo máxima quando os eventos são equipropráveis. Logo, ataques que utilizam uma porta ou IP em específico podem reduzir significativamente a entropia. Por exemplo, ataques DoS (*Denial of Service*) e *Port Scan* afetam principalmente as dimensões de IP e porta [57], concentrando a distribuição em um IP. Ataques de DDoS (*Distributed Denial of Service*), por exemplo, concentram vários IPs de origem distintos em um IP de destino, o que pode ser verificado ao utilizar a análise de entropias. Para tanto, a entropia é adequada para analisar o comportamento do tráfego da rede.

4.3 Geração do DSNSF usando Algoritmos Genéticos

Para a criação do sistema proposto, alguns detalhes devem ser definidos, como o número de semanas analisadas, o intervalo de tempo e os parâmetros do Algoritmo Genético. Como o tráfego de uma rede é contínuo, é necessário o uso de intervalos de tempo para gerar uma predição do tráfego da rede no dado intervalo.

O intervalo de tempo selecionado deve balancear dois aspectos importantes: quantidade de informações coletadas e tempo de resposta. Caso um intervalo de tempo curto seja selecionado, pode não conter informações o suficiente para estabelecer um padrão, o que resulta em uma predição sem aplicabilidade. Se um intervalo de tempo muito longo é definido, o tempo de resposta para remediar uma possível anomalia na rede é grande. Portanto, um intervalo de tempo aceitável é de um minuto, pois incorpora uma quantidade de informações suficiente e um bom tempo de resposta [58].

A codificação do cromossomo usada na geração do DSNSF é numérica, uma vez que todas as dimensões analisadas podem ser representadas quantitativamente. Além disso, caso uma codificação de caracteres binários fosse utilizada, seria necessário converter a cadeia de caracteres binários para um número, ou realizar o cálculo do *fitness* em caracteres binários, em ambos os casos um custo computacional seria adicionado. O cromossomo poderia ser representado por um vetor numérico, em que cada posição representa uma dimensão. Porém, não há necessidade, uma vez que uma dimensão não depende da outra para a geração do DSNSF e, dessa forma, as otimizações podem ser realizadas em paralelo.

A função *fitness* usada é a distância Euclidiana entre o cromossomo e as entradas usadas na geração do DSNSF. Essa é uma métrica tradicional e amplamente aplicada para a detecção de anomalias [59], portanto ela foi selecionada como a função *fitness*. O cálculo da distância Euclidiana é dada por:

$$f = \sqrt{\sum_{i=1}^{n}(y - x_i)^2}, \tag{4.3}$$

em que y é o valor do cromossomo, x_i representa cada elemento dos dados de entrada do intervalo de tempo e n é o número de entradas.

Com os parâmetros definidos, o GADS é aplicado em cada uma das dimensões analisadas e representado no Algoritmo 2. Para a geração do DSNSF são necessários n vetores de entrada, em que n é o número de semanas utilizadas para gerar o DSNSF. Cada um dos vetores de entrada possui 1 valor para cada intervalo de um minuto, totalizando 1440 pontos (24 horas $\cdot$ 60 minutos = 1440 minutos). Usando os dados de entrada, são calculados os limites inferior e superior, que são usados para gerar a população inicial. Até que o critério de parada seja atingido, o processo de seleção, *crossover* e mutação são aplicados, retornando o melhor cromossomo no fim de todas as iterações. A saída final do algoritmo é um vetor com 1440 elementos, em que cada elemento é o DSNSF do intervalo de um minuto.

O algoritmo apresentado gera a caracterização do tráfego de um dia inteiro para o propósito de análise. Cada intervalo de tempo é otimizado independente dos outros intervalos de tempo. Assim, a otimização é mais rápida do que codificar o cromossomo como a predição do dia todo e os intervalos de tempo como genes do cromossomo. Além disso, é possível realizar essa análise em tempo real, otimizando cada intervalo de tempo assim que os fluxos sejam coletados.

4.4 Cálculo de Limiares usando EWMA

A detecção de anomalias com a Lógica *Fuzzy* requere o cálculo dos limiares, indicando se o comportamento da rede se adequa aos padrões estabelecidos para um dado intervalo de tempo. Para tanto, é usado o EWMA [60, 61] para estabelecer os limiares

Algoritmo 2 Algoritmo Genético usado para a geração de DSNSF.

Require: Conjunto de n vetores com 1440 posições cada
Ensure: Vetor representando o DSNSF com 1440 posições

```
1: for all dados de entrada do
2:     Calcular o limite inferior
3:     Calcular o limite superior
4: for i = 1 : 1440 do
5:     Gerar a população para o intervalo de tempo
6:     for j = 1 : número máximo de gerações do
7:         Calcular o fitness da população
8:         for k = 1 : número de cromossomos na população do
9:             Seleção (torneio)
10:            Crossover (média dos cromossomos)
11:            Mutação (adiciona um pequeno valor)
12:    Calcular o fitness da população
13:    Adicionar o melhor cromossomo ao vetor do DSNSF
14: return DSNSF
```

inferior e superior. O EWMA necessita de um histórico do tráfego para a sua aplicação. O conjunto de dados usado para gerar o DSNSF também é utilizado pelo EWMA para calcular os limiares.

O EWMA usa pesos exponenciais de acordo com a distância temporal entre o elemento analisado e as observações de entrada. O EWMA é calculado por:

$$z_i = \lambda x_i + (1 - \lambda) z_{i-1}, \tag{4.4}$$

em que z_i é o valor da média ponderada para o instante i, λ é uma constante entre 0 e 1 e x_i é o valor observado no instante de tempo i. Para o EWMA, o instante de tempo é em relação ao tráfego usado para a geração do DSNSF, possuindo valores de 1 a n, em que n é o número de semanas utilizadas para a análise.

O termo z_0 é o alvo da análise; para o GADS seria o valor do tráfego real da dimensão analisada. Em certas ocasiões, a média pode ser o z_0.

A seguinte equação é utilizada para calcular a variância de z_i:

$$\sigma_{z_i}^2 = \sigma^2 \left(\frac{\lambda}{2 - \lambda} \right) [1 - (1 - \lambda)^{2i}], \tag{4.5}$$

em que σ^2 é a variância dos dados usados para a geração do DSNSF. Pode-se observar que, quanto maior o valor de i, o termo $[1 - (1 - \lambda)^{2i}]$ converge para 1.

Portanto, se $[1 - (1 - \lambda)^{2i}] = 1$, então o desvio padrão de z_i pode ser reduzido,

como:

$$\begin{aligned}\sigma_{z_i} &= \sqrt{\sigma^2 \frac{\lambda}{(2-\lambda)} \cdot 1} \\ &= \sigma \sqrt{\frac{\lambda}{(2-\lambda)}}\end{aligned} \tag{4.6}$$

No gráfico de controle do EWMA tradicional, a média aritmética é utilizada como o centro para o cálculo dos limiares. Porém, para esse trabalho, o DSNSF gerado pelo Algoritmo Genético servirá como o centro. O cálculo dos limiares inferior e superior em um dado instante é dado por:

$$limiares = DSNSF \pm L\ \sigma\ \sqrt{\frac{\lambda}{(2-\lambda)}}, \tag{4.7}$$

calculado usando o termo resultante da Equação 4.6, em que L é a amplitude dos limiares.

É possível observar que os valores de λ e L devem ser definidos para gerar os limiares usando o EWMA. De acordo com [62], os valores mais aplicados de λ e L são, respectivamente, $0.1 \leq \lambda \leq 0.25$ e $L = 3$.

O cálculo do EWMA resulta no intervalo de valores das flutuações normais entre o DSNSF gerado e o tráfego atual da rede. Com isso, é possível analisar se o tráfego atual da rede comporta dentro dos limites esperados.

4.5 Detecção de Anomalias com Lógica Fuzzy

De acordo com os trabalhos de Molnar e Moczar[63] e Lakhina, Crovella e Diot[57], a maioria das anomalias presentes no tráfego de rede afeta mais de duas dimensões. Portanto, uma análise compreendendo combinações das dimensões é necessária. Uma forma de atingir isso seria analisar o número de dimensões em que o tráfego real está fora dos limiares definidos, como é feito no trabalho de Carvalho *et al.*[64]. Porém, essa metodologia não leva em consideração os desvios entre o DSNSF e o tráfego real. Por exemplo, se todas as dimensões apresentarem desvios significantes do esperado, mas estão dentro dos limiares, um alarme não é gerado, mesmo se uma anomalia estivesse presente. Portanto, neste trabalho é apresentada uma aplicação da lógica Fuzzy na etapa de geração de alarmes.

O processo de "fuzzyficação" apresentado neste trabalho utiliza o DSNSF e os limiares de cada dimensão analisada como parâmetros para a função de pertinência. A função de pertinência usada neste trabalho é uma adaptação da função Gaussiana, e é

dada por:

$$\zeta_k = 1 - \mathrm{e}^{\frac{-(x_k - \hat{x_k})^2}{2\theta_k^2}}, \tag{4.8}$$

em que k representa a dimensão do tráfego da rede, x_k é o tráfego real, $\hat{x_k}$ é o DSNSF e θ_k é o limiar obtido pelo EWMA. Esse processo resulta em seis saídas, que são agregadas em um único valor por uma soma dos graus de pertinência. Assim, o desvio entre o tráfego esperado e o real é quantificado, o que não é feito caso a detecção de anomalias seja baseada em dimensões anômalas. O processo de "defuzzyficação" compara se o resultado dessa soma se encontra acima de um limiar geral Γ, definido na Seção 4.6. Assim sendo, um alarme é gerado, cabendo ao responsável pela rede tratar o possível problema.

Os graus de pertinência são utilizados para definir se uma anomalia está presente no tráfego de rede. Com isso, as regras apresentadas nas Equações 4.9 e 4.10 são usadas para decidir se um intervalo de tempo inclui uma anomalia ou não.

$$\textbf{Regra 1} : \mathit{SE} \sum_{k=1}^{6} \zeta_k \geq \Gamma \ \mathit{ENTÃO} \text{ “}\textbf{anômalo}\text{”} \tag{4.9}$$

$$\textbf{Regra 2} : \mathit{SE} \sum_{k=1}^{6} \zeta_k < \Gamma \ \mathit{ENTÃO} \text{ “}\textbf{normal}\text{”} \tag{4.10}$$

O sistema apresentado para a detecção de anomalias utiliza a previsão e o desvio tolerável para calcular o grau de pertinência nas seis dimensões analisadas. Ao agregrar os graus de pertinência em um único valor, o desvio geral é avaliado.

4.6 Parâmetros do GADS

Antes de aplicar o GADS é necessário definir os parâmetros usados. A Tabela 2 expõe os parâmetros utilizados na execução dos testes apresentados no Capítulo 5.

Foram realizados alguns testes para definir o número de semanas usadas para a geração do DSNSF. A Figura 11 apresenta a média dos *Normalized Mean Square Errors* (NMSE) nas seis dimensões dos dias úteis de quatro semanas distintas:

- 2012b: 1° a 5 de Outubro de 2012;
- 2012a: 22 a 26 de Outubro de 2012;
- 2015a: 5 a 9 de Outubro de 2015;
- 2015b: 19 a 23 de Outubro de 2015.

Tabela 2 – Parâmetros usados para a geração do DSNSF.

Parâmetro	**Valor/Método**
Intervalo de tempo	60 segundos
Semanas utilizadas para análise	4
Função fitness	Distância Euclidiana
Número máximo de gerações	40
Seleção	Torneio
Crossover	Média aritmética entre os cromossomos selecionados
Mutação	3% de chance de adicionar ou subtrair 1% do valor do cromossomo
λ (EWMA)	0,25
L (EWMA)	3,0
Γ	3,9644

O eixo x da Figura 11 é o número de semanas utilizadas para a geração do DSNSF, e o eixo y representa a média do NMSE em todas as dimensões dos dias analisados. É possível observar que, em cada conjunto, a diferença dos erros em relação ao número de semanas usadas é menor do que 1%. Os resultados dos dados de 2012 são um pouco contraditórios, uma vez que apresentam comportamentos opostos. Essa inconsistência pode ter sido causada por alguma diferença nos dados de entrada ou nos processos aleatórios envolvidos na geração do DSNSF. Porém, o resultado com os dados de 2015 mostra que o uso de oito semanas ao contrário de duas apresenta um erro menor.

O uso de poucas semanas para a geração do DSNSF é preferencial, pois requere um menor volume de dados armazenados. Em contrapartida, caso anomalias estejam incorporadas nesses dados, elas podem afetar a geração do DSNSF, resultando em uma predição infiel ao real. Portanto, esses dois aspectos devem ser levados em consideração ao definir o número de semanas utilizados para a análise. Com isso, foi selecionado quatro semanas para a análise, uma vez que a diferença nos erros é baixa. Esse valor equilibra os aspectos mencionados anteriormente, produzindo um resultado adequado.

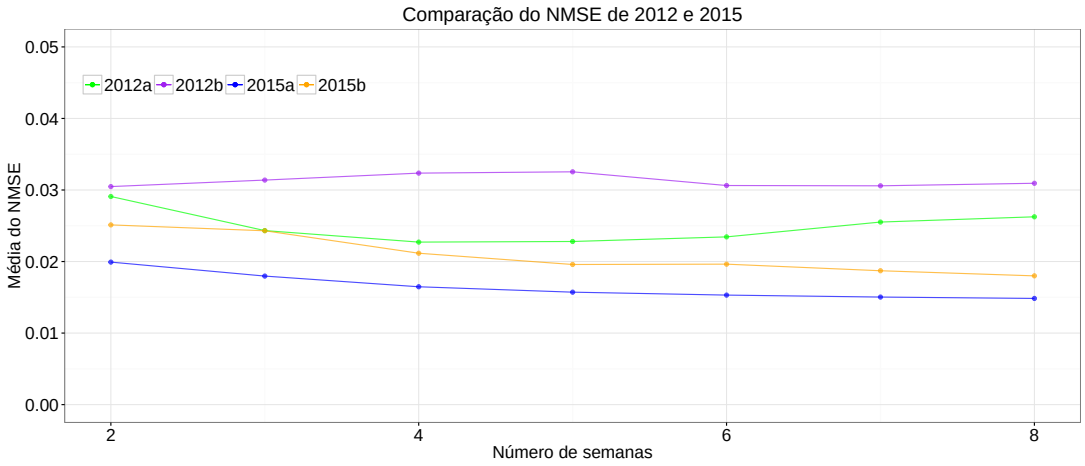


Figura 11 – Teste de NMSE usando de 2 a 8 semanas para a geração do DSNSF.

O número máximo de iterações foi definido como 40, pois é o suficiente para que haja a convergência dos cromossomos presentes na população. A Figura 12 mostra um exemplo da convergência do *fitness*. Nos experimentos realizados, a convergência ocorre em até 15 gerações para uma população de 20 cromossomos e seleção por torneio. Caso a convergência não ocorra em 40 gerações, o algoritmo finaliza a sua execução, retornando a melhor solução encontrada.

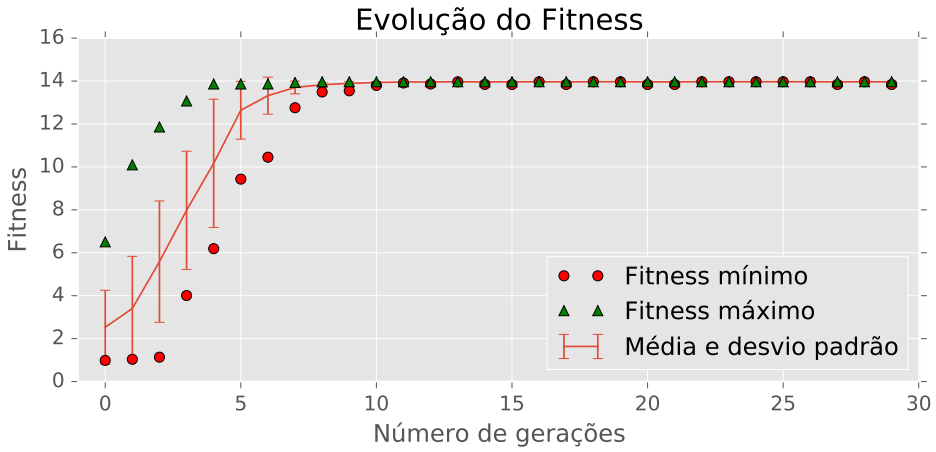


Figura 12 – Evolução do *fitness* da população.

Os métodos de seleção mais utilizados são a Roleta e o Torneio [65]. Neste trabalho o Torneio foi escolhido por possuir uma baixa complexidade computacional, apesar de possuir uma menor taxa de convergência em relação ao método da Roleta.

Os parâmetros selecionados para o EWMA são valores um pouco acima dos sugeridos pela literatura. Foram selecionados os valores de 0,25 e 3,0 para λ e L, respectivamente, por gerarem limiares um pouco maiores a fim de diminuir a taxa de falsos positivos.

Para definir o valor de corte Γ, foram analisados cinco dias da semana (segunda-feira a sexta-feira) com anomalias de *DoS* (*Denial of Service*), *DDoS* e *Flash Crowd*. Com

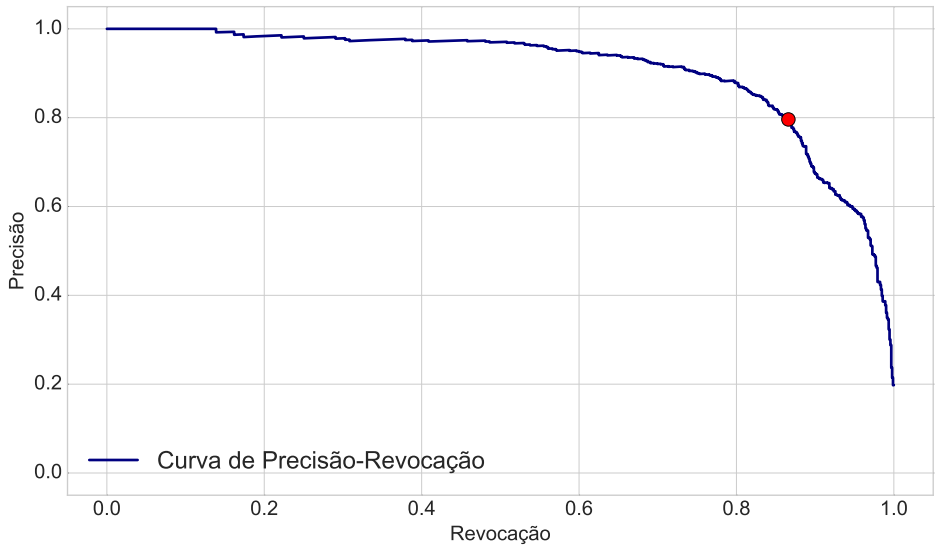


Figura 13 – Curva de Precisão-Revocação para definir o valor de Γ.

isso, foi gerada a curva de *Precision-Recall* para encontrar Γ, que varia o valor do limiar e calcula as métricas de precisão (P) e revocação (R) definidas na Seção 5.1. O valor de corte ótimo encontra-se no limiar em que a soma entre a precisão e a revocação possui o maior valor, ou seja $argmax(P_l + R_l)$, sendo l o limiar. A Figura 13 apresenta a curva gerada com o valor corte $\Gamma = 3,9644$.

Os parâmetros definidos nesta seção são utilizados para executar os testes apresentados no Capítulo 5, em que os experimentos são realizados.

4.7 Exemplo Numérico

Para compreender o funcionamento do sistema proposto, um exemplo numérico de todas das etapas de caracterização e detecção de anomalias é demonstrado. Uma população reduzida de quatro cromossomos é utilizada para o propósito de exemplificação. Dado o vetor de entrada com o tráfego de rede de 4 dias $x = \{x_1 = 150, x_2 = 40, x_3 = 180, x_4 = 165\}$, os operadores genéticos de seleção (torneio), *crossover* (média) e mutação são realizados. Além disso, os cálculos dos limiares pelo EWMA e do grau de pertinência também são expostos.

