Universidade Estadual de Londrina

MATHEUS PEREIRA DE NOVAES

SISTEMA DE DETECÇÃO E MITIGAÇÃO DE ANOMALIAS EM REDES DEFINIDAS POR SOFTWARE

Londrina
2019

MATHEUS PEREIRA DE NOVAES

SISTEMA DE DETECÇÃO E MITIGAÇÃO DE ANOMALIAS EM REDES DEFINIDAS POR SOFTWARE

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
2019

MATHEUS PEREIRA DE NOVAES

SISTEMA DE DETECÇÃO E MITIGAÇÃO DE ANOMALIAS EM REDES DEFINIDAS POR SOFTWARE

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

_____

Prof. Dr. Sylvio Barbon Jr.
Universidade Estadual de Londrina – UEL

_____

Prof. Dr. Luiz Fernando Carvalho
Universidade Tecnológica Federal do Paraná – UTFPR

Londrina, 13 de março de 2019.

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

Novaes, Matheus Pereira.
Sistema de Detecção e Mitigação de Anomalias em Redes Definidas por Software / Matheus Pereira Novaes. - Londrina, 2019.
111 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, 2019.
Inclui bibliografia.

1. Redes Definidas por Software - Tese. 2. Otimização de Exame de Partículas - Tese. 3. Detecção de Anomalias - Tese. I. Lemes Proença Jr, Mario . II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. III. Título.

Este trabalho é dedicado aqueles que acreditam que o trabalho de hoje é o futuro do amanhã.

AGRADECIMENTOS

A Deus, por ter me dado discernimento e sabedoria ao longo desta jornada e sem ele acredito que não teria alcançando meus objetivos.

A minha família por ter me dado todo suporte necessário e que sempre estiveram ao meu lado. Meu pai Ademir, a minha mãe Dirce e minha irmã Mayara.

Ao meu orientador Prof. Dr. Mario Lemes Proença Jr. pela oportunidade e confiança, me aceitando como orientando desde a iniciação científica até a orientação do mestrado. Obrigado por ter compartilhado do seu tempo e conhecimento e mostrar que o trabalho duro gera bons frutos.

Aos membros do grupo de pesquisa de redes pelo apoio e contribuição na condução deste trabalho: Luiz, Cinara, Marcos, Anderson e Gustavo.

As amizades que a UEL me proporcionou, que de certa forma estes amigos estiveram ao meu lado tanto em momentos de distração quanto de dificuldade.

A todos os professores e técnicos da UEL que de alguma forma contribuíram para a minha formação acadêmica.

A FA/CAPES, pela bolsa concedida.

"Mas buscai primeiro o Reino de Deus, e a sua justiça, e todas essas coisas vos serão acrescentadas."
(Bíblia Sagrada, Mateus 6, 33)

NOVAES, M. P.. **Sistema de Detecção e Mitigação de Anomalias em Redes Definidas por Software**. 2019. 111 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2019.

RESUMO

Redes de computadores tornaram-se estruturas dinâmicas e complexas. Em consequência deste fato, a configuração e o gerenciamento de toda essa estrutura é uma atividade desafiadora. Rede Definidas por Software (*SDN*) é um novo paradigma de rede que, por meio de uma abstração de planos de rede, busca separar o plano de dados e o plano de controle, e tendo como objetivo superar as limitações no que diz respeito à configuração da infraestrutura de rede. Assim como no ambiente tradicional de rede, o ambiente *SDN* também está suscetível a vulnerabilidades de segurança. Este trabalho apresenta um sistema de detecção e mitigação de ataques distribuídos de negação de serviços (Distributed Denial of Service - DDoS) e ataques de Portscan em ambientes *SDN* chamado (*Particle Swarm Optimization for Digital Signature – PSO-DS*). O *PSO-DS* é composto por três módulos: caracterização, detecção e mitigação de anomalias. O sistema foi submetido a testes em um ambiente emulado de rede usando o emulador Mininet, os controladores *SDN* Floodlight e Pox e para a coleta dos dados o protocolo *OpenFlow*. Os resultados obtidos demonstram a eficiência do sistema para auxiliar no gerenciamento da rede, detectar e mitigar a ocorrência de ataques.

Palavras-chave: Redes Definidas por Software. Otimização de exame de Partículas. Detecção de Anomalias.

NOVAES, M. P.. **Anomaly Detection and Mitigation System over Software- Defined Network**. 2019. 111 pp. Master's Thesis (Master in Science in Computer Science) – Universidade Estadual de Londrina, Londrina, 2019.

ABSTRACT

Computer networks have become dynamic and complex structures. Considering this, the configuration and management of this whole structure is a challenging activity. Software-Defined Networking is a new network paradigm that, through abstraction of network plans, seeks to separate the data and control plan, and aims to change the limitations of the network infrastructure configuration. The SDN environment is also susceptible to security vulnerabilities. This work presents a system for the detection and mitigation of Distributed Denial of Service (DDoS) and Portscan attacks over SDN environments: Particle Swarm Optimization for Digital Signature (*PSO-DS*) .The PSO system consists of three modules: characterization; anomaly detection, and mitigation. The system was submitted for testing within an emulated SDN environment using the Mininet emulator, SDN Floodlight and Pox controller. For the data collection, the OpenFlow protocol was used. The results show the system efficiency in assisting network management, detecting and mitigate attacks' occurrences.

Keywords: Software-Defined Network. Particle Swarm Optimization. Anomaly Detection

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACODS	*Ant Colony Optimization for Digital Signature*
ADTW	*Adaptive Dynamic Time Warping*
API	*Application Programming Interface*
ARIMA	*Autoregressive Integrated Moving Average*
ATLANTA	*Anomaly deTection and machine LeArNing Traffic classifICation for software-defined networking*
CA	*Classification Accuracy*
CC	*Correlation Coefficient*
CCN	*Content-Centric Networks*
CE	*Classification Error*
CNN	*Convolutional Neural Network*
CPT	*Conditional Probability Table*
DAG	*Directed Acyclic Graph*
DARPA	*Defense Advanced Research Projects Agency*
DDoS	*Distributed Denial of Service*
DoS	*Denial of Service*
DSNSF	*Digital Signature of Network Segments*
FN	*False Negative*
FP	*False Positive*
GA	*Genetic Algorithm*
HIDS	*Host-Based Intrusion Detection System*
IDS	*Intrusion Detection System*
IETF	*Internet Engineering Task Force*
IoT	*Internet of Things*

IP	*Internet Protocol*
IPFIX	*IP Flows Information Export*
ISP	*Internet Service Provide*
LSTM	*Long Short-Term Memory*
NB-DS	*Naive Bayes for Digital Signature*
NBI	*Northbound Interface*
NIB	*Network Information Base*
NIDS	*Network Intrusion Detection Systems*
NMSE	*Normalised Mean Square Error*
NOS	*Network Operating System*
PCA	*Principal Component Analysis*
PSO-DS	*Particle Swarm Optimization for Digital Signature*
RNN	*Recurrent Neural Network*
SBI	*Southbound interface*
SDN	*Software Defined Network*
SI	*Swarm Intelligence*
SNMP	*Simple Network Management Protocol*
SSL	*Secure Socket Layer*
SVM	*Support Vector Machine*
TLS	*Transport Layer Security*
TN	*True Negative*
TP	*True Positive*

LISTA DE SÍMBOLOS

$H(x)$	Entropia de Shannon
J	Janela deslizante
n	número de amostras
X	Vetor n-dimensional da amostra
C	Número de cluster
P	Conjunto de partículas
v_p	Velocidade para partícula p
x_p	Atual posição da partícula p
w	Coeficiente de inércia
c_1	Constante de aceleração 1
c_2	Constante de aceleração 2
r_1	Número aleatório definidos no intervalo $[0, 1]$
r_2	Número aleatório definidos no intervalo $[0, 1]$
p_{best_p}	Melhor posição que partícula p já ocupou
g_{best}	Solução global
F	Função objetivo
c_j	Valor do centro do cluster j
x_a	Valor do atributo a ser clusterizado
μ	Média
k	Parâmetro de desvios padrões
σ	Desvio padrão
D	Quantidade de classes
$P(D_i\|\mathbf{X})$	Probabilidade de $\mathbf{X}$ pertencer a classe D_i

SUMÁRIO

1 INTRODUÇÃO

A quantidade de aplicações e o número de dispositivos conectados que utilizam a Internet como meio de transmissão de dados está aumentando em um ritmo acelerado. Aplicações Web (por exemplo, *Internet banking*, rede sociais, *e-commerce*), proliferação de dispositivos móveis e as novas tecnologias, como *Internet of Things* (IoT), estão cada vez mais ganhando popularidade. Qualidade de serviço e a segurança requeridas pelas aplicações se tornaram um problema para as atividades do gerente de rede devido à complexa infraestrutura do modelo tradicional de rede, pois a configuração é feita por meio de soluções proprietárias e os planos de rede (dados e controle) são acoplados nos dispositivos [1].

Redes Definidas por Software (*SDN*) é uma arquitetura de rede emergente para projetar redes futuras e satisfazer as novas demandas das aplicações existentes. O novo paradigma de rede *SDN* tem como principal característica a separação entre os planos de rede, ou seja, o plano de controle e o plano de dados são desacoplados dos dispositivos de rede por meio de uma abstração de planos [2]. A separação de planos permite controlar, alterar e gerenciar o comportamento da rede por intermédio de uma interface de software flexível, ao contrário do modelo tradicional de rede, em que os dispositivos são caixas proprietárias fechadas, limitando a flexibilidade no que diz respeito à configuração e controle interno [3].

Novos recursos de gerenciamento e monitoramento, tais como centralização do controle e programação da rede, que podem melhorar o desempenho e reduzir os gargalos do modelo legado estão presentes nas redes *SDN*. Apesar do novos recursos, as redes *SDN* também estão sujeitas às ameaças e vulnerabilidades de segurança. Na arquitetura SDN a inteligência da rede é centralizada no controlador, assim como qualquer sistema de gerenciamento centralizado, o controlador pode ser um alvo ideal para ataques de Negação de serviço (*Denial of Service – DoS*) [4][1]. O ataque de *DoS* tem como finalidade o esgotamento de algum recurso, seja ele a nível de servidor em que o atacante através de inúmeras solicitações busca indisponibilizar algum tipo de serviço ou a nível de infraestrutura em que o atacante satura um *link* de rede [5].

A gerência da segurança das informações da rede é uma atividade com alto grau de complexidade, sendo de substancial importância assegurar a disponibilidade, confiabilidade e integridade dos serviços de redes oferecidos aos usuários finais. È indispensável o emprego de ferramentas para dar suporte às atividades do gerente em relação a segurança e disponibilidade da rede e que, de algum modo, possam facilitar a identificação e isolamento de eventos anômalos [6].

É comum empregar soluções de rede preventivas como, por exemplo, firewall e/ou soluções reativas, como antivírus. Porém, as aplicações destes tipos de soluções não são suficientes contra ameaças de agentes maliciosos. *Intrusion Detection System - IDS*, são dispositivos reativos que buscam detectar e impedir à ocorrência de ataques. Os *IDS* são classificados em duas maneiras: baseado em assinatura de ataque e no perfil normal da rede. A primeira delas possui uma base de dados contendo padrões sobre as anomalias conhecidas. A segunda técnica consiste em gerar um perfil do tráfego da rede como sendo normal, não dependendo de conhecimento prévio sobre as anomalias. A principal desvantagem da técnica fundamentada no perfil da rede é a ocorrência indevida de alarmes, gerando falso-positivos. Muitas vezes o tráfego de usuários legítimos é detectado como sendo anômalo, fazendo com que falsos alarmes sejam disparados ao administrador da rede [7][8].

A detecção de anomalias pode ser aplicada em várias áreas como, por exemplo, detecção de fraudes em cartões de crédito, patologias clínicas na área médica ou na detecção de intrusões de sistemas. É uma ampla área de estudo que pode ser aplicada a qualquer situação em que haja um comportamento esperado. Conforme descrito por Bhuyan [9], anomalias em redes de computadores estão relacionadas a fatores em que determinadas atividades da rede tenham um comportamento divergente do esperado. Alguns fatores, tais como equipamentos com defeito, sobrecarga do tráfego, ataques de negação de serviço e intrusão de agentes maliciosos podem produzir um desvio do comportamento da rede. Estes eventos são considerados anômalos e prejudicam o funcionamento da rede inviabilizando a entrega padrão dos serviços aos usuários.

Nos últimos anos, com o aumento das ameaças de segurança e com o enorme volume de tráfego, várias abordagens foram propostas para detecção de anomalias. Tais abordagens foram desenvolvidas sob diferentes perspectivas de diversos domínios de pesquisa como, por exemplo, método baseado em aprendizado de máquina [10], heurística [11][12], clusterização [13][6] e análise estatística [14][15]. Em linhas gerais, as técnicas de detecção de anomalias pretendem reconhecer padrões de tráfego sensíveis por meio de mudanças repentinas no volume de tráfego esperado ou mudanças inesperadas na distribuição de certas características do tráfego de rede, como endereços IP e portas.

O sistema proposto neste trabalho possui como finalidade a caracterização do tráfego, a detecção e mitigação de ataques DDoS e Portscan em Redes Definidas por Software. O sistema utiliza como princípio o conceito de *Digital Signature of Network Segments* (DSNS) introduzido por Proença [16] e Proença *et al.* [17]. Este conceito aplica uma técnica eficiente para criar um modelo que caracteriza o perfil da rede utilizando dados históricos. A caracterização proposta por Proença *et al.* foi idealizada para o ambiente tradicional de rede e utilizou uma base histórica de semanas passadas do tráfego contendo objetos *MIB* (*Management Information Base*) provenientes do protocolo de gerenciamento

SNMP (*Simple Network Management Protocol*). Por outro lado, a caracterização proposta neste trabalho utiliza atributos de fluxos IP (*Internet Protrocol*) coletados do controlador *SDN* e a predição da assinatura de rede é realizada empregando uma janela deslizante do tráfego, consequentemente, o sistema proposto dispensa a utilização de uma base de dados para gerar a assinatura. A partir do perfil normal da rede é possível reconhecer comportamentos que diferem do esperado e auxiliar na fase de detecção de anomalias da metodologia apresentada neste trabalho.

Os eventos anômalos podem afetar o funcionamento da rede. Além de detectar e identificar a ocorrência de anomalias, é fundamental tomar contramedidas para minimizar os efeitos causados por elas. A detecção de ataques em conjunto com o emprego de políticas de mitigação auxilia na resiliência da rede para garantir os níveis mínimos de operabilidade e a qualidade dos serviços [18]. Desse modo, o sistema desenvolvido neste trabalho implementa políticas para minimizar dos eventos anômalos encontrado.

O sistema é composto por três módulos com diferentes funcionalidades. O primeiro módulo, que é responsável pela caracterização do tráfego de rede, é chamado *Particle Swarm Optimization for Digital Signature* (*PSO-DS*) e emprega organização dos dados de fluxos IP em *cluster* para extração do comportamento normal do tráfego. Neste trabalho as dimensões de fluxos analisadas são pacotes por segundo, bits por segundo, entropia de IP de origem, entropia de IP de destino, entropia de porta de origem e entropia de porta de destino, totalizando seis assinaturas que representam o comportamento normal do tráfego. O segundo módulo aplica a desigualdade de Bienaymé-Chebyshev que resulta na geração de limiares superiores e inferiores de normalidade fundamentados no perfil normal gerado no módulo de caracterização. E por fim, o módulo de mitigação, o qual é responsável por minimizar os danos provocados por um ataque.

As principais contribuições deste trabalho são:

- Um sistema para detecção de anomalias e mitigação de *DDoS* e *Portscan* em redes *SDN*;
- Janela de análise para detecção de anomalias com intervalo de tempo de 1 (um) e 5 (cinco) segundos;
- Comparação do método de detecção de anomalias proposto com métodos clássicos presentes na literatura.

O restante deste trabalho encontra-se organizado pelos seguintes capítulos: no Capítulo 2 são apresentados os trabalhos relacionados. No Capítulo 3 encontram-se conceitos de redes *SDN*. O Capítulo 4 é composto por conceitos e definições de sistemas de detecção de intruso e anomalias em redes de computadores. O Capítulo 5 apresenta as técnicas

aplicadas para o desenvolvimento do sistema proposto neste trabalho. No Capítulo 6 são apresentados os resultados do sistema através de testes de desempenho. E por fim, no Capítulo 7 são apresentadas as considerações finais sobre o trabalho proposto e as aplicações futuras.

2 TRABALHOS RELACIONADOS

Redes Definidas por Software está sendo considerada a próxima arquitetura de rede [19]. No entanto, devido à arquitetura *SDN*, sabe-se que o gerenciamento de fluxos da rede é centralizado e executado por um controlador que está sujeito a ameaças de segurança, por exemplo, ataques de *DDoS*. Portanto, a segurança das redes *SDN* permanece indefinida e soluções relacionadas à detecção e mitigação de ataques podem ser encontradas na literatura.

De acordo com Aleroud e Alsmadi [20], quando a lógica de encaminhamento dos pacotes é centralizada e alocada no controlador, os agentes maliciosos exploram vulnerabilidades no controlador, nos *links* de comunicação entre o controlador e os dispositivos de encaminhamentos e na memória dos *switches*. Um *switch* tem memória limitada, quando está sob ataque, o número de fluxos recebido pelo dispositivo aumenta consideravelmente, ocupando toda a capacidade de armazenamento da tabela de encaminhamento. Diversos estudos têm sido desenvolvidos para criar mecanismos de defesa para suprir estas vulnerabilidades presentes na arquitetura SDN. Silva *et al.* [21] introduziram um *framework* chamado ATLANTIC (*Anomaly deTection and machine LeArNing Traffic classifICation for software-defined networking*) para detecção, classificação e mitigação de eventos anômalos em redes *SDN*. Garg e Garg [22] apresentaram um mecanismo adaptativo para atualizar dinamicamente as políticas para agregação das entradas dos fluxos e para a detecção de anomalias, de modo que a sobrecarga de monitoramento possa ser reduzida e as anomalias possam ser detectadas com maior precisão. Mousavi e St-Hilaire [23] aplicaram uma técnica para detecção de *DDoS* utilizando a entropia. O principal objetivo dos autores é detectar um ataque em seu estágio inicial, pois a detecção realizada no início do ataque é possível aplicar políticas de mitigação antes que o controlador seja completamente inundado por pacotes maliciosos.

Carvalho *et al.* [24] apresentaram uma nova abordagem para detectar e mitigar ataques de DDoS em ambientes *SDN*. O sistema proposto pelos autores é composto por quatro etapas: a primeira está relacionada à coleta e armazenamento de registros de fluxo IP; a segunda etapa é a geração de um perfil de rede normal com base nos dados coletados usando o método *ACODS* (Otimização de Colônia de Formigas para Assinatura Digital); a terceira etapa corresponde à detecção de anomalias comparando o comportamento real da rede com o perfil gerado usando *ADTW* (*Adaptive Dynamic Time Warping*) para detectar eventos suspeitos que diferem do comportamento esperado; finalmente, no quarto passo são aplicadas políticas de mitigação. De acordo com os resultados apresentados pelos autores, o sistema se mostrou eficiente nas fases de detecção e mitigação de eventos anômalos.

Um mecanismo de defesa e detecção de ataques de *DDoS* em ambientes de redes *SDN* baseado em *deep learning*, foi proposto por Li *et al.* [25]. O modelo apresentado pelos autores é composto pelas seguintes camadas: camada de entrada, camada recursiva para frente, camada recursiva reversa, camada oculta totalmente conectada e camada de saída. Na construção do modelo foi empregada *recurrent neural network* (*RNN*), *long short-term memory* (*LSTM*) e *convolutional neural network* (*CNN*). De acordo com o resultado obtido pelo modelo após a detecção, o controlador *SDN* gera políticas de descartes e as emite para os *switches*. Para a condução dos testes foi empregado o *dataset* ISCX [26] para treinar o modelo de detecção e verificar a arquitetura de defesa por meio de ataques de *DDoS* em tempo real. De acordo com os resultados apresentados, o método de defesa apresentado obteve uma taxa de acurácia de 98%.

Os ataques de *DDoS* não afetam apenas as redes do usuários finais, mas também afetam as redes dos Provedores de Serviços de Internet (*Internet Service Provider – ISP*). Todo o tráfego malicioso destinado a vítima passa através da rede *ISP*, potencialmente congestionando os *links* dentro da rede [27]. Sahay *et al.* [28] propuseram um *framework* de mitigação de *DDoS* baseado em *SDN* chamado *ArOMA*. O *framework* destina-se a integrar de forma sistemática e contínua diferentes módulos de mitigação de *DDoS*, que são distribuídos entre o *ISP* e seus clientes. Em particular, os módulos de monitoração do tráfego e detecção de anomalias foram implantados no lado do cliente, enquanto o mecanismo de mitigação é executado no lado do *ISP*. Vale ressaltar que o *framework* proposto pelos autores enfatiza na mitigação, uma vez que a detecção do ataque é feita do lado do cliente. Portanto, foi assumida como premissa que o cliente detecta fluxos suspeitos de serem ataques e compartilha os alertas com o *ISP* por meio de um canal de comunicação entre os controladores *SDN* implementados no *ISP* e o cliente. A mitigação foi realizada por meio de mecanismos de políticas implementadas no *ISP*. De acordo com o alerta emitido pelo cliente, a mitigação é aplicada para minimizar o dano do ataque. Por meio de simulações experimentais o *framework* se mostrou efetivo da mitigação de ataques de *DDoS*.

Os sistemas de detecção de intruso de rede (*Netwok Intrusion Detection Systems – NIDS*) são essenciais nas infraestruturas modernas de rede, em que a sua principal funcionalidade é monitorar e identificar o tráfego indesejável. A maioria dos *NIDS* comerciais são fundamentados em assinaturas, em que um conjunto de regras é usado para determinar o tráfego malicioso [29]. Estes sistemas são eficientes contra ameaças já conhecidas, mas a detecção fundamentada em assinatura falha quando os ataques são desconhecidos ou modificados para violar alguma das regras estabelecidas. Outra abordagem de *NIDS* são os fundamentados em anomalias em que é caracterizado um perfil como sendo normal. Esta abordagem constitui-se em analisar o tráfego da rede para definir limiares máximos e mínimos do comportamento normal. Depois de gerado este perfil normal, uma eventual anomalia é detectada quando no monitoramento em algum ponto do segmento o com-

portamento do tráfego diverge do esperado. A principal desvantagem desta técnica é a possibilidade de alta taxa de falso-positivo. Muitas vezes, tráfego de usuários legítimos é detectado como sendo anômalos, fazendo com que falsos alarmes sejam disparados ao administrador da rede. Muitos estudos na literatura vêm sendo desenvolvidos para reduzir esta taxa.

Hamamoto *et al.* [30] propuseram um sistema de detecção de anomalias aplicado às redes de larga escala. Os autores utilizaram a abordagem de *DSNSF* (Assinatura Digital de Segmento de Rede usando análise de fluxo) para gerar assinaturas do comportamento de rede normal aplicando *GA* (Algoritmo Genético). Além disso, a lógica Fuzzy foi usada juntamente com os *DSNSFs* gerados para detectar comportamentos anômalos na rede analisada. Para validar o sistema proposto, foram utilizados dados reais coletados da Universidade Estadual de Londrina utilizando sFlow. Com o uso de ferramentas de simulação de eventos anômalos, três anomalias diferentes foram injetadas nos dados reais da rede: *DoS*, *DDoS* e *Flash Crowd*. O sistema proposto se mostrou eficiente, com taxas de precisão acima de 96%. Diferentes trabalhos também aplicaram à abordagem de *DSNSF* através do uso de diferentes técnicas, como *PCA* (*Principal Component Analysis*) [31], *ARIMA* (*Autoregressive Integrated Moving Average*) [32] e *ACO* [33]. No entanto, as caracterizações do tráfego nesses trabalhos utilizaram uma abordagem que analisa de duas a quatro semanas de dados para reconhecimento de padrões e geração do perfil normal no ambiente tradicional de rede. No entanto, no modelo proposto no presente trabalho o método *PSO-DS* gera o perfil de rede apenas utilizando amostras de uma janela deslizante, não tendo necessidade de uma fase de treinamento para a caracterização do tráfego.