Primeiramente, é necessário calcular os limites inferior e superior, nesse caso eles são respectivamente $l_{inferior} = 40$ e $l_{superior} = 180$. Com isso, um conjunto de população inicial é gerada tal que os valores dos cromossomos encontram-se no intervalo $[40, 180]$. Para esse exemplo, a população inicial $\mathbf{p_{inicial}}$ é gerada aleatoriamente, sendo ela $\langle c_1 = 71, c_2 = 161, c_3 = 68, c_4 = 168\rangle$. Com uma população inicial gerada é necessário calcular os valores de *fitness* para que as operações genéticas possam ser realizadas. Para essa população, os valores de *fitness* são $\mathbf{f_{p_{inicial}}} = \langle f_1 = 167, f_2 = 123, f_3 = 170, f_4 = 130\rangle$.

Nota-se que os valores foram truncados para facilitar a exemplificação. A Equação 4.11 apresenta o cálculo do valor de *fitness* para c_1.

$$
\begin{aligned}
f &= \sqrt{\sum_{i=1}^{n} (y - x_i)^2} \\
&= \sqrt{(c_1 - x_1)^2 + (c_1 - x_2)^2 + (c_1 - x_3)^2 + (c_1 - x_4)^2} \\
&= \sqrt{(71 - 150)^2 + (71 - 40)^2 + (71 - 180)^2 + (71 - 165)^2} \\
&= \sqrt{(-79)^2 + (31)^2 + (-109)^2 + (-94)^2} \\
&= \sqrt{6241 + 961 + 11881 + 8836} \\
&= \sqrt{27919} \\
&= \lfloor 167,0897 \rfloor \\
&= 167
\end{aligned}
\tag{4.11}
$$

Seguindo o algoritmo proposto para a geração do DSNSF, primeiro é feita a seleção dos cromossomos. No método de seleção aplicado, dois cromossomos aleatórios são escolhidos, e o que possuir o melhor *fitness* é selecionado. Para o DSNSF, o melhor *fitness* é o elemento com a menor distância dos dados de entrada. Supondo que os cromossomos c_1 e c_2 sejam escolhidos, c_2 é o cromossomo selecionado, uma vez que c_2 possui uma distância menor, pois $f_2 < f_1$. O processo de seleção continua até que o número de pares de cromossomos selecionados seja igual ao tamanho da população.

Com os cromossomos selecionados, o processo de *crossover* inicia. No algoritmo implementado, o *crossover* é feito pela média entre os valores dos cromossomos. Assumindo que os cromossomos c_2 e c_4 foram selecionados, o *crossover* resulta em um cromossomo $c_{r_{2,4}}$ com o valor de 164. O *crossover* é realizados para todos os pares de cromossomos selecionados, mantendo o tamanho da população inicial.

Por fim, o processo de mutação é feito aleatoriamente, selecionando um valor entre $[0, 1]$. Caso esse valor seja menor do que a taxa de mutação, $0,03$ (3% de chance de mutação), um valor correspondente a 1% do valor do cromossomo é adicionado ou subtraído. Portanto, há uma chance de 1,5% de adicionar e 1,5% de subtrair 1% do valor do cromossomo. Como exemplo, suponha que o cromossomo $c_{r_{2,4}}$ sofra uma mutação, o valor resultante pode ser 165, caso seja adicionado, ou 163 caso seja subtraído. Com isso, a população sofre pequenas modificações que podem levar a soluções mais promissoras e até mesmo evitar mínimos locais.

Assim, o *fitness* da nova população é calculado, substituindo a população antiga. Esse laço continua até que o número máximo de gerações seja atingido, finalizando a execução do algoritmo e retornando o melhor cromossomo obtido. O valor do cromossomo representa o tráfego esperado para uma dimensão em um intervalo de tempo, que

é o DSNSF. Para calcular o grau de pertinência, é necessário obter o valor do limiar pelo EWMA. Nota-se que o cálculo do EWMA não requer o DSNSF, portanto, esse processo pode ser feito paralelo à geração do DSNSF. O desvio padrão (σ) da entrada é 55, utilizando a Equação 4.7 essa entrada, resulta no valor 62. Assim, é possível utilizar a Equação 4.8 para calcular o grau de pertinência. A Equação 4.12 mostra o cálculo do EWMA utilizando como entrada o x de exemplo.

$$\begin{aligned} EWMA &= L\sigma\sqrt{\frac{\lambda}{2-\lambda}} \\ &= 3 \cdot 55 \cdot \sqrt{\frac{0,25}{2-0,25}} \\ &= 165 \cdot \sqrt{0,1428} \\ &= 165 \cdot 0,3779 \\ &= \lfloor 62.3641 \rfloor \\ &= 62 \end{aligned} \tag{4.12}$$

Com os valores do DSNSF e do EWMA obtidos, o grau de pertinência ζ_k que representa o nível de anomalia presente em uma dimensão pode ser calculado. Como parâmetros para o cálculo do grau de pertinência, o DSNSF, o EWMA e o valor do tráfego real são utilizados. Para demonstrar a obtenção do grau de pertinência, suponha que o tráfego tenha sido de 125. Assim, a Equação 4.13 apresenta o cálculo do grau de pertinência.

$$\begin{aligned} \zeta_k &= 1 - e^{\frac{-(x_k - x^{'}_k)^2}{2 \cdot \theta_k^2}} \\ &= 1 - 2,7182^{\frac{-(125-140)^2}{2 \cdot 62^2}} \\ &= 1 - 2,7182^{\frac{-(-15)^2}{(2) \cdot 3844}} \\ &= 1 - 2,7182^{\frac{-225}{7688}} \\ &= 1 - 2,7182^{-0,0292} \\ &= 1 - 0.9711 \\ &= 0,0289 \end{aligned} \tag{4.13}$$

Nesse exemplo, o grau de pertinência obtido é de $0,0289$, o que é um valor baixo, indicando que o tráfego observado está condizente com o DSNSF e possui uma variação baixa do esperado. No entanto, o sistema de detecção não analisa somente uma dimensão, seria necessário avaliar os graus de pertinência das outras dimensões para determinar se de fato não existem anomalias no tráfego de rede. Para tanto, suponha que as análises das

outras dimensões tenham sido feitas e que os valores de pertinência $\boldsymbol{\zeta}$ de todas as dimensões tenham sido $\langle \zeta_1 = 0,0289, \zeta_2 = 0,2892, \zeta_3 = 0,7865, \zeta_4 = 0,9253, \zeta_5 = 0,8346, \zeta_6 = 0,5179 \rangle$. Fazendo o somatório dos níveis de pertinência de todo as dimensões, obtém-se o valor $3,3824$, que é menor do que o limiar geral $\Gamma = 3,9644$, indicando que uma anomalia não está presente no tráfego da rede. Supondo agora, que ζ_1 seja um valor maior, por exemplo $0,6289$, o valor do somatório dos graus de pertinência seria $3,9824$, o que excede o limiar Γ. Nesse caso, um alarme seria disparado, e o gerente de redes notificado para tomar as devidas providências.

O procedimento detalhado nesse exemplo é feito para todos os intervalos de tempo em todas as dimensões, disparando alarmes quando necessário.

4.8 Considerações do Capítulo

Neste capítulo foi apresentado o sistema proposto para a detecção de anomalias em redes. O sistema possui duas etapas para atingir o objetivo: geração do DSNSF e a detecção de anomalias. A primeira etapa é realizada utilizando um Algoritmo Genético que gera o tráfego esperado para as seis dimensões usadas na análise. Com base nisso, a detecção é feita com a Lógica *Fuzzy*, que utiliza limiares calculados com o EWMA, além do DSNSF. No Capítulo 5 são apresentados os experimentos realizados para validar o sistema proposto.

5 RESULTADOS

O trabalho apresentado usa o tráfego da rede coletado por meio de protocolos baseados em fluxos IP, como sFlow e IPFIX. A partir do histórico do tráfego de rede, um Algoritmo Genético é aplicado para implementar um IDS baseado em detecção de anomalias. O sistema apresentado gera o comportamento esperado do tráfego em um segmento da rede para comparar com o real. Além disso, o EWMA é utilizado para a geração dos limiares do tráfego, indicando flutuações normais entre o tráfego previsto e o real. Com essas análises, a Lógica *Fuzzy* é utilizada para classificar o tráfego em um determinado intervalo de tempo como anômalo ou normal. Assim, a caracterização do tráfego é retroalimentada com o tráfego real a cada vez que é executada, se adaptando de acordo com o comportamento recente da rede.

Para analisar o sistema apresentado, foram utilizados cinco cenários. Em cada cenário, os dados coletados e as técnicas avaliadas são diferentes. No primeiro cenário, a caracterização do tráfego gerada com o GADS é analisada de uma forma geral. O cenário 2 apresenta a análise do desempenho da detecção de anomalias usando a Lógica *Fuzzy*. No terceiro cenário, o sistema proposto é utilizado em uma rede com o tráfego baixo comparado com o da UEL. O quarto cenário compara o sistema de detecção de anomalias proposto com o ACODS (*Ant Colony Optimization for Digital Signature*), que utiliza algoritmos diferentes para a geração do DSNSF e detecão. E, por fim, o Cenário 5 aplica o sistema *Fuzzy* de detecção em dois DSNSFs gerados por Algoritmos diferentes: GADS e FADS. A Tabela 3 apresenta um sumário dos experimentos realizados.

Tabela 3 – Cenários utilizados para a execução dos testes.

Cenário	**Dados**	**Protocolo**	**Análise**
Cenário 1	Tráfego de rede da UEL de 2012 e 2015	sFlow	Caracterização do tráfego usando GADS
Cenário 2	Tráfego de rede da UEL de 2016	sFlow	Detecção de anomalias usando Fuzzy-GADS
Cenário 3	Tráfego de rede da UTFPR (Toledo)	IPFIX	Detecção de anomalias usando Fuzzy-GADS
Cenário 4	Tráfego de rede da UEL de 2016	sFlow	Detecção de anomalias usando Fuzzy-GADS e ACODS
Cenário 5	Tráfego de rede da UEL de 2016	sFlow	Detecção de anomalias usando Fuzzy-GADS e Fuzzy-FADS

Para avaliar a capacidade de detecção de anomalias do sistema apresentado, anomalias de rede foram injetadas artificialmente nos arquivos de fluxos IP. No total, foram utilizados três tipos de anomalias: DoS, DDoS e *Flash Crowd*. As anomalias foram injetadas usando a ferramenta *Scorpius* [66]. O *Scorpius* é uma ferramenta que auxilia na análise de detecção de anomalias, injetando fluxos anômalos nos arquivos que representam os fluxos coletados por meio do protocolo sFlow. Essa ferramenta simula anomalias de DoS, DDoS, *Port Scan* e *Flash Crowd*. Assim, é possível utilizar coletas de redes reais para analisar o desempenho de sistemas de detecção de anomalias, o que acarreta em um aumento de *datasets* disponíveis para a análise. Neste trabalho, essa ferramenta foi utilizada para simular anomalias nos cenários de experimentos 2, 3, 4 e 5.

5.1 Métricas de Avaliação

As métricas utilizadas para medir o desempenho do sistema proposto foram: *Normalized Mean Square Error* (NMSE), Correlação, teste de Bland-Altman, precisão, acurácia, *F-measure* e taxa de falsos positivos. Essas métricas foram usadas nos diferentes cenários para analisar a performance do sistema apresentado.

O NMSE [67] é o erro quadrático médio normalizado entre o tráfego real e o DSNSF para os 1440 (24 horas) intervalos analisados em um dia. Com isso, é possível observar o desvio total do DSNSF gerado pelo Algoritmo Genético. Um erro baixo indica que o DSNSF obteve um bom desempenho para caracterizar o tráfego da rede. A Equação 5.1 representa o NMSE, em que n é o número de observações, x_i e y_i são, respectivamente, o tráfego observado e o DSNSF para a amostra no intervalo i.

$$NMSE = \frac{n \sum_i (x_i - y_i)^2}{\sum_i x_i \sum_i y_i} \tag{5.1}$$

Uma outra métrica que também é utilizada para medir o desempenho em termos de erros é o teste de Bland-Altman, proposto por Bland e Altman [68]. A diferença entre o DSNSF e o tráfego real deve estar contido no intervalo definido por $\overline{d} \pm 1,96\sigma$, em que $\overline{d}$ é a média da diferença entre o tráfego e o DSNSF, e o σ é o desvio padrão de $\overline{d}$. Os autores afirmam que no mínimo 95% das diferenças devem estar dentro desses limites para um bom ajuste. A Figura 14 apresenta um exemplo de gráfico de Bland-Altman para a dimensão de bits, em que 94,23% dos pontos encontram-se na faixa ótima, ficando um pouco abaixo do ideal.

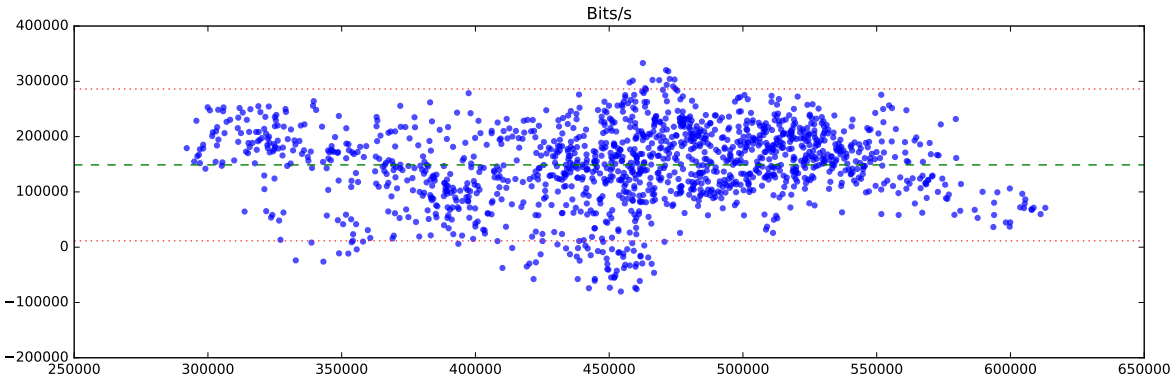


Figura 14 – Exemplo de teste de Bland-Altman para a dimensão de bits.

O coeficiente de correlação r indica a relação entre dois conjuntos de dados, com o valor variando entre -1 a 1, sendo que -1 indica que os conjuntos são opostamente relacionados, 0 são totalmente não relacionados e 1 são completamente relacionados. A Equação 5.2 apresenta o cálculo de r, em que $\mathbf{x}$ e $\mathbf{y}$ são os vetores de dados a serem

analisados e n é a quantidade de valores no vetor.

$$r = \frac{n\sum xy - (\sum x)(\sum y)}{\sqrt{n\sum x^2 - (\sum x)^2}\sqrt{n\sum y^2 - (\sum y)^2}} \tag{5.2}$$

As métricas que avaliam o desempenho estatístico (acurácia, precisão, *F-measure* e taxa de falsos positivos) são amplamente utilizadas em problemas de classificação em que se avaliam o número de erros e acertos nos resultados da classificação. Com isso, é possível analisar a capacidade de corretamente detectar anomalias do sistema apresentado. Para melhor compreender os valores de verdadeiro positivo (T_p), verdadeiro negativo (T_n), falso positivo e (F_p) e falso negativo (F_n), toma-se como exemplo a detecção de anomalias em tráfego de redes. Nesse caso, verdadeiro positivo são as instâncias anômalas que foram corretamente detectadas, enquanto verdadeiro negativo são as instâncias sem anomalias corretamente identificadas. Falso positivo ou alarme falso são as instâncias não anômalas detectadas como anômalas, e o falso negativo são as instâncias anômalas não detectadas. Essas métricas servem como base para várias outras, como as apresentadas nas Equações 5.3, 5.4, 5.5, 5.6 e 5.7, que apresentam respectivamente o cálculo da acurácia, taxa de falsos positivos, precisão, revocação (ou taxa de verdadeiros positivos) e *F-measure*.

$$A = \frac{(T_p + T_n)}{(T_p + T_n + F_p + F_n)} \tag{5.3}$$

$$TFP = \frac{F_p}{(F_p + T_n)} \tag{5.4}$$

$$P = \frac{T_p}{T_p + F_p} \tag{5.5}$$

$$R = \frac{T_p}{T_p + F_n} \tag{5.6}$$

$$F1 = 2\frac{PR}{P + R} \tag{5.7}$$

A acurácia representa a proporção de classificações corretas [69] e a taxa de falsos positivos mede o erros gerado na classificação de anomalias (falsos alarmes). O *F-measure* é calculado em termos da precisão e da revocação. A precisão indica a exatidão dos resultados, e a revocação mede a completude dos alarmes corretos, sendo o *F-measure* a média harmônica entre as duas métricas [70]. Assim, a precisão e a revocação são resumidas em um valor escalar, providenciando uma outra medida para analisar a performance dos resultados.

Uma outra métrica utilizada é a curva ROC (*Receiver Operation Characteristics*), que mostra as taxas de verdadeiro positivo e falso positivo [71]. Com isso, é possível analisar os acertos dos intervalos anômalos e os alarmes falsos gerados pela classificação. A área sob a curva ROC também pode ser calculada, indicando qual sistema obteve um melhor desempenho.

5.2 Cenário 1

Esse experimento apresenta uma análise geral do sistema apresentado para a geração do DSNSF utilizando tráfego o coletado na UEL. Primeiramente, os meses de outubro de 2012 e 2015 são analisados usando NMSE do DSNSF criado utilizando Algoritmos Genéticos, mostrando uma diferença no comportamento do tráfego da rede da UEL. Então, uma semana de outubro de 2015 é analisada mais profundamente, apresentando a correlação e o teste de Bland-Altman, além do NMSE.

5.2.1 Outubro de 2012 e 2015

Os dados para essa análise foram coletados usando o protocolo sFlow com amostragens diferentes para os dois anos. Os dados de 2012 foram coletados com a amostragem de 1:256, ou seja, capturando 1 pacote a cada 256, enquanto em 2015, os dados foram coletados com a amostragem de 1:512. No mês de outubro de 2012, foram analisados 21 dias úteis no total. Além dos fins de semana, os feriados também foram removidos (12 e 15 de outubro), pois a universidade não funciona nesses dias. A Figura 15 mostra os valores do NMSE dos testes realizados para o mês de outubro de 2012. Observando a figura é possível concluir que o NMSE para as dimensões de pacotes e entropias fica abaixo de 5%, com exceção de alguns *outliers* identificados, representados por círculos azuis no gráfico. Para bits, o NMSE ficou em torno de 7,5%, com *outliers* atingindo valores acima de 20%. Isso nos mostra que para esse teste, a dimensão de bits é mais imprevisível, uma vez que o erro para essa dimensão obteve valores elevados em comparação com as outras dimensões.

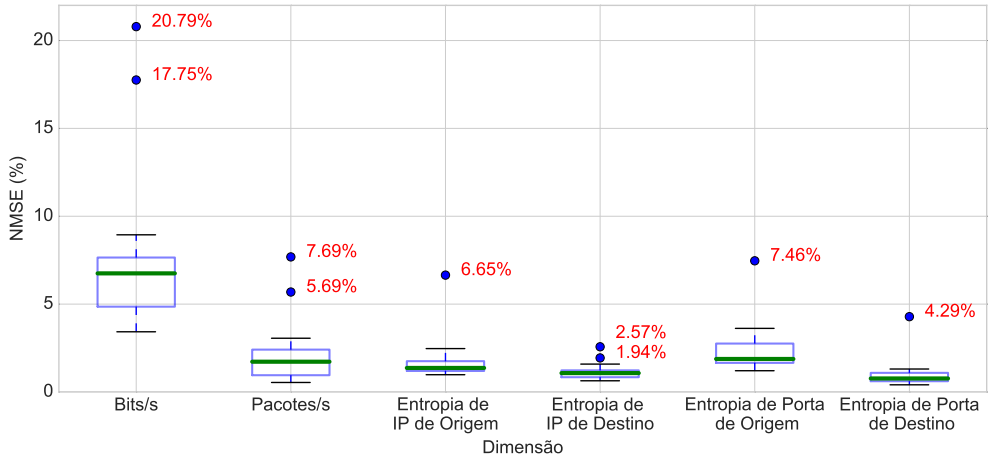


Figura 15 – NMSE das 6 dimensões analisadas do mês de outubro de 2012.

A Tabela 4 indica os *outliers* identificados em cada uma das dimensões. Nos atributos de bits e pacotes, os dias 5 e 30 foram identificados como *outliers*. O dia 23 foi identificado como *outlier* para as entropias de IPs e portas, e o dia 05 para a entropia de IP de destino. Visto que os *outliers* se concentram nos dias 5, 23 e 30 de outubro, pode ser

que algum problema na rede possa ter ocorrido; por consequência esses dias apresentaram valores elevados de erro, levando a serem rotulados como *outliers* em algumas dimensões.

Tabela 4 – *Outliers* identificados na análise de outubro de 2012.

Dimensão	Data	Valor do NMSE
Bits	05/10/2012	20,79%
Bits	30/10/2012	17,75%
Pacotes	05/10/2012	7,69%
Pacotes	30/10/2012	5,69%
Entropia de IP de Origem	23/10/2012	6,65%
Entropia de IP de Destino	05/10/2012	2,57%
Entropia de IP de Destino	23/10/2012	1,94%
Entropia de Porta de Origem	23/10/2012	7,46%
Entropia de Porta de Destino	23/10/2012	4,29%

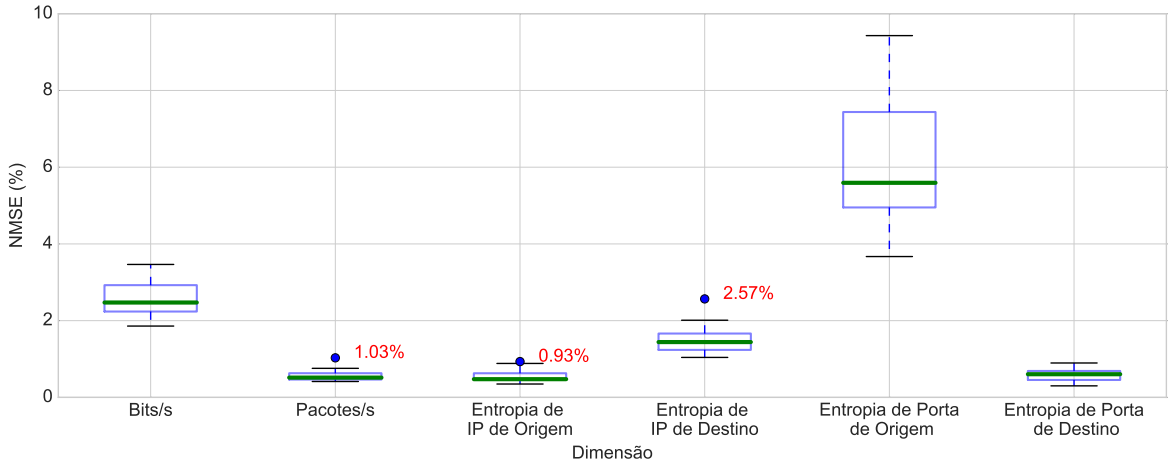


Figura 16 – NMSE das 6 dimensões analisadas do mês de outubro de 2015.

No mês de outubro de 2015, foram analisados 19 dias úteis no total. Além dos fins de semana, foram removidos os feriados do mês (12, 15 e 28 de outubro de 2015). A Figura 16 apresenta os NMSEs calculados no mês avaliado e a Tabela 5 mostra os *outliers* identificados. Em comparação com os resultados obtidos no cenário de 2012, os erros em geral foram menores, exceto em relação à porta de origem que possui valores dos erros mais dispersos e a média maior. Também é possível observar que o número de *outliers* detectados foi significantemente menor do que os resultados de 2012.

Tabela 5 – *Outliers* identificados na análise de outubro de 2015.

Dimensão	**Data**	**Valor do NMSE**
Pacotes	23/10/2015	1,03%
Entropia de IP de Origem	14/10/2015	0,93%
Entropia de IP de Destino	30/10/2015	2,57%

A diferença mais marcante entre as duas análises realizadas é em relação aos valores da entropia de porta de origem. Em outubro de 2012, esses valores estão entre 3% e 4%, enquanto em outubro de 2015 os valores da mesma dimensão estão entre 4% e 8%. Para tanto, uma investigação foi feita em relação a essa diferença e foi observada que há uma menor diversidade de portas de origem no tráfego de 2015.

Tabela 6 – Número de fluxos e portas de origem observados nos dias 24/10/2012 e 29/10/2015.

Observações	**24/10/2012**	**29/10/2015**
Número mínimo de fluxos observados no dia	104	51
Número máximo de fluxos observados no dia	178	86
Média de fluxos observados no dia	120	63
Número mínimo de portas distintas	17	3
Número máximo de portas distintas	67	30
Média de portas distintas	37	12
Número de portas distintas no intervalo de tempo das 8:59 às 9:00	39	16
Portas mais frequentes no intervalo de tempo das 8:59 às 9:00	porta 80, 51 fluxos; porta 8080, 25 fluxos porta 443, 7 fluxos,	porta 443, 35 fluxos; porta 80, 22 fluxos

A Tabela 6 apresenta as informações do número mínimo, máximo e média de fluxos e número mínimo, máximo, e média de portas de origem distintas para intervalos de um minuto dos dias 24/10/2012 e 29/10/2015. Também é apresentado o número de portas distintas e as mais frequentes no intervalo de tempo das 8h59 às as 9h.

Ao analisar a Tabela 6, nota-se que o número de fluxos coletados e de portas de

origem distintas em cada intervalo de tempo em 2012 é próximo ao dobro da quantia que é coletada em 2015. Além disso, houve uma mudança das portas mais frequentes no intervalo de tempo avaliado. As outras portas omitidas na tabela possuem a frequência de uma observação, somente nos dados de 2012 que há outras duas portas que aparecem duas e três vezes. Com isso, os valores da entropia de porta de origem é baixo para o cenário de 2015, e uma pequena mudança nessa dimensão faz com que o DSNSF não seja tão eficiente quanto nos resultados obtidos em 2012. Vale ressaltar que a entropia analisa o nível de dispersão ou concentração das informações, o que acarreta na variação da entropia de portas de origem.

Outra diferença notável entre os dois cenários é a frequência das portas de origem. No cenário de 2015, a porta mais frequente nos fluxos coletados é 443, que é reservada para o protocolo HTTPS (*Hypertext Transfer Protocol Secure*), enquanto em 2012 é a porta 80, reservada para o protocolo HTTP (*Hypertext Transfer Protocol*). Isso mostra uma mudança no protocolo usado para acessar a Internet, o que anteriormente era feito através do protocolo HTTP, agora é feito por HTTPS. Essa concentração de fluxos originando na porta 443 também auxilia na diferença entre os valores de entropia de porta de origem observados em 2012 e 2015.

Essa diferença entre o comportamento do tráfego entre os dois anos mostra que a a rede de computadores é dinâmica e adaptativa. Conforme novas tecnologias e conceitos nessa área vão surgindo, existe a necessidade de adaptar as técnicas de análise e gerenciamento da mesma. Para tanto, algoritmos que fazem a análise do tráfego de rede baseado no histórico recente são ideais, uma vez que com isso é possível acompanhar as possíveis mudanças que possam vir a ocorrer.

5.2.2 Análise da semana de outubro de 2015

Os dias 05 a 09 de outubro de 2015 são analisados apresentando o DSNSF e o tráfego real da rede. Para melhor avaliar o desempenho do DSNSF gerado pelo Algoritmo Genético são apresentados os valores de NMSE, correlação e do teste de Bland-Altman da semana analisada.

As Figuras 17 a 21 apresentam os DSNSFs dos dias analisados. Em alguns intervalos de tempo, o tráfego da rede ultrapassa o limiares estipulados pelo EWMA, mas, em geral, o tráfego se comporta de uma forma condizente com o DSNSF. Percebe-se que o intervalo das 8:00 as 18:00 possui um tráfego mais padronizado, uma vez que os valores dos limiares são menores, indicando um baixo desvio padrão entre os dados de entrada. O desempenho do sistema pode ser melhor analisado pelos valores de correlação, NMSE e Bland-Altman.

A correlação do DSNSF com o tráfego diário é apresentada na Figura 22, e estão agrupadas por atributo. A entropia do IP de destino possui uma correlação menor em

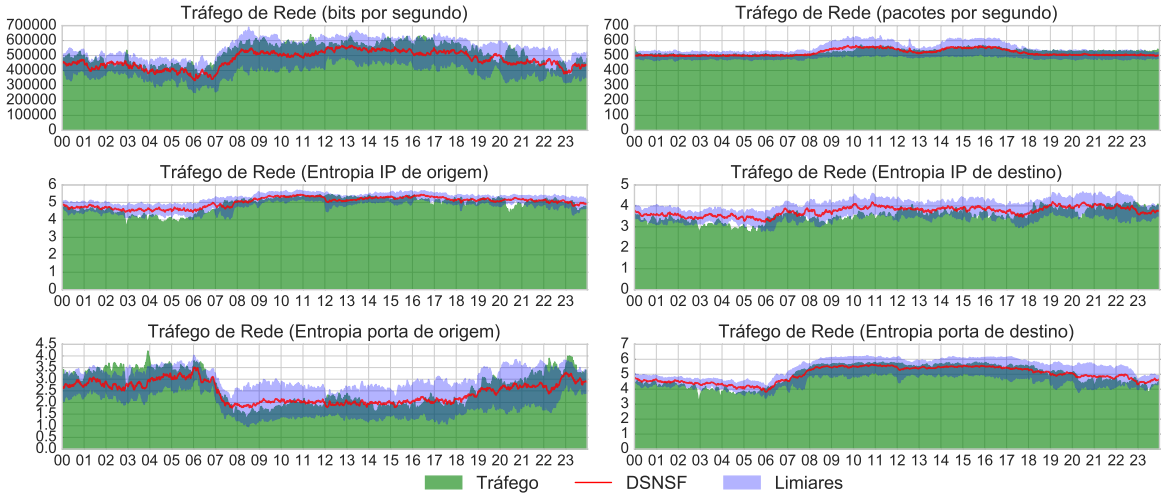


Figura 17 – DSNSF, tráfego real e limiares do dia 05/10/2015.

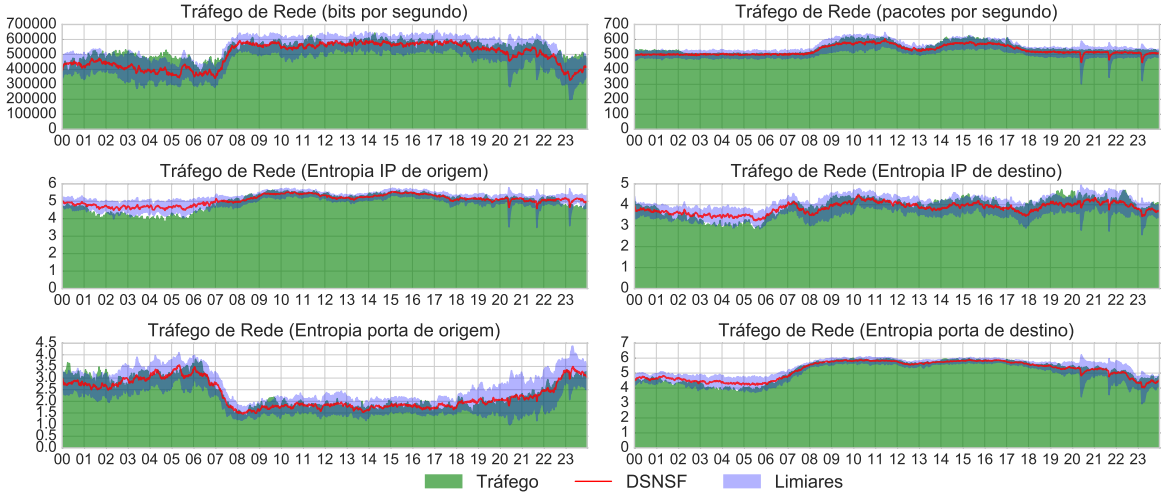


Figura 18 – DSNSF, tráfego real e limiares do dia 06/10/2015.

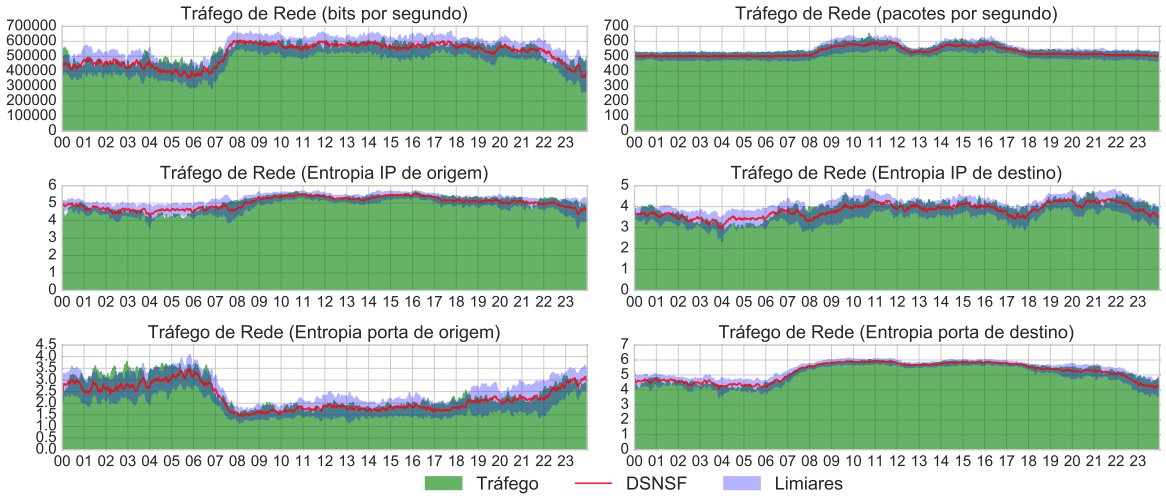


Figura 19 – DSNSF, tráfego real e limiares do dia 07/10/2015.

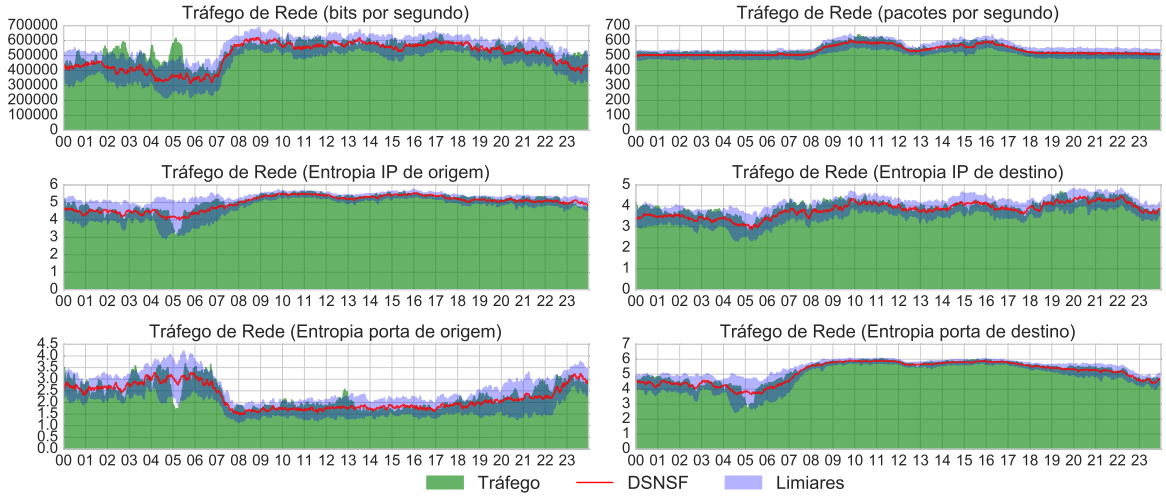


Figura 20 – DSNSF, tráfego real e limiares do dia 08/10/2015.

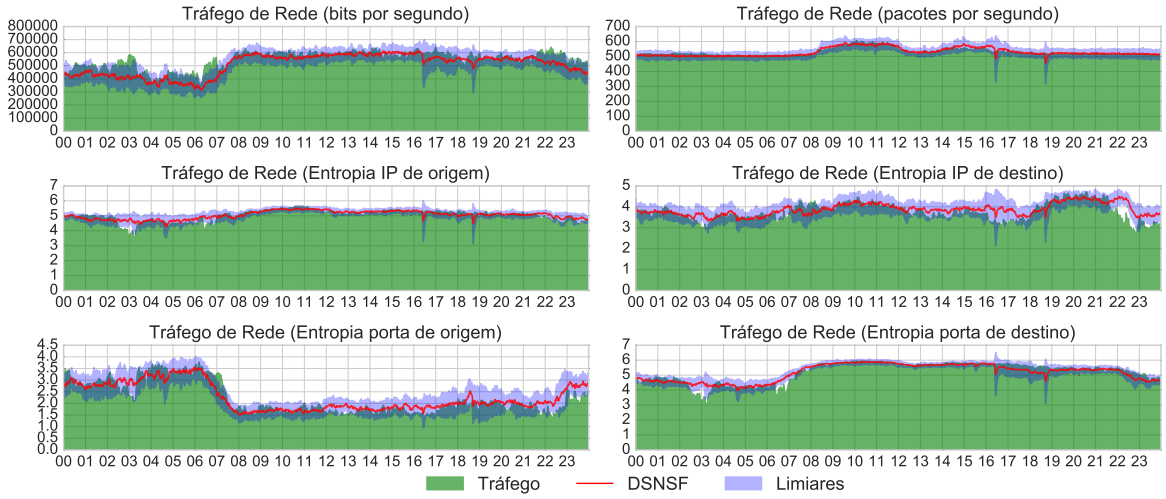


Figura 21 – DSNSF, tráfego real e limiares do dia 09/10/2015.

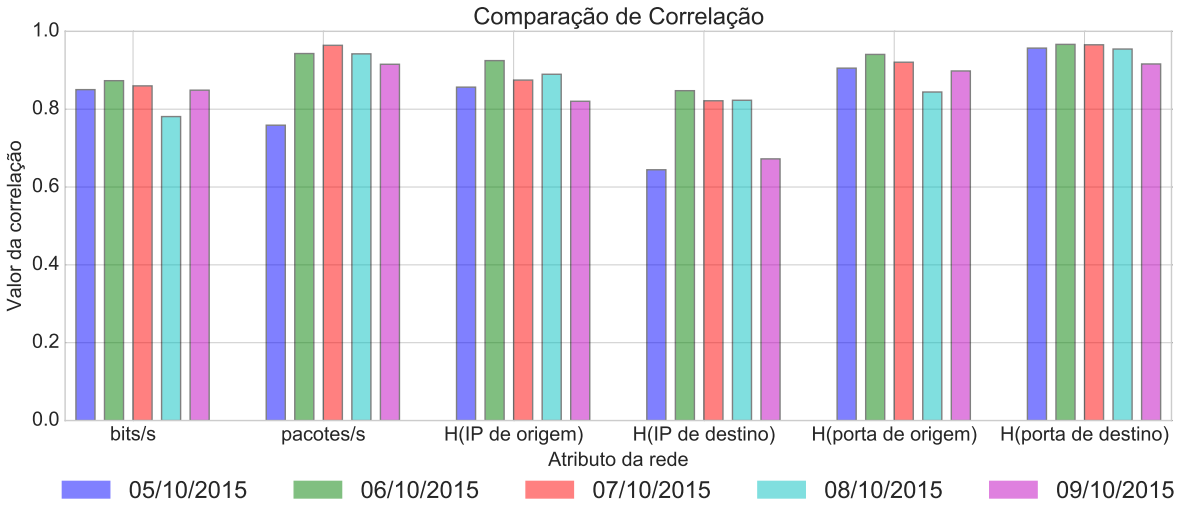


Figura 22 – Correlação dos dias 05 à 09 de outubro de 2015.

comparação com os outros atributos, especialmente em bits. Em contrapartida, a entropia da porta de destino possui a correlação elevada para todos os dias. Devido às variações presentes no tráfego, a correlação não atinge valores acima de 0,9 para todos os atributos.