Diversas abordagens aplicam algoritmos de clusterização para criação de sistemas de detecção de anomalias. O K-means é um algoritmo amplamente aplicado para classificação e detecção de anomalias. No entanto, um dos problemas é a convergência local e a sensibilidade à seleção dos centróides de cada cluster. Karami e Guerrero-Zapata [34] apresentaram um novo sistema de detecção de anomalias para Redes Concentradas em Conteúdo (*Content-Centric Networks-CCN*) que funciona em duas etapas. Na primeira etapa foi aplicada uma abordagem híbrida do *PSO* e o K-means com duas funções objetivos, uma para determinar o quanto os clusters estão separados e a otimização local para determinar o número ideal de clusters. Na segunda etapa foi aplicada a lógica fuzzy para a classificação de detecção de anomalias. Resultados experimentais demonstram que o algoritmo proposto pode atingir o número ideal de clusters, clusters bem separados, assim como aumentar a alta taxa de detecção e diminuir a taxa de falso-positivos.

Algumas abordagens de detecção de anomalias aplicam a combinação de técnicas de aprendizado de máquina que são chamados métodos *ensamble*. A aplicação de vários algoritmos é utilizada para obter um melhor desempenho preditivo. Aburomman e Reaz [35] propuseram um *IDS* aplicando três diferentes técnicas: *SVM* (*Support Vector Ma-*

chine), *PSO* e *K-NN* (*k-Nearest Meighbour*). Na fase de treinamento foram aplicados seis classificadores *K-NN* e seis classificadores *SVM* no mesmo conjunto de dados. Então foi aplicado o *PSO* que combina a saída dos 12 classificadores para chegar a um classificador final. Para a validação do sistema proposto os autores utilizaram cinco subgrupos aleatórios do conhecido conjunto de dados do KDD99. A métrica de avaliação utilizada foi a acurácia em que a média se encontra no valor de 92%.

A Tabela 1 apresenta o conjunto de dados, o ambiente de rede utilizado e a acurácia para os trabalhos discutidos neste capítulo. Alguns destes trabalhos a acurácia não foi avaliada pelos autores.

Tabela 1 – Trabalhos discutidos.

Trabalho	**Dados**	**Ambiente**	**Acurácia**	**Técnica**
Aleroud e Alsmadi [20]	Fluxo IP emulado	*SDN*	88,7%	Assinatura de anomalias
Silva *et al.* [21]	Fluxo IP emulado	*SDN*	-	K-means e SVM
Garg e Garg [22]	Fluxo IP emulado	*SDN*	96%	Regras
Carvalho *et al.* [24]	Fluxo IP emulado	*SDN*	-	ACO e ADTW
Li *et al.* [25]	Pacotes Dataset ISCX2012	*SDN*	98%	RNN, LSTM, CNN
Sahay *et al.* [28]	Fluxo IP emulado	*SDN*	-	Assinatura de anomalias
Hamamoto *et al.* [30]	Fluxo IP real	Tradicional	96,5%	GA e Fuzzy
Fernandes *et al.* [31]	Fluxo IP real	Tradicional	96,6%	PCA
Pena *et al.* [32]	Fluxo IP real	Tradicional	89%	ARIMA
Carvalho *et al.* [33]	Fluxo IP real	Tradicional	96,26%	ACO e ADTW
Karami e Guerrero-Zapata [34]	Pacotes emulado	CCN	-	PSO, K-means e Fuzzy
Aburomman e Reaz [35]	KDD 99	Tradicional	92%	K-NN, SVM e PSO

Os trabalhos apresentados fornecem soluções para detecção de anomalias no ambiente tradicional e no ambiente *SDN*. No entanto, essas soluções não apresentam um tempo de análise próximo ao tempo real. O sistema proposto nesta dissertação explora a coleta e a análise do tráfego utilizando diferentes tempos de coleta para detecção, classificação e aplica políticas de mitigação de acordo com o tipo de anomalia detectada.

3 REDES DEFINIDAS POR SOFTWARE

Redes de computadores tornaram-se estruturas dinâmicas e complexas. Em consequência deste fato, a configuração e o gerenciamento de toda esta estrutura é uma atividade desafiadora. As estruturas de rede geralmente são compostas por um grande número de dispositivos de rede, tais como *switches*, roteadores e *appliance* de rede (*firewalls*, sistemas de detecção de intrusão, etc.). Cada um destes dispositivos possuem complexos protocolos implementados. É da responsabilidade do administrador de rede configurar todos os elementos que compõem a estrutura de rede para atender às diversas políticas de alto nível para responder aos inúmeros eventos de rede que possam ocorrer. No entanto, a configuração de rede permanece difícil, pois, para atender as especificações das políticas de alto nível são necessárias implementações distribuídas em termos de configurações de baixo nível [36].

As infraestruturas das redes são compostas por equipamentos proprietários, fechados e de alto custo. A configuração destes dispositivos é complexas, pois cada um dos dispositivos possui uma documentação e inúmeros comandos, e a forma de configuração varia de um fabricante para o outro. Também vale ressaltar que esses equipamentos possuem *APIs* fechadas, consequentemente limita a flexibilidade em relação à configuração e ao controle interno. Outro fator complexo é que o modelo de rede atual é integrado verticalmente. O plano de controle (responsável pela inteligência da rede, ou seja, define quais as rotas os pacotes vão tomar ao longo da rede) e o plano de dados (responsável por encaminhar os pacotes na rede) estão incorporados dentro dos dispositivos de rede [2].

Redes definidas por software (*Software-Defined Networking – SDN*) é um paradigma de rede que por meio de uma abstração de planos de rede busca separar o plano de dados e o plano de controle com o objetivo de mudar as limitações relacionadas à configuração da infraestrutura de rede.

Neste capítulo serão apresentados alguns conceitos sobre redes *SDN*. Inicialmente serão apresentados os estudos iniciais desenvolvidos até a consolidação paradigma de rede *SDN*, os detalhes dos componentes presentes na arquitetura, e por fim serão apresentados os atuais controladores de redes disponíveis.

3.1 Evolução Histórica de Redes Programáveis

A história das redes programáveis pode ser dividida em três partes [37]: (1) redes ativas (de meados da década de 90 até o início dos anos 2000) que introduziram funções programáveis à rede, levando a uma maior inovação; (2) separação entre o plano de controle e o de dados (de meados de 2001 até 2007) desenvolveu estudos de interfaces

abertas entre o plano de controle e o plano de dados; (3) protocolo *Openflow* e sistemas operacionais de rede (teve início em 2007 e sua consolidação em 2013) representaram a primeira adoção de interface aberta e desenvolvimento de meios para a separação do plano de controle e do plano de dados escalável e prática.

Em meados da década de 90 ocorreram avanços em relação à quantidade de aplicações e recursos, ultrapassando as simples funcionalidades propostas no início das redes que eram apenas transferência de arquivos e o envio e recebimento de e-mails. Inúmeras aplicações e maior utilização pelo público atraíram pesquisadores com interesse para desenvolver e testar a implantação de novas ideias com intuito de melhorar os serviços de rede. Os pesquisadores deram início ao desenvolvimento e testes de novos protocolos de rede em pequenas configurações de experimentação e comportamento simulado em redes maiores. No entanto, a motivação e o financiamento não continuaram, pois, levar as ideias para o IETF (*Internet Engineering Task Force*) para a padronização destes protocolos era um processo lento que acarretou na frustração de diversos pesquisadores.

A rede ativa representou uma outra etapa importante para o controle de rede visando uma interface de programação (*Application Programming Interface - API*) de rede, que expôs recursos como, por exemplo, processamento, armazenamento e filas de pacotes em nós individuais, dando suporte à construção de funcionalidades personalizadas para aplicar a um subconjunto de pacotes que passa pelo nó.

A comunidade de rede buscou dois modelos de programação [38][39]:

- O modelo de cápsula: em que o código executado nos nós era carregado na interface de entrada (*In-Band*) dos pacotes de dados;
- O modelo de roteador/*switch* programável: em que o código era executado por um mecanismo de rede fora-de-banda (*Out-of-Band*), ou seja, regiões onde não se trafega dados de uma rede.

Um importante acelerador no ecossistema de rede ativa foi o interesse das agências de financiamento, em particular pela *DARPA* (Agência de Projetos de Pesquisa Avançada de Defesa dos Estados Unidos) desde meados da década de 90 até o início dos anos 2000 [40]. Embora nem todas as pesquisas em redes ativas tenham sido financiadas pela *DARPA*, o programa de financiamento apoiou uma coleção de projetos e, talvez mais importante, encorajou a convergência em uma terminologia e conjuntos de componentes de redes ativos para que os projetos pudessem contribuir para um todo [41].

No começo dos anos 2000, tentativas iniciais de separar os planos de controle e de dados foram relativamente pragmáticas, mas representaram um ponto de partida do desacoplamento rígido convencional da Internet para a computação dos caminhos e o encaminhamento de pacotes. Os esforços para separar os planos de controle e de dados

resultaram em vários conceitos que foram transferidos para os projetos *SDN* subsequentes [37]:

- Controle logicamente centralizado usando uma interface aberta para o plano de dados: o *ForCES*, um grupo de pesquisa do *IETF*, propôs uma interface padrão e aberta ao plano de dados para permitir a inovação no *software* do plano de controle. O *SoftRouter* [42] usou a *API ForCES* para permitir que um controlador separado instalasse as entradas da tabela de encaminhamento no plano de dados, permitindo a remoção completa da funcionalidade de controle dos roteadores. No entanto, o *ForCES* não foi adotado pelos principais fornecedores de roteadores, o que dificultou a implantação incremental. O *OpenFlow* também enfrentou desafios e restrições similares de compatibilidade com versões anteriores, em particular, a especificação *OpenFlow* baseou-se em compatibilidade com *switches* de *commodities*.

- Gerenciamento de estado distribuído: os controladores de rotas logicamente centralizados enfrentaram desafios envolvendo gerenciamento de estado distribuído. Um controlador logicamente centralizado deve ser replicado para tratar as falhas do controlador e garantir a escalabilidade, mas a replicação pode apresentar um potencial estado inconsistente entre as replicações. Os pesquisadores exploraram os possíveis cenários de falhas e requisitos de consistência. No caso de controle de roteamento, as réplicas do controlador não precisavam de um protocolo geral de gerenciamento do estado, uma vez que cada réplica eventualmente calcularia as mesmas rotas. Para uma melhor escalabilidade, cada instância do controlador pode ser responsável por uma parte separada da topologia. Essas instâncias de controle poderiam trocar informações de roteamento entre si. Os desafios surgiram novamente vários anos depois no contexto de controladores *SDN* distribuídos, exigindo soluções mais sofisticadas para o gerenciamento de estado distribuído [43].

Antes do surgimento do protocolo *OpenFlow*, as ideias subjacentes à *SDN* já enfrentavam uma tensão entre a visão de redes totalmente programáveis e o pragmatismo que permitiriam essa implementação na prática. O *OpenFlow* conseguiu o equilíbrio entre estes dois objetivos, permitindo mais funções do que os controladores de rotas já existentes. Embora depender do *hardware* dos *switches* existentes limitasse um pouco a flexibilidade, o *OpenFlow* foi quase imediatamente implantável. A criação da *API OpenFlow* foi seguida rapidamente pelo *design* de plataformas de controle como o *NOX* que permitiu a criação de diversos novos aplicativos de controle.

Embora o *OpenFlow* tenha incorporado muitos dos princípios do trabalho anterior sobre a separação do plano controle e os plano de dados, também ofereceu várias contribuições intelectuais adicionais [37]: generalização dos dispositivos e funções de rede; visão de um sistema operacional de rede, e técnicas de gerenciamento de estado distribuídas.

3.2 Arquitetura

O termo Redes Definidas por *Software* (*Software-Defined Networking-SDN*) foi originalmente utilizado para representar as abstrações e trabalhos entorno do protocolo *OpenFlow* na Universidade de *Stanford*. Conforme é definida, *SDN* é a uma arquitetura de rede onde o estado de encaminhamento no plano de dados é gerenciado remotamente por um plano de controle desacoplado do dispositivo de rede. A Figura 1 apresenta a arquitetura proposta para as redes SDN.

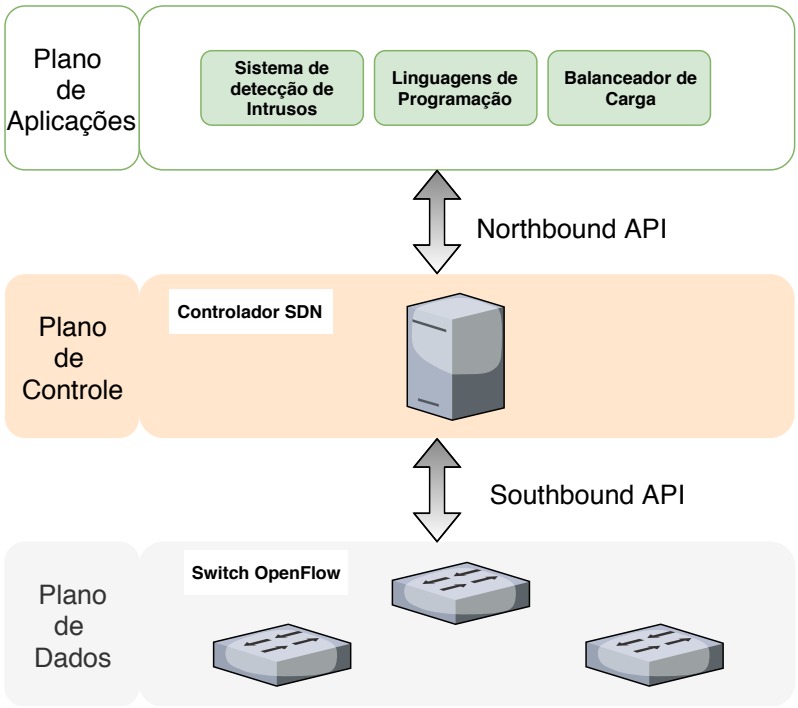


Figura 1 – Arquitetura das Redes Definidas por Software (Adaptada de [2]).

Conforme foi definida em [2], a arquitetura *SDN* é dividida em quatro pilares:

1. Os planos de controle e os planos de dados são desacoplados. A funcionalidade de controle é removida dos *switches*, que se tornaram simples elementos de encaminhamentos de pacotes.

2. As decisões de encaminhamento são baseadas em fluxos, ao invés de serem baseadas em destino. Um fluxo é definido por um conjunto de campos de pacote que atuam como um filtro de correspondência e um conjunto de ações (instruções). No contexto de *SDN*, um fluxo é uma sequência de pacotes entre a origem e o destino. Todos os pacotes de um fluxo recebem políticas de serviço idênticas nos dispositivos de encaminhamento. A abstração de fluxo permite unificar o comportamento de diferentes tipos de dispositivos de rede, incluindo roteadores, *switches*, *firewalls* e *appliance* de rede. A programação de fluxo permite uma flexibilidade sem precedentes, limitada apenas às capacidades das tabelas de fluxos implementadas.

3. Toda lógica de controle desacoplada do *switch* é movida para o chamado controlador *SDN* ou sistema operacional de rede (*Network Operating System - NOS*). O *NOS* é uma plataforma de *software* que funciona em tecnologia de servidor de *commodities* e fornece recursos e abstrações essenciais para facilitar a programação de dispositivos de encaminhamento com base em uma visão global da topologia de rede centralizada e lógica.

4. A rede é programável através de aplicativos de software executados sobre o *NOS*, o qual interage com os dispositivos do plano de dados subjacentes. Esta é uma característica fundamental da rede *SDN*, ou seja, é considerada como a principal proposta de valor.

A comunicação entre o plano de dados e o plano de controle é feita pela *Sothbound API* que está localizada nos *switches OpenFlow* e a comunicação entre o plano de controle e as aplicações é feita pela *Northbound API* que está localizada no controlador [44].

Seguindo as concepções introduzidas em [45], redes *SDN* são definidas por três fundamentais abstrações: (i) encaminhamento, (ii) distribuição e (iii) especificação. A abstração de encaminhamento deve permitir qualquer comportamento de encaminhamento desejado pelo controlador. O *OpenFlow* é uma realização desta abstração, que pode ser vista a um "driver de dispositivo"em um sistema operacional [2].

A abstração da distribuição deve proteger as aplicações *SDN* das mudanças de estado distribuído da rede, que pode ser um problema de controle distribuído logicamente centralizado. Sua realização requer uma camada de distribuição comum que na *SDN* se encontra no *NOS*. Essa camada possui duas funções essenciais. Primeiro, é responsável por instalar os comandos de controle nos *switches*. Em segundo lugar, ele coleta as informações de *status* sobre a camada de encaminhamento (*switches* e nos *links*) para oferecer uma visão global dos aplicativos de rede junto ao controlador [2].

A abstração de especificação deve permitir que a aplicação da rede expresse o comportamento desejado sem ser responsável por implementar esse comportamento em si. Isto pode ser alcançado através de soluções de virtualização, bem como linguagens de programação de rede. Essas abordagens mapeiam as configurações abstratas que as aplicações expressam com base em um modelo simplificado e abstrato da rede em uma configuração física para a visão de rede global exposta pelo controlador *SDN* [2].

3.2.1 Plano de Dados

No plano de dados (*data plane*), os dispositivos de encaminhamento (*switch*, roteador) são responsáveis pelas atividades de encaminhamento de pacotes e a implementação das demais ações relacionadas aos pacotes. O encaminhamento de pacotes é uma das funções primárias do plano de dados. A programação do plano de dados permite várias

funções, tais como, a inspeção profunda de pacotes, o processamento dentro da rede, a engenharia de tráfego e detecção de anomalias. Através da separação entre o plano de dados e o plano de controle é possível oferecer suporte a novos protocolos para satisfazer os requisitos destas funções, pois o modelo de rede *SDN* possui uma infraestrutura aberta tendo flexibilidade para adaptação a novos protocolos [46].

Os principais elementos presentes na arquitetura *SDN* são os controladores e os dispositivos de encaminhamento. Os pacotes são encaminhados pelos *switches*, tendo como base a entrada das tabelas de fluxos presentes nos *switches*. As tabelas fluxos são compostas por várias entradas de fluxos e seu cabeçalho consiste em seis campos: *Match*, *Priority*, *Counters*, *Action*, *Timeouts* e *Cookie*. O campo *Match* é utilizado para encontrar a correspondência dos pacotes que pertencem ao mesmo fluxo, que consiste na porta de entrada e nos campos do cabeçalho do pacote. O campo *Priority* é aplicado para garantir a correspondência de prioridade da entrada de fluxo. Os campos *Counters* são utilizados para manter a estatística de utilização dos pacotes de um fluxo. O campo *Action* define como os pacotes serão processados e para onde serão encaminhados. O *Idle_timeout* é o tempo inativo do fluxo antes da expiração e *Hard_timeout* é o tempo que o fluxo será mantido no *switch*. O *Cookie* é um campo escolhido pelo controlador para filtrar estatísticas, modificação ou exclusão de um fluxo.

Um *switch Openflow* consiste em três partes principais: a tabela de fluxo, um canal seguro para comunicação com o controlador, geralmente utilizando *TLS* (*Transport Layer Security*) [47] ou *SSL* (*Secure Socket Layer*) [48] e o protocolo *OpenFlow* [49] que é utilizado para comunicação e gerência dos *switches*. Os *switches OpenFlow* podem ser de dois tipos: puro ou híbrido. Os *switches OpenFlow* puro não possuem recursos do modelo legado rede ou controle acoplado e dependem de um controlador para as decisões de encaminhamento. Já os *switches* híbridos suportam o *OpenFlow* e oferece suporte para operações e protocolos tradicionais. A maioria dos *switches* comerciais disponíveis atualmente são chamados de *OpenFlow enable*.

3.2.2 Plano de Controle

Os controladores *SDN* estão no centro da arquitetura, eles atuam como uma camada intermediária entre as aplicações pela interface de borda norte (*Northbound Interface* – NBI) e o plano de dados pela interface de borda sul (*Southbound interface* - SBI) [50]. O controlador desempenha um papel principal atuando como a inteligência da rede e fornece uma visão abstrata e centralizada da rede em sua totalidade. O controlador fornece as aplicações ou aos usuários a capacidade de interagir com o controle da rede e gerenciar os serviços e os recursos relacionados à rede. A rede pode ter mais de um controlador, de modo que cada controlador é responsável por controlar um domínio de *switches* da rede [43].

O protocolo *OpenFlow* define uma *API* para a comunicação entre o controlador e os *switches OpenFlow*. O controlador através de mensagens *OpenFlow* realiza a manipulação das tabelas de fluxos dos *switches* adicionando, atualizando e excluindo as entradas de fluxos. Isto acontece de forma reativa, ou seja, quando o controlador recebe um pacote do *switch*, ou de forma proativa dependendo da implementação do controlador. O controlador manipula as tabelas de fluxos para determinar como os pacotes serão manipulados pelos *switches OpenFlow*. Quando um pacote chega ao *switch OpenFlow*, a ação a ser tomada é feita com base na comparação entre o cabeçalho do pacote e a regra que deve ser seguida segundo a tabela de entrada de fluxo. Então os contadores são atualizados e a ação correspondente é realizada.

Caso não encontre nenhuma regra de correspondência entre o pacote e as regras da tabela de entrada de fluxo, o pacote é encaminhado por completo ao controlador. Se houver uma correspondência do pacote na tabela de entrada de fluxos então é encaminhado ao controlador apenas o cabeçalho dos pacotes. Os pacotes que chegam ao controlador, usualmente, correspondem ao primeiro pacote de um fluxo. Dependendo do tipo de aplicação, o controlador pode instalar uma regra no *switch* para que todos os pacotes de um determinado fluxo seja enviado ao controlador para a realização do tratamento individual de cada um deles. Este tipo de pacote corresponde a pacotes de controle (ICMP, DNS, DHCP) ou de roteamento (OSPF, BGP) [51].

Atualmente, diversos tipos de controladores *OpenFlow* foram desenvolvidos no contexto das redes SDN. A Tabela 2 sumariza as principais implementações dos controladores *SDN*.

Tabela 2 – Implementações dos controladores SDN.

Controlador	**Implementação**	**Desenvolvedor**
NOX	C++	Nicira
POX	Python	Nicira
Trema	Ruby/C	NEC
Mastro	Java	Universidade Rice
Beacon	Java	Stanford
Onix	Python/C	Nicira/Google/NEC
SNAC	C++	Nicira
Ryu	Python	NTT
MUL	C	Kulcloud
Floodlight	Java	BigSwitch

- **NOX**: é o controlador original *OpenFlow*, e a sua linguagem original é o C++. Na sua primeira versão, fornecia uma *API* para utilização de *scripts* em Python. O NOX é um sistema operacional simples para redes programáveis e que provê primitivas para o gerenciamento de eventos e para as funções de comunicação com os *switches*.

O NOX fornece uma interface programável de alto nível para o desenvolvimento para dispositivos de encaminhamento e aplicações. Foi projetado para suportar pequenas redes de alguns *hosts* e até mesmo para grandes redes comerciais de centenas de *switches*.

- **POX**: o controlador POX é uma outra plataforma de controle SDN, foi desenvolvido com base no NOX. Ele é escrito completamente em Python, resultando em uma interface mais amigável e simples. O POX é mais adequado em casos que o desenvolvedor queira utilizar a linguagem de programação Python.

- **Trema**: é um *framework* de controlador *OpenFlow* que fornece uma interface para criação de controladores *SDN* utilizando Ruby e C. O escopo do projeto do Trema é auxiliar os desenvolvedores a criar de maneira simples seus próprios controladores. Não possui como objetivo uma implementação específica de controlador *SDN*.

- **Mastro**: é um ambiente escalável de controlador *OpenFlow* para criação de controladores *SDN* utilizando Java. O Maestro foi projetado para explorar o paralelismo em um servidor, com objetivo de melhorar o sistema de transferência. Busca reduzir o esforço dos programadores no que diz respeito ao gerenciamento e paralelização.

- **Beacon**: o Beacon é um controlador *OpenFlow*, multi-plataforma, modular baseado em Java, que oferece suporte à operação baseada tanto em eventos quanto em *thread*. Fornecendo uma estrutura para controle de dispositivos de rede utilizando o protocolo *OpenFlow* e um conjunto de aplicativos internos que fornecem a funcionalidade de plano de controle comum. O sistema operacional tem uma estrutura modular que permite que o controlador seja atualizado em tempo de execução sem a necessidade de interromper outras atividades de encaminhamento de pacotes.

- **Onix**: Onix é um controlador de rede desenvolvido em conjunto pela Google, NEC e Nicira, possui com objetivo o controle de redes de grandes dimensões. É um controlador distribuído que permite a escalabilidade da rede, particionando a estrutura de controle de rede em múltiplos controladores. A visão da topologia de rede é mantida em uma estrutura denominada *NIB* (*Network Information Base*) que é núcleo do sistema, em que para manter a escalabilidade é distribuído nos servidores de controle.