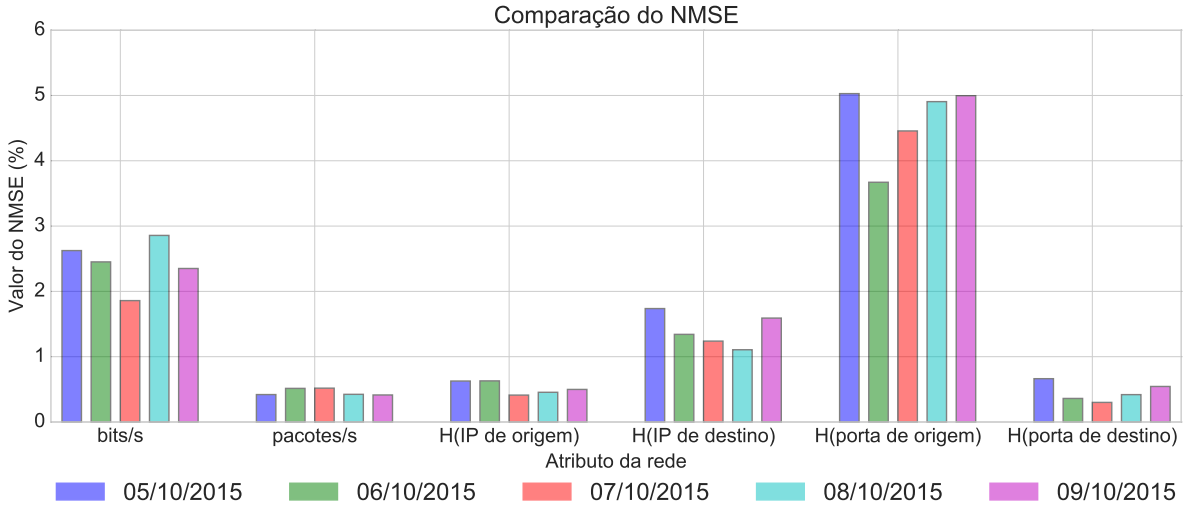


Figura 23 – NMSE dos dias 05 à 09 de outubro de 2015.

A Figura 23 apresenta o NMSE dos dias analisados, que estão agrupados por dimensão. Nessa métrica, a maioria dos dias avaliados obtiveram o erro abaixo de 3%, com a exceção da entropia da porta de origem, que chegou a atingir valores de até 5%. Os erros em pacotes por segundo, entropia de IP de origem e de porta de destino obtiveram valores abaixo de 1%, indicando que o DSNSF ficou próximo ao tráfego real nessas dimensões.

Tabela 7 – Teste de Bland-Altman para os dias 05 à 09 de outubro de 2015.

Atributo	**05/10**	**06/10**	**07/10**	**08/10**	**09/10**
Bits/s	95,06%	95,34%	92,70%	94,30%	94,65%
Pacotes/s	94,30%	96,25%	95,34%	94,86%	94,86%
H(IP de origem)	93,47%	94,86%	92,84%	94,23%	94,93%
H(IP de destino)	95,27%	93,95%	94,72%	96,52%	94,16%
H(porta de origem)	96,04%	94,23%	94,58%	95,13%	93,88%
H(porta de destino)	96,87%	92,70%	92,36%	94,72%	94,09%

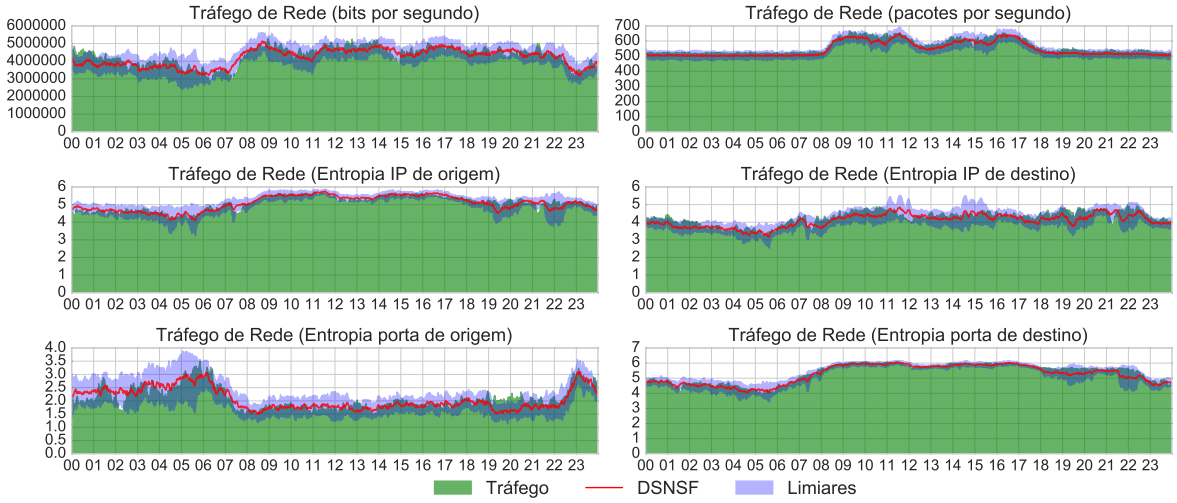
A Tabela 7 mostra o resultado da análise usando o Bland-Altman. Os valores indicam a porcentagem de intervalos de tempo que encontram-se na faixa estabelecida pelo teste ($\overline{d} \pm 1,96\sigma$). Todos os valores se encontram acima de 90%, com a maioria dos valores entre 93% e 95%, sendo o ideal valores acima de 95%. Ainda assim, o DSNSF exibiu um bom resultado, atingindo valores muito próximos do ideal.

Esse cenário analisa somente a geração do DSNSF usando o algoritmo genético, a etapa de detecção de anomalias será avaliada nos outros cenários.

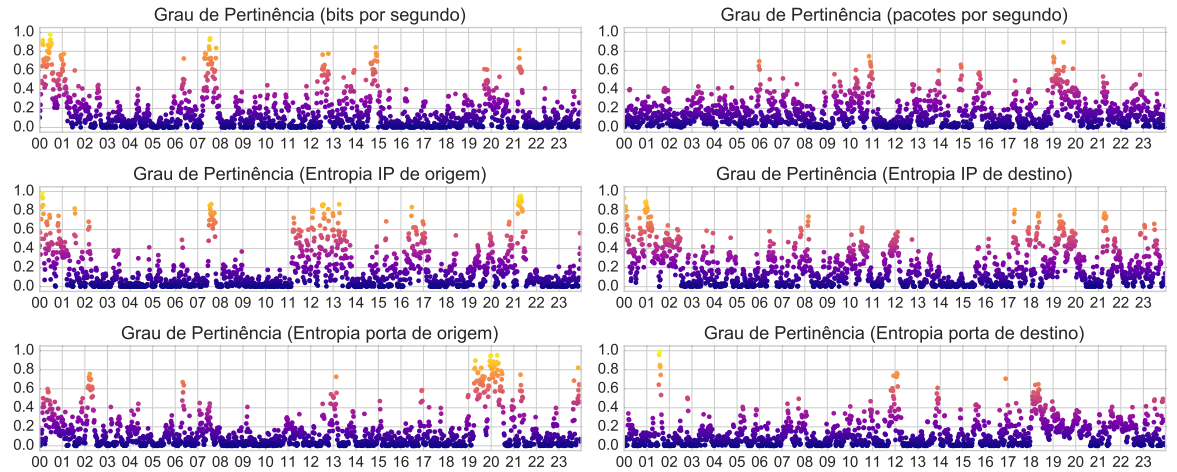
5.3 Cenário 2

Para analisar o desempenho da detecção de anomalias do sistema apresentado, os dias 04 a 08 de julho de 2016 foram utilizados para os testes desse cenário, sendo eles dias úteis da semana. No dia 07 de julho, anomalias de DoS, DDoS e *Flash Crowd* foram injetadas usando a ferramenta Scorpius. As figuras apresentam o tráfego de rede em conjunto com o DSNSF e os limiares, os graus de pertinência e os alarmes gerados pelo sistema proposto ao longo do dia.

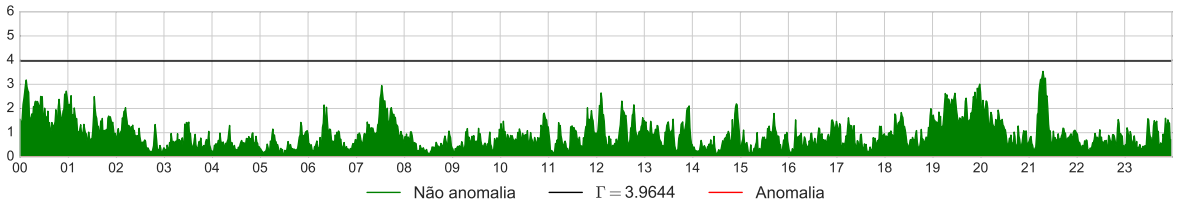
A análise realizada para o dia 04 de julho (segunda-feira) é apresentada na Figura 24. Pode ser observado na Figura 24a que os atributos de bits, pacotes e entropia da porta de origem mudam o comportamento a partir das 8h até as 18h, que é o intervalo de tempo em que a universidade é mais ativa. Os graus de pertinência apresentados na Figura 24b mostram que existem alguns intervalos que possuem o grau pertinência elevado, chegando próximo a 1. Porém, em geral, a rede se enquadra no comportamento esperado, isso pode ser observado pelos alarmes gerados apresentados na Figura 24c, indicando que não há anomalias presentes no tráfego do dia 04 de julho de 2016.



(a) DSNSF, tráfego real e limiares do dia 04 de julho de 2016.



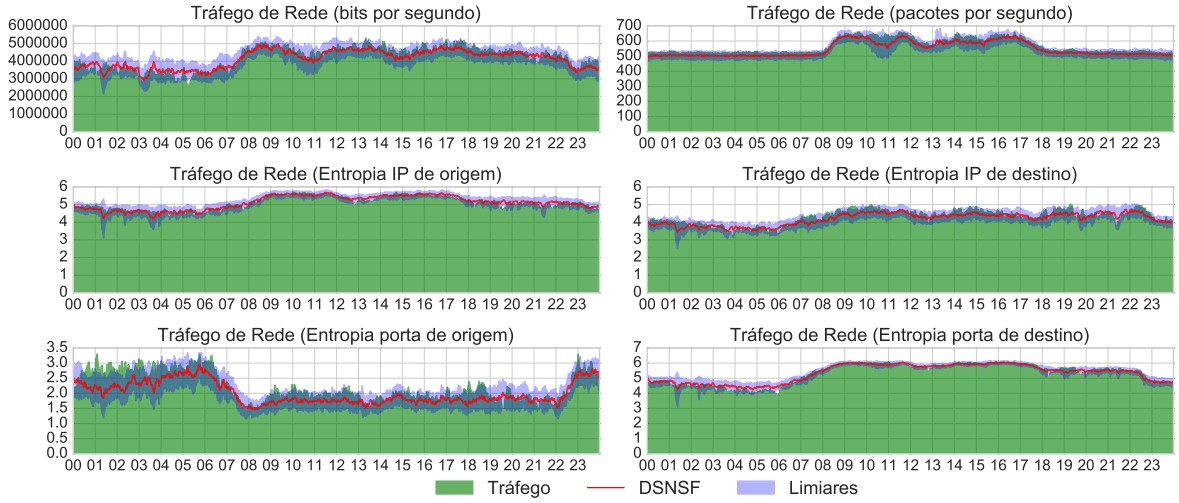
(b) Graus de pertinência do dia 04 de julho de 2016.



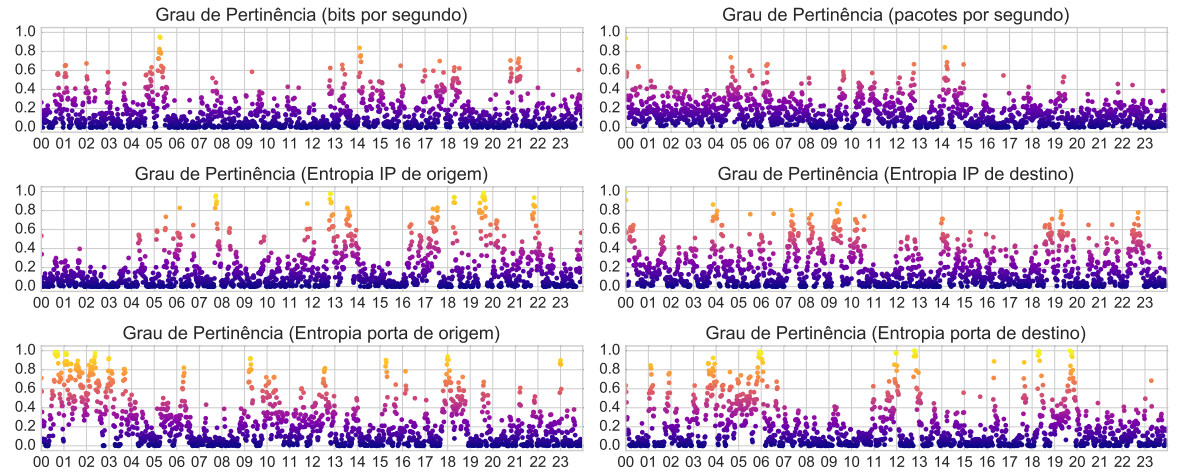
(c) Alarmes gerados para o dia 04 de julho de 2016.

Figura 24 – Análise do dia 04/07/2016.

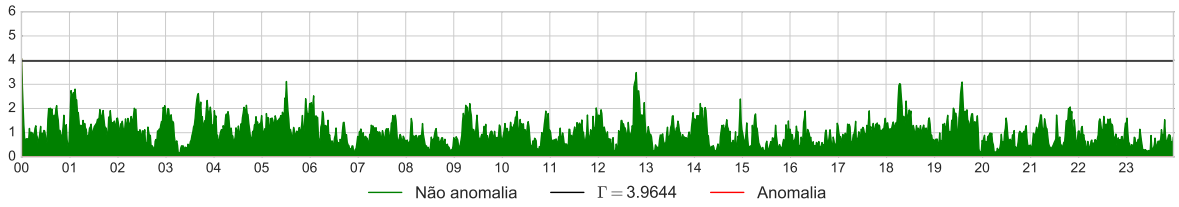
A Figura 25 apresenta a análise os resultados para o dia 05 de julho (terça-feira). O DSNSF e o tráfego são exibidos na Figura 25a apresentando o comportamento comum para uma terça-feira. A Figura 25b indica que há intervalos de tempo com desvios significativos do esperado. Porém, a Figura 25c mostra que não houve anomalias no tráfego ao longo do dia.



(a) DSNSF, tráfego real e limiares do dia 05 de julho de 2016.



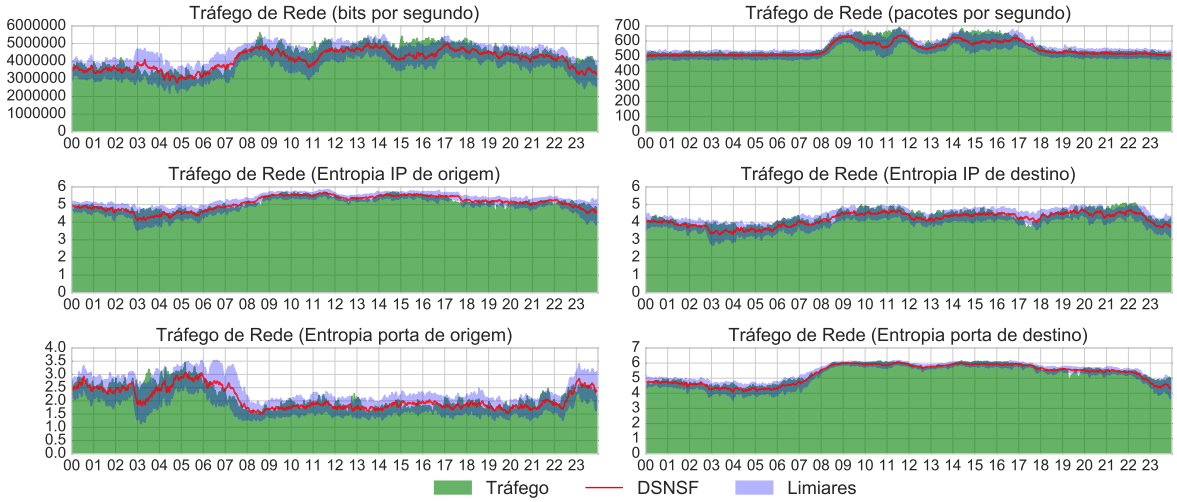
(b) Graus de pertinência do dia 05 de julho de 2016.



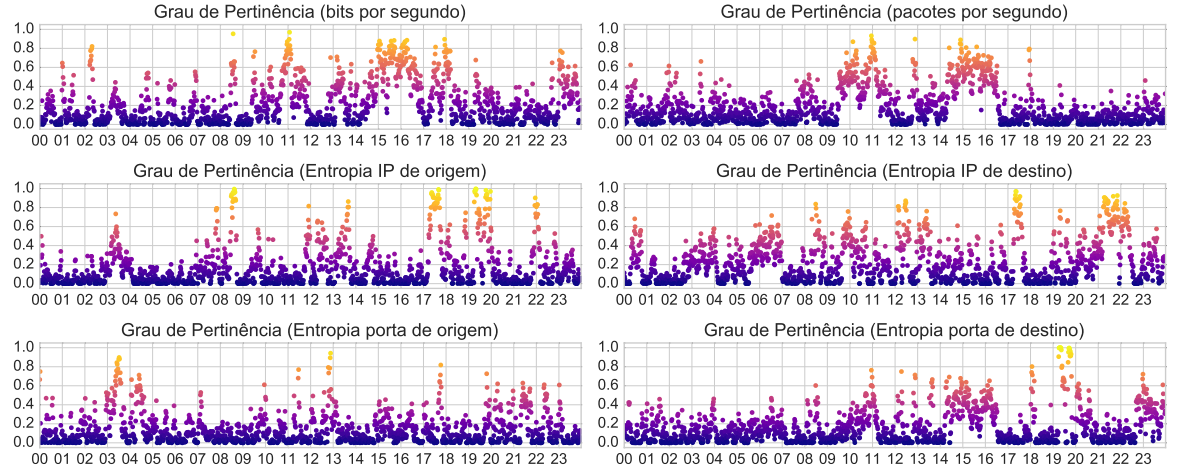
(c) Alarmes gerados para o dia 05 de julho de 2016.

Figura 25 – Análise do dia 05/07/2016.

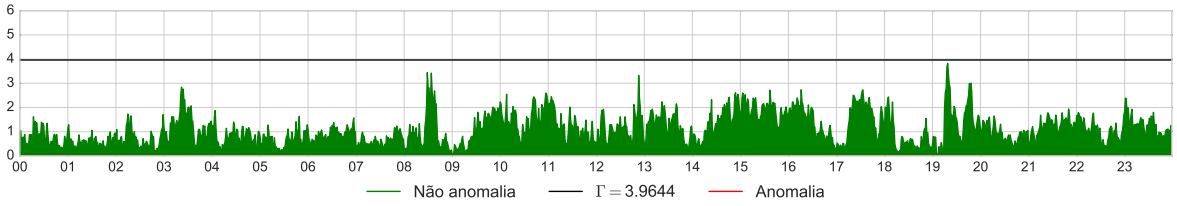
O dia 06 de julho (quarta-feira) apresentou algumas diferenças entre o DSNSF e o tráfego real que persistiram por um longo intervalo de tempo em bits e pacotes. Isso pode ser observado nas Figuras 26a e 26b no intervalo das 9h às 11h e 14h às 17h. Apesar dos graus de pertinência elevados nesses intervalos de tempo para dois atributos, o comportamento da rede ainda se manteve normal, como indicado pelos alarmes na Figura 26c.



(a) DSNSF, tráfego real e limiares do dia 06 de julho de 2016.



(b) Graus de pertinência do dia 06 de julho de 2016.

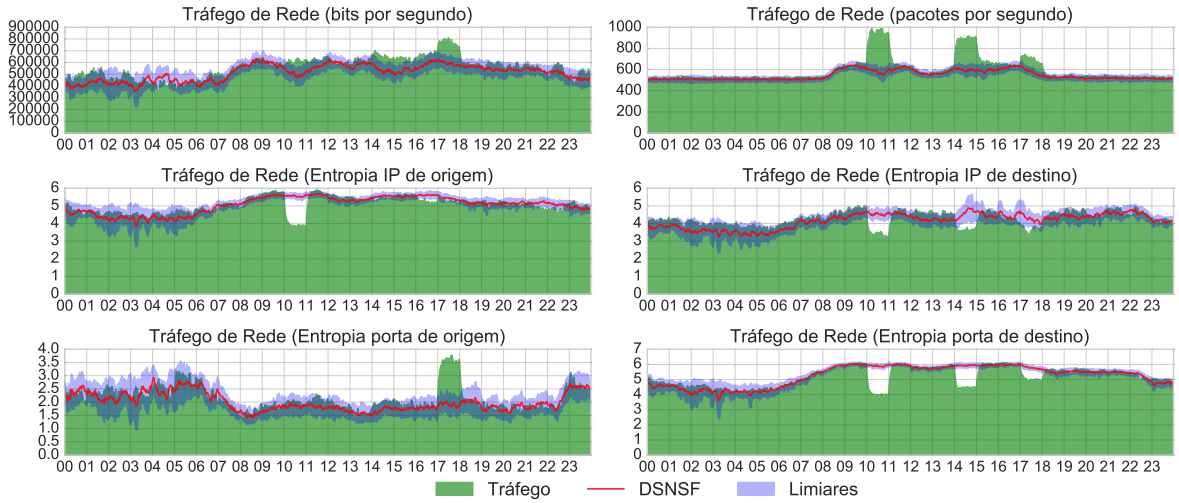


(c) Alarmes gerados para o dia 06 de julho de 2016.

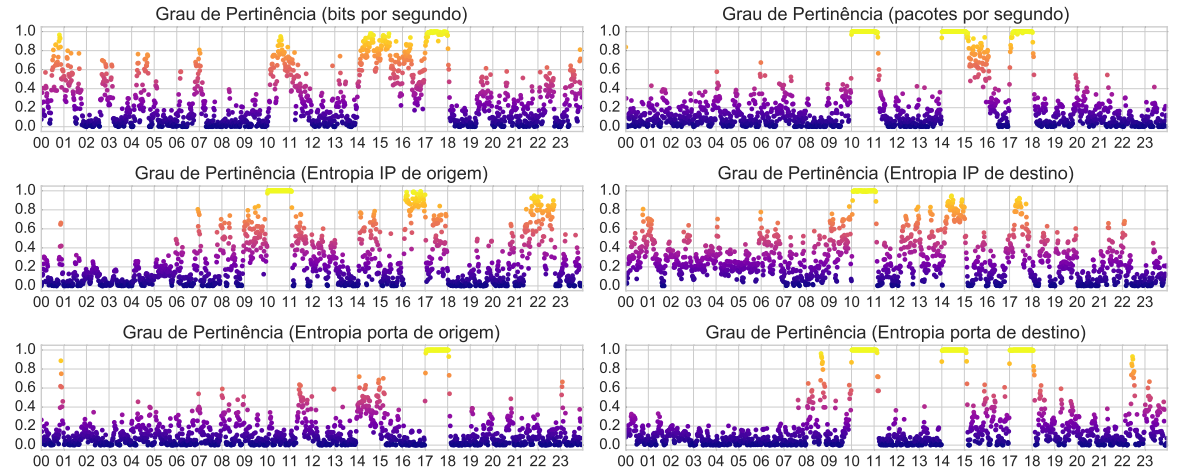
Figura 26 – Análise do dia 06/07/2016.

Para o dia 07 de julho (quinta-feira), anomalias de DoS foram injetadas no intervalo das 10h às 11h, DDoS das 14h às 15h e *Flash Crowd* das 17h às 18h. Essas anomalias podem ser notadas ao analisar a Figura 27. Os intervalos anômalos podem ser identificados analisando as Figuras 27a e 27b. O tráfego da rede apresenta desvios significativos do comportamento esperado com valores que ultrapassam os limiares. Vários atributos possuem o grau de pertinência elevado para os intervalos anômalos. A Figura 27c apresenta os intervalos de tempo em que anomalias foram detectadas. Para compreender melhor

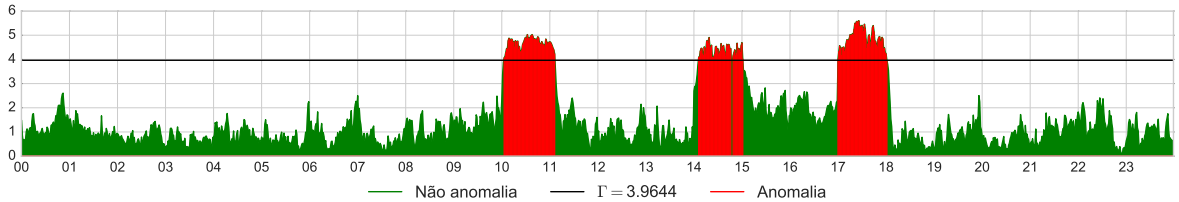
o resultado dessa análise, as métricas de precisão, acurácia, *F-measure* e taxa de falsos positivos foram calculadas, os valores são apresentados na Tabela 8. Como pode ser observado, o sistema apresentado obteve um bom desempenho para esse cenário, com altos valores para precisão, acurária e *F-measure* e uma baixa taxa de falsos positivos.



(a) DSNSF, tráfego real e limiares do dia 07 de julho de 2016.



(b) Graus de pertinência do dia 07 de julho de 2016.



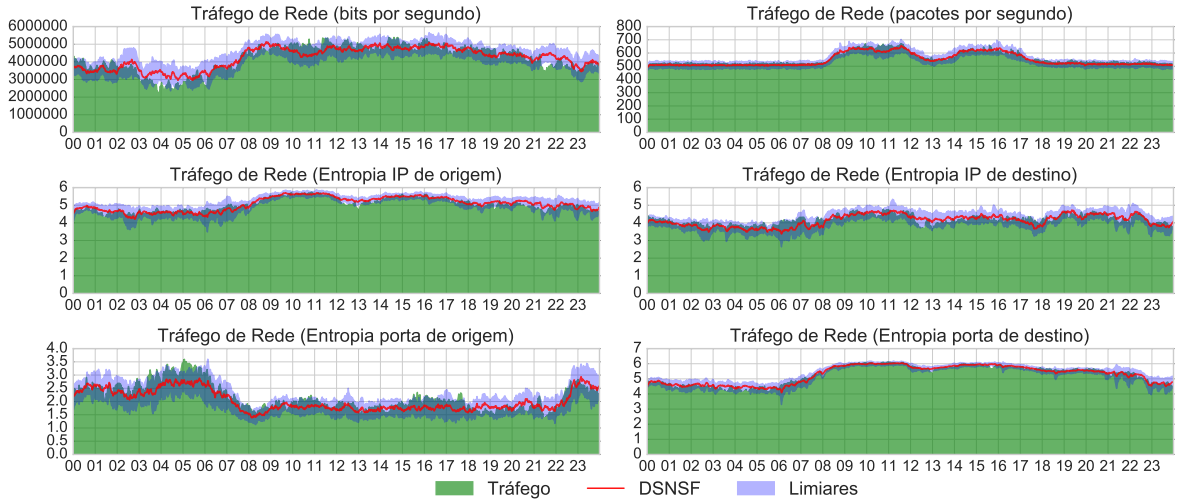
(c) Alarmes gerados para o dia 07 de julho de 2016.

Figura 27 – Análise do dia 07/07/2016.

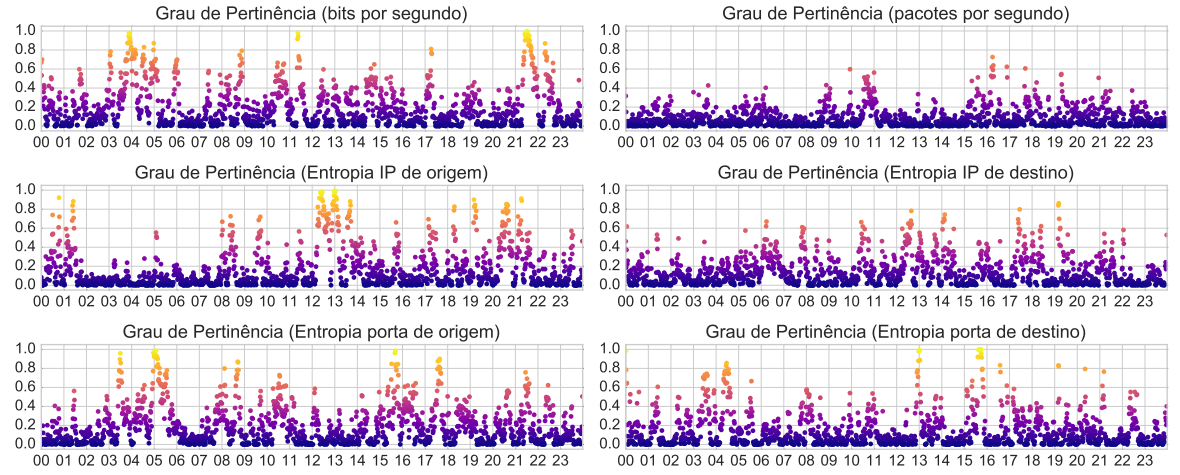
Tabela 8 – Desempenho do sistema proposto na análise do dia 07/07/2016.

Métrica	**Valor**
Precisão	95,55%
Acurácia	98,68%
F-measure	94,76%
Taxa de Falsos Positivos	0,63%

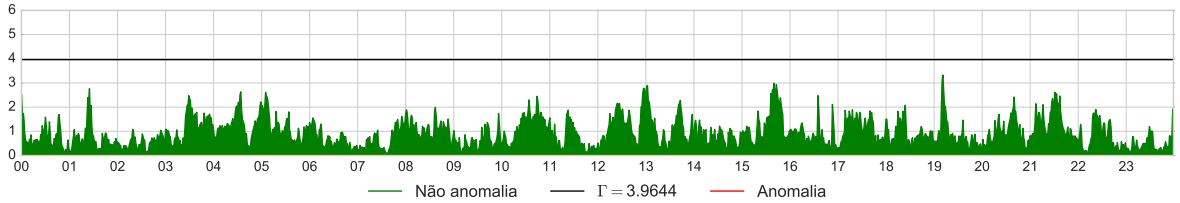
A análise do dia 08 de julho (sexta-feira) apresentou um comportamento normal, como pode ser observado na Figura 28. Não houve nenhum evento classificado como anômalo para alertar o responsável da rede, conforme observado na Figura 28c. O tráfego também não apresentou nada de incomum como indica a Figura 28a. A Figura 28b mostra que alguns intervalos possuem um grau de pertinência elevado, porém, em geral, ainda estão dentro do padrão normal da rede pela análise realizada.



(a) DSNSF, tráfego real e limiares do dia 08 de julho de 2016.



(b) Graus de pertinência do dia 08 de julho de 2016.

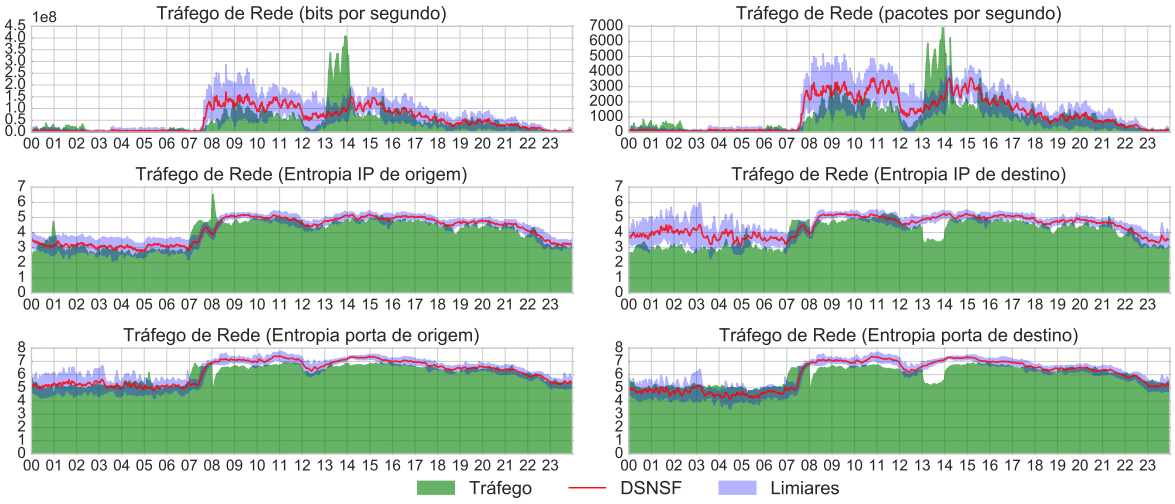


(c) Alarmes gerados para o dia 08 de julho de 2016.

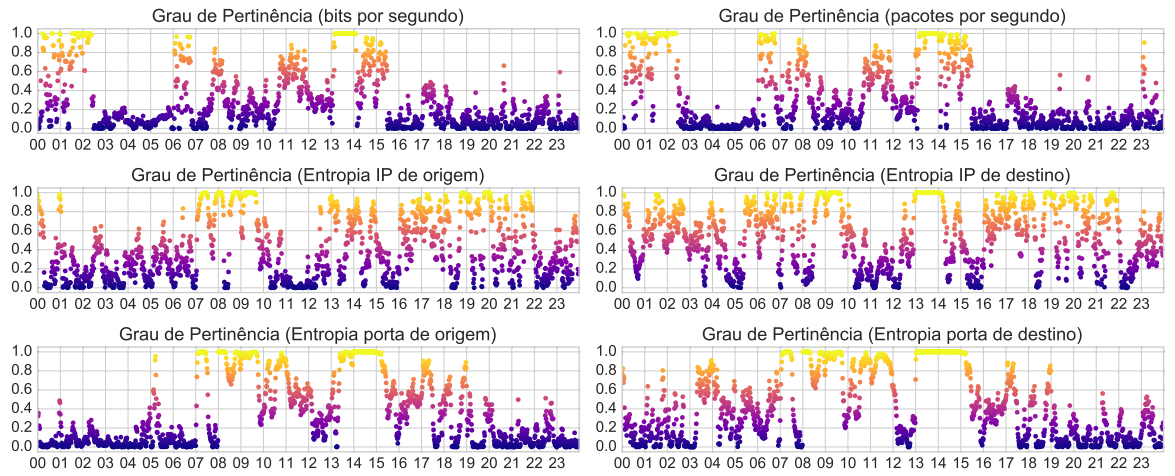
Figura 28 – Análise do dia 08/07/2016.

5.4 Cenário 3

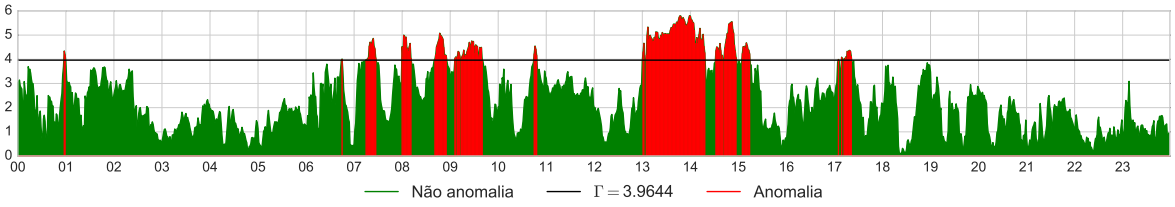
Esse cenário utiliza o sistema proposto para analisar uma rede com tráfego baixo. Os dados utilizados para essa análise foram coletados da UTFPR-Toledo em 2013 com o protocolo IPFIX, sem amostragem, armazenando todos os fluxos que trafegam pela rede. Para esse cenário somente o dia 29 de abril de 2013 (terça-feira) foi analisado. Nesse dia foram injetadas anomalias de DDoS no intervalo das 13h às 14h.



(a) DSNSF, tráfego real e limiares do dia 29 de abril de 2013.



(b) Graus de pertinência do dia 29 de abril de 2013.



(c) Alarmes gerados para o dia 29 de abril de 2013.

Figura 29 – Análise do dia 29/04/2013.

A Figura 29 apresenta os resultados da análise. Os alarmes gerados podem visua-

lizados na Figura 29c, que indica a presença de vários alarmes falsos. Também pode ser observada a diferença entre o tráfego real e o DSNSF na Figura 29a presente em todos os atributos analisados. Nessa figura é possível notar que os tráfegos dos atributos de volume (bits e pacotes) aumentam significantemente quando a universidade inicia as suas atividades. Ao avaliar os limiares desses atributos, também se pode concluir que há uma grande variação do tráfego de volume entre dias diferentes, uma vez que os limiares do EWMA utilizam o desvio padrão. A Figura 29b mostra os graus de pertinência elevados ao longo do dia para todos os atributos, que resultam nos alarmes falsos. Também é possível visualizar o comportamento anômalo do tráfego do período no intervalo de tempo das 13h às 14h.

Tabela 9 – Desempenho do sistema proposto na análise do dia 29/04/2013.

Métrica	**Valor**
Precisão	23,64%
Acurácia	79,86%
F-measure	24,87%
Taxa de Falsos Positivos	12,33%

A Tabela 9 apresenta os resultados alcançados pelo sistema proposto neste cenário. Ele conseguiu detectar a maioria dos intevalos anômalos de DDoS, porém, o sistema apresentou uma taxa de falsos positivos de 12,33%, um valor muito elevado para detecção de anomalias em redes. Esses falsos alarmes afetam as métricas de precisão e *F-measure*. A acurácia também obteve um valor baixo, uma vez que muitos intervalos normais foram classificados anômalos.

O objetivo desse cenário é demonstrar o desempenho do sistema proposto em uma rede de porte menor. Com os resultados adquiridos, pode-se concluir que ele não é propício para esse cenário, devido à alta taxa de falsos positivos em um dia da semana comum. Para que esse sistema possa ser aplicado com eficiência nesse cenário, seria necessário ajustar os parâmetros do Algoritmo Genético e da Lógica *Fuzzy* para incluir mais flexibilidade na análise do tráfego de rede. Porém, com isso existe o risco do sistema não reter a capacidade de detectar anomalias com precisão.