- **SNAC**: o SNAC é um controlador OpenFlow baseado no NOX-0.4 que utiliza C++. Ele suporta uma interface gráfica com o usuário e uma linguagem de definição de políticas e usa um gerenciador de políticas de fácil manipulação baseado na Web para gerenciar a rede, configurar dispositivos e monitorar eventos.

- **Ryu**: o Ryu é um sistema operacional *SDN* que visa fornecer controle logicamente centralizado e APIs para criar novos aplicativos de gerenciamento e controle de rede.

A linguagem utilizada pelo controlador é o Pyhton. É uma estrutura *SDN* baseada em componentes que suporta *OpenFlow* v1.0, v1.2 e v1.3.

- **MUL**: MUL é um controlador *OpenFlow* que suporta uma infraestrutura multi-thread. Foi desenvolvido pela Kucloud e oferece suporte para linguagem C. Suporta *OpenFlow* v1.0 e v1.3.

- **Floodlight**: o Floodlight é um controlador *SDN* desenvolvido pela BigSwitch baseado em Java. O controlador oferece suporte a uma ampla variedade de *switches OpenFlow* virtuais e físicos e pode operar com redes mistas *OpenFlow* e não *OpenFlow*.

3.3 Conclusão do Capítulo

O ambiente tradicional de rede é composto de diversos equipamentos e as configurações destes dispositivos diferem de um fabricante para outro. Os planos de dados e controle são acoplados nos dispositivos de rede limitando a configuração e o controle interno. As Redes *SDN* surgem para tornar flexível o controle e a gerência da infraestrutura de rede.

Neste capítulo foi apresentado um resumo da evolução histórica das redes programáveis. Os estudos tiveram início com a introdução de rede ativas até a consolidação da separação entre o plano de controle e do plano de dados de modo escalável e prático. Também foram apresentados os conceitos e elementos que compõem a arquitetura *SDN*. E por fim, foram apresentadas as principais implementações de controladores *SDN*.

No próximo capítulo serão apresentados os conceitos de sistema de detecção de intrusos e os tipos de anomalias que afetam o comportamento normal da rede. Também serão exploradas as diferentes técnicas aplicadas para a construção deste tipo de sistema.

4 SISTEMA DE DETECÇÃO DE INTRUSO E ANOMALIAS EM REDES DE COMPUTADORES

Os sistemas de detecção de intrusos possuem duas classificações [52]: baseado em *host* (*Host-Based Intrusion Detection System - HIDS*) e o baseado em rede (*Netwok-Based Intrusion Detection System – NIDS*). Devido a constante evolução e a necessidade de novas ferramentas para aplicação na detecção de intrusão surgiram o sistemas de prevenção de intrusão (*Intrusion Prevention System – ISP*), os *ISP* são dispositivos reativos que não buscam apenas detectar, mas também impedir a ocorrência de ataques. Os *IDS* híbridos (*Hybrid – IDS*) são uma combinação entre *HIDS* e *NIDS* aproveitando as melhores características de cada um dos tipos. A utilização de *IDS* híbridos é importante para a proteção contra diferentes tipos de ameaças, tanto ataques realizados internamente quanto realizados externamente.

4.1 Host-Based Intrusion Detection System – HIDS

Os *HIDS* realizam o monitoramento através da coleta de informações de arquivos de *logs* sobre as atividades de um único sistema ou *host*. Esses agentes baseados em *host*, que às vezes são chamados de sensores, normalmente são instalados em uma máquina que é considerada suscetível a possíveis ataques. Os sensores funcionam coletando dados sobre eventos que ocorrem no sistema que está sendo monitorado. Esses eventos podem ter acessos e alterações em arquivos do sistema, alterações de privilégios dos usuários, quais os programas estão sendo executados, e demais atividades que forem realizadas [53]. Todos os dados gerados são registrados por um mecanismo chamado de trilhas de auditoria [54].

As informações coletadas por meio de trilhas de auditoria baseadas em *host* contém dados úteis sobre o sistema e seus usuários. Por exemplo, as trilhas de auditoria podem conter informações sobre os responsáveis por um evento, bem como quaisquer objetos relacionados a esse evento. Sendo essencial para identificar qual processo deu início a um evento, e por consequência identificar os usuários associados a esse evento. Em ataques maliciosos gerados internamente, a identificação do gerador do ataque é útil para buscar medidas punitivas contra o usuário. Como os sistemas baseados em *host* podem monitorar o acesso a informações em termos de “quem acessou o quê”, esses sistemas podem rastrear atividades mal-intencionadas ou impróprias para um usuário específico.

Os *HIDS* também possuem diversas desvantagens. Uma delas é que eles não podem ver o tráfego de rede. Conforme mencionado anteriormente, os sistemas baseados em *host* são altamente dependentes dos registros do sistema. Qualquer vulnerabilidade existente a

este sistema enfraquecerá a integridade do sensor baseado em *host*. Se um intruso puder encontrar e explorar uma destas fraquezas, isso pode levar a um ataque que é difícil de capturar e uma vulnerabilidade difícil de corrigir. Uma vez que as trilhas de auditoria são usadas como fonte de informação, elas podem ocupar um espaço de armazenamento significativo, além de aumentar a carga do servidor de hospedagem.

4.2 Network-Based Intrusion Detection System – NIDS

Os sistemas de detecção intrusos baseados em rede realiza o monitoramento através da coleta de informações da própria rede. O *NIDS* verifica ataques ou comportamentos irregulares da rede inspecionando os cabeçalhos dos pacotes ou aplicando protocolos patronizados para o monitoramento. O *SNMP* (*Simple Network Management Protocol*) [55], *NetFlow* [56], *sFlow* [57] e *IPFIX* (*IP Flows Information Export*) [58] são alguns exemplos de protocolos de monitoramento do tráfego da rede.

Os *NIDS* possuem 2 abordagens para detecção de intrusos [59]:

- **Padrões de assinatura**: a detecção de intruso baseada em padrões de assinatura é necessária uma grande base de dados contendo as assinaturas para determinar se uma anomalia veio a ocorrer. Para a detecção é necessário que o sistema monitore as atividades da rede e caso encontre um conjunto de atributos que estejam na base de assinaturas é disparado um alarme para o administrador da rede. Uma das grandes vantagens deste tipo de abordagem é a diminuição da taxa de falsos-positivos que um é grande desafio enfrentado na detecção de anomalias. Pelo fato de que só será detectada uma anomalia de uma assinatura pré-estabelecida, ou seja, esta anomalia já tenha ocorrido antes. A desvantagem desta técnica é que o tamanho da base de assinatura pode ser muito grande, e isto pode afetar no desempenho para a detecção em tempo real. Outra desvantagem é a constante atualização da base de assinaturas, pois uma anomalia que não esteja registrada não será detectada.

- **Detecção de anomalia**: em que é caracterizado um perfil como sendo normal, constitui-se em analisar o tráfego passado da rede para definir limiares máximos e mínimos do comportamento normal. Depois de gerado este perfil normal, uma eventual anomalia é detectada quando no monitoramento em algum ponto do segmento é ultrapassado um destes limiares pré-definidos. Uma das principais vantagens deste tipo de metodologia é a detecção de anomalias não conhecidas, pois os modelos criados são baseados no comportamento normal. Outra vantagem é que não há necessidade de uma base com informações das anomalias, isto faz com que a detecção em tempo real seja uma vantagem em relação à detecção de anomalias utilizando padrões de assinatura. Porém, a desvantagem da aplicação desse tipo de aborda-

gem é a ocorrência de falso-positivos em que o tráfego legítimo é detectado como anômalo.

4.2.1 Anomalias em Redes

Na literatura as anomalias de redes podem ser classificadas em duas categorias [60]. A primeira está relacionada com problemas de falhas e desempenho, também definida no trabalho de Zarpelão [61] como sendo anomalias em que não há presença de agentes maliciosos. Já a segunda classificação está relacionada a problemas de segurança, ou seja, ligados a ataques como o propósito de violar a segurança da rede.

Anomalias causadas por falhas e desempenho podem ocorrer como, por exemplo, em casos em que muitos usuários fazem requisições a um determinado servidor de arquivo ou quando algum nó da rede venha a falhar, sobrecarregando outro nó da rede. De acordo com [61], as seguintes causas que provocam este tipo de anomalia são:

- ***Flash crowd***: ocorre quando um elevado número inesperado de usuários envia massivas solicitações a um serviço *Web*, provando um pico no tráfego. Este tipo de situação geralmente acontecem quando grandes eventos mundiais ocorrem, fazendo com que inúmeros usuários realizem solicitações às plataformas de notícia em busca de informações [62][63].

- ***Babbling node***: é uma situação em que um nó envia pequenos pacotes para diversos nós da rede, entrando em loop infinito. Casos como este ocorrem quando o nó envia solicitações para verificar relatório de *status* [60].

- **Tempestade de *Broadcast***: é uma reação em cadeia em que um grande volume de pacotes *broadcast* trafegam pela rede. Essa situação normalmente ocorre quando um nó falha e inicia o envio de pacotes *broadcast*, fazendo com que os demais nó repliquem a transmissão destes pacotes [64].

- **Congestionamento**: inúmeros pacotes sendo transmitidos podem provocar um pico no tráfego da rede. Quando um congestionamento ocorre pode provocar atrasos na transmissão e também a perda de pacotes [65]. Os itens mencionados anteriormente também causam congestionamento [61].

- ***Bugs* no *software* de roteamento**: pacotes que chegam até os equipamentos de roteamento podem apresentar alguma deformação em seu formato, muitas vezes os *softwares* dos equipamentos não estão preparados para operar com esse tipo de situação e acabam causando erros de encaminhamento, afetando a carga de tráfego da rede monitorada [66][61].

- **Erros de configuração**: erros em configurações de servidores podem dificultar o acesso dos usuários a determinados serviços oferecidos pelo mesmo, isto pode

acarretar em congestionamento nos canais de transmissão da rede. Conforme já mencionado, o congestionamento é um dos fatores que levam a rede apresentar comportamentos anômalos [60][61].

Por outro lado, anomalias causadas por questões de segurança estão relacionadas com a presença de agentes maliciosos sobrecarregando os níveis de tráfego da rede. Esta situação ocorre onde o agente sobrecarrega um determinado serviço com objetivo de torná-lo indisponível. Também quando um alto volume de tráfego malicioso é injetado, prejudicando o tráfego legítimo dos usuários, tendo como consequência a saturação da capacidade do *link* da rede [60].

As principais causas desse tipo de anomalia são [61]:

- **Ataque de negação de serviço**: nos ataques de negação de serviço (*Denial of Service - DoS*), o agente malicioso sobrecarrega o tráfego com o objetivo de indisponibilizar um serviço. Este grande volume de tráfego gerado faz com que toda a largura de banda fique ocupada, gerando grande lentidão nos serviços de rede. Outra prática é constituída por ataques distribuídos (*Distributed Denial of Service - DDoS*), ou seja, é um ataque realizado em grupo [67].
- ***Worm***: é um programa que tem a capacidade de multiplicar-se automaticamente enviando cópias para outros computadores ao longo da rede. Devido a sua grande capacidade de gerar e propagar cópias de si mesmo, consequentemente, afeta o desempenho da rede [67][68].
- ***Portscan***: esta técnica é utilizada por gerentes de redes para saber o *status* das portas, se elas estão abertas, escutando ou fechadas. No entanto, esta técnica também é utilizada por agentes maliciosos para descobrirem alguma vulnerabilidade a fim de realizarem invasões ao sistema [61][67].

4.2.2 Técnicas para detecção de anomalias

A detecção de anomalias é uma tarefa importante na análise de dados que possui como objetivo detectar anormalidades em um determinado conjunto de dados. É uma área de grande interesse de pesquisa de mineração de dados, pois envolve a descoberta de padrões dentro de um conjunto de dados [8]. No trabalho de Ahmed *et al.* [8] a taxonomia das técnicas utilizadas para detecção de anomalias é dividida em quatro categorias:

- **Técnicas de classificação**: através de uma base de dados rotulados, o classificador é treinado para identificar, em uma nova observação, a ocorrência de uma anomalia.
- **Estatística**: um modelo estatístico é aplicado para criar um perfil de eventos normais.

- **Clusterização**: refere-se a algoritmos de aprendizado não supervisionado que não exigem dados rotulados e separa as amostras de acordo com a suas características para classificá-las
- **Teoria da Informação**: várias medidas, como entropia, entropia condicional, entropia relativa, ganho de informação e custo de informação são usadas para explicar as características de um conjunto de dados.

4.2.2.1 Técnicas de classificação

As técnicas baseadas em classificação dependem de dados rotulados sobre as caraterísticas dos ataques de rede, pois as técnicas de aprendizado supervisionado necessitam de uma fase de treinamento do modelo para aprender quais são as características de um ataque para posteriormente serem empregadas na fase de identificação e classificação de eventos anômalos. Esse tipo de abordagem é capaz de detectar as anomalias que foram fornecidas na fase de treinamento do modelo. Isso demonstra um ponto fraco no emprego deste tipo de técnica, pois o sistema estará vulnerável a novos tipos de ataques que constantemente aparecem em novas versões e possuem comportamentos diferentes.

Algumas técnicas encontradas na literatura para a detecção de anomalias são:

- **Rede Neural Artificial (RNA)**: são algoritmos de aprendizado de máquina que são inspirados no comportamento dos neurônios biológicos. Essa abordagem é aplicável em diversos cenários para o reconhecimento de padrões. As redes neurais são divididas em camadas e os neurônios são interligados através de conexões denominadas pesos sinápticos. Os sinais de entradas são propagados até a camada de saída através da ligação entre os neurônios. A capacidade de uma rede neural para classificação de dados também tem sido aplicada na detecção de anomalias [69].
- **Rede Bayesiana**: uma rede bayesiana é uma abordagem eficiente para modelar um domínio que contenha incerteza. Uma variável aleatória discreta é representada usando um grafo acíclico direcionado (DAG), onde cada nó reflete o estado da variável aleatória e contém uma tabela de probabilidade condicional (CPT). A tarefa da CPT é fornecer a probabilidade de um nó estar em um estado específico. A rede Bayesiana é utilizada para problemas de classificação e tem sido aplicada em sistemas de detecção de anomalias [70].
- **Máquina de Vetores de Suporte (SVM)**: O algoritmo SVM padrão é uma técnica de aprendizagem supervisionada, que requer dados rotulados para criar uma regra de classificação. No entanto, ele também pode ser adaptado como um algoritmo de aprendizado sem supervisão, pelo qual ele tenta separar todo o conjunto de dados de treinamento de sua origem, enquanto o SVM supervisionado regularmente tenta separar duas classes por um hiperplano [71].

4.2.2.2 Modelos Estatísticos

Os modelos estatísticos aplicados para detecção de anomalias têm sido empregados devido ao baixo custo computacional e a grande eficiência que eles possuem para identificar comportamentos anômalos no tráfego de rede. Algumas técnicas podem ser citadas:

- **Modelo de Mistura (*Mixture Model*)**: o modelo tem sido utilizado no processo de identificação de comportamentos anormais da rede baseado na análise de pacotes. Os modelos de misturas são um tipo de modelo de densidade que compreende em um número de funções de componentes, geralmente gaussianas. A distribuição de vetores de características é extraída dos pacotes da rede e modelado por uma densidade de mistura gaussiana a fim de obter os dados para modelagem estatística [72].

- **Wavelet**: a wavelet é uma abordagem de processamento de sinais eficiente devido à capacidade de decompor os sinais do tráfego de rede, permitindo simultaneamente auxiliar no reconhecimento de diferentes ruídos e tendências. A abordagem de processamento de sinais tem sido aplicado em diversos trabalhos para auxiliar na detecção de anomalias [73].

- **Análise de Componentes Principais (PCA)**: a ideia principal da PCA é reduzir a dimensionalidade de um conjunto de dados composto por um grande número de variáveis correlacionadas. O processo é realizado através da transformação em um novo conjunto de variáveis, os componentes principais, que são não correlacionados e organizados de modo que os primeiros componentes retenham a maior parte da variação presente em todas as variáveis originais, ou seja, os dados de entrada podem ser representados por conjunto reduzido de dimensões sem muita perda de informação [74].

- **Holt-Winters:** é um método estatístico de previsão aplicado a séries temporais caracterizado pela presença de tendência linear e periodicidade, que é fundamentado no método da Média Móvel Exponencialmente Ponderada (EWMA). O emprego desta técnica gera uma previsão do comportamento normal da rede, com a aplicação de intervalos de confiança, a assinatura gerada pode ser comparada com o tráfego real para detecção de comportamentos anômalos do tráfego [75].

4.2.2.3 Clusterização e Metaheurísticas

Os algoritmos de clusterização são técnicas de aprendizado de máquina não supervisionado aplicadas para agrupar as instâncias de dados e extrair informações dos conjuntos. O K-means é um algoritmo clássico para clusterização e metaheurísticas têm sido empregadas em conjunto para otimizar o processo de clusterização. Alguns exemplos dessas aplicações para a detecção de anomalias são:

- **Algoritmo Genético**: no trabalho de Hernandes *et al.* [76] foi aplicado algoritmo genético para otimizar a clusterização de fluxos de redes tradicionais para gerar uma assinatura do comportamento normal da rede. Posteriormente, a assinatura foi aplicada para auxiliar no processo de detecção de comportamentos anômalos da rede.

- **Otimização por Exame de Partículas**: Lima *et al.* [77] aplicaram o *PSO* em combinação com o K-means para gerar um *baseline* para gerenciamento de *backbone* aplicado a dados *SNMP* da rede. O objetivo da aplicação do *PSO* foi melhorar a qualidade das soluções da clusterização e no cálculo dos centróides dos clusters.

- **Otimização da Colônia de Formigas**: Assis *et al.* [78] visam otimizar a eficiência da clusterização minimizando distância Euclidiana entre os fluxos de dados, buscando soluções de agrupamentos dos dados para extração de padrões, comportamentos e características do tráfego da rede para aplicar na detecção de anomalias.

- **Algoritmo de Vaga-lume**: Salmen *et al.* [79] aplicaram a metaheurística para otimizar a organização dos dados em *cluster* e buscar o centróide que melhor representa o conjunto de dados do tráfego. Os centróides encontrados representam o comportamento normal do tráfego e posteriormente é comparado com o tráfego real para detecção de ataques.

5 SISTEMA DE DETECÇÃO E MITIGAÇÃO DE ANOMALIAS

O sistema de detecção e mitigação de anomalias proposto neste trabalho é dividido em três etapas:

1. A primeira etapa é caracterização do tráfego, extraindo a assinatura do segmento de rede pela organização dos dados de fluxos em *clusters*;
2. A segunda é a detecção de anomalias utilizando a desigualdade de Chebyshev;
3. E por fim, o módulo de mitigação que é ativado quando na etapa anterior uma anomalia é detectada.

A Figura 2 ilustra o diagrama esquemático de operação do sistema de detecção e mitigação de anomalias proposto neste trabalho. O sistema foi desenvolvido no plano de aplicações. A primeira etapa é a caracterização do tráfego, no qual os atributos de fluxos são pré-processados e as assinaturas do comportamento normal da rede são geradas. A etapa seguinte é o módulo de detecção de anomalias, que determina dinamicamente limiares de normalidade do tráfego e caso o comportamento seja diferente do esperado, um evento anômalo é detectado. Quando ocorre uma anomalia, os IPs e portas contidos no intervalo de análise são considerados suspeitos. A etapa 3 é responsável pela aplicação das políticas de mitigação e recebe como entrada os fluxos suspeitos determinados na etapa anterior. Neste conjunto de fluxos, o módulo mitigação aplica a contramedida mais adequada para minimizar os efeitos de um ataque.

5.1 Clusterização de Dados

A clusterização é uma técnica de aprendizado de máquina não supervisionado para extração de padrões em agrupamentos ou *clusters*. Esta técnica é empregada principalmente na mineração de dados e seu objetivo é encontrar e quantificar em subconjuntos os elementos semelhantes que constituem uma base dados não rotulados [80]. Um agrupamento pode ser descrito considerando a homogeneidade interna e a separação externa, isto é, os elementos que pertencem ao mesmo agrupamento possuem padrões semelhantes entre si, e por outro lado, apresentam padrões diferentes dos demais agrupamentos.

Os agrupamentos gerados no processo de clusterização podem ser de dois tipos: exclusivo e não-exclusivo. Um agrupamento exclusivo gera subconjuntos únicos, isto é, partições de um conjunto de dados em que cada elemento pertença exclusivamente a um *cluster*. Em contrapartida, o agrupamento não-exclusivo permite que uma amostra pode

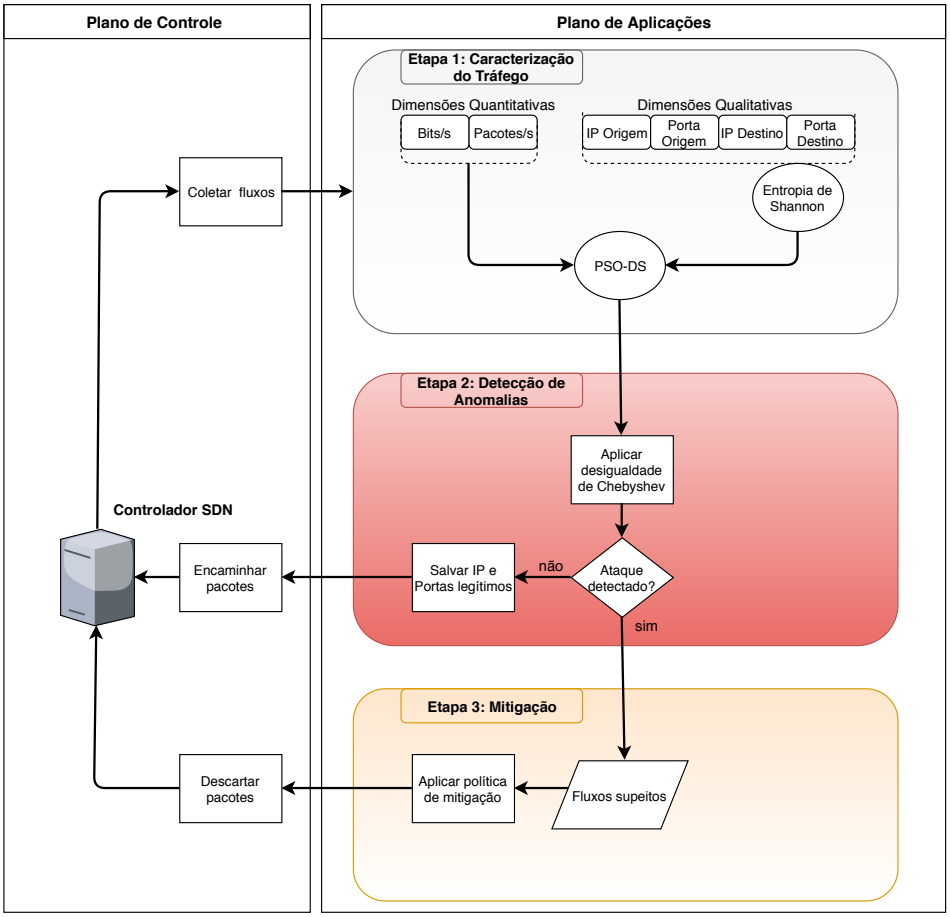


Figura 2 – Arquitetura do sistema proposto.

ser associada a mais de um subconjunto [81]. Um problema de clusterização com agrupamento exclusivo consiste em um conjunto finito de dados $\mathbf{B}$ composto por N amostras. Cada amostra é representada por um vetor n-dimensional $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, em que x_i corresponde cada um dos atributos (dimensões ou variáveis). O processo de clusterização resulta na separação $\mathbf{B}$ em K-*clusters* e cada partição é definida como $C = \{C_1, C_2, \ldots, C_K\}$ $(K \leq N)$ e a seguintes restrições devem ser satisfeitas [82]:

$$C_i \neq \emptyset, \quad i = 1, \ldots, K \tag{5.1}$$

$$\bigcup_{i=1}^{K} C_i = \mathbf{B} \tag{5.2}$$

$$C_i \bigcap C_j = \emptyset, \quad i, j = 1, \ldots, K \quad e \quad i \neq j \tag{5.3}$$

A condição representada na Equação (5.1) significa que todos os *clusters* são constituídos por pelo menos uma amostra, ou seja, não existe *cluster* vazio. A segunda condição designada na Equação (5.2) define que todas as amostras que pertencem ao conjunto de dados é associada a algum *cluster*. E a última condição presente na Equação (5.3) ressalta

a característica do agrupamento exclusivo em que uma amostra pertença exclusivamente a um *cluster* (grupos disjuntos).