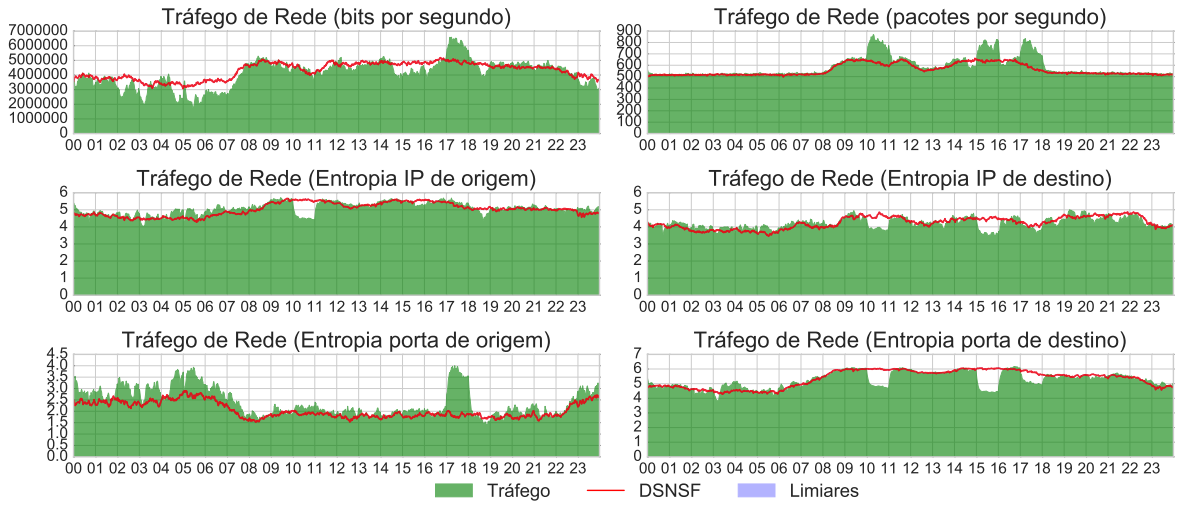
5.5 Cenário 4

Para validar o desempenho do sistema apresentado, ele foi comparado com outro trabalho que possui o mesmo objetivo, detecção de anomalias em tráfego de rede com análise de fluxos IP. Para tanto, foram introduzidos anomalias de DoS, DDoS e *Flash Crowd* utilizando a ferramenta Scorpius no tráfego do dia 3 de outubro de 2016. O ataque de DoS foi injetado das 10h às 11h, DDoS das 15h às 14h e, por fim, o *Flash Crowd* das 17h às 18h.

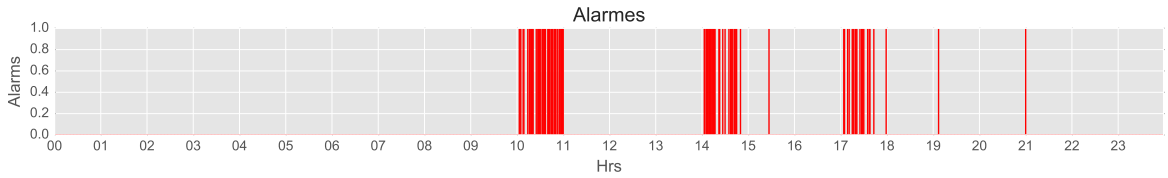
O sistema utilizado para comparar com o Fuzzy-GADS foi o ACODS (*Ant Colony Optimization for Digital Signature*) [64], desenvolvido na dissertação de mestrado de Carvalho[72] Esse sistema foi desenvolvido recentemente e apresentou bons resultados nas métricas avaliadas. Ao usar *clusterização*, é possível identificar e quantificar similaridades na organização dos intervalos de tempo do histórico do tráfego. O processo de *clusterização* é conduzido pelo ACODS, uma modificação da metaheurística *Ant Colony Optimization* (ACO). A vantagem de utilizar o ACODS é que o processo de *clusterização* é um algoritmo de aprendizado não-supervisionado, viabilizando o sistema de funcionar de uma forma autonômica. Além disso, ele possibilita a construção de soluções que não são dadas pelo ótimo local, o que é um problema que ocorre em certos algoritmos de clusterização. Assumindo que o tráfego normal da rede pode sofrer deslocamentos devido a mudanças nas atividades dos usuários, o ACODS implementa o *Adaptive Dynamic Time Warping* (ADTW) para lidar com essas situações. Essa metodologia é uma modificação do *Dynamic Time Warping*, usado para reconhecimento de padrões e amplamente aplicado em reconhecimento de voz para encontrar o melhor alinhamento entre duas séries temporais. Uma diferença essencial entre os sistemas comparados é a forma na qual a detecção de anomalias é realizada. No ACODS, isso é feito analisando-se o número de dimensões anômalas, desconsiderando a diferença entre o observado e o esperado. Em contrapartida, o Fuzzy-GADS usa a Lógica Fuzzy para quantificar os desvios e os agregam em um único valor. Então, esse valor é comparado com um valor de corte, determinando se existe uma anomalia.

A Figura 30 mostra a análise da detecção de anomalias com o sistema ACODS. O DSNSF, apresentado na Figura 30a, gerado pelo ACODS, é semelhante ao DSNSF gerado pelo GADS na Figura 31a, com um erro significante no horário das 2h às 7h. Porém, observa-se pelos alarmes gerados na Figura 30b que não teve anomalias nesse horário. Os alarmes gerados encontram-se nos intervalos de tempo em que as anomalias foram injetadas. Entretanto, muitas das anomalias não foram detectadas, o ACODS apresentou dificuldades para a detecção, especialmente de *Flash Crowd*. Além disso, houve alguns alarmes falsos emitidos, o que diminui a sua performance.

A Figura 31 apresenta os gráficos gerados da análise proveniente do sistema Fuzzy-GADS. A Figura 31a apresenta o tráfego real em comparação com o DSNSF e os limiares,



(a) DSNSF, tráfego real e limiares do dia 03 de agosto de 2016 com o ACODS.



(b) Alarmes gerados para o dia 03 de agosto de 2016 com o ACODS.

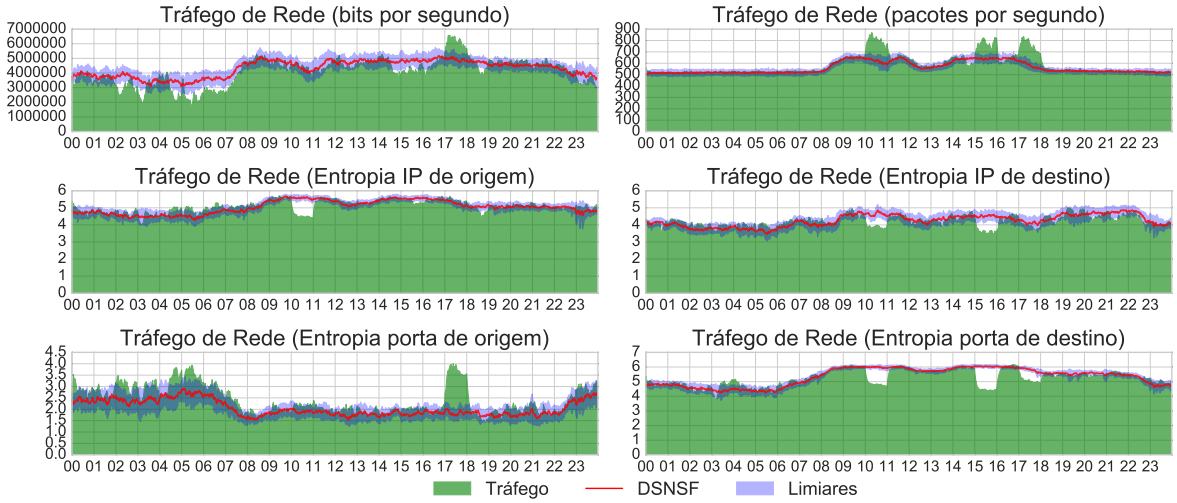
Figura 30 – Análise do dia 03/08/2016 com o ACODS.

a maior diferença entre o tráfego esperado e o real também se encontram entre as 2:00 até 7:00 horas. Entretanto, no horário comercial o tráfego é mais condizente com o DSNSF, exceto nos intervalos em que as anomalias foram injetadas. Isso também pode ser observado na Figura 31b, em que os graus de pertinência de cada dimensão analisada são apresentados. O alarme geral mostrado na Figura 31c expõe que o Fuzzy-GADS detectou a maioria das anomalias injetadas, apresentando somente um falso positivo próximo à 0h.

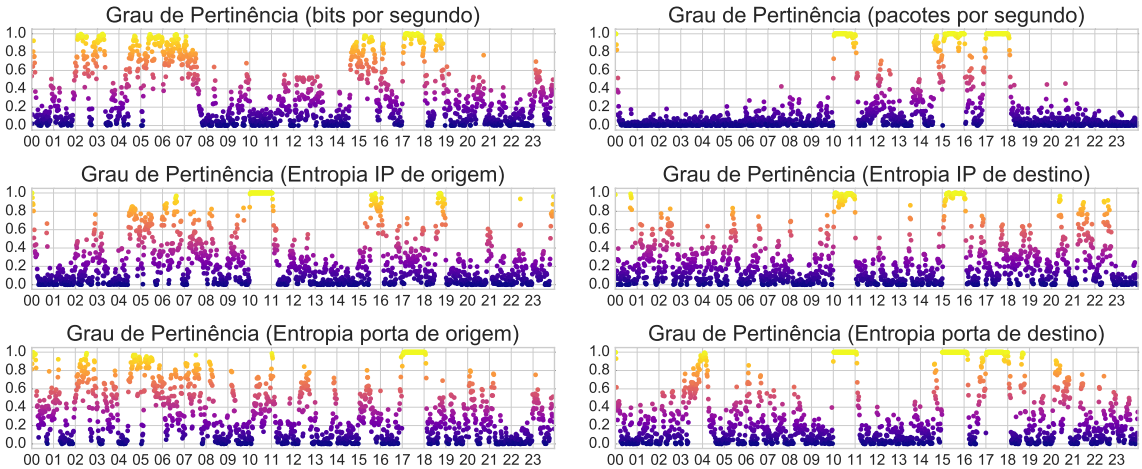
Tabela 10 – Desempenho do método proposto e do ACODS para o dia 03/08/2016.

Métrica	**Fuzzy-GADS**	**ACODS**
Precisão	94,41%	67,03%
Acurácia	98,33%	89,44%
F-measure	93,37%	44,52%
Taxa de Falsos Positivos	0,79%	2,38%

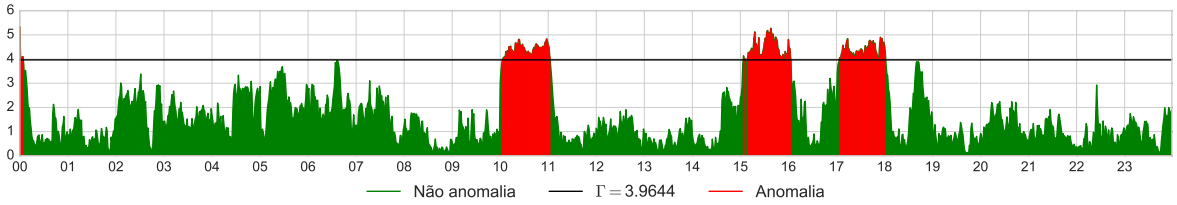
A Figura 32 apresenta a curva ROC dos dois sistemas analisados. Nota-se que a falha da detecção de alguns intervalos anômalos do ACODS influenciou a curva ROC. O Fuzzy-GADS atingiu uma área sob a curva de 0,9939, enquanto o ACODS teve o valor



(a) DSNSF, tráfego real e limiares do dia 03 de agosto de 2016 com o GADS.



(b) Graus de pertinência do dia 03 de agosto de 2016.



(c) Alarmes gerados para o dia 03 de agosto de 2016 com a Lógiaca *Fuzzy*.

Figura 31 – Análise do dia 03/08/2016 com o Fuzzy-GADS.

de 0,9287. Com isso, podemos concluir que o Fuzzy-GADS teve um desempenho melhor comparado ao ACODS. Além da curva ROC, também são utilizadas outras métricas de desempenho.

A Tabela 10 apresenta os desempenhos obtidos pelo Fuzzy-GADS e o ACODS no cenário descrito. Observa-se que o Fuzzy-GADS obteve um desempenho superior ao ACODS, especialmente nas métricas de precisão, acurácia e *F-measure*. A taxa de falsos positivos teve uma variação menor, porém o sistema proposto obteve um melhor resultado

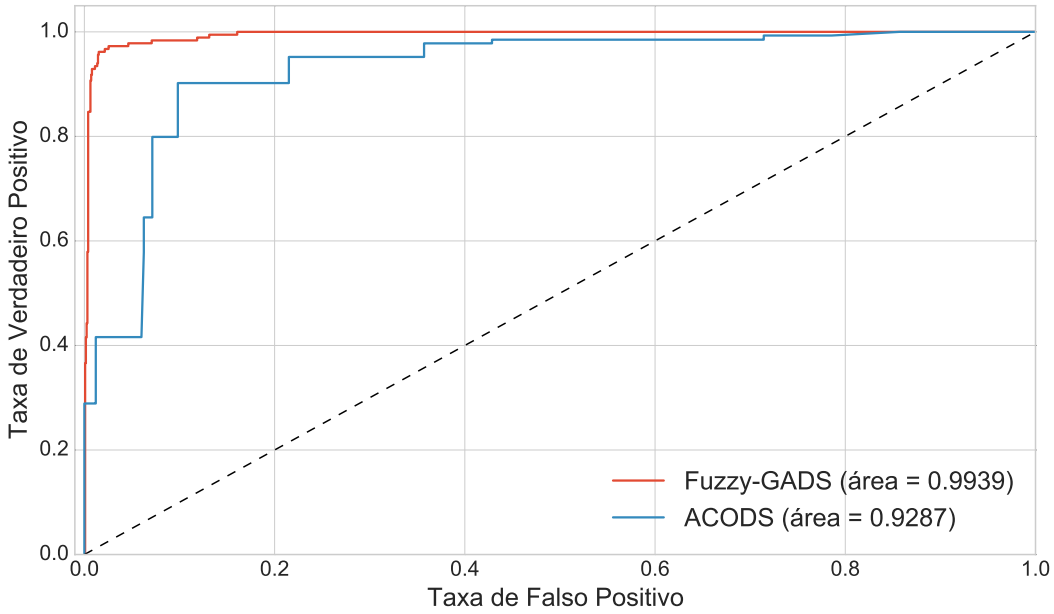


Figura 32 – Curva ROC do Fuzzy-GADS e ACODS.

nesse aspecto. Esses resultados indicam que o sistema apresentado nesse trabalho possui a capacidade de detecção de anomalias silimiar a de um método desenvolvido recentemente.

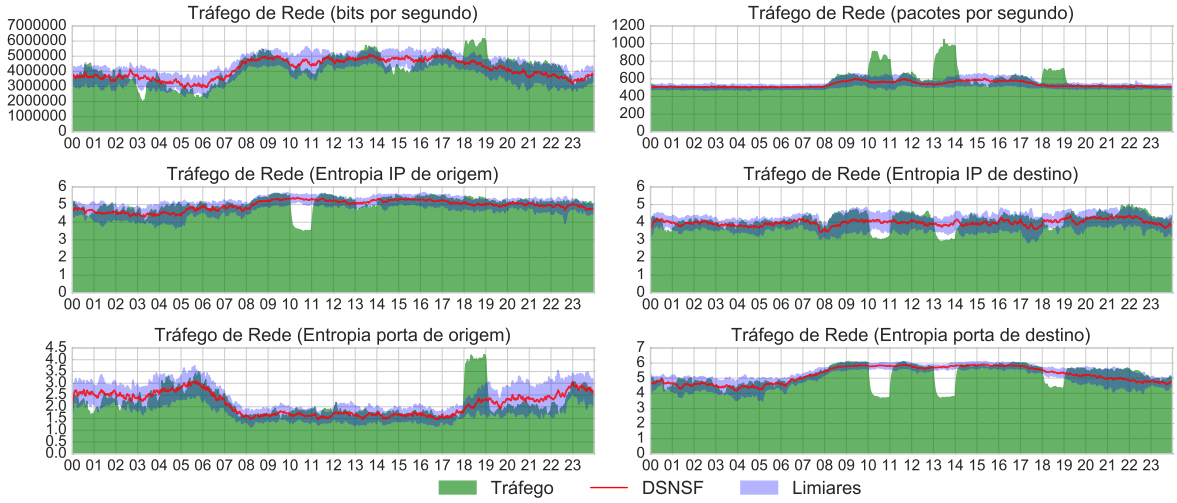
5.6 Cenário 5

O Cenário 3 (Seção 5.4) compara o Fuzzy-GADS com o ACODS, ambos os sistemas fazem a geração do DSNSF e a detecção de anomalias de formas distintas. No entanto, no Cenário 5 é aplicada a etapa de detecção de anomalias com a Lógica *Fuzzy* apresentada na Seção 4.5 em dois DSNSFs gerados por algoritmos distintos. Um dos algoritmos é o GADS e o outro é o FADS (*Firefly Algorithm for Digital Signature*) [73]. Os limiares utilizados para os dois DSNSFs foram gerados pelo EWMA para a detecção de anomalias.

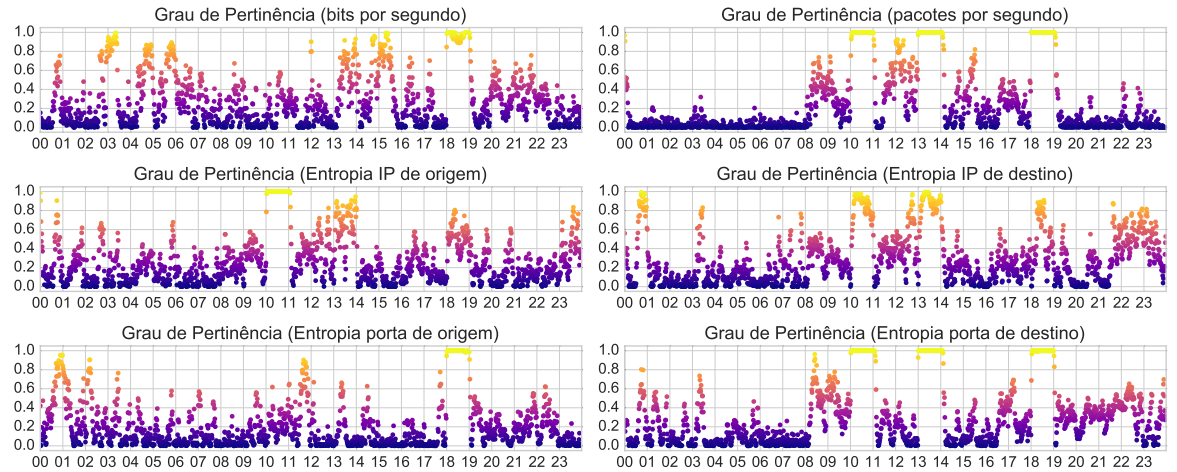
O FADS utiliza o *Firefly Algorithm* (FA) para definir os centroides do algoritmo de *clusterização k-means*, que é utilizado para gerar o DSNSF. Um problema consequente da aplicação do *k-means* é a convergência para o mínimo local, que ocorre quando os centroides são inicializados muito próximos um ao outro. Assim como os Algoritmos Genéticos, o FA é uma metaheurística que visa otimizar uma operação, no caso é a definição dos centroides. A função objetivo que os autores utilizaram também é baseada na distância euclidiana. Os testes realizados no trabalho dos autores usam um intervalo de tempo de cinco minutos para a extração das dimensões da rede, isso foi adaptado para o nosso experimento que utiliza intervalos de 1 minuto.

O tráfego analisado nesse cenário é do dia 15 de setembro de 2016 (quinta-feira). As anomalias de DoS, DDoS e *Flash Crowd* foram injetadas nos intervalos das 10h as 11h, 13h às 14h e das 18h até às 19h respectivamente.

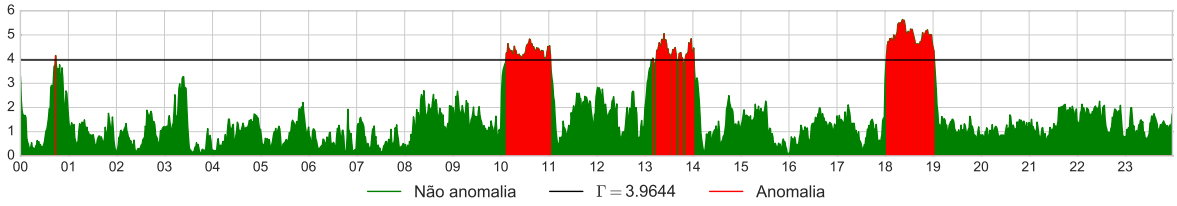
A análise do Fuzzy-GADS é apresentada na Figura 33. Percebe-se na Figura 33a que os intervalos de tempo em que as anomalias foram injetadas apresentam um comportamento distinto do DSNSF, com um brusco aumento ou queda do tráfego nos atributos afetados. Os graus de pertinência na Figura 33b refletem as anomalias presentes, condizentes com o comportamento do tráfego da rede. Os alarmes gerados pelo Fuzzy-GADS são apresentados na Figura 33c, com 1 falso positivo próximo à 1h e as detecções corretas nos intevalos anômalos. No intervalo das 13h às 14h, em que o DDoS foi injetado, existem alguns pontos em que a detecção falhou.



(a) DSNSF, tráfego real e limiares do dia 15 de setembro de 2016.



(b) Graus de pertinência do dia 15 de setembro de 2016.

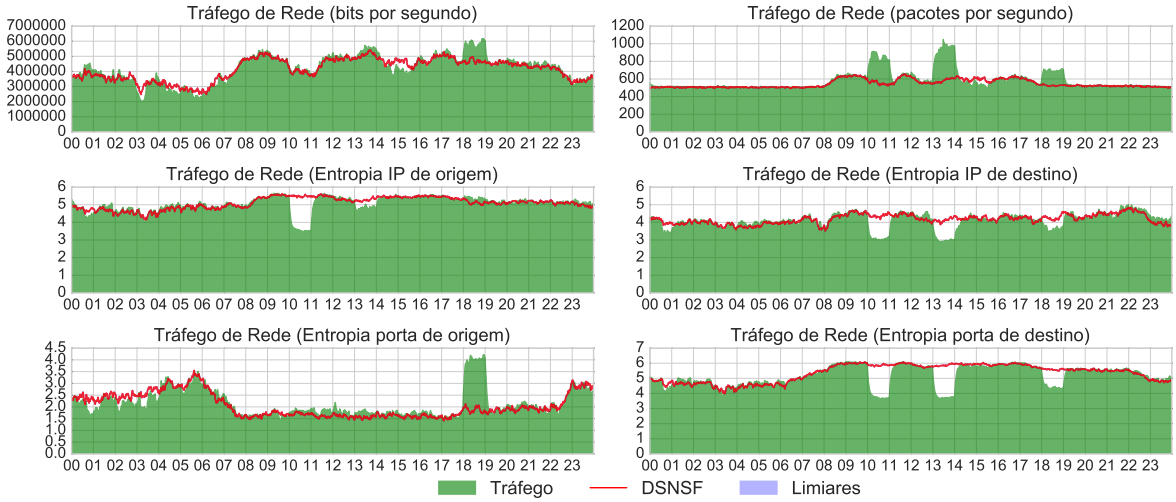


(c) Alarmes gerados para o dia 15 de setembro de 2016.

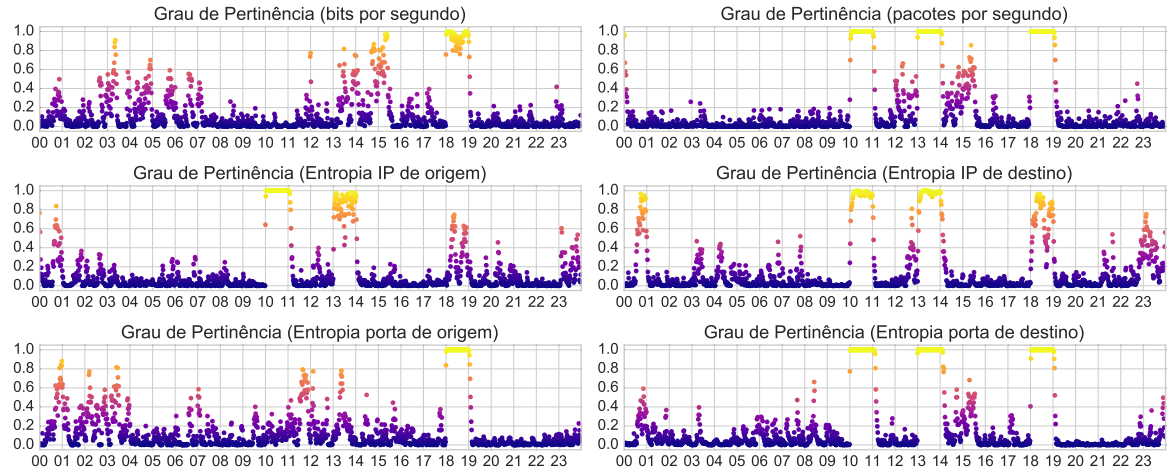
Figura 33 – Análise do dia 15/09/2016 aplicando o Fuzzy-GADS.

A Figura 34 apresenta a análise com o Fuzzy-FADS. Assim como na Figura 33 é possível perceber a mudança do comportamento do tráfego com as anomalias. Ao observar a Figura 33a, é possível notar que o DSNSF gerado pelo FADS é mais fiel ao tráfego do que o DSNSF gerado pelo GADS, o que afeta o desempenho apresentado na Tabela 11. Isso também pode ser notado na Figura 34b, em que os graus de pertinência possuem uma oscilação menor e valores baixos para os intervalos normais. Os alarmes apresentados na Figura 34c dizem respeito às anomalias, com nenhum falso positivo visível fora

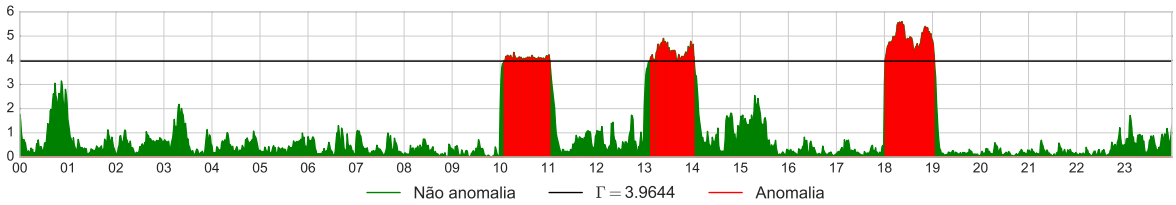
desses intervalos. Os falsos positivos encontram-se nos intervalos de tempo imediatamente anterior ou posterior às anomalias.



(a) DSNSF, tráfego real e limiares do dia 15 de setembro de 2016.



(b) Graus de pertunência do dia 15 de setembro de 2016.



(c) Alarmes gerados para o dia 15 de setembro de 2016.

Figura 34 – Análise do dia 15/09/2016 aplicando o Fuzzy-FADS.

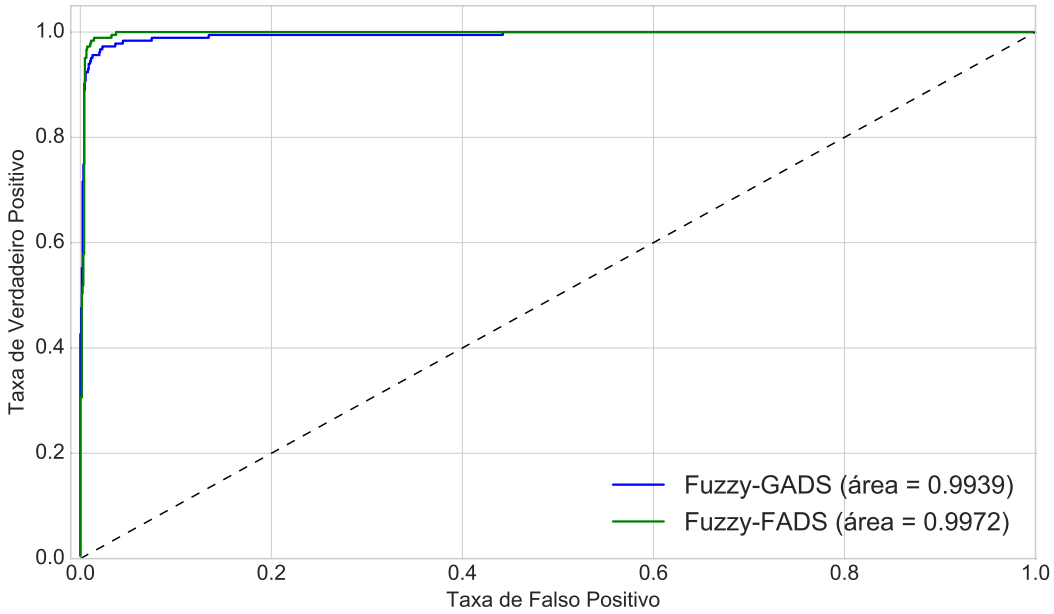


Figura 35 – Curva ROC do Fuzzy-GADS e Fuzzy-FADS.

A Figura 35 apresenta a curva ROC dos dois métodos aplicados neste cenário. Pela figura, percebe-se que ambos os sistemas obtiveram um bom desempenho. Porém, o Fuzzy-FADS obteve uma área sob a curva maior do que o Fuzzy-GADS, o que indica um maior número de acertos com poucos falsos positivos. Portanto, o Fuzzy-FADS teve um desempenho melhor, isso também pode ser observado na Tabela 11, que expõe as outras métricas avaliadas neste cenário.

Tabela 11 – Desempenho do Fuzzy-GADS e do Fuzzy-FADS para o dia 15/09/2016.

Métrica	**Fuzzy-GADS**	**Fuzzy-FADS**
Precisão	96,85%	96,55%
Acurácia	97,63%	98,54%
F-measure	90,05%	94,11%
Taxa de Falsos Positivos	0,39%	0,47%

A Tabela 11 apresenta os desempenhos atingidos pelos sistemas utilizados na análise. Em termos de precisão e taxa de falsos positivos os dois algoritmos apresentaram o desempenho similar, as diferenças mais acentuadas estão nas métricas de acurácia e *F-measure*. Isso indica que o FADS obteve maior sucesso para caracterizar o tráfego da rede, enquanto o GADS apresentou discrepâncias maiores com o tráfego real. O Fuzzy-GADS teve uma taxa de falsos positivos um pouco menor do que o Fuzzy-FADS, sendo uma

métrica essencial para a análise de sistemas de detecção de anomalias em redes. Em geral, o Fuzzy-FADS alcançou um melhor desempenho, sendo somente inferior na precisão e na taxa de falsos positivos por um valor baixo.

Esse cenário demonstra que a Lógica *Fuzzy* para a detecção de anomalias funciona independente do método utilizado para a geração do DSNSF. Assim, é possível testar o algoritmo de detecção de anomalias com vários outros sistemas que estabelecem o perfil normal do tráfego. Também é importante notar que o cálculo dos graus de pertinência não dependem somente do DSNSF, mas também dos limiares. Os limiares calculados pelo EWMA obtiveram bons resultados como parâmetro para a função de pertinência para as análises demonstradas. Outro ponto positivo desse método de detecção é o baixo custo computacional, uma vez que é somente uma função e não depende de iterações ou otimizações.

5.7 Considerações do Capítulo

Neste capítulo, o sistema proposto foi analisado e comparado com outros sistemas presentes na literatura que operam utilizando o mesmo tipo de dados (fluxos IP). Em alguns cenários, o local e o protocolo de coleta também foram diferentes, agregando valor às análises realizadas. Além disso, essas abordagens também usam metaheurísticas para a geração do DSNSF. O método apresentado obteve bons resultados quando analisados isoladamente, alcançando baixos erros e alta correlação para a caracterização. Nos cenários em que a detecção também foi analisada, ele obteve alta acurácia, precisão, *F-measure* e baixa taxa de falsos positivos.

Tabela 12 – Resumo dos desempenhos das detecções de anomalias apresentadas nos resultados.

	Cenários					
Métrica	**2**	**3**	**4 (GADS)**	**4 (ACODS)**	**5 (GADS)**	**5 (FADS)**
Precisão	95,55%	23,64%	94,41%	67,03%	96,85%	96,55%
Acurácia	98,68%	79,86%	98,33%	89,44%	97,63%	98,54%
F-measure	94,76%	24,87%	93,37%	44,52%	90,05%	94,11%
Taxa de Falsos Positivos	0,63%	12,33%	0,79%	2,38%	0,39%	0,47%

Nos cenários em que o Fuzzy-GADS foi comparado com outros algoritmos, ele atingiu bons resultados. No Cenário 4, em que a comparação foi realizada com o ACODS,

o sistema proposto obteve uma acurácia significantemente superior. Entretanto, ele foi inferior ao Fuzzy-FADS no Cenário 5, que atingiu valores maiores para acurácia e *F-measure*. A Tabela 12 apresenta um resumo dos desempenhos alcançados pelos sistemas nos cenários em que a detecção de anomalias foi realizada.

A complexidade computacional do sistema proposto depende da implementação do Algoritmo Genético, uma vez que os cálculos do EWMA e dos graus de pertinência são equações, portanto possuem complexidade constante. Entretanto, o Algoritmo Genético possui a complexidade mais elevada, dependendo do número de gerações, do tamanho da população e dos operadores. Considerando que esses valores já são previamente definidos, o seu tempo de execução será sempre igual para uma configuração de *hardware* em específico. Nos testes realizados, a geração do DSNSF para um intervalo é menor do que 1 segundo para uma dimensão, possibilitando a aplicação em tempo real do sistema proposto.

6 CONCLUSÃO

Com a utilização extensiva das redes de computadores para as mais variadas tarefas, tornou-se necessário o desenvolvimento de técnicas e métodos sofisticados para auxiliar em sua gerência. Para tanto, pesquisas envolvendo detecção de intrusões e falhas na rede vêm sendo desenvolvidas. Um IDS usa informações coletadas sobre o tráfego da rede para analisar se esse comportamento é anômalo. Existem três principais abordagens para a implementação de um IDS: baseado em assinaturas, baseado em detecção de anomalias e híbrido.

Este trabalho apresenta um IDS baseado em detecção de anomalias usando Algoritmos Genéticos, EWMA e Lógica *Fuzzy*. O sistema apresentado usa o conceito de DSNSF para criar uma previsão do comportamento da rede para um dado intervalo de tempo. Para tanto, utiliza-se o tráfego de 4 semanas anteriores para criar o DSNSF.

O Algoritmo Genético usado faz uma predição do comportamento do tráfego esperado em seis dimensões: bits por segundo, pacotes por segundo, entropias de IP de origem e destino e entropias de portas de origem e destino. A entropia de Shannon foi usada para medir o nível de dispersão ou concentração de IPs e portas. Para extrair essas informações, é necessário definir um intervalo no qual os fluxos são agrupados para o cálculo da média de bits e pacotes por segundo, além das entropias. Nos testes realizados, foi definido um intervalo de 60 segundos para agrupar os fluxos IP. Nesse intervalo de tempo, há fluxos o suficiente para estabelecer padrões e, caso uma anomalia esteja presente, o tempo de resposta de um minuto é adequado para tratar os possíveis problemas na rede.

A entrada utilizada para a geração do DSNSF com o Algoritmo Genético também é usada para calcular os limiares usando o EWMA. O EWMA usa observações no histórico para gerar limiares inferiores e superiores. Se o comportamento da rede ultrapassar, pode ser um indicativo de uma anomalia. Quanto mais recente é a observação, maior é o peso que ela possui para a geração dos limiares, assim o tráfego da semana anterior é mais relevante do que o tráfego de quatro semanas anteriores.

Com os DSNSFs e os limiares estipulados, a detecção de anomalias é realizada pela Lógica *Fuzzy*. Para cada dimensão da rede analisada, os graus de pertinência são calculados, indicando o nível de anomalia para a dada dimensão. Assim, os graus de pertinência de todas as dimensões são agregrados por uma soma, resultando no nível geral de anomalia do intervalo de tempo. Esse nível geral é então comparado com o valor de corte e, caso ele seja superior, um alarme que notifica o responsável pela rede é disparado. Assim o gerente de redes investiga e identifica o possível problema com base na análise providenciada.

Para validar o sistema apresentado, foram realizados experimentos com dados coletados de diferentes anos e redes. No primeiro cenário, a geração do DSNSF pelo Algoritmo Genético utiliza dados da UEL de 2012 e 2015. Os dados de 2012 foram coletados com a amostragem de 1 pacote a cada 256 e os dados de 2015 de 1 pacote a cada 512. Assim, o número de fluxos IP disponíveis nos dados de 2012 é aproximadamente o dobro do que em 2015. Em sua maioria, os valores do NMSE atingidos foram abaixo de 5%. Para a análise de 2012, a dimensão de bits por segundo apresentou um erro relativamente maior comparado com as outras dimensões. Com os dados de 2015, a entropia de porta de origem apresentou maior erro. Devido a isso, foi feita uma comparação entre as informações de fluxos coletados nos dois cenários. Nessa comparação, foi possível observar a mudança nas portas de origem mais utilizadas entre o intervalo de outubro de 2012 e outubro de 2015. Essa mudança ocorreu devido ao uso da porta 443 (HTTPS) em 2015, pois muitos serviços começaram a utilizá-la como alternativa à porta 80 (HTTP), o que acarretou em um maior erro na entropia da porta de origem em 2015. Isso demonstra que as redes de computadores estão sujeitas a mudanças e métodos para gerência e monitoramento também devem evoluir de forma dinâmica.

Ainda no primeiro cenário, a caracterização de uma semana do tráfego coletado em 2015 foi analisada. As métricas para a sua avaliação foram o NMSE, correlação e o teste de Bland-Altman. A maioria dos intervalos de tempo analisados teve o tráfego real entre os limiares definidos pelo EWMA. Os desvios mais notáveis entre o tráfego real e a caracterização encontram-se nas horas em que a universidade não está em funcionamento.

No segundo cenário, a detecção de anomalias utilizando o GADS, EWMA e Lógica *Fuzzy* foi analisada. Os dados utilizados nesse cenário foram coletados no ano de 2016 na UEL com o protocolo sFlow. Para tanto, o sistema apresentado foi aplicado em 5 dias úteis de uma semana de julho, em que anomalias foram injetadas artificialmente pela ferramenta Scorpius em um dos dias. No dia em que as anomalias foram injetadas, o Fuzzy-GADS apresentou um desempenho de alta acurácia (98,68%), alta precisão (95,55%) e uma baixa taxa de falsos positivos (0,63%). Esse resultado indica que o sistema proposto conseguiu classificar corretamente a maioria dos intervalos anômalos com poucos alarmes falsos.

O terceiro cenário utiliza os dados coletados na UTFPR-Toledo em 2013. Essa rede é menor do que a da UEL, então o tráfego e os números de usuários e ativos de rede são reduzidos em comparação aos dados dos Cenários 1 e 2. O objetivo desse cenário foi medir o desempenho do sistema proposto em redes de pequeno porte. Pelos resultados obtidos, percebe-se que o desempenho não é otimizado para redes desse tamanho, uma vez que foram disparados vários alarmes falsos. Então, pode-se concluir que não é viável a aplicação do sistema apresentado para tais redes.

Enquanto os Cenários 1 a 3 apresentam a análise somente do sistema proposto, os Cenários 4 e 5 fazem comparativos com outros métodos de detecção de anomalias

presentes na literatura. O Cenário 4 compara o sitema proposto com o ACODS, que funciona similarmente, utilizando dados coletados de protocolos baseados em fluxos IP com a geração de DSNSF. Na comparação realizada, o Fuzzy-GADS alcança resultados superiores ao ACODS nas métricas avaliadas, especialmente em precisão e *F-measure*. Uma diferença essencial entre os dois sistemas comparados, é a forma na qual a detecção de anomalias é realizada. No ACODS, o número de dimensões anômalas é utilizado para decidir se existe uma anomalia, enquanto o Fuzzy-GADS quantifica os desvios em graus de pertinência e os agregam em uma análise mais adequada. Isso auxiliou o Fuzzy-GADS a obter um desempenho melhor.

O Cenário 5 aplica a etapa de detecção de anomalias da Lógica *Fuzzy* no GADS e no FADS, comparando os resultados. Em geral, o Fuzzy-FADS apresentou uma performance superior. A única métrica em que o Fuzzy-GADS foi melhor é a taxa de falsos positivos. Conforme foi demonstrado nesse cenário, a etapa de detecção de anomalias funciona modularmente, ou seja, com os DSNSFs gerados, é possível utilizar o sistema *Fuzzy* apresentado para a detecção de anomalias.