A aplicação de técnicas de clusterização no processo de caracterização do tráfego de rede possui vantagens em relação aos demais modelos de aprendizado de máquina. A principal vantagem da clusterização é capacidade de extrair padrões do tráfego sem conhecimento prévio do conjunto de dados, ou seja, as amostra não possuem rótulos (ataque ou normal). Esta característica é fundamental na detecção de anomalias, visto que, constantemente novas técnicas de ataques são propostas por agentes maliciosos e são aplicadas de diferentes maneiras contra os sistemas de rede, portanto, há uma necessidade de utilização de técnicas capazes detectar comportamento desconhecidos [83].

Em contrapartida, encontrar a solução ótima para um agrupamento é um problema NP-completo, o número de possíveis combinações para particionar um conjunto de dados em *c clusters* cresce exponencialmente [84]. Deste modo, a aplicabilidade de abordagens capazes de explorar um espaço de busca amplo e tornar o processo de clusterização efetivamente melhor pode ser fundamental. As metaheurísticas inspiradas na natureza empregam uma população para explorar o espaço de busca e, assim, a sua aplicação em conjunto com os algoritmos de clusterização vem sendo amplamente explorados para garantir maior probabilidade de obter partições de *cluster* ideais [85].

A methaheurístca *Particle Swarm Intelligence* [86], inspirada no comportamento social de uma revoada de pássaros, é amplamente utilizada em diversos trabalhos em que são abordados problemas de otimização devido à sua forma simples de implementação e taxa de convergência mais rápida em relação as demais metaheurísticas bio-inspiradas [87][88]. A característica intrínseca de auto-organização do *PSO* permite acelerar o processo de clusterização, encontrar soluções de boa qualidade e evitar ótimos locais.

5.2 Particle Swarm Optimization

Diversos estudos foram desenvolvidos aplicando *Swarm Intelligence* (SI) para propor métodos com o objetivo de resolver problemas complexos de otimização que são difíceis de resolver utilizando algoritmos clássicos de otimização. *Swarm Intelligence* são metaheurísticas inspiradas na natureza baseadas no comportamento coletivo da inteligência de agentes naturais relacionadas às formas de interação e organização no ambiente. Alguns exemplos dessas metaheurísticas bio-inspiradas são *Ant Colonization Optimization* (*ACO*) [89], *Particle Swarm Optimization* (*PSO*) [86], *Bat-Inspired Algorithm* (*BA*) [90] e *Firefly Algorithm* (*FA*) [91].

A metaheurística *Particle Swarm Optimization* foi introduzida por Eberhart e Kennedy em 1995 [86]. Inspirada pelo comportamento social de uma revoada de pássaros ou de um cardume de peixes, introduzindo uma nova abordagem para a otimização de

funções. No *PSO* uma revoada de pássaro é inicializada aleatoriamente no espaço de busca, cada pássaro é uma possível solução do problema e sua qualidade é avaliada pelo função de aptidão, também conhecida como função objetivo. Conforme ilustra a Figura 3, em que cada pássaro é referido como uma partícula e o conjunto de partículas é chamado enxame.

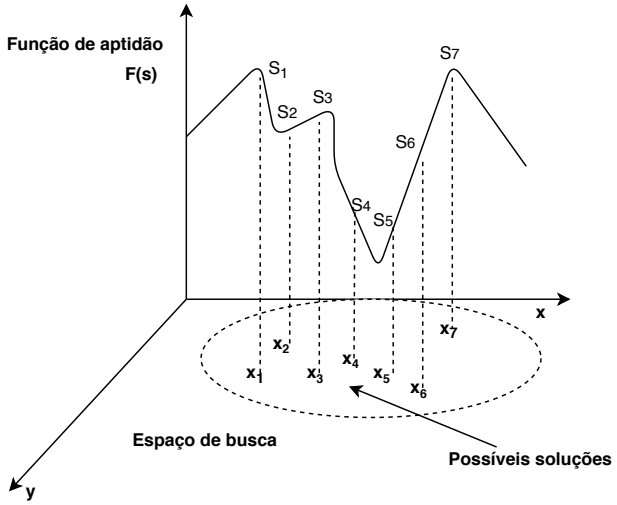


Figura 3 – Ilustração do processo de otimização do Particle Swarm Optimization.

O *PSO* busca otimizar uma função específica, conhecida como função objetivo. A cada iteração a função objetivo é utilizada para avaliar a eficiência das soluções geradas, ou seja, guia o movimento das partículas em direção à solução do problema. De outro modo, a função objetivo avalia o quão próximo as partículas estão da solução, isto é, avalia a performance das soluções, em que cada partícula possui uma velocidade de atualização que orienta seu movimento ao longo do espaço de busca. Considere uma população de partículas P, em que v_p e p_p representam a velocidade e a posição da partícula p. O movimento de cada partícula é realizado atualizando a sua velocidade de deslocamento e a sua posição através da Equação (5.4) e da Equação (5.5), respectivamente.

$$v_{p+1} = \overbrace{wv_p}^{Inércia} + \overbrace{c_1 r_1 \left(p_{best_p} - x_p\right)}^{Termo\ \ Cognitivo} + \overbrace{c_2 r_2 \left(g_{best} - x_p\right)}^{Termo\ \ Social} \tag{5.4}$$

$$p_{p+1} = p_p + v_p, \tag{5.5}$$

em que w é o coeficiente de inércia c_1 e c_2 são as constantes de aceleração, r_1 e r_2 são números aleatórios definidos no intervalo $[0, 1]$, p_{best_p} é a melhor posição que partícula p já ocupou até aquela iteração e g_{best} representa a solução global daquele momento do enxame. De acordo com [86], c_1 e c_2 podem ser usados como 2.05 e w igual a 0.5.

A Equação (5.4) da atualização da velocidade é composta por três termos: Inércia, cognitivo e social. O termo de inércia faz com que a partícula siga a sua própria direção

com a mesma velocidade. Por meio do termo cognitivo, a partícula consegue aprender com a própria experiência e força a partícula voltar a uma região anterior do espaço de busca que seja melhor que a atual. E por fim, o termo social faz com que a partícula siga em direção a regiões que apresentam boas soluções no espaço de busca e permite a troca de informações com o melhor indivíduo do enxame. A interpretação geométrica do movimento de uma partícula é ilustrado na Figura 4 e demonstra como cada um dos termos da equação da velocidade influência no deslocamento da partícula.

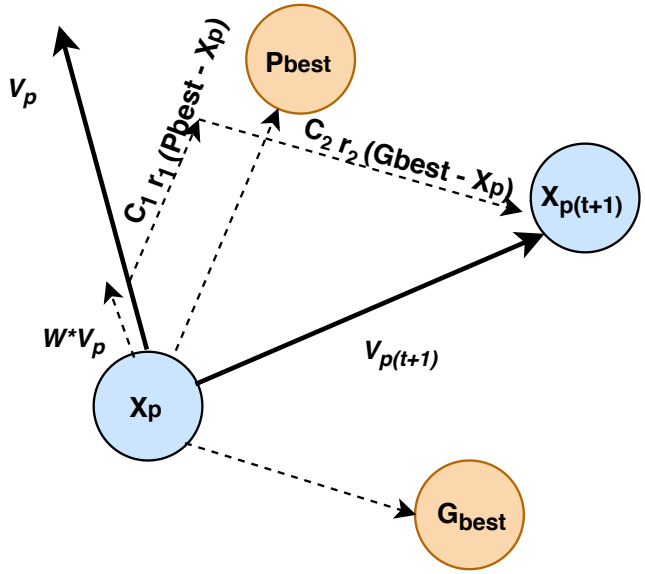


Figura 4 – Movimentação de uma partícula em um espaço de busca.

As partículas p_{best_p} e g_{best} são avaliadas a cada iteração utilizando a função objetivo e elas só serão atualizadas caso a solução atual apresentar um resultado melhor que os valores já encontrados até aquela iteração [92]. As partículas p_{best_p} e g_{best} são atualizadas seguindo as seguintes comparações, através da função objetivo f, definidas nas equações 5.6 e 5.7, respectivamente:

$$p_{best_p} = p'_{best_p} \qquad se \qquad f(p'_{best_p}) < f(p_{best_p}) \tag{5.6}$$

$$g_{best} = g'_{best} \qquad se \qquad f(g'_{best}) < f(g_{best}) \tag{5.7}$$

5.3 Módulo de Caracterização

O módulo de caracterização tem como finalidade gerar a assinatura do comportamento normal da rede e seu emprego é essencial para o gerenciamento e segurança da rede. A caracterização torna mais confiável e segura as decisões de gerência em relação a possíveis problemas que venham ocorrer na rede. Obter uma previsão próxima ao comportamento real é um passo importante para a detecção de tráfego anômalo, pois a assinatura

gerada delimita os limites de normalidade de uma amostra do tráfego em um determinado instante do segmento de rede observado [93].

As caracterizações das assinaturas são geradas a partir de dados de fluxos IP's que são coletados dos *switches* da rede *SDN* pelo controlador utilizando o protocolo *OpenFlow*. Na especificação do *OpenFlow* é definido duas abordagens distintas para coleta de estatísticas de fluxos dos *switches*: *push-based* e *pull-based*. Na abordagem *push-based* o controlador recebe passivamente dos *switches* relatórios de fluxos ativos e suas estatísticas. Por outro lado, no método *pull-based*, o controlador periodicamente envia mensagens do tipo *Read_State* para os *switches* para coleta de estatística da entrada da tabela de fluxos. Como este tipo de abordagem não requer nenhuma configuração adicional, ele tem sido utilizado de maneira ampla em diversas aplicações *SDN* [94][95][96]. Portanto, neste trabalho o método utilizado para extração de estatística utilizado é o *pull-based*.

Quando o *switch* recebe a solicitação do controlador ele envia a reposta utilizando uma mensagem *OpenFlow* do tipo *Multipart* que contém um conjunto de registros. E cada registro é formado por uma estrutura chamada *ofp_flow_stats*. Conforme detalhada na Tabela 3, esta estrutura é composta por alguns atributos [97].

Um dos aspectos importantes para implementação do módulo de caracterização do tráfego é o tempo que as informações são coletadas do controlador. De acordo com Giotis *et al.* [98], para ambientes de rede *SDN* o tempo preciso de coleta dos dados pode ser realizado a cada 30 segundos. Porém, o tempo de coleta pode ser determinado por alguns aspectos do ambiente de rede, com o atraso da comunicação, a quantidade de tráfego e a disposição da topologia da rede. Desse modo, o tempo de coleta dos dados fica a critério do administrador de rede e pode ser definido conforme a sua necessidade. No sistema proposto neste trabalho, os tempos de coleta utilizados foram 1, 5, 30 e 60 segundos.

Tabela 3 – Registro coletado do switch contendo os atributos de estatística de um fluxo.

Atributo	Descrição
uint16_t length	Comprimento da entrada do registro.
uint8_t table_id	ID da tabela de fluxo de origem.
uint32_t duration_sec	Tempo que o fluxo está vivo em segundos.
uint32_t duration_nsec	Tempo que o fluxo está vivo nanossegundo além do duration_sec.
uint16_t priority	Prioridade da entrada
uint16_t idle_timeout	Número de segundos inativos antes da expiração.
uint16_t hard_timeout	Número de segundos antes da expiração.
uint64_t packet_count	Número de pacotes no fluxo.
uint64_t byte_count	Número de bytes no fluxo.
struct ofp_match match	Informações sobre o fluxo: Endereço IP de origem e destino; Porta de origem e destino; E protocolo de transporte.

Dentre os atributos que são coletados pelo controlador foram selecionados os se-

guintes atributos: bytes/s, pacotes/s, endereço IP de origem, endereço IP de destino, porta de origem e porta de destino. Esses atributos de fluxos foram exaustivamente analisados e empregados na caracterização do tráfego de redes de altas velocidades e apresentaram bons resultados para descrever e compreender melhor o comportamento da rede [99][100][101]. As dimensões bytes e pacotes são atributos quantitativos, ou seja, atributos de volume que são capazes de fornecer informações em relação à quantidade de informações que estão trafegando ao longo da rede. Os demais são atributos nominais e fornecem informações qualitativas, isto é, estes atributos permitem compreender quais os dispositivos que estão se comunicando e quais serviços estão sendo acessados por eles. A utilização desses atributos é fundamental para identificar possíveis atacantes e é indispensável na utilização do módulo de mitigação para minimizar os danos causados por um ataque.

Os atributos IP e porta são dados nominais e para realizar um análise quantitativa é necessária a transformação desses atributos por meio do cálculo da entropia. Portanto, neste trabalho foi empregada a Entropia de Shannon [102] que permite extrair informações da concentração e dispersão desses atributos de fluxo. Para este propósito, dado um conjunto de amostras do descritor $X = \{x_1, x_2, \ldots, x_n\}$ em que x_i representa o número de ocorrência da amostra i no intervalo de tempo. A entropia H para X é definida na Equação (5.8) como:

$$H(X) = -\sum_{i=1}^{N} \left(\frac{x_i}{S}\right) log_2 \left(\frac{x_i}{S}\right), \tag{5.8}$$

em que $S = \sum_{i=1}^{N} x_i$ é o somatório de todas as ocorrências presentes no histograma.

Por meio da utilização da entropia para caracterização do tráfego é possível identificar ataques. Por exemplo, em um ataque de DDoS ocorre uma concentração da entropia de endereço IP e porta de destino da vítima; a entropia da porta de origem fica dispersa pelo fato de múltiplos atacantes utilizarem portas aleatórias de origem [103].

Posteriormente, ao garantir que todos atributos de fluxos coletados estão representados de maneira quantitativa, inicia-se o processo de geração das assinaturas do tráfego. O perfil normal da rede é gerado por meio do agrupamento dos dados de fluxos em *cluster*. A clusterização permite agrupar os dados semelhantes por meio de uma medida de afinidade bem definida em um *clusters*, minimizando a variância dentro deste grupo e maximizando em relação aos demais agrupamentos e também é uma ferramenta poderosa para extração de padrões sem conhecimento prévio das amostras.

5.3.1 Particle Swarm Optimization for Digital Signature

Na literatura, diversos trabalhos foram desenvolvidos utilizando o *PSO* para otimização no processo de clusterização [104][105][106]. O *PSO* foi desenvolvido para ser um método simples de otimização que pode ser implementado em poucas linhas de código. O

método requer apenas operações matemáticas primitivas e, em termos de custo computacional, implica em um algoritmo de baixo custo capaz de encontrar regiões ótimas em espaços multidimensionais de busca [107]. Na aplicação de diversos problemas o *PSO* tem a capacidade rápida de convergência, sendo um método otimizador eficaz [108]. Baseada na abordagem empregada neste trabalho que utiliza a caracterização multidimensional do tráfego, a convergência rápida das soluções torna-se um fator primordial.

Com base nas características e vantagens da aplicação do PSO, foi desenvolvida uma técnica para caracterização do tráfego chamada *Particle Swarm Optimization for Digital Signature* (*PSO-DS*). Por meio da organização dos dados de fluxo da rede em *clusters*, o *PSO-DS* foi aplicado para minimizar a distância intra-*clusters*, isto é, a soma da distância entre os dados de fluxo e seu respectivo centróide. A Equação (5.9) descreve a distância intra-*clusters*.

$$F = \sum_{j=1}^{C} \sqrt{\sum_{a=1}^{J} (c_j - x_a)^2}, \tag{5.9}$$

no qual C representa o número de clusters e J representa o tamanho da janela deslizante de fluxos que serão clusterizados. Já a variável c_j indica o valor do centro do cluster j e x_a o valor do atributo a ser clusterizado. Cada ponto da assinatura é obtido fazendo a média entre os centróides dos C clusters obtidos após o processo de otimização do *PSO*. O Algoritmo 1 ilustra o processo para geração das assinaturas

Algoritmo 1 *PSO-DS* utilizado para caracterização do tráfego.

Entrada: Conjunto de informações extraídas dos fluxos da janela J
Saída: Um vetor contendo a assinatura do instante t_i

1: **para** para cada atributo de fluxo **faça**
2: Calcular o limite inferior da janela deslizante
3: Calcular o limite superior da janela deslizante
4: Gerar população para intervalo de tempo
5: **enquanto** número máximo de iterações ou a condição de parada não for satisfeita **faça**
6: Avaliar a função objetivo da partícula (5.9)
7: Atualizar o $pBest$ (5.6)
8: Atualizar $gBest$ (5.7)
9: Atualizar a velocidade da partícula (5.4)
10: Atualizar a posição da partícula (5.5)
$C_{ja} \leftarrow$ média dos k clusters da melhor solução
11: **retorna** C

O *PSO* é um algoritmo executado de forma iterativa, isto é, os critérios de paradas devem ser definidos para que o programa não execute indefinidamente. No *PSO-DS* foram adotados dois critérios de paradas. O primeiro critério de parada está relacionado com a

qualidade das soluções geradas, caso o g_{best} em três iterações consecutivas não altere o seu valor então a execução do processo é interrompida. O segundo critério diz respeito ao número máximo de iterações. Quando o número de 100 iterações é executado, o sistema é interrompido para que sua execução não ocorra indeterminadamente. Esses dois critérios de paradas adotados se mostraram suficientes para a produção de boas soluções para os DSNSFs gerados.

5.3.2 Parâmetros Utilizados pelo Módulo de Caracterização

Conforme mencionado nas seções anteriores, o módulo de caracterização do sistema aplica a metaheurística *PSO*. Como toda metaheurística possui parâmetros envolvidos para a geração das soluções, é necessária a calibragem desses parâmetros para garantir a qualidade das soluções. Com objetivo de garantir a eficiência do módulo de caracterização e obter um bom desempenho do *PSO* na realização dessa tarefa, os valores dos parâmetros utilizados foram definidos, conforme ilustra a Tabela 4. A escolha desses valores foi feita por meio de teste de exaustão, aplicando valores dentro de um limite aceitável para obtenção de resultados satisfatórios. O conjunto de dados utilizados para a escolha dos parâmetros é composto por 9 dias de emulação, totalizando 216 horas de análise do tráfego de rede. Os testes e os resultados aplicados são demonstrados nas subseções seguintes.

Tabela 4 – Parâmetros do módulo de caracterização.

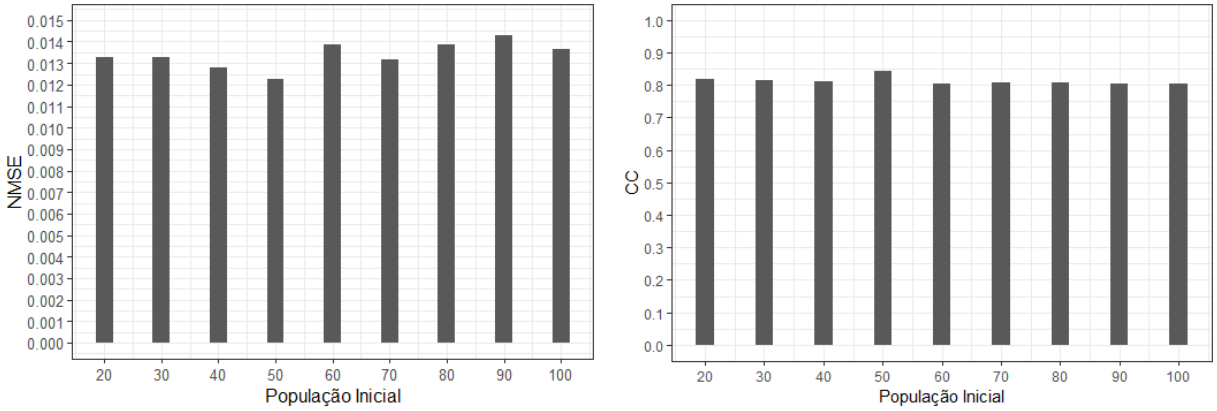
Parâmetro	**Valor/Método**
População Inicial	50 partículas
Tamanho da Janela deslizante	5 amostras anteriores
Função Objetivo	Distância Intra-*cluster*
Número de *cluster*	2 *clusters*
Critério de Parada	50 iterações ou 3 iterações consecutivas sem alterar o g_{best}

5.3.2.1 População Inicial

O primeiro passo a ser realizado de todo algoritmo de otimização baseado enxame é a escolha da população de indivíduos, ou seja, as possíveis soluções para o problema. Sabendo que a população de indivíduos é gerada de modo aleatório, o tamanho da população deve ser escolhido com cautela. A escolha muito pequena de uma população pode afetar a capacidade do enxame de explorar diferentes regiões do espaço de solução. Por outro lado, uma população muito grande de indivíduos afeta diretamente o custo computacional do sistema, aumentando consideravelmente o tempo levado para a caracterização do tráfego.

Através de testes empíricos aplicados por Bratton e Kennedy [109], foi demonstrado que o número de partículas que compõem o enxame pode influenciar o desempenho

resultante dependendo do problema que está sendo otimizado. Os autores definem que o número de indivíduos compreende entre 20 e 100 partículas. Considerando o custo computacional e a eficácia do sistema, foram realizados testes para os valores do número de partículas. As métricas aplicadas foram o erro quadrático médio (*Normalized Mean Square Error – NMSE*) e o Coeficiente de Correlação (*Correlation Coefficient – CC*) para o *DSNSF* gerado em um dia para cada um dos valores entre 20 e 100 partículas. A primeira métrica calcula a diferença absoluta entre o *DSNSF* gerado e o tráfego real. A segunda métrica mensura a relação entre as tendências demonstrada entre o *DSNSF* e o movimento real do tráfego. Os resultados dos valores das métricas avaliadas *NMSE* e *CC* são ilustrados na Figura 5. Na Figura 5a são apresentados os resultados da avaliação do *NMSE* e o melhor resultado obtido foi de uma população inicial de 50 partículas, alcançando um valor de *NMSE* de aproximadamente 0.012. Os resultados da avaliação do *CC* são apresentados na Figura 5b. O melhor resultado também foi de uma população inicial de 50 partículas e o resultado foi de 0.85. Portanto, o número da população inicial escolhido como parâmetro foi igual a 50 partículas.



(a) Avaliação de NMSE.

(b) Avaliação de CC.

Figura 5 – Avaliação quantidade da população inicial aplicando o NMSE e o CC.

5.3.2.2 Tamanho da janela deslizante

Nesta seção é avaliado o tamanho da janela deslizante J utilizada pelo módulo de caracterização para a geração das assinaturas do comportamento do tráfego da rede. Para a geração do *DSNSF* são utilizados os dados das últimas coletas anteriores do tráfego real. Para verificar a convergência de J, foram feitas análises de NMSE e CC.

Os valores avaliados para J compreendem entre 2 e 30 coletas anteriores do tráfego real. O valores encontrados para o NMSE e CC encontram-se na Tabela 5. De acordo com os testes aplicados, pode-se observar que a melhor janela deslizante foi de $J = 5$ coletas anteriores.

Tabela 5 – NMSE e CC para o tamanho da janela deslizante.

Janela Deslizante	**NMSE**	**CC**
2	0.014	0.713
3	0.015	0.733
5	0.013	0.818
10	0.018	0.740
15	0.020	0.714
20	0.021	0.699
25	0.020	0.692
30	0.022	0.684

5.3.2.3 Validação do número de Clusters

Ao aplicar um processo de clusterização, é relevante avaliar o número de *clusters* apropriado para a manipulação do conjunto de dados. Dessa forma, a técnica Silhouette [110] foi empregada para validar o número de clusters utilizados pelo *PSO-DS*. A aplicação desta técnica fornece uma representação gráfica da disposição dos elementos dentro de cada cluster. A representação gráfica da Silhouette tem utilidade quando a medida de proximidade está em uma escala (como no caso da Distância Intra-*cluster*) e quando se está buscando clusters compactos e claramente separados.

O retorno da função compreende entre -1 $\leq f(s) \leq$ 1. Essa função possui três interpretações para os resultados. Quando $f(s)$ está próximo de -1 significa que o objeto i foi classificado de maneira errada, ou seja, esse elemento deveria ser atribuído a outro cluster. Quando valor de $f(s)$ tende a zero é um caso intermediário, isso significa que o elemento i poderia ser atribuído a mais de um cluster. O melhor caso é quando $f(s)$ está próximo de 1, isto é, indica a existência de uma alta semelhança entre o elemento i e os demais elementos pertencentes ao cluster. Os gráficos presentes na Figura 6 apresentam os valores de C utilizados para validação do número de clusters.

A partir da análise de C valores utilizados, nota-se que o melhor valor encontrado ocorre quando foram aplicados dois clusters. Conforme os gráficos apresentados na Figura 6, a função não obteve nenhum retorno zero ou negativo, ou seja, nenhum elemento foi atribuído ao cluster de maneira errada.

5.4 Módulo para detecção de Anomalias

O módulo para detecção de anomalias é baseado na desigualdade de Bienaymé-Chebyshev. A desigualdade de Bienaymé- Chebyshev é utilizada para encontrar comportamentos que diferem da assinatura gerada para as dimensões de dados quantitativos e qualitativos. A desigualdade de Bienaymé-Chebyshev determina um limiar da porcentagem de dados que existem dentro do número k de desvios padrão em relação a média. A

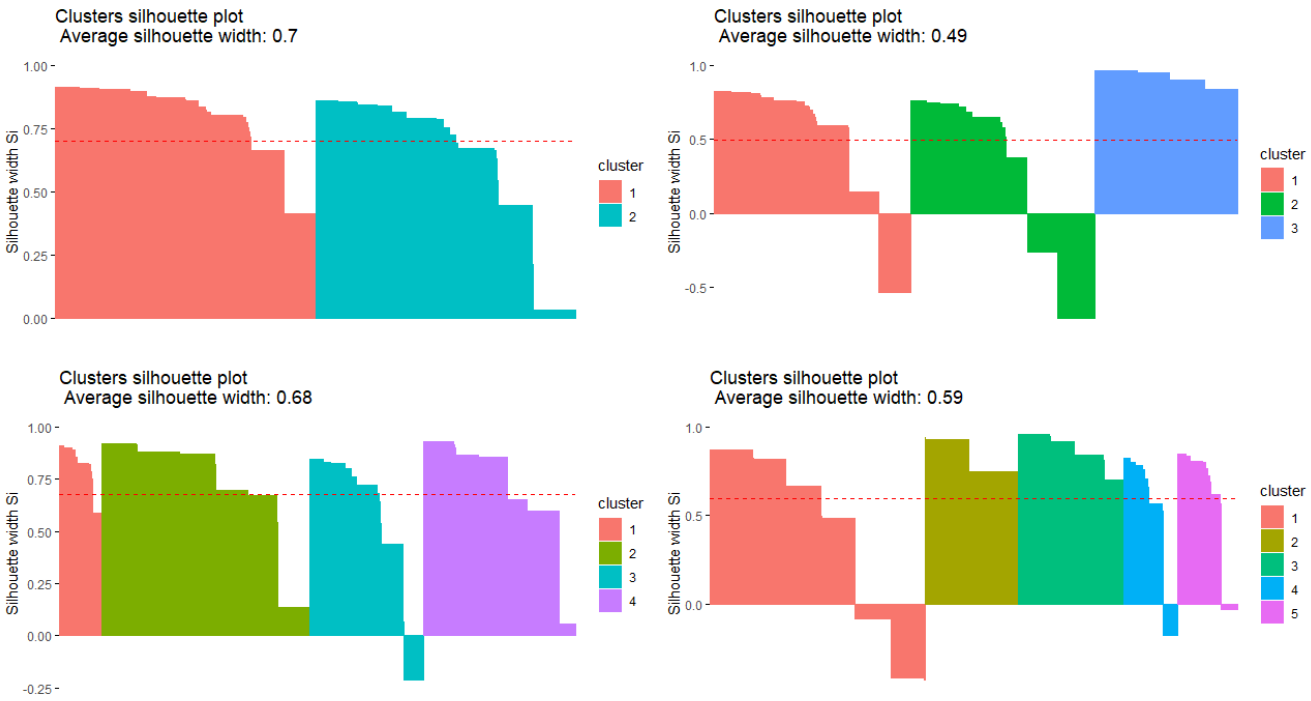


Figura 6 – Gráfico de Sillhouette para C valores de *clusters*.

desigualdade pode ser aplicada para detecção de *outliers* [111] quando não se conhece a distribuição dos dados.

A fórmula para a desigualdade de Bienaymé-Chebyshev é representada na Equação (5.10):

$$P(\mid X - \mu \mid \geq k\sigma) \leq \frac{1}{k^2}, \tag{5.10}$$

em que X é uma variável aleatória, μ é a média, $k > 0$ é o parâmetro de desvios e σ é o desvio padrão. Definindo o parâmetro $k = 4.47$ na Equação (5.10), a probabilidade resultante será igual a 0.05, que é o ponto de corte usual de significância estatística para verificar a discrepância de uma hipótese em relação aos dados observados [112].

Diversos trabalhos anteriores aplicaram a desigualdade de Bienaymé-Chebyshev para detecção de anomalias em redes tradicionais e obtiveram bons resultados. Kruegel e Vigna [113] propuseram um *IDS* para detectar ataques contra servidores e aplicações *web*. Villamarin e Brustoloni [114] aplicaram a desigualdade para detectar *botnet* em redes corporativas e de provedores de acesso. Já Sakib e Huang [115] apresentaram uma abordagem para detecção de *botnets* baseada em *HTTP*.

Para a formulação do módulo de detecção de anomalias foi realizada uma adaptação da desigualdade de Bienaymé-Chebyshev com objetivo de criar limiares superiores e inferiores dinamicamente a cada janela deslizante J para determinar o que é considerado comportamento normal do tráfego com base no DSNSF gerado. A utilização apenas da assinatura gerada como parâmetro para detecção de anomalias resultaria em uma alta

taxa de falsos alarmes, pois o tráfego da rede é dinâmico e qualquer flutuação em seu comportamento sem a utilização de um grau de tolerância seria identificado como anômalo. As equações (5.11) e (5.12) são utilizadas para determinar os limiares superiores e inferiores, respectivamente. A dimensão é detectada como sendo anômala caso o tráfego real seja maior que o limiar $UPPER$ ou menor que o limiar $LOWER$.

$$UPPER = DSNSF + k\sigma, \tag{5.11}$$

$$LOWER = DSNSF - k\sigma, \tag{5.12}$$

Após a detecção individual de cada um dos atributos de fluxos é necessário definir quando de fato ocorreu uma anomalia geral. De acordo com a literatura, cada tipo de anomalia afeta de modo diferente o comportamento dos atributos de fluxos do tráfego. Podem ser citado, como exemplo, um ataque de *DoS* no qual ocorre uma maior concentração dos atributos de IP de origem e porta de destino, diferente de uma anomalia de *Flash Crowd* em que ocorre uma dispersão dos atributos de IP de origem e porta de origem. De acordo com seguintes trabalhos [116][117][118], a Tabela 6 ilustra os tipos de anomalias e os atributos que são afetados.

Tabela 6 – Tipos de anomalias e atributos afetados.

Tipo de anomalias	**Atributos afetados**
DDoS	Aumento no volume de bits e pacotes, decréscimo das entropias de IP de origem e destino decréscimo da entropia de porta de destino
Portscan	Aumento no volume de pacotes decréscimo das entropias de IP de origem e destino aumento da entropia de porta de destino

Com base no comportamento dos atributos durante a ocorrência de uma anomalia, será considerado ao menos três atributos que sejam detectados como anormais para que o módulo de mitigação seja ativado, ou seja, se em um intervalo de tempo quaisquer três atributos do tráfego forem detectados como anômalos então de fato um alarme é disparado e naquele intervalo em análise será considerado a ocorrência de uma anomalia. O Algoritmo 2 ilustra o processo de detecção de anomalias.

5.5 Módulo de Mitigação

A detecção e a identificação de anomalias são etapas essenciais para garantir a operabilidade e os serviços disponíveis pelos sistemas de rede. Após a ocorrência de um evento anômalo, deve ser adotado um mecanismo para minimizar os efeitos provocados

Algoritmo 2 Desigualdade de Chebyshev para detecção de anomalias.

```
1: para Cada atributo de fluxo faça
2:     Calcular o limiar UPPER (5.11)
3:     Calcular o limiar LOWER (5.12)
4:     se (Dimensão é maior que UPPER ou Dimensão é menor que LOWER) então
5:         Dimensão é anômala
6:     senão
7:         Dimensão é normal
8: se (Tráfego tem comportamento de DDoS) então
9:     Tráfego classificado como DDoS
10: senão
11:     se (Tráfego tem comportamento de Portscan) então
12:         Tráfego classificado como Portscan
13:     senão
14:         Tráfego classificado como normal
```

por esse evento. O processo usual para diminuir os efeitos causados por ataques é por meio da mitigação, aplicando políticas autonômicas sem necessidade da interferência humana e tem como propósito garantir a resiliência e operabilidade da rede. Deste modo, o sistema proposto é composto por um módulo responsável por identificar os fluxos anômalos e políticas de mitigação são tomadas.

As políticas de mitigação são estruturadas utilizando o modelo **Evento-Condição-Ação** (ECA), que é considerado adequado para o gerenciamento dinâmico de políticas. Neste modelo, o Evento refere-se a uma anomalia específica e está associado a um conjunto de regras. Essas regras são descritas como um conjunto de **Condição** que correspondem ao contexto em que a anomalia ocorreu. E por fim, a **Ação** é a contramedida tomada em relação aos fluxos identificados como maliciosos [119].

O método principal utilizado nas aplicações para mitigação de ataques em ambientes *SDN* é modificar a entrada da tabela de fluxo do *switch* ou adicionar uma nova entrada na tabela do *switch*. Após a detecção de um ataque, algumas características devem ser identificadas, por exemplo, IP de origem, IP destino, número porta de origem, número porta de destino e o tipo de protocolo. Essas características auxiliam na identificação do atacante e são fundamentais para tomar as contramedidas para minimizar o dano de um ataque. Uma nova entrada na tabela de fluxos pode ser instalada com base em uma ou mais dessas características, sinalizando que os pacotes que pertencem ao fluxo são de um ataque. E as ações tomadas podem ser o descarte desses pacotes, bloqueio do tráfego anômalo e/ou um redirecionamento a um *honeypot* [98].

Fundamentado nos conceitos apresentados, o módulo de mitigação do sistema é composto por duas políticas para mitigar as anomalias detectadas. Após o módulo de detecção disparar um alarme, o módulo de mitigação entra em ação. O primeiro passo é identificar os fluxos suspeitos do intervalo de análise. A identificação dos fluxos suspeitos

é feita com base na análise dos endereços IP e portas que compõem o intervalo anômalo. São considerados fluxos suspeitos todos aqueles que se destinam ao endereço de IP que mais recebe fluxo.

Com a identificação dos fluxos suspeitos, caso o **Evento** disparado pelo módulo de detecção for um ataque de *DDoS*, será feito um descarte dos fluxos com base nos endereços IP de origem que aparecem com maior frequência nos fluxos suspeitos e que possuem simultaneamente a mesma porta de destino. Quando **Evento** disparado for um ataque de Portscan, o processo de identificação do atacante é feito por meio do endereço de IP de origem que apresenta a maior variedade de porta de destino. Esse IP é considerado atacante e todos os seus fluxos serão descartados. Os endereços IP permitem identificar os *host* que estão envolvidos no evento anômalo e as portas identificam quais os serviços afetados. O processo de mitigação é mostrado no Algoritmo 3.

Algoritmo 3 Processo de Mitigação.

Entrada: Fluxos suspeitos
Saída: Descarte de pacotes anômalos

```
1: se Ataque de DDoS então
2:     Identificar o endereço de IP de destino que mais recebeu fluxos
3:     Identificar nesses fluxos os endereços IP dos atacantes que possuem a mesma porta de destino
4:     se IPs e Portas estão na Safe List então
5:         Encaminhar os pacotes
6:     senão
7:         Descartar pacotes
8: se Ataque Portscan então
9:     Identificar o endereço de IP de destino que mais recebeu fluxos
10:    Identificar nesses fluxos o IP de origem que apresenta maior variedade de porta de destino
11:    se IPs e Portas estão na Safe List então
12:        Encaminhar os pacotes
13:    senão
14:        Descartar pacotes
```

Existem anomalias que não são causadas por agentes maliciosos, mas possuem o mesmo comportamento de um ataque. Por exemplo, uma anomalia de *Flashcrowd* possui as mesma características de um ataque de DDoS, no entanto, são usuários realizando requisições legítimas. No trabalho de Giotis *et al.* [98], os autores recomendam a implementação de um mecanismo que mantém uma lista de atributos de fluxos IP de usuários legítimos. Dessa forma, foi implementado um mecanismo chamado *Safe List* que mantém uma lista de endereços de IP e portas dos 5 minutos anteriores do intervalo de tempo analisado [15]. Portanto, essa lista é verificada antes de iniciar o processo de mitigação.

6 RESULTADOS

Neste capítulo serão descritos os testes e experimentos realizados. Os testes aplicados têm como objetivo verificar a eficiência do sistema pra caracterização do tráfego, para detecção e mitigação de anomalias em redes *SDN*. Para realizar esses testes, foram utilizados dados simulados gerados pelo emulador Mininet [120], uma ferramenta que permite a criação de redes virtuais realistas compostas por controladores, *hosts*, *links* e *switches* em uma única máquina virtual. O Mininet usa uma virtualização leve na criação de topologias personalizadas de código aberto e, é amplamente aplicado para pesquisa e desenvolvimento de soluções *SDN*. Para garantir que o cenário emulado seja o mais próximo possível de um ambiente *SDN* real, com altas taxas de tráfego passando pela rede, os experimentos utilizaram uma ferramenta chamada Scapy [121] para injetar tráfego na rede emulada.

6.1 Cenários

Para implementar o mecanismo de caracterização, detecção mitigação de anomalias, utilizamos dois controladores e duas topologias distintas para a geração dos cenários de rede *SDN*. A Figura 7 mostra as topologias utilizadas para a condução dos experimentos. Duas máquinas foram utilizadas para a emulação do tráfego, uma para o emulador Mininet e a outra para o Controlador SDN. A primeira máquina possui 8 GB de memória RAM, 4 núcleos de processamento com frequência de 2,21 GHz e sistema operacional Linux Ubuntu 16.04. A segunda máquina possui 4 GB de memória RAM, 4 núcleos de processamento com frequência 3.10 GHZ.

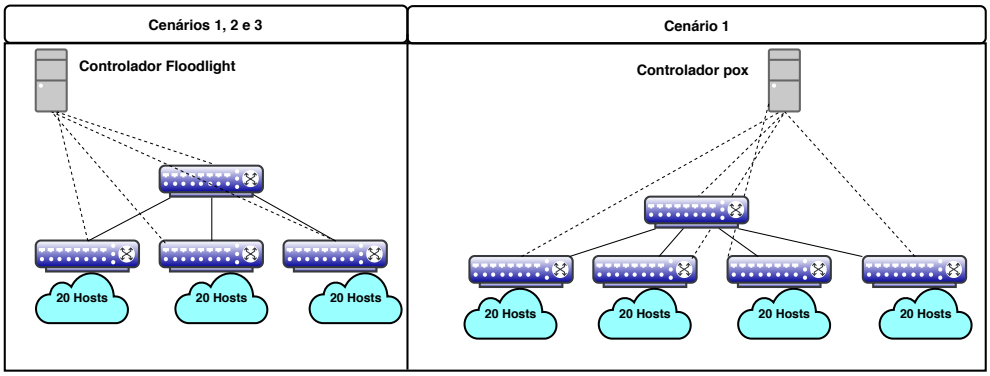


Figura 7 – Topologias emuladas para os cenários 1, 2, 3 e 4.

O primeiro controlador é o Floodlight, um controlador baseado em Java. O segundo controlador utilizado foi o POX, outra plataforma de controle *SDN* baseada em Python. A topologia emulada utilizando o controlador Floodlight é composta de quatro *switches*

em uma topologia em forma de árvore, em que um *switch* raiz conecta os outros três, cada um conectando vinte hosts diferentes. A segunda topologia emulada utilizando o controlador POX também é uma topologia em árvore, mas é formada por cinco *switches*, sendo um deles raiz que conecta os demais, formando quatro sub-redes com 20 *hosts* cada. Finalmente, os dados são coletados e gerenciados utilizando o protocolo *OpenFlow*. Os cenários avaliados para caracterização do tráfego são descritos na Tabela 7.

Tabela 7 – Cenário 1 para avaliação do módulo de caracterização.

Cenário	**Controlador**	**Topologia**	**Dias emulados**
1	Floodlight	- 1 switch raiz - 3 switches folha - 20 host por sub-rede - Total de Host 60	5
2	POX	- 1 switch raiz - 4 switches folha - 20 host por sub-rede - Total de Host 80	4

O cenário 2 foi utilizado para validar a eficiência do módulo de detecção de anomalias do sistema. Neste cenário foram incorporadas ao tráfego anomalias sintéticas geradas para emular o comportamento de ataques específicos. O tráfego malicioso injetado ao tráfego foi produzido utilizando o hping3 [122], um *assembler* e analisador de pacotes TPC/IP usado principalmente com uma ferramenta de segurança. As anomalias incorporadas ao tráfego em diferentes períodos do dia foram ataques de DDoS. Os ataques foram realizados em diferentes intensidade e duração. Um sumário dos ataques realizados encontra-se na Tabela 8.

Tabela 8 – Cenário 2 para avaliação do módulo de detecção de anomalias.

Cenário 2	**Ataque 1**	**Ataque 2**
Dia 1	Tipo: DDoS Atacantes: 4 IPs atacantes: 10.0.0.10-10.0.0.13 IP da vítima: 10.0.0.40 Horário: 11:00:00 - 12:01:00	Tipo: DDoS Atacantes: 11 IPs atacantes: 10.0.0.20-10.0.0.30 Ip da vítima:10.0.0.10 Horário: 11:00:00 - 13:05:00
Dia 2	Tipo: DDoS Atacantes:10 IPs atacantes: 10.0.0.21-10.0.0.30 IP da vítima: 10.0.0.50 Horário: 15:00:00 - 16:00:00	Tipo: DDoS Atacantes: 4 IPs atacantes: 10.0.0.11 , 10.0.0.14 Ip da vítima: 10.0.0.1 Horário: 15:00:00 - 16:00:00

No cenário 3 é apresentado um estudo de caso em que o intervalo de análise é reduzido para 5 segundos. Este cenário é composto de dois dias de observação com duração de 24 horas. Nestes dias emulados foram incorporados períodos de ataques de DDoS e Portscan com diferentes intensidades. As informações dos ataques são detalhadas na Tabela 9. Neste estudo de caso, além da avaliação do módulo de detecção, também é avaliado o módulo de mitigação.

O resumo dos cenários aplicados para avaliação do sistema é descrito na Tabela 10.

Tabela 9 – Cenário 3: Descrição dos ataques

	Ataque 1	**Ataque 2**	**Ataque 3**
Dia 1	Tipo: DDoS Atacantes: 5 IPs atacantes: 10.0.0.20 - 10.0.0.24 IP da vítima: 10.0.0.58 Horário: 07:00:00 - 07:59:40	Tipo: DDoS Atacantes: 10 IPs atacantes: 10.0.0.21 - 10.0.0.30 IP da vítima: 10.0.0.10 Horário: 12:00:00 - 13:02:20	Tipo: DDoS Atacantes: 15 IPs atacantes: 10.0.0.21 - 10.0.0.35 IP da vítima: 10.0.0.1 Horário: 15:00:00 - 15:59:05
Dia 2	Tipo: Portscan IP atacante: 10.0.0.60 IP da vítima: 10.0.0.30 Portas: 1 - 14961 Tempo entre os pacotes: 0.2 Horário: 06:00:00 - 06:59:40	Tipo: Portscan IP atacante: 10.0.0.41 IP da vítima: 10.0.0.35 Portas: 1 - 19999 Tempo entre os pacotes: 0.1 Horário: 10:00:00 - 10:46:30	Tipo: Portscan IP atacante: 10.0.0.35 IP da vítima: 10.0.0.1 Portas: 1 - 19126 Tempo entre os pacotes 0.15 Horário: 16:00:00 - 16:56:55

Tabela 10 – Cenários aplicados para avaliação do sistema.

Cenário	Avaliação	Dias avaliados
1	Caracterização	9
2	Detecção de DDoS em 60 segundos	2
3	Detecção e Mitigação de ataques de DDoS e Portscan em 5 segundos	2
4	Detecção e Mitigação de ataques de DDoS e Portscan em 1 segundo	1

6.2 Métricas de Avaliação

As métricas aplicadas para avaliação do módulo de caracterização do tráfego foram NMSE [123] e Coeficiente de Correlação. O NMSE é o erro médio quadrático produzido entre o DSNSF e o tráfego real. Essa métrica produz valores que vão de zero ao infinito, valores próximos a zero indicam uma boa caracterização do tráfego enquanto valores elevados indicam que a previsão realizada pelo módulo caracterização diverge do comportamento real do tráfego. Para o cálculo do NMSE entre duas séries temporais utiliza a Equação (6.1). A métrica CC é aplicada para avaliar a relação entre as tendências entre o DSNSF e movimento real do tráfego. Os resultados para o CC exprime o grau de correlação através de valores situados entre -1 e 1. Quando o CC se aproxima de 1, nota-se que o DSNSF e o movimento real possuem maior similaridade, ou seja, os DSNSF acompanha as tendências do tráfego. Para o cálculo do CC entre duas variáveis utiliza-se a Equação (6.2).

$$NMSE = \frac{1}{n}\frac{\sum_i (X_i - Y_i)^2}{\overline{XY}} \tag{6.1}$$

$$CC = \frac{\sum_i (X_i - \overline{X})(Y_i - \overline{X})}{\sqrt{\sum_i (X_i - \overline{X})^2 (Y_i - \overline{X})^2}} \tag{6.2}$$

Para medir os resultados de desempenho do sistema analisado para detecção de anomalias, foram aplicadas a seguintes métricas estatísticas clássicas [124]: Taxa de Verdadeiro Positivo e Negativo (TP e TN), Taxa de Falso Positivo e Negativo (FP e FN), Precisão, Acurácia de Classificação (CA) e Classificação de Erro (CE). Na Tabela 11

é apresentada uma matriz de confusão, ilustrando todas as possibilidades de saída do sistema na classificação do tráfego na avaliação das métricas.

Tabela 11 – Matriz de confusão.

	DSNSF		
Tráfego Real		Normal	Anomalia
	Normal	TN	FP
	Anomalia	FN	TP

A interpretação dos resultados presentes na matriz de confusão são:

- **Taxa de Verdadeiro-Positivo (TP):** a TP indica que o tráfego foi classificado como anômalo e de fato ocorreu uma anomalia;
- **Taxa de Verdadeiro-Negativo (TN):** a TN significa que o tráfego foi classificado como normal e não houve a ocorrência de um evento anômalo;
- **Taxa de Falso-Positivo (FP):** a FP indica que o tráfego foi identificado como anômalo, mas era normal;
- **Taxa de Falso-Negativo (FN):** a FN expressa um erro do sistema, o tráfego foi identificado como normal e na verdade ocorreu uma anomalia.

A acurácia representa a relação de proporcionalidade de amostras classificadas corretamente (TP e TN), ou seja, essa métrica avalia a capacidade do sistema de identificar e disparar alarmes quando um evento anômalo realmente ocorreu e a capacidade de não disparar um alarme quando no intervalo analisado do tráfego não houve a ocorrência de uma anomalia. A acurácia é expressa pela Equação (6.3):

$$CA = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.3}$$

A métrica CA pode levar a resultados tendenciosos em caso que a base de dados está desbalanceada, isto é, a base contém mais amostras normais que anômalas e o sistema classifica todas as amostras normais corretamente e classifica de modo errado as amostras anômalas. A métrica Precisão pode ser aplicada para resolver esse resultado tendencioso e enfatizar na classificação correta das amostras anômalas. A Precisão é expressa na Equação (6.4):

$$\text{Precisão} = \frac{TP}{(TP + FP)} \tag{6.4}$$

A classificação de erro mede a portagem de amostras que foram classificadas de maneira incorreta (FP e FN). A métrica toma como base quantidade de intervalos legítimos do tráfego identificados pelo sistema como anômalo e quantidade de intervalos analisados e identificados pelo sistema como normal, no entanto, houve a ocorrência de uma anomalia e o sistema não disparou um alarme. A classificação de erro do sistema é ilustrada na Equação (6.5):

$$CE = \frac{FP + FN}{TP + TN + FP + FN} \tag{6.5}$$

A *Receiver Operating Characteristics* (ROC) [125] pode ser a combinação das taxas TP e FP que possibilita uma análise visual da capacidade do sistema na detecção de comportamentos anômalos. No entanto, para melhor quantificar a eficiência entre vários classificadores, calcula-se a área sobre a curva (*Area Under the Curve* - AUC) ROC. Aquele que apresentar o escalar de maior valor é o que tem mais aptidão para classificar as amostras.

A eficiência do módulo de mitigação é feita por meio da aplicação do teste estatístico chamado de Teste de McNemar e também pela taxa de pacote descartados. O Teste de MacNemar é um teste não paramétrico que a sua aplicação é realizada por meio de amostras pareadas de dados nominais. Ele é aplicado para tabelas de contingência 2x2 com um traço dicotômico, ou seja, dois comportamentos (e.g. anômalo e normal) com objetivo de verificar se as frequências marginais são iguais ou não [126]. Na Tabela 12 é ilustrado um exemplo genérico de uma tabela de contingência 2x2 que apresenta o resultado de dois testes em uma amostra de n indivíduos.

Tabela 12 – Tabela de contingência 2x2 genérica.

	Teste 2 positivo	Teste 2 negativo	Total da linha
Teste 1 positivo	a	b	a+b
Teste 1 negativo	c	d	c+d
Total da coluna	a+c	b+d	n

A hipótese nula indica que as probabilidades para cada resultado são iguais, isto é, não houve alteração nas frequências marginais e $p_a + p_b = p_a + p_c$ e $p_c + p_d = p_b + p_d$, em que p_a, p_b, p_c, p_d indicam as probabilidades teóricas de ocorrências nas células com o rótulo correspondente. A hipótese nula e a hipótese alternativa são representadas, respectivamente, por:

$$H_0 : p_b = p_c$$
$$H_1 : p_b \neq p_c$$

A fórmula do teste de MacNemar se origina da fórmula do Qui-quadrado:

$$\chi^2 = \frac{(b-c)^2}{b+c} \tag{6.6}$$

χ^2 tem uma distribuição qui-quadrado com 1 grau de liberdade. Se o resultado de χ^2 é significativo, isto é, que $p_b \neq p_c$ o que significa que as frequências marginais são significativamente diferentes umas das outras e a hipótese nula é rejeitada.

6.3 Cenário 1: Avaliação do módulo de caracterização

Conforme descritos na Tabela 7, esses experimentos foram aplicados para avaliar a eficiência do módulo de caracterização do sistema na geração do *DSNSF*. Na topologia 1 o *PSO-DS* foi aplicado utilizando o controlador *SDN* Floodlight para gerar a assinatura de cinco dias de tráfego. Já na topologia 2 o *PSO-DS* foi aplicado utilizando o controlador *SDN* POX para a caracterização de quatro dias de tráfego. Os dados para essa análise foram coletados de ambos os controladores utilizando o protocolo *OpenFlow*. A coleta e geração do *DSNSF* foram realizadas no intervalo de 60 segundos. As métricas aplicadas para validar a eficiência do módulo de caracterização foram o NMSE e o Coeficiente de Correlação.

A Figura 8 representa o DSNSF gerado para os seis atributos para o dia de número 1. Na Figura 9 temos as assinaturas geradas para o dia de número 2. A caracterização do dia 3 é demonstrada na Figura 10. A caracterização realizada para o quarto dia é ilustrada na Figura 11. E por fim, na Figura 12 são os DSNSF gerados para o quinto dia. Os dados do tráfego real são exibidos na cor verde e os DSNSF gerados são representados pela cor azul para cada um dos atributos de fluxos analisados. Nas figuras apresentadas nota-se que há um ajustamento entre o comportamento real do tráfego e DNSNF, sendo possível certificar a eficiência do sistema ao realizar as assinaturas do tráfego da rede.

É possível observar que o DNSF conseguiu acompanhar o crescimento e do decaimento do tráfego em todos os atributos. Nos atributos bits e pacotes é possível notar a evidência deste acompanhamento, às 8h o tráfego começa a aumentar e às 21h ocorre um decaimento do tráfego e ambas as situações a assinatura prevista pelo sistema conseguiu se adequar a essas mudanças do tráfego.

A Figura 13 apresenta o gráfico que contém os valores obtidos da realização do teste de NMSE para as seis dimensões de fluxos analisadas e os respectivos dias avaliados. É possível notar uma taxa de erro muito baixa para todas as dimensões de fluxo. Bits foi a dimensão que apresentou a menor taxa de erro médio para os cinco dias analisados, com um valor médio de 0,0066. A dimensão que apresentou a maior taxa de erro médio foi a de entropia da porta de origem, com um taxa de erro médio de 0,014. Em média, para

Figura 8 – Tráfego real e DSNSF dia 1 Floodlight.

Figura 9 – Tráfego real e DSNSF dia 2 Floodlight.

todas as dimensões e dias analisados, a taxa de erro foi de 0,0097, uma taxa próximo de zero que indica uma boa caracterização do tráfego esperado.

O coeficiente de correlação foi aplicado para mensurar quanto o DSNSF está correlacionado com tráfego real, indicando se o DSNSF gerado é capaz de se ajustar as tendências do tráfego, ou seja, acompanhar o crescimento ou decaimento das dimensões de fluxos. A Figura 14 apresenta o gráfico contendo os valores obtidos da avaliação de CC para os atributos de fluxo para cada um dos cinco dias. A partir do gráfico apresentado

Figura 10 – Tráfego real e DSNSF dia 3 Floodlight.

Figura 11 – Tráfego real e DSNSF dia 4 Floodlight.

é possível notar que os atributos quantitativos (bits/s e pacotes/s) são aqueles que apresentaram os maiores valores de CC. O valor médio para os cinco dias para bits e pacotes foram 0,9451 e 0,9441, respectivamente. Esses valores foram próximos de 1, o que indica que o tráfego e o DSNSF gerados para esses atributos estão altamente correlacionados. Por outro lado, na análise para as entropias geradas verificou-se valores médio do CC ficaram entre 0,6853 e 0,8586. Para todos dias analisados, o valor médio encontrado para o CC entre o tráfego real e o DSNSF foi de 0,8520. Estes resultados apontam a capacidade

Figura 12 – Tráfego real e DSNSF dia 5 Floodlight.

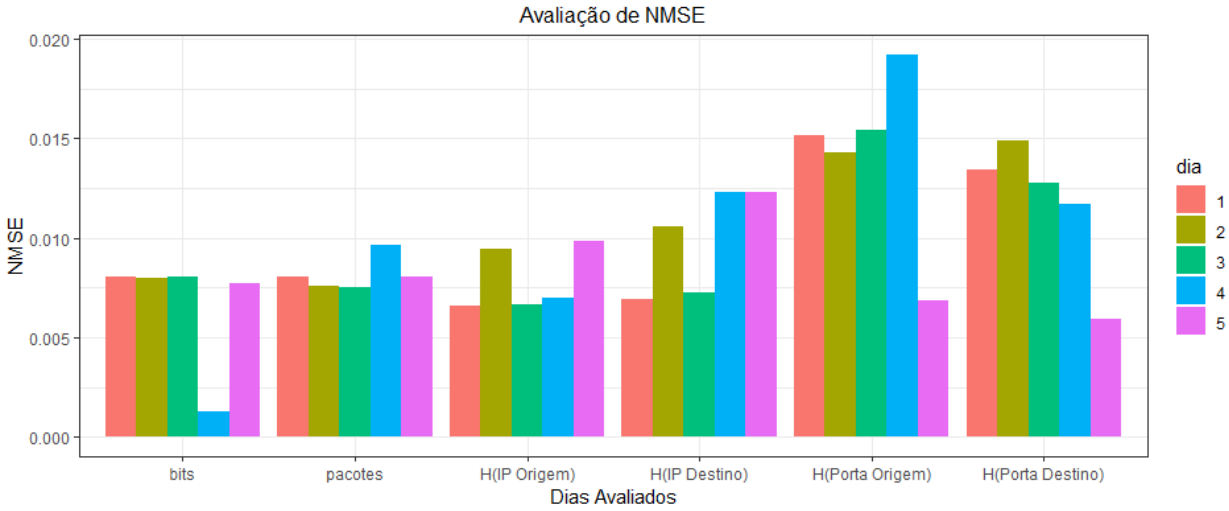


Figura 13 – Avaliação de NMSE para os 5 dias de tráfego.

do sistema de gerar uma assinatura capaz de acompanhar as tendências de aumento e diminuição do tráfego analisado.

As seguintes assinaturas utilizando o controlador POX são: os gráficos presentes na Figura 15 representam o DSNSF para o dia 1; O dia 2 e suas respectivas assinaturas geradas são representadas na Figura 16; Na Figura 17 contém as assinaturas para o terceiro dia de análise; e por fim, a Figura 18 compõe os gráficos referentes as assinaturas do quarto dia de análise. As assinaturas referentes ao cenário 2, em que foi utilizado o controlador POX e a coleta e caracterização do tráfego, foi realizada no intervalo de 30 segundos, totalizando 2880 intervalos em uma análise de 24 horas de caracterização, isto é, a assinatura para um dia completo. Foi realizada a caracterização dos respectivos

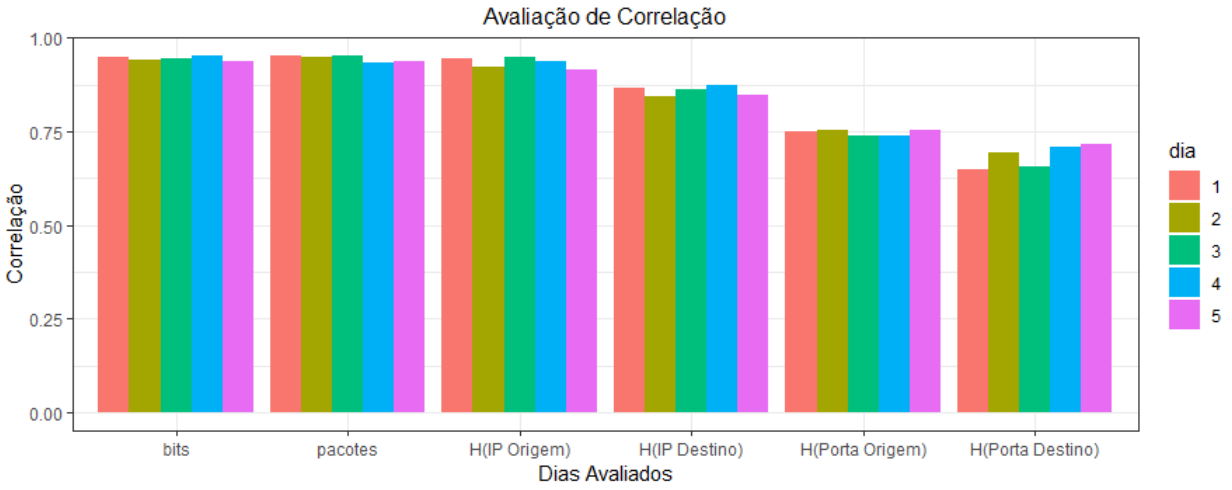


Figura 14 – Avaliação de CC para os 5 dias de tráfego.

atributos de fluxos para cada um dos quatro dias analisados.

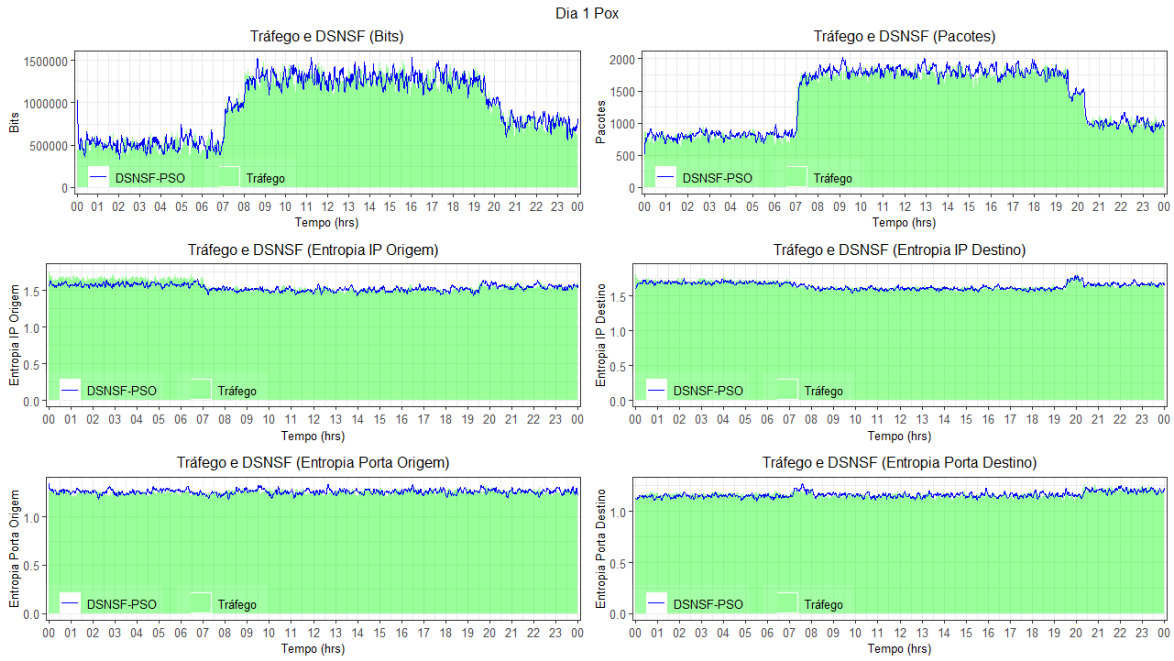


Figura 15 – Tráfego real e DSNSF dia 1 POX.

Na caracterização do tráfego no controlador POX é possível observar com uma evidência maior a capacidade do DSNSF de se adaptar as tendências do tráfego, pois ocorre um aumento significativo nas dimensões de bits e pacotes a partir das 7h e um decréscimo a partir das 20h. No momento de emular o funcionamento da rede, essas atenuações foram realizadas para simular o comportamento real de um dia de trabalho, pois no horário comercial ocorre uma demanda maior na utilização da rede pelos usuários. É possível notar essas atenuações do tráfego nos quatro dias e as assinaturas geradas conseguiram acompanhar as tendências, produzindo uma boa caracterização.

A Figura 19 apresenta o resultado do teste de NMSE para os quatro dias analisados

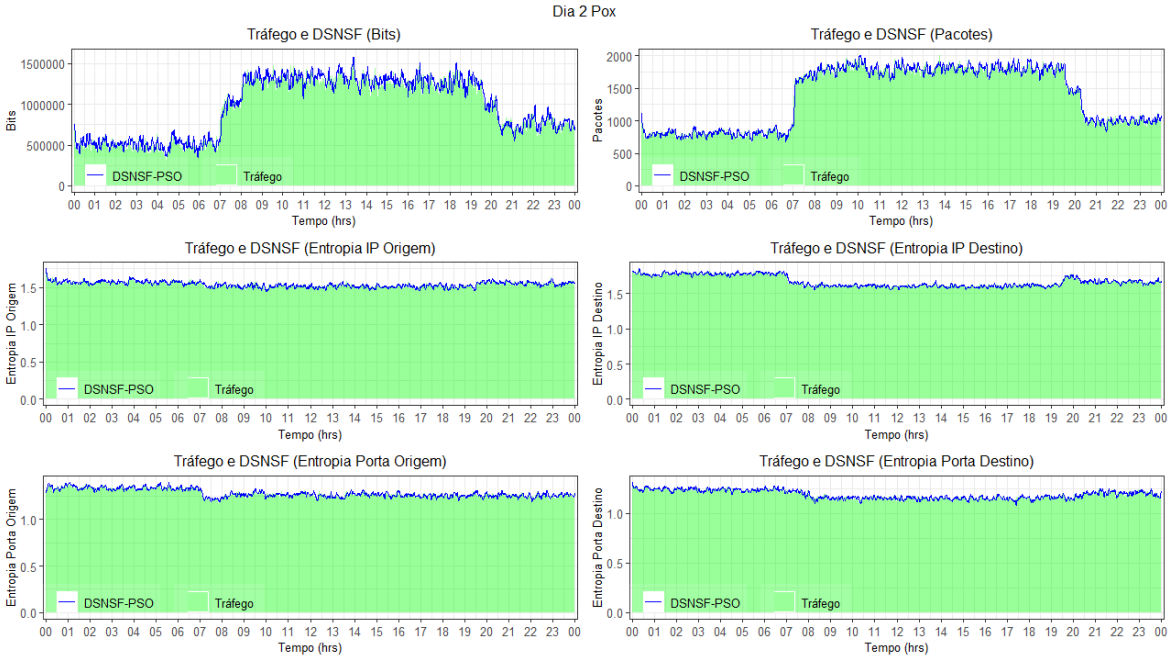


Figura 16 – Tráfego real e DSNSF dia 2 POX.

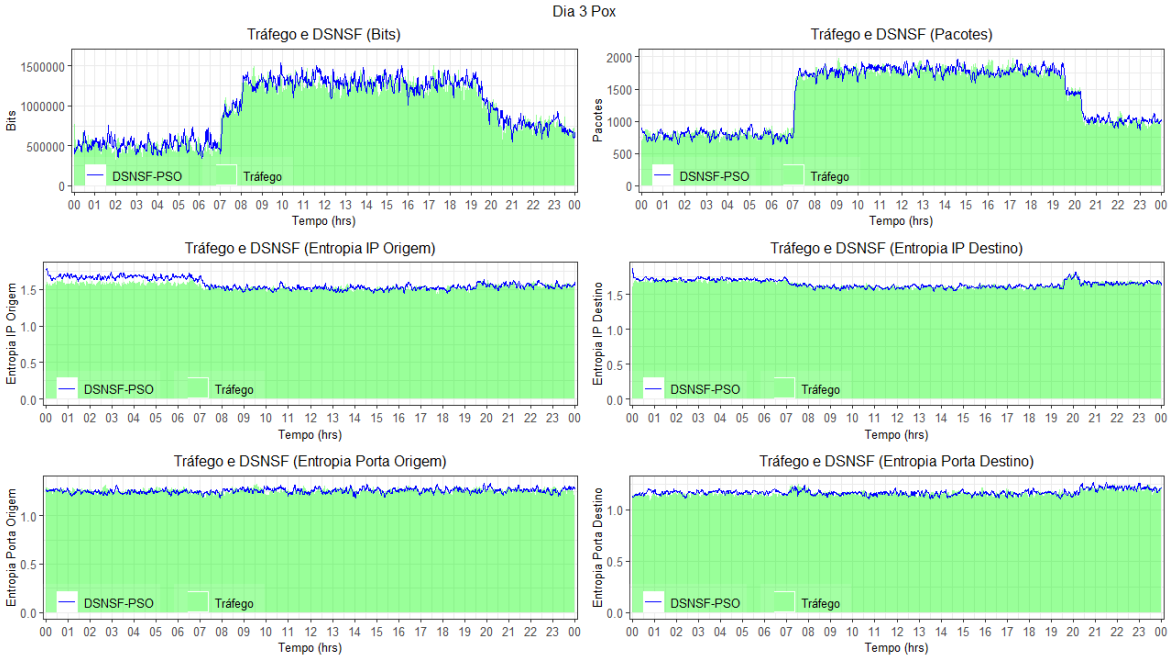


Figura 17 – Tráfego real e DSNSF dia 3 POX.

e foram separados pelos atributos de fluxo. Os valores resultantes do teste são próximos de zero, indicando que o tráfego previsto e o tráfego real estão de acordo. A dimensão de fluxo que apresentou o maior erro médio foi a de entropia da porta de origem, um valor de erro médio de 0,01383. Igualmente a caracterização utilizando o Floodlight, as dimensões que obtiveram o menor erro médio para o POX foram bits e pacotes, obtendo um valor de erro de 0,00374 e 0,00215, respectivamente. No geral, o erro médio para os quatro dia foi de 0,007125.

Figura 18 – Tráfego real e DSNSF dia 4 POX.

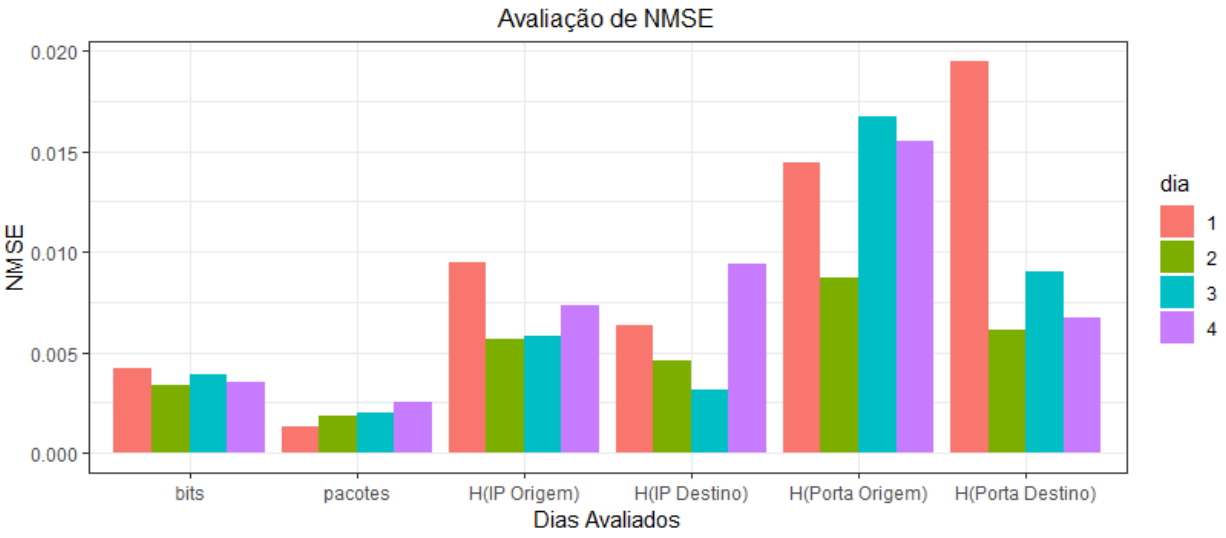


Figura 19 – Avaliação de NMSE para os 4 dias de tráfego do POX.

Os resultados referentes ao teste de CC utilizando o POX são apresentados na Figura 20. Os valores obtidos são muito próximos de 1, ou seja, indicando que a assinatura e o tráfego previsto possuem uma correlação linear positiva. O CC produzido para as dimensões de bits e pacotes foram 0,98, sendo as dimensões que obtiveram o melhor resultado. Assim como no teste aplicado para o NMSE, a dimensão de entropia de porta de origem obteve o menor CC, um valor de 0,7310. Porém, o CC médio de todas as dimensões foi de aproximadamente de 0,9.

Conforme os resultados apresentados para os dois cenários utilizando controladores e topologias distintas, é possível observar que o módulo de caracterização do *PSO-DS*

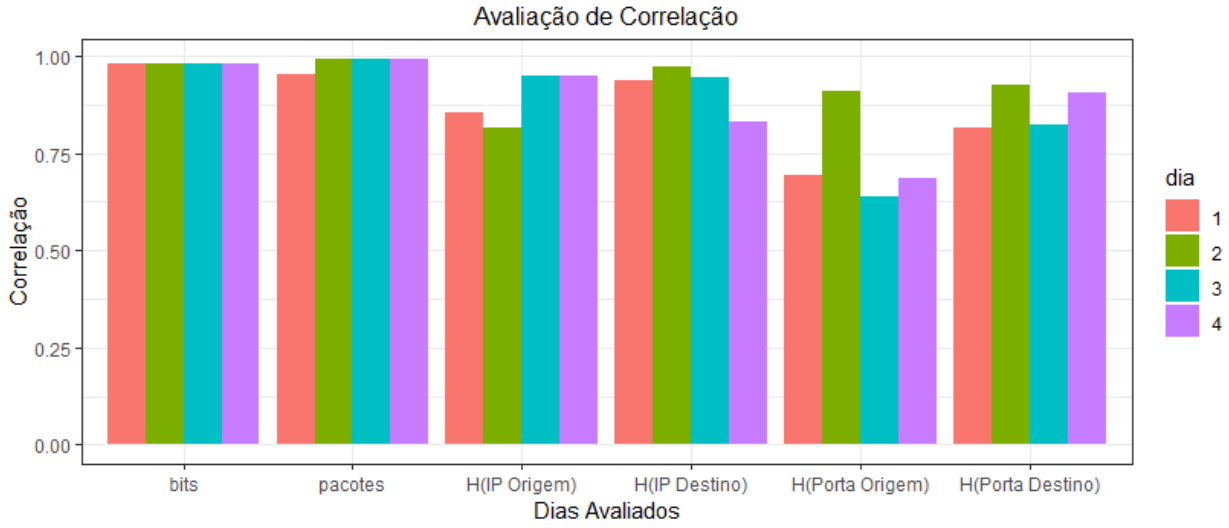


Figura 20 – Avaliação de CC para os 4 dias de tráfego do POX.

obteve resultados próximos para ambos os cenários avaliados. As assinaturas geradas utilizando o controlador POX para as métricas avaliadas obtiveram resultados levemente superiores se comparado com as assinaturas geradas utilizando o Floodlight. Em ambos os cenários a caracterização foi realizada de maneira eficiente, produzindo boas assinaturas.

6.4 Cenário 2: Avaliação da detecção de DDoS

Para realizar a análise de desempenho do módulo de detecção de anomalias do *PSO-DS*, foram utilizados dois dias do controlador Floodlight para os testes deste cenário. O dia 1 possui dois ataques de DDoS com diferentes intensidade e períodos do dia, variando a quantidade de *hosts* atacantes e o *host* de destino. O primeiro ataque corresponde a quatro *hosts* atacantes e a duração do ataque é das 11:00:00 às 12:01:00. O segundo ataque foi realizado das 15:00:00 às 16:00:00 com um total de 10 *hosts* atacante.

O dia 2 deste cenário também possui dois ataques de DDoS em período e intensidade distintas. O primeiro ataque foi realizado entre às 11:00:00 e 13:00:00, o número de atacantes são de 10 *hosts*. O segundo ataque de DDoS foi realizado das 15:00:00 às 16:00:00 por quatro atacantes. As figuras a seguir apresentam o tráfego real, o DSNSF, os alarmes detectados por dimensão, e por fim, o alarme geral sinalizado os períodos do dia em que foram detectados os eventos anômalos.

A Figura 21 e a Figura 22 apresentam o resultado da análise feito pelo sistema para o dia 1 e para o dia 2, respectivamente. Cada uma das dimensões apresentam os alarmes detectados individualmente, mas vale ressaltar que um alarme geral apenas será disparado quando pelo menos três dimensões acusarem a ocorrência de uma anomalia no mesmo intervalo analisado. É possível notar que a previsão gerada não incorporou as anomalias no seu processo de aprendizado. Essa característica é fundamental para uma boa taxa de

detecção de anomalias pelo sistema. Sabe-se que a caracterização feita pelo sistema utiliza uma janela deslizante, caso no processo de geração do DSNSF o sistema incorporasse a anomalia no seu aprendizado, o sistema de detecção iria considerar o tráfego anômalo como normal. Como poder ser observado, o método apresentado foi capaz de detectar e gerar alarmes quando as anomalias ocorreram. O *PSO-DS* gerou alarmes precisamente quando os ataques começaram, embora alguns alarmes falso-positivos tenham sido acionados em alguns pontos.

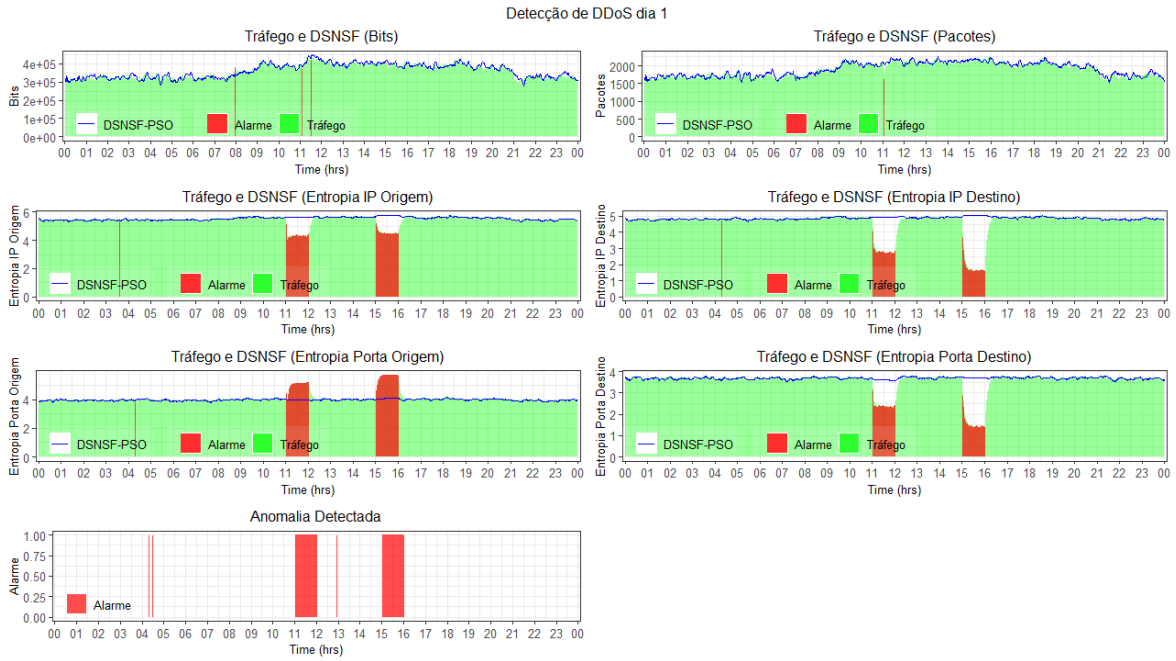


Figura 21 – Alarmes gerados pelo módulo de detecção para ocorrência de DDoS do dia 1.

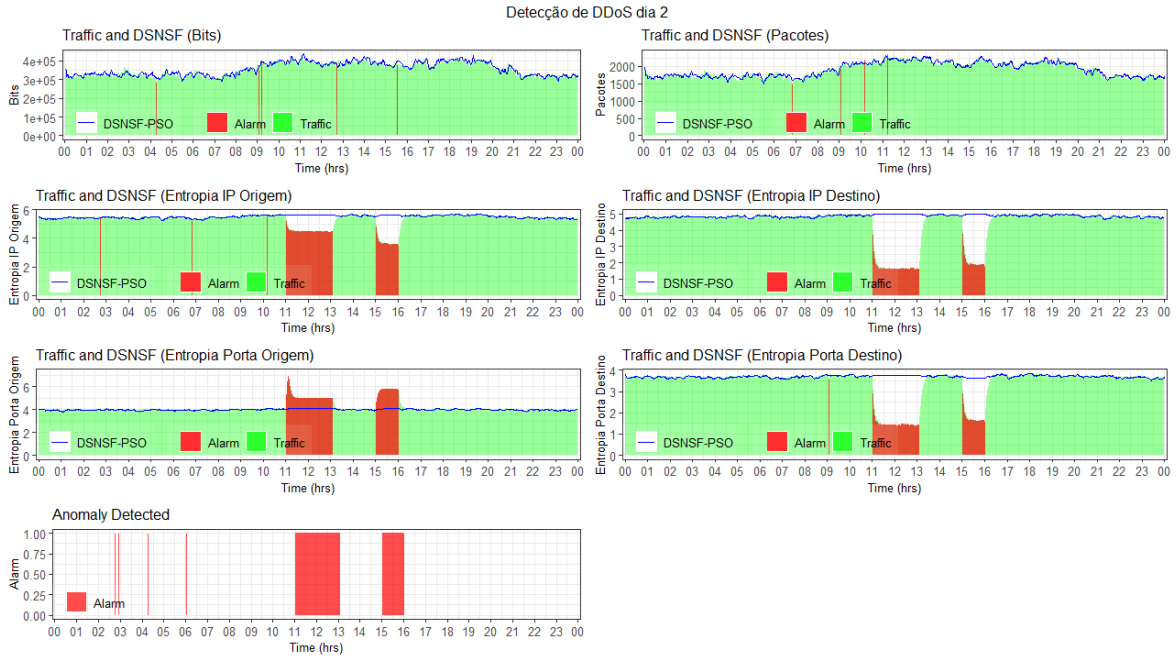


Figura 22 – Alarmes gerados pelo módulo de detecção para ocorrência de DDoS do dia 2.

O modulo de detecção proposto neste trabalho foi comparado com dois modelos clássicos presentes na literatura. O k-NN [127] é um clássico algoritmo de aprendizado de máquina baseado nos vizinhos mais próximo que foi amplamente empregado para detecção e classificação de eventos anômalos. O segundo método é o K-means [128], uma abordagem de aprendizado não supervisionada baseada na organização dos dados em *cluster* e que também foi aplicada para detecção de anomalias.

Os resultados das métricas aplicadas para avaliar o desempenho do sistema encontram-se na Tabela 13. O sistema conseguiu detectar aproximadamente todos os intervalos que apresentam ataques de DDoS. A taxa verdadeiro positivo foi de 98,93% para o dia 1 e de 98,37% para o dia 2, isto significa que para ambos os cenários houve uma ótima taxa de acerto para identificar onde ocorrem as anomalias. A taxa de falso-positivo foi baixa, para o dia 1 apresentou uma taxa de falso-positivo de 1,03% e para o dia 2 uma taxa de falso-positivo de 0,6%. Além disso, os resultados apontam ótimos valores para as métricas de classificação de acurácia, erro e AUC, com taxas de 98,94%, 1,05% e 99,29% para o dia 1 e 99,30%, 0,69% e 99,48% para o dia 2. O *PSO-DS* demonstrou resultados superiores ao K-means e resultados similares ao K-NN para os dois dias avaliados.

Tabela 13 – Resultado comparativo na detecção de ataques de DDoS no ambiente SDN.

	PSO-DS (Dia 1 e Dia 2)	**K-means (Dia 1 e Dia 2)**	**K-NN (Dia 1 e Dia2)**
TP	98,93% e 98,37%	85,24% e 74,43%	97,54% e 100%
TN	98,96% e 99,39%	45,59% e 56,81%	99,77% e 99,60%
FP	1,03% e 0,60%	54,40% e 43,18%	0,22% e 0,39%
FN	1,06% e 1,62%	14,75% e 27,56%	2,45% e 0%
Acurácia	98,94% e 99,30%	85,70% e 73,00%	99,58% e 99.65%
Precisão	98,96% e 99,39%	61,04% e 62,64%	99,76% e 99,60%
Erro	1,05% e 0,69%	34,57% e 35,37%	1,34% e 0,19%
AUC	99,29% e 99,48%	64,62% e 65,27%	99,47% e 99,92%

6.5 Cenário 3: Detecção e Mitigação de DDoS e Portscan

O caso de estudo proposto no cenário 3, o intervalo de análise do tráfego é realizado a cada 5 segundos. Isto é, a cada 5 segundos o controlador envia solicitações aos *switches*, obtendo os fluxos da rede para realizar o processo de análise pelo sistema proposto. Este cenário é composto por dois dias (48 horas) de emulação do comportamento da rede. Durante a emulação foram realizados ataques de *DDoS* e *Portscan* aplicando em diferentes períodos e intensidades ao longo do dia. A aplicação das anomalias busca avaliar a capacidade de detecção destes eventos anômalos pelo sistema. A topologia de rede utilizada é a mesma aplicada no cenário 1 e os módulos do sistema foram propostos utilizando o controlador Floodlight.

Considerando que o tempo de análise é realizado a cada 5 segundos, um fator importante é mensurar o tempo gasto pelos módulos do sistema de acordo com o tipo de anomalia e a política de mitigação. A Figura 23 mostra a quantidade de tempo em segundos para a realização de todo o processo do sistema para o dia um em que há a ocorrência de ataques de DDoS. Neste dia de observação o tempo máximo de processamento de todas as etapas do sistema não ultrapassou dois segundos. Já na Figura 24 é mostrado o tempo de processamento do sistema para o segundo dia de análise em que há a presença de intervalos que ocorrem ataques de Portscan. Pode ser notado que em ambos os dias de análise o tempo de processamento das etapas do sistema ficaram próximos a 1.5 segundos e em alguns pontos ocorreram picos próximos a 2 segundos, isso demonstra que o sistema ao realizar uma coleta do tráfego consegue executar as rotinas de caracterização, detecção e mitigação antes da próxima coleta. Na Figura 23 e na Figura 24, em vermelho é indicado os intervalos que ocorreram os ataques e pode ser notado que a ocorrência dos ataques não influenciou no tempo de processamento do módulos do sistema.

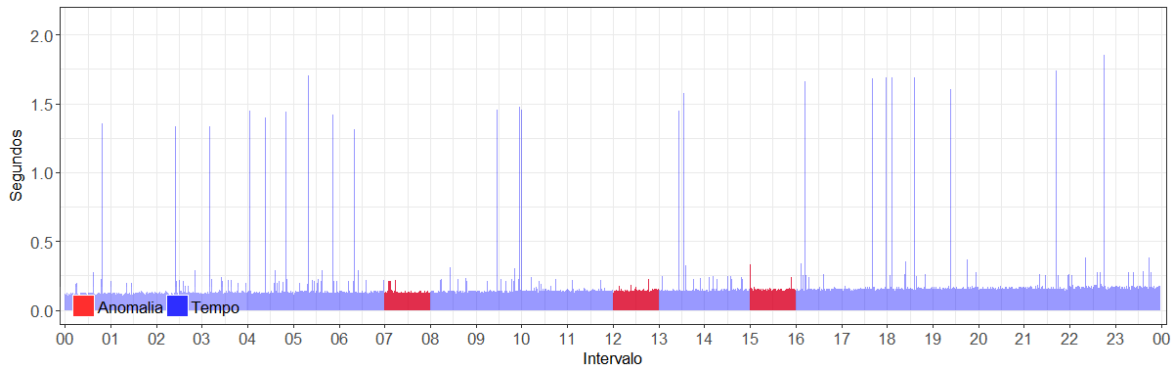


Figura 23 – Tempo de processamento do dia 1 cenário 3 contendo ataques de DDoS.

Os gráficos presentes na Figura 25 apresentam os resultados obtidos pelo sistema durante o processo caracterização e detecção do dia 1 do cenário 3. A área em verde representa o tráfego real emulado que foi coletado pelo controlador, assinatura gerada pelo módulo de caracterização é representada pela linha azul e em vermelho temos os

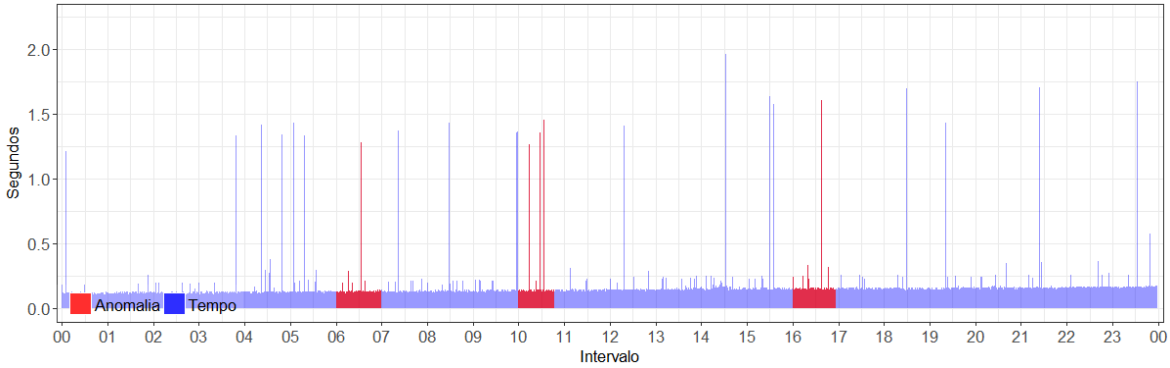


Figura 24 – Tempo de processamento do dia 1 cenário 3 contendo ataques de Portscan.

períodos de ataques detectados pelo sistema para cada atributo em análise. Nos períodos em que houve a ocorrência dos ataques pode ser observado um aumento considerado do tráfego, principalmente nos períodos em que os ataques foram realizados de forma mais intensa. A taxa de pacotes teve um aumento de aproximadamente 70% entre 15 e 16 horas, que corresponde o período em que foi realizado o ataque de DDoS mais intenso. Em um ataque DDoS são enviados inúmeros pacotes por diversos atacantes para uma vítima a um serviço e este comportamento pode ser notado nos atributos de entropia de IP e porta destino, pois no momento do ataque ocorreu uma concentração desses atributos.

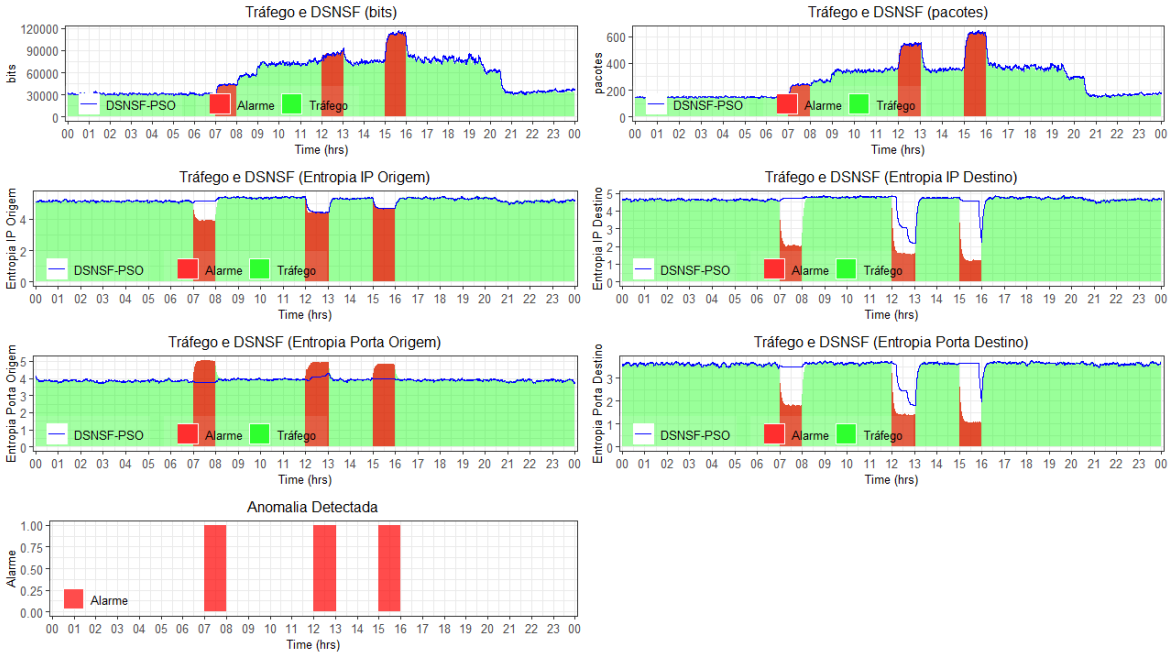


Figura 25 – Análise do tráfego com períodos de ataques de DDoS.

A Figura 26 apresenta os gráficos com os seis atributos analisados e os alarmes gerados pelo sistema do dia 2 do cenário 3. O tráfego gerado é representado pela área em verde e em azul temos as assinaturas geradas pelo módulo de caracterização proposto. Nos gráficos de cada dimensão as áreas em vermelho representam os alarmes, ou seja,

o momento que o módulo de detecção do sistema identificou a ocorrência de alguma anomalia. Neste dia de análise além do tráfego normal emulado foram realizados três períodos de ataques de Portscan variando o intervalo de varredura das portas e o tempo de envio entre os pacotes. Conforme pode ser notado, as alterações no comportamento do tráfego no momento da ocorrência do ataque são sucintas.

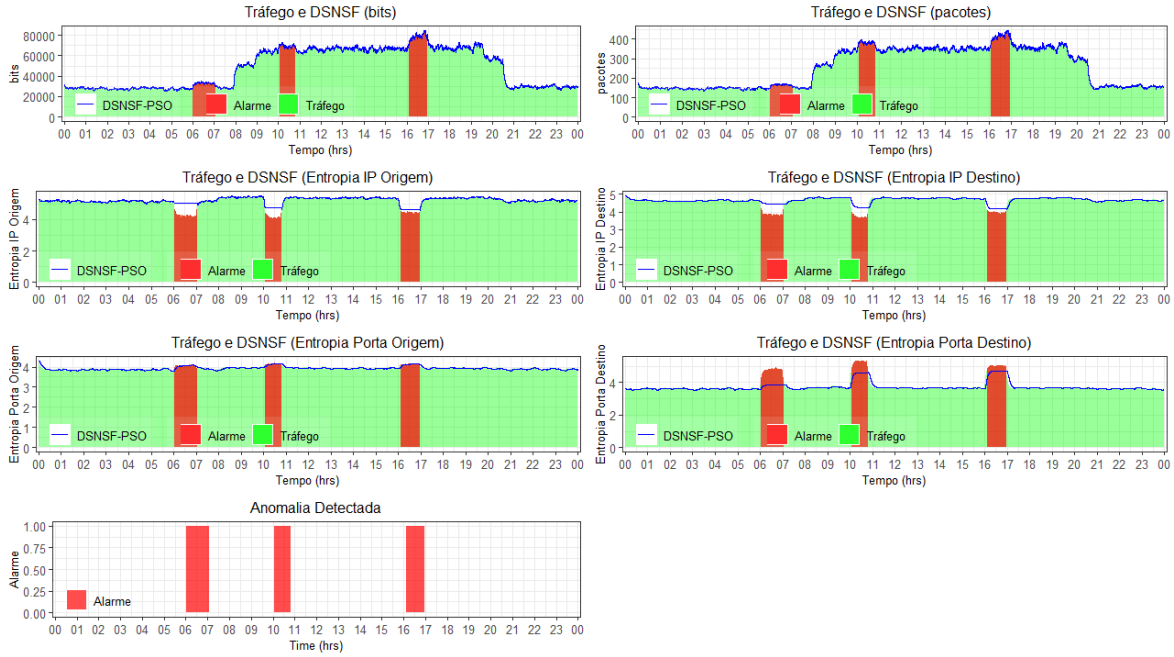


Figura 26 – Análise do tráfego com períodos de ataques de Portscan.

Uma avaliação mais detalhada do módulo de detecção foi aplicada comparando o resultado do modelo proposto com o K-means e o K-NN. Para a fase de treinamento do K-NN foi aplicado 10-*fold cross-validation* e o número ótimo de vizinho encontrado foi igual a 17. E para o método K-means o número de clusters empregados foram três representando cada classe do tráfego em análise (normal, DDoS, Portscan). Foram gerados um conjunto de dados para treinamento de ambos os métodos e esses dados são detalhados na Tabela 14.

Tabela 14 – Amostras utilizadas para o treinamento dos métodos K-NN e K-means.

	Quantidade de amostras
Normal	14851
DDoS	1424
Portscan	1005
Total	17280

A Tabela 15 apresenta as métricas de desempenho para os dois dias do cenário 3, comparando com os métodos de classificação. O módulo de detecção do sistema conseguiu detectar a maioria dos intervalos contendo os ataques de DDoS e Portscan, a taxa de

Precisão foi de aproximadamente 98%. Outro resultado importante foi a taxa de falsos alarmes disparados pelo sistema e a avaliação para essa métrica foi uma taxa de FP próximo a 2%. O K-NN obteve desempenho melhor que o K-means para todas as métricas comparadas e obteve o valor de Precisão de 96% que é próximo ao obtido pelo método proposto. É notável que o método proposto neste trabalho obteve desempenho superior em comparação aos outros métodos.

Tabela 15 – Resultado dos métodos comparados para deteceção de DDoS e Portscan.

	PSO-DS	**K-means**	**K-NN**
TP	99,66%	64,90%	76,84%
TN	97,85%	58,62%	96,90%
FP	2,15%	41,38%	3,10%
FN	0,34%	35,20%	23,16%
Acurácia	98,06%	59,36%	94,50%
Precisão	97,88%	61,06%	96,12%
Erro	1,94%	40,64%	5,50%
AUC	98,75%	61,71%	86,87

Uma análise mais aprofundada para comparação entre os métodos de classificação pode ser realizada por meio do cálculo da AUC. A Figura 27 expressa o resultado obtido para AUC para os métodos comparados. Poder ser notado que o *PSO-DS* possui o maior equilíbrio entre a taxa de TP e FP e obteve um valor de aproximadamente 98%. Já o K-means e o K-NN obtiveram 61% e 86%, respectivamente.

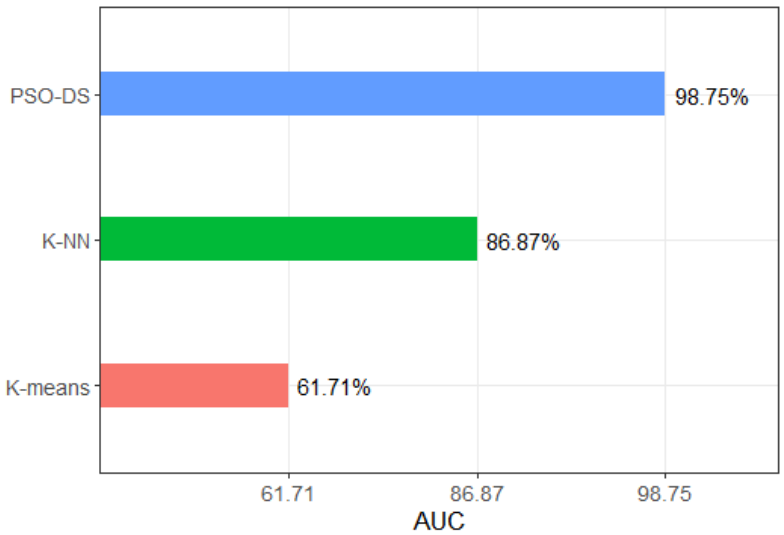


Figura 27 – AUC gerada para as curvas ROC dos modelos comparados.

6.5.1 Avaliação do módulo de mitigação

Após a identificação da ocorrência de um ataque de DDoS ou Portscan pelo módulo de detecção de anomalias do sistema, o módulo de mitigação é ativado para tomar as

contramedidas adequadas para os ataques detectados. A Figura 28 apresenta o comportamento do tráfego do dia 1 em que há a ocorrência de ataques de DDoS com módulo de mitigação desativado e compara o seu comportamento quando a mitigação está ativada. O tráfego gerado sem aplicação da política de mitigação é representado pela área em verde e a linha em azul mostra o tráfego após aplicado a mitigação contra o ataque de DDoS. Similar a figura anterior, a Figura 29 mostra a aplicação da mitigação para o dia 2 em que houve a ocorrência de ataques de Portscan. Visualmente em ambos os dias é possível notar quando os ataques são mitigados, os valores dos atributos em análise retornam ao seu comportamento esperado.

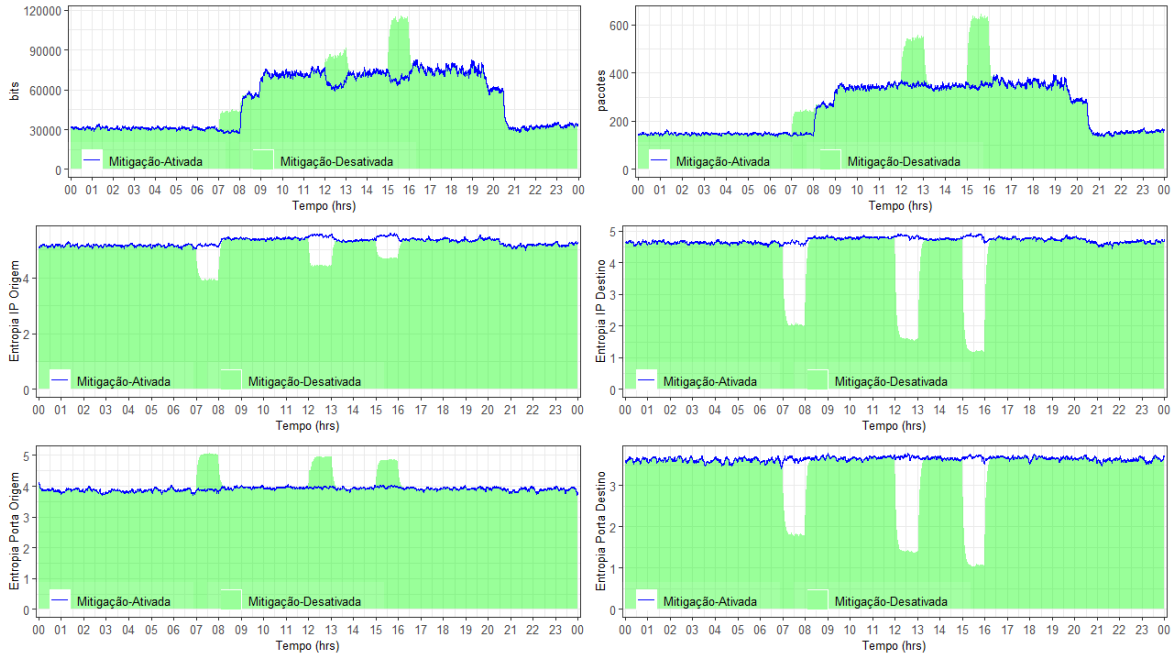


Figura 28 – Análise do tráfego com o módulo de mitigação desativado e ativado na ocorrência de ataques de DDoS.

Para verificar a eficiência do módulo de mitigação foi realizado o teste estatístico de MacNemar. Na Tabela 16 e na Tabela 17 temos as tabelas de contingência para a mitigação de *DDoS* e *Portscan*, respectivamente. As tabelas tem como objetivo apresentar o que foi detectado como normal e anômalo antes da mitigação – normal (antes) e anômalo (antes) e o que foi detectado como normal e anômalo depois da mitigação – normal (depois) e anômalo (depois). O teste de MacNemar aplicado possui nível de significância de $\alpha = 5\%$ e a hipótese nula de que as frequências marginais são iguais, o que indicaria que a mitigação não foi eficaz. Aplicando o teste na Tabela 16 e na Tabela 17, os resultados de p-valor para a mitigação de *DDoS* e *Portscan* foram abaixo de $2.2 \times e^{-16}$ que é menor que o valor de α, logo, a hipótese nula é rejeitada, o que indica que houve sim diferença nas frequências e que então a mitigação foi eficaz em minimizar os efeitos das ameaças.

Por fim, foi avaliado a taxa de pacotes anômalos descartados após a aplicação das políticas de mitigação. No dia 1 foram descartados 99,5% dos pacotes que eram de origem

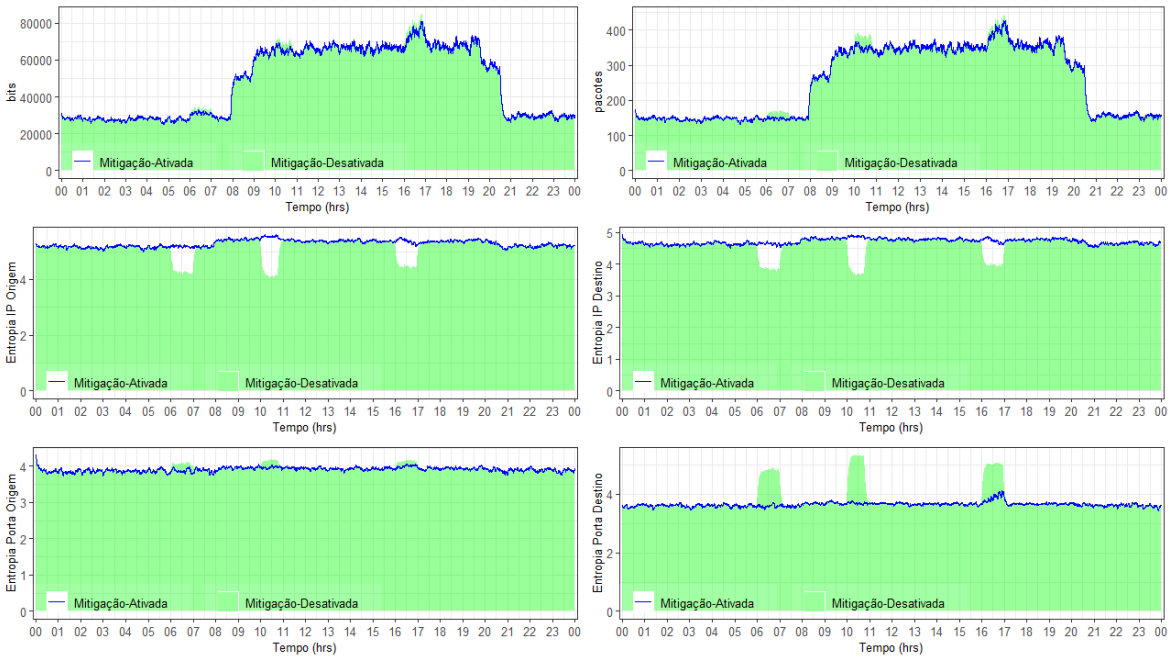


Figura 29 – Análise do tráfego com o módulo de mitigação desativado e ativado na ocorrência de ataques de Portscan.

Tabela 16 – Tabela de contingência aplicada para mitigação de DDoS.

	normal (depois)	**anômalo (depois)**
normal (antes)	15358	6
anômalo (antes)	1869	47

Tabela 17 – Tabela de contingência aplicada para mitigação de Portscan.

	normal (depois)	**anômalo (depois)**
normal (antes)	15068	40
anômalo (antes)	2158	14

de ataques de DDoS e no dia 2 a taxa de descartes de pacotes de ataques de Portscan foi de 94,8%. Desse modo, é possível verificar que o módulo de mitigação foi eficiente nas políticas tomadas contra os ataques de DDoS e Portscan.

6.6 Cenário 4: Tempo de coleta dos dados em 1 segundo

Neste cenário é apresentado um estudo de caso reduzindo o tempo de análise do tráfego para um segundo. O objetivo da redução deste intervalo é aproximar o tempo de análise ao tempo real. Com um tempo reduzido espera-se realizar a detecção de eventos anômalos no instante inicial do ataque. A identificação no instante inicial de um evento anômalo favorece a tomada de contramedidas para minimizar os danos e evitar que os recursos da rede, como os links de comunicação e as entradas das tabelas de fluxos dos

switches sejam saturados.

Por meio da análise de tempo de processamento realizada no cenário anterior, foi verificado que existem intervalos com picos de processamento próximos a dois segundos e a sua utilização para a análise do tráfego a cada um segundo não atenderia as exigências de processamento. Para este propósito, no módulo de caracterização e detecção foi realizada uma implementação do classificador Naive Bayes. O Naive Bayes (NB) é um classificador estatístico bem conhecido, que determina a probabilidade de uma amostra pertencer a uma determinada classe [129]. Algoritmos de classificação são importantes objetos de pesquisa aplicados para o reconhecimento e classificação de anomalias em redes de computadores [130]. O NB utiliza uma técnica simples de aprendizado supervisionado e com baixo custo computacional. É uma técnica eficiente capaz de trabalhar com um grande número de atributos, mostrando-se eficaz no emprego de soluções em ambientes de rede *SDN* [131][132]. O classificador NB é baseado no teorema de Bayes que é expresso matematicamente pela Equação (6.7)

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{6.7}$$

O Naive Bayes para Assinatura Digital (*NB-DS*) é aplicado para caracterização e detecção de anomalias do tráfego da rede em ambientes *SDN*. Para a fase treinamento do *NB-DS* os dados de Fluxos IP são coletados do controlador *SDN* e para cada uma das seis dimensões de fluxos analisadas são extraídos arquivos separados para a caracterização do padrão normal e anômalo do tráfego. O *NB-DS* é capaz de detectar ataques de DDoS e Portscan. A Figura 30 ilustra o processo de operacional do *NB-DS*.

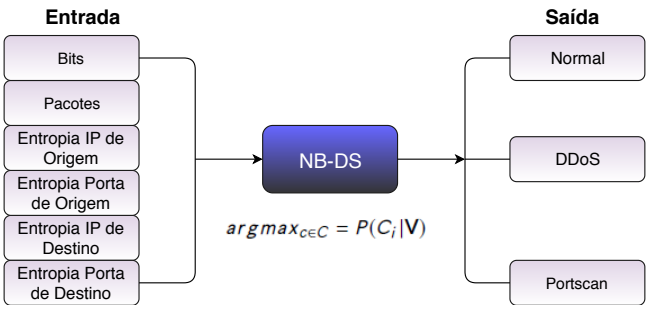


Figura 30 – *NB-DS* processo operacional.

O processo de treinamento do NB consiste em submeter o classificador a um conjunto de dados de treinamento T, com cada uma das classes rotuladas. Existem D classes, D_1, D_2, ..., D_d. Neste trabalho temos três classes, ou seja, o comportamento normal do tráfego da rede, ataque de DDoS e ataque de Portscan . Cada uma das amostras presentes em T é representada por um vetor n-dimensional, $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, representando cada um dos valores coletados dos atributos.

Dado uma amostra $\mathbf{x}$, o classificador NB vai realizar a predição que amostra $\mathbf{x}$ pertence à classe com maior probabilidade posterior, condicionada a esta amostra. Isto implica que $\mathbf{x}$ será classificado como D_i se e somente se

$$P(D_i|\mathbf{x}) > P(D_j|\mathbf{x}) \quad \text{para} \quad 1 \leq j \leq d, j \neq i \tag{6.8}$$

Consequentemente, a classe que maximiza $P(D_i \mid \mathbf{x})$ é encontrada. A probabilidade $P(D_i \mid \mathbf{x})$ para cada classe D_i é calculada utilizando o Teorema de Bayes

$$P(D_i|\mathbf{x}) = \frac{P(\mathbf{x}|D_i)P(D_i)}{P(\mathbf{x})} \tag{6.9}$$

A probabilidade $P(\mathbf{x})$ é a mesma para todas as classes, devido a isto, somente $P(\mathbf{x}|D_i)P(D_i)$ será maximizado. O cálculo de $P(\mathbf{x}|D_i)$ pode ser computacionalmente alto para ser computado se o conjunto de dados é composto por muitos atributos. O custo computacional no cálculo de $P(\mathbf{x}|D_i)P(D_i)$ é reduzido assumindo a suposição que os valores dos atributos são condicionalmente independentes. Isto é,

$$P(\mathbf{x}|D_i) \approx \prod_{y=1}^{n} P(x_y|D_i) \tag{6.10}$$

Quando os valores dos atributos utilizados forem contínuos, $P(x_y|D_i)$ pode ser estimado utilizando uma distribuição Gaussiana com média μ e desvio padrão σ, definida por:

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(v-\mu)^2}{2\sigma^2}} \tag{6.11}$$

$$P(x_y|C_i) = g(x_i, \mu_i, \sigma_i), \tag{6.12}$$

em que μ_i e σ_i são respectivamente a média e o desvio padrão dos valores dos atributos de cada uma das classes D_i presentes no conjunto de treinamento T.

A classificação de uma nova amostra do segmento do tráfego de rede pelo *NB-DS* é realizada de acordo com (6.13).

$$c = argmax_{d \in D} = P(D_i|\mathbf{x}) \tag{6.13}$$

Para avaliação deste cenário foi emulada uma nova topologia de rede. Conforme representado na Figura 31, os elementos da topologia encontram-se dispostos no formato de estrela, em que seis *switches* são conectados a um *switch* central. A topologia possui um total de 120 *hosts*, cada *switch* possui 20 *hosts* conectados.

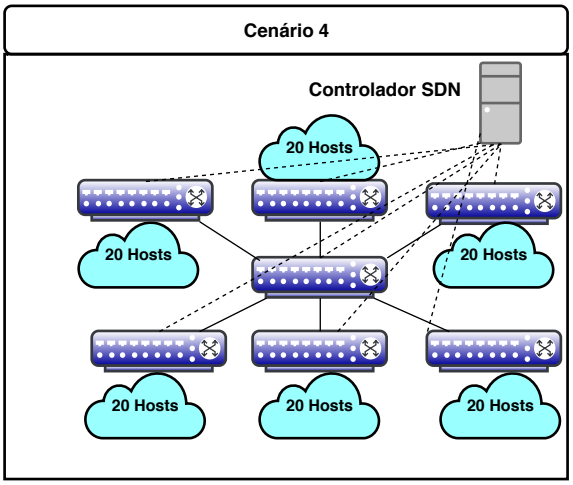


Figura 31 – Topologia de rede emulada do cenário 4.

Neste cenário foram emulados dois dias (48 horas) de funcionamento da rede, a quantidade de amostras do tráfego para as classes é ilustrada na Tabela 18. O primeiro dia possui tráfego com comportamento normal, ataques de DDoS e Portscan e foi utilizado para a fase de treinamento do *NB-DS*. Para o treinamento foi aplicado o *holdout* no conjunto de amostras do dia 1, utilizando 75% para treinamento e 25% para teste. O segundo dia foi aplicado para avaliar a eficiência do sistema para detecção e classificação dos eventos anômalos do tráfego. A emulação do segundo dia contém dados com comportamento normal e anômalo, durante emulação foram realizados dois períodos de ataques, o primeiro de DDoS e o segundo de PortScan. As informações sobre os ataques são apresentadas detalhadamente na Tabela 19.

Tabela 18 – Conjunto de dados utilizados no cenário.

	Amostras dia 1	**Amostras dia 2**
Normal	78300	78420
DDoS	4500	4380
Portscan	3600	3600
Total	86400	86400

O NB-DS, desenvolvido neste cenário, foi comparado com o K-means e o K-NN. O resultado das métricas aplicadas para avaliar a eficiência é mostrado na Tabela 20. Conforme pode ser observado o *NB-DS* se mostrou eficiente para a detecção dos ataques de DDoS e PortScan, obtendo um taxa de Precisão 99,98%. O K-NN foi o que obteve o melhor desempenho em relação a métricas avaliadas. Comparando o cálculo de AUC, o NB-DS e o K-NN obtiveram resultados próximo, 99,94% e 99,99 respectivamente. Demonstrando que o *NB-DS* e o K-NN tiveram um bom equilíbrio entre as taxas de detecção e os falsos-alarmes.

Tabela 19 – Informações dos parâmetros de ataque do cenário 4.

Tipo de Ataque	**Parâmetros**
DDoS	Atacantes: 15 IP atacantes: 10.0.0.11 - 10.0.0.25 IP da vítima: 10.0.0.50 Horário: 10:32 - 11:45
Portscan	IP atacante: 10.0.0.20 IP da vítima: 10.0.0.82 Portas: 1 - 30000 Tempo entre os pacotes : 0.08 Horário: 16:45 - 17:45

Tabela 20 – Resultado da detecção de DDoS e Portscan pelo *NB-DS*.

	NB-DS	**K-means**	**K-NN**
TP	99,98%	98,60%	99,91%
TN	99,88%	54,99%	100%
FP	0,01%	45,00%	0,00%
FN	0,11%	1,39%	0,08%
Acurácia	99,87%	54,91%	99,99%
Precisão	99,98%	68,66%	100%
Erro	0,06%	40,91%	0,008%
AUC	99,94%	76,80%	99,99%

A Figura 32, por meio da matriz de confusão, ilustra o resultado da classificação do tráfego realizada pelo NB-DS. A matriz de confusão apresenta os valores corretos e os preditos pelo classificador. É possível identificar quais os acertos e os erros para cada classe, assim, permitindo avaliar a eficiência do método na detecção de anomalias. A diagonal principal da matriz mostra a quantidade de amostras que foram classificadas de maneira correta e os demais valores apresentam os erros cometidos pelo classificador. Os ataques de DDoS foram classificados corretamente, não havendo erro na sua classificação. Na classificação do tráfego normal apenas 12 amostras foram classificadas de maneira errada, essas amostras foram classificadas como DDoS. O ataque de Portscan foi a classe que apresentou o maior erro na sua classificação, duas amostras foram classificadas como DDoS e nove amostras foram classificadas como tráfego normal.

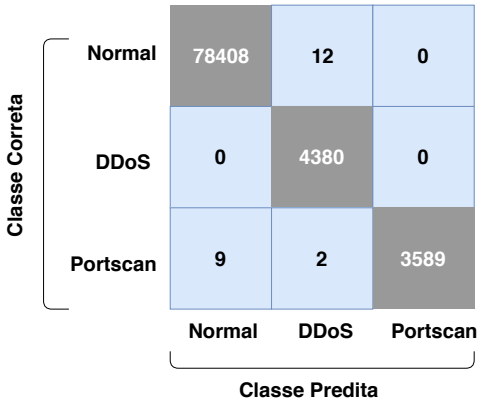


Figura 32 – Matriz de confusão da classificação do tráfego pelo *NB-DS*.

6.6.1 Avaliação do módulo de mitigação para o cenário 4

O módulo de mitigação também foi avaliado para este cenário. A partir dos alarmes gerados pelo processo de classificação do *NB-DS* foram aplicadas as políticas de mitigação. A Figura 33 apresenta os atributos do tráfego em verde sem a aplicação da mitigação e em azul é a mostrado o tráfego após a aplicação da mitigação. No período entre 10:32:00 e 11:45:00 temos a ocorrência de um ataque de DDoS, neste período é notável o aumento da taxa de pacotes e bits quando o módulo de mitigação está desabilitado, mas com a ativação do módulo o tráfego tende a voltar a sua normalidade devido aos descarte dos pacotes anômalos. O período do ataque de Portscan entre 16:45 e 17:45 provoca pequenas alterações no comportamento do tráfego, com aplicação da mitigação os atributos afetados também voltam a sua normalidade.

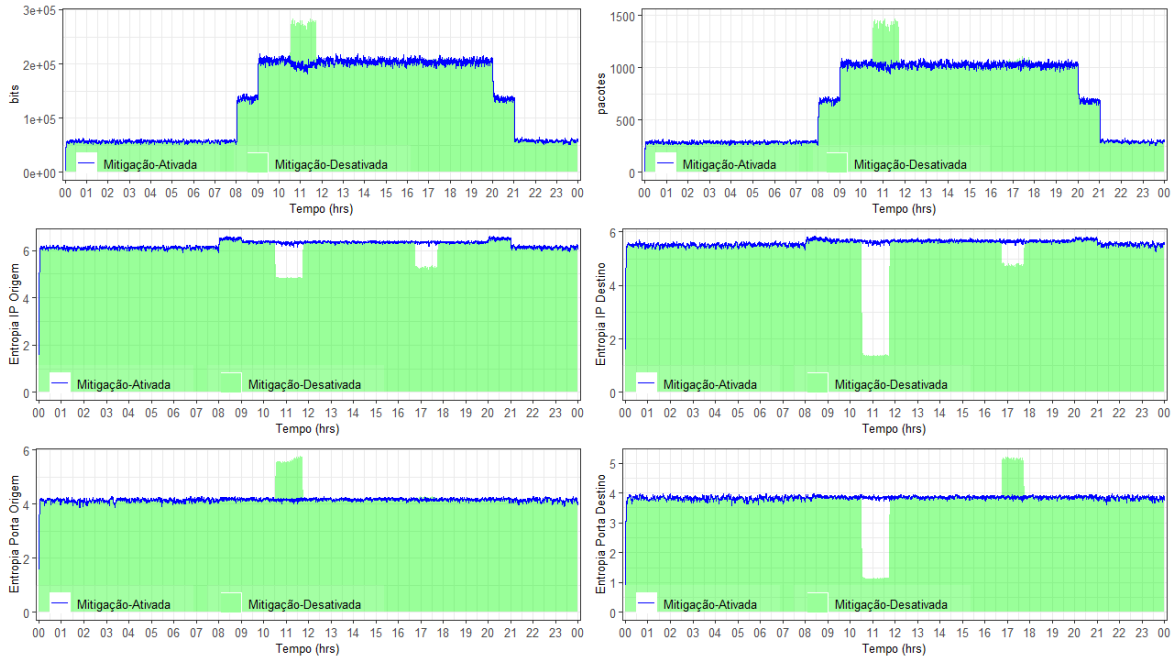


Figura 33 – Análise do tráfego com o módulo de mitigação desativado e ativado na ocorrência de ataques de DDoS e Portscan.

Neste cenário, a análise da mitigação por meio do teste de McNemar e das taxas de pacotes descartados também foram aplicados. O nível de significância para o teste de McNemar foi de $\alpha = 5\%$. A Tabela 21 fornece as informações sobre a classificação do tráfego entre anômalo e normal antes e após o processo de mitigação. Aplicando o teste nas informações da tabela, os resultados de p-valor foram abaixo de $2.2 \times e^{-16}$ que é menor que o valor de α, logo, a hipótese nula neste cenário também foi rejeitada, o que indica que houve sim diferença nas frequências e que, então, a mitigação foi eficaz em minimizar os efeitos das ameaças. E, a taxa pacotes anômalos descartados pelo sistema foi de 99.95%, o que demonstra que praticamente todos os pacotes de anômalos foram descartados.

Tabela 21 – Tabela de contingência aplicada para avaliação da mitigação.

	normal (depois)	**anômalo (depois)**
normal (antes)	78417	0
anômalo (antes)	7952	31

6.7 Custo Computacional *PSO-DS* e *NB-DS*.

A clusterização é um problema NP-completo, mas o seu custo pode ser reduzido com um número fixo de iterações e com a aplicação do *PSO* foi possível reduzir o custo computacional. A complexidade computacional assintótica do *PSO* está relacionada com o número de instruções executadas. Os algoritmos evolucionários têm custo computacional $O(N^*P^*F)$ para cada iteração, em que N é dimensionalidade do problema, P é o tamanho da população e F é custo da função objetivo. A execução do *PSO-DS* é multiplicada por seis, pois o processo de caracterização é feito de maneira separada para cada atributo de fluxo.

O Naive