Os resultados mostram que o sistema proposto obtém um bom desempenho para redes de grande porte. Isso corrobora as afirmações presentes na literatura indicando que a Lógica *Fuzzy* é adequada para ser aplicada em detecção de anomalias em rede. Isso ocorre pelas imprecisões presentes em informações coletadas sobre o tráfego de rede. Além disso, as operações e comportamentos dos usuários estão sujeitos a mudanças repentinas, agregando mais incerteza ao cenário de análise do tráfego da rede. Analisando esses resultados e características da rede, é possível confirmar que a aplicação do sistema apresentado é pertinente para o ambiente de detecção de anomalias em redes.

O sistema de detecção de anomalias apresentado neste trabalho possui as seguintes contribuições:

- **Aplicação de Algoritmo Genético**: a metaheurística de Algoritmo Genético foi utilizada para caracterizar o comportamento normal do tráfego;
- **Aplicação da Lógica *Fuzzy***: na etapa de detecção de anomalias a Lógica *Fuzzy* foi aplicada, apresentando eficácia no reconhecimento de comportamentos anômalos do tráfego;
- **Comparação com outro sistema de detecção**: a detecção de anomalias do sistema desenvolvido obteve um desempenho superior ao ACODS, um sistema aplicado para o mesmo propósito;
- **Avaliação da detecção de anomalias com Lógica *Fuzzy***: a detecção de anomalias foi aplicada com dois DSNSFs diferentes, gerados pelo GADS e FADS. Os

resultados mostram bons desempenhos utilizando ambos os DSNSFs aplicados. Assim, a metodologia usando Lógica *Fuzzy* aplicada na detecção pode ser utilizada em conjunto com sistemas que criam a caracterização do tráfego;

- **Análise de tráfegos reais**: o sistema proposto foi aplicado no tráfego de duas redes reais de diferentes escalas e protocolos de coleta. Essa análise demonstra a capacidade de detecção em ambientes reais de redes.

Neste trabalho foi aplicado o conceito de Algoritmos Genéticos, EWMA e lógica Fuzzy para detectar anomalias em tráfego de redes com a análise de fluxos IP. Além disso, o sistema apresentado pode ser usado em tempo real, uma vez que os dados são coletados com um protocolo padronizado e necessita de intervalos de 1 minuto para gerar a caracterização. A análise dos cenários expostos no capítulo de resultados indica que o Fuzzy-GADS possui um bom desempenho, apresentando altas taxas de precisão, acurácia e *F-measure*, ao mesmo tempo que alcança uma baixa taxa de falsos positivos para uma rede de grande porte, que são propriedades desejáveis para a detecção de anomalias em rede de computadores.

Para os trabalhos futuros, outras formas de aplicação da Lógica *Fuzzy* para a etapa de detecção de anomalias serão estudadas, visando melhorar o seu desempenho. Também será estudada uma forma de aplicação do sistema em Redes Definidas por *Software*, que é paradigma que funciona diferentemente das redes tradicionais, separando o plano de controle do plano de dados, possibilitanto uma gerência mais unificada da rede.

REFERÊNCIAS

[1] CASE, J. *et al.* RFC, *Simple Network Management Protocol (SNMP).* Fremont, CA, USA: RFC Editor, 1989. 1–34 p. RFC 1098. (Internet Request for Comments, 1098). Obsoleted by RFC 1157. Disponível em: <https://www.rfc-editor.org/rfc/rfc1098.txt>.

[2] PHAAL, P.; PANCHEN, S.; MCKEE, N. *InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks.* [S.l.]: IETF, 2001. RFC 3176 (Informational). (Request for Comments, 3176).

[3] CLAISE, B. *Cisco Systems NetFlow Services Export Version 9.* RFC Editor, 2004. RFC 3954. (Request for Comments, 3954). Disponível em: <https://rfc-editor.org/rfc/rfc3954.txt>.

[4] ZSEBY, T. *et al. Requirements for IP Flow Information Export (IPFIX).* RFC Editor, 2015. RFC 3917. (Request for Comments, 3917). Disponível em: <https://rfc-editor.org/rfc/rfc3917.txt>.

[5] CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 41, n. 3, p. 15:1–15:58, jul. 2009. ISSN 0360-0300.

[6] VERISIGN Distributed Denial of Service Trends Report. [S.l.]. Disponível em: <https://www.verisign.com/>.

[7] CIBERSECURITY Predictions 2017. [S.l.]. Disponível em: <www.pandasecurity.com/mediacenter>.

[8] WU, S. X.; BANZHAF, W. The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing*, v. 10, n. 1, p. 1 – 35, 2010. ISSN 1568-4946.

[9] ANDERSON, J. P. *Computer security threat monitoring and surveillance.* [S.l.], 1980.

[10] DENNING, D. E. An intrusion-detection model. *IEEE Transactions on Software Engineering*, SE-13, n. 2, p. 222–232, Feb 1987. ISSN 0098-5589.

[11] AL-JARRAH, O.; ARAFAT, A. Network intrusion detection system using attack behavior classification. In: *Information and Communication Systems (ICICS), 2014 5th International Conference on.* [S.l.: s.n.], 2014. p. 1–6.

[12] AHMED, M.; MAHMOOD, A. N.; HU, J. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, v. 60, p. 19 – 31, 2016. ISSN 1084-8045. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1084804515002891>.

[13] YU, Y. A survey of anomaly intrusion detection techniques. *J. Comput. Sci. Coll.*, Consortium for Computing Sciences in Colleges, USA, v. 28, n. 1, p. 9–17, out. 2012. ISSN 1937-4771. Disponível em: <http://dl.acm.org/citation.cfm?id=2379703.2379707>.

[14] GONG, R. H.; ZULKERNINE, M.; ABOLMAESUMI, P. A software implementation of a genetic algorithm based approach to network intrusion detection. In: *Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Network*. [S.l.: s.n.], 2005. p. 246–253.

[15] BHUYAN, M. H.; BHATTACHARYYA, D. K.; KALITA, J. K. Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys Tutorials*, v. 16, n. 1, p. 303–336, First 2014. ISSN 1553-877X.

[16] ZADEH, L. A. Fuzzy logic, neural networks, and soft computing. *Commun. ACM*, ACM, New York, NY, USA, v. 37, n. 3, p. 77–84, mar. 1994. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/175247.175255>.

[17] ANIL, S.; REMYA, R. A hybrid method based on genetic algorithm, self-organised feature map, and support vector machine for better network anomaly detection. In: *Computing, Communications and Networking Technologies (ICCCNT),2013 Fourth International Conference on*. [S.l.: s.n.], 2013. p. 1–5.

[18] KDD Cup 1999 data. [S.l.]. Disponível em: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.

[19] JONGSUEBSUK, P.; WATTANAPONGSAKORN, N.; CHARNSRIPINYO, C. Network intrusion detection with fuzzy genetic algorithm for unknown attacks. In: *The International Conference on Information Networking 2013 (ICOIN)*. [S.l.: s.n.], 2013. p. 1–5. ISSN 1550-445X.

[20] KUANG, F.; XU, W.; ZHANG, S. A novel hybrid kpca and svm with ga model for intrusion detection. *Applied Soft Computing*, v. 18, p. 178 – 184, 2014. ISSN 1568-4946. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1568494614000477>.

[21] NARSINGYANI, D.; KALE, O. Optimizing false positive in anomaly based intrusion detection using genetic algorithm. In: *MOOCs, Innovation and Technology in Education (MITE), 2015 IEEE 3rd International Conference on*. [S.l.: s.n.], 2015. p. 72–77.

[22] JADIDI, Z. *et al.* Flow-based anomaly detection using neural network optimized with gsa algorithm. In: *2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops*. [S.l.: s.n.], 2013. p. 76–81. ISSN 1545-0678.

[23] MKUZANGWE, N. N. P.; MCDONALD, A.; NELWAMONDO, F. V. Implementation of anomaly detection algorithms for detecting transmission control protocol synchronized flooding attacks. In: *Fuzzy Systems and Knowledge Discovery (FSKD), 2015 12th International Conference on*. [S.l.: s.n.], 2015. p. 2137–2141.

[24] CHAUDHARY, A.; TIWARI, V. N.; KUMAR, A. Design an anomaly based fuzzy intrusion detection system for packet dropping attack in mobile ad hoc networks. In: *Advance Computing Conference (IACC), 2014 IEEE International*. [S.l.: s.n.], 2014. p. 256–261.

[25] AZIZ, A. S. A. *et al.* Detectors generation using genetic algorithm for a negative selection inspired anomaly network intrusion detection system. In: *Computer Science and Information Systems (FedCSIS), 2012 Federated Conference on.* [S.l.: s.n.], 2012. p. 597–602.

[26] Fernandes Jr., G.; RODRIGUES, J. J.; Proença Jr., M. L. Autonomous profile-based anomaly detection system using principal component analysis and flow analysis. *Applied Soft Computing*, v. 34, p. 513 – 525, 2015. ISSN 1568-4946. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1568494615003191>.

[27] Fernandes Jr., G. *et al.* Network anomaly detection using {IP} flows with principal component analysis and ant colony optimization. *Journal of Network and Computer Applications*, v. 64, p. 1 – 11, 2016. ISSN 1084-8045. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1084804516000618>.

[28] ZHAO, X.; WANG, G.; LI, Z. Unsupervised network anomaly detection based on abnormality weights and subspace clustering. In: *2016 Sixth International Conference on Information Science and Technology (ICIST).* [S.l.: s.n.], 2016. p. 482–486.

[29] WANG, S.; ZHAO, A. An improved hybrid genetic algorithm for traveling salesman problem. In: *Computational Intelligence and Software Engineering, 2009.* [S.l.: s.n.], 2009. p. 1–3.

[30] XING, Y. *et al.* An improved adaptive genetic algorithm for job-shop scheduling problem. In: *Natural Computation, 2007.* [S.l.: s.n.], 2007. v. 4, p. 287–291.

[31] DORIGO, M.; BIRATTARI, M.; STUTZLE, T. Ant colony optimization. *IEEE Computational Intelligence Magazine*, v. 1, n. 4, p. 28–39, Nov 2006. ISSN 1556-603X.

[32] ALWESHAH, M.; ABDULLAH, S. Hybridizing firefly algorithms with a probabilistic neural network for solving classification problems. *Applied Soft Computing*, v. 35, p. 513 – 524, 2015. ISSN 1568-4946. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1568494615003671>.

[33] CHUANG, L.-Y. *et al.* Operon prediction using chaos embedded particle swarm optimization. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 10, n. 5, p. 1299–1309, set. 2013. ISSN 1545-5963. Disponível em: <http://dx.doi.org/10.1109/TCBB.2013.63>.

[34] NI, J. *et al.* An improved shuffled frog leaping algorithm for robot path planning. In: *2014 10th International Conference on Natural Computation (ICNC).* [S.l.: s.n.], 2014. p. 545–549. ISSN 2157-9555.

[35] ĐURASEVIć, M.; JAKOBOVIć, D.; KNEžEVIć, K. Adaptive scheduling on unrelated machines with genetic programming. *Applied Soft Computing*, p. –, 2016. ISSN 1568-4946. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1568494616303519>.

[36] KOTA, L.; JARMAI, K. Mathematical modeling of multiple tour multiple traveling salesman problem using evolutionary programming. *Applied Mathematical*

Modelling, v. 39, n. 12, p. 3410 – 3433, 2015. ISSN 0307-904X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0307904X14006386>.

[37] 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.

[38] ON the Origin of Species by Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life. reissue. Murray, London: [s.n.], 1859.

[39] DUBROVIN, V.; FEDORCHENKO, E.; ZHYLENKO, I. Genetic algorithms operators' adjustings optimumness research. In: *Modern Problems of Radio Engineering, Telecommunications and Computer Science, 2008 Proceedings of International Conference on*. [S.l.: s.n.], 2008. p. 67–69.

[40] DAVIS, L. Handbook of genetic algorithms. 1991.

[41] NELSON, A. L.; BARLOW, G. J.; DOITSIDIS, L. Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, v. 57, n. 4, p. 345 – 370, 2009. ISSN 0921-8890. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0921889008001450>.

[42] THIERENS, D.; GOLDBERG, D. Convergence models of genetic algorithm selection schemes. In: _____. *Parallel Problem Solving from Nature — PPSN III: International Conference on Evolutionary Computation The Third Conference on Parallel Problem Solving from Nature Jerusalem, Israel, October 9–14, 1994 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994. p. 119–129. ISBN 978-3-540-49001-2. Disponível em: <http://dx.doi.org/10.1007/3-540-58484-6_256>.

[43] GOLDBERG, D. E.; DEB, K. A comparative analysis of selection schemes used in genetic algorithms. *Foundations of genetic algorithms*, v. 1, p. 69–93, 1991.

[44] ZADEH, L. Fuzzy sets. *Information and Control*, v. 8, n. 3, p. 338 – 353, 1965. ISSN 0019-9958. Disponível em: <http://www.sciencedirect.com/science/article/pii/S001999586590241X>.

[45] ROSS, T. J. *Fuzzy Logic with Engineering Applications*. [S.l.]: John Wiley & Sons, 2009.

[46] TSOUKALAS, L. H.; UHRIG, R. E. *Fuzzy and Neural Approaches in Engineering*. 1st. ed. New York, NY, USA: John Wiley & Sons, Inc., 1996. ISBN 0471160032.

[47] ZADEH, L. A. Is there a need for fuzzy logic? In: *NAFIPS 2008 - 2008 Annual Meeting of the North American Fuzzy Information Processing Society*. [S.l.: s.n.], 2008. p. 1–3.

[48] SILER, W.; BUCKLEY, J. J. *Fuzzy expert systems and fuzzy reasoning*. [S.l.]: John Wiley & Sons, 2005.

[49] Proença Jr., M. L.; ZARPELÃO, B. B.; MENDES, L. S. Anomaly detection for network servers using digital signature of network segment. In: *Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop (AICT/SAPIR/ELETE'05)*. [S.l.: s.n.], 2005. p. 290–295.

[50] Proença Jr., M. L. *Baseline aplicado a gerência de redes*. Tese (Doutorado) — UNICAMP - Faculdade de Engenharia Elétrica e de Computação, 2005.

[51] LIMA, M. F. *et al.* Networking anomaly detection using dsns and particle swarm optimization with re-clustering. In: *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*. [S.l.: s.n.], 2010. p. 1–6. ISSN 1930-529X.

[52] ADANIYA, M. H. A. C. *et al.* Anomaly detection using dsns and firefly harmonic clustering algorithm. In: *2012 IEEE International Conference on Communications (ICC)*. [S.l.: s.n.], 2012. p. 1183–1187. ISSN 1550-3607.

[53] ASSIS, M. V. O. de *et al.* Holt-winters statistical forecasting and aco metaheuristic for traffic characterization. In: *2013 IEEE International Conference on Communications (ICC)*. [S.l.: s.n.], 2013. p. 2524–2528. ISSN 1550-3607.

[54] FERNANDES, G. *et al.* Digital signature to help network management using principal component analysis and k-means clustering. In: *2013 IEEE International Conference on Communications (ICC)*. [S.l.: s.n.], 2013. p. 2519–2523. ISSN 1550-3607.

[55] PENA, E. H. M. *et al.* Correlational paraconsistent machine for anomaly detection. In: *2014 IEEE Global Communications Conference*. [S.l.: s.n.], 2014. p. 551–556. ISSN 1930-529X.

[56] SHANNON, C. E. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 5, n. 1, p. 3–55, jan. 2001. ISSN 1559-1662. Disponível em: <http://doi.acm.org/10.1145/584091.584093>.

[57] LAKHINA, A.; CROVELLA, M.; DIOT, C. Mining anomalies using traffic feature distributions. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 35, n. 4, p. 217–228, ago. 2005. ISSN 0146-4833. Disponível em: <http://doi.acm.org/10.1145/1090191.1080118>.

[58] FERNANDES, G. *et al.* Network anomaly detection using ip flows with principal component analysis and ant colony optimization. *J. Netw. Comput. Appl.*, Academic Press Ltd., London, UK, UK, v. 64, n. C, p. 1–11, abr. 2016. ISSN 1084-8045. Disponível em: <http://dx.doi.org/10.1016/j.jnca.2015.11.024>.

[59] WELLER-FAHY, D. J.; BORGHETTI, B. J.; SODEMANN, A. A. A survey of distance and similarity measures used within network intrusion anomaly detection. *IEEE Communications Surveys Tutorials*, v. 17, n. 1, p. 70–91, Firstquarter 2015. ISSN 1553-877X.

[60] MATIAS, R. *et al.* Comparison analysis of statistical control charts for quality monitoring of network traffic forecasts. In: *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*. [S.l.: s.n.], 2011. p. 404–409. ISSN 1062-922X.

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

[62] MONTGOMERY, D. *Introduction to Statistical Quality Control*. [S.l.]: Wiley, 2008. ISBN 9780470169926.

[63] MOLNAR, S.; MOCZAR, Z. Three-dimensional characterization of internet flows. In: *2011 IEEE International Conference on Communications (ICC)*. [S.l.: s.n.], 2011. p. 1–6. ISSN 1550-3607.

[64] CARVALHO, L. F. *et al.* Unsupervised learning clustering and self-organized agents applied to help network management. *Expert Systems with Applications*, v. 54, p. 29 – 47, 2016. ISSN 0957-4174.

[65] HAUPT, R. L.; HAUPT, S. E. *Practical genetic algorithms*. [S.l.]: John Wiley & Sons, 2004.

[66] ASSIS, M. V. O.; JR., M. L. P. Scorpius: sflow network anomaly simulator. *Journal of Computer Science*, v. 11, n. 4, p. 662–674, 2015.

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

[68] BLAND, J. M.; ALTMAN, D. Statistical methods for assessing agreement between two methods of clinical measurement. *The Lancet*, v. 327, n. 8476, p. 307 – 310, 1986. ISSN 0140-6736. Originally published as Volume 1, Issue 8476. Disponível em: <//www.sciencedirect.com/science/article/pii/S0140673686908378>.

[69] FIORE, U. *et al.* Network anomaly detection with the restricted boltzmann machine. *Neurocomputing*, v. 122, p. 13 – 23, 2013. ISSN 0925-2312. Advances in cognitive and ubiquitous computingSelected papers from the Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2012). Disponível em: <http://www.sciencedirect.com/science/article/pii/S0925231213005547>.

[70] FARSHCHI, M. *et al.* Metric selection and anomaly detection for cloud operations using log and metric correlation analysis. *Journal of Systems and Software*, p. –, 2017. ISSN 0164-1212. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121217300596>.

[71] WITTEN, I. H. *et al. Data Mining: Practical machine learning tools and techniques*. [S.l.]: Morgan Kaufmann, 2016.

[72] CARVALHO, L. F. *Metaheurística Ant Colony Optimization e Análise de Fluxos IP Aplicados à Detecção de Anomalias e à Gerência de Redes*. Dissertação (Mestrado) — Universidade Estadual de Londrina, 2014.

[73] SALMEN, F. *et al.* Using firefly and genetic metaheuristics for anomaly detection based on network flows. In: *The Eleventh Advanced International Conference on Telecommunications AICT 2015*. [S.l.: s.n.], 2015. p. 113–118. ISSN 2308-4030.

TRABALHOS PUBLICADOS PELO AUTOR

1. A. H. Hamamoto, L. F. Carvalho and M. L. Proença, **ACO and GA metaheuristics for anomaly detection**, 2015 34th International Conference of the Chilean Computer Science Society (SCCC), Santiago, 2015, pp. 1-6., **(Qualis B3)**

2. (Em revisão - R2) A. H. Hamamoto, L. F. Carvalho, L. H. D. Sampaio, T. Abrão, M. L. Proença Jr., **Network Anomaly Detection System using Genetic Algorithm and Fuzzy Logic**, 2017, Journal of Expert Systems With Applications, **(Qualis A1)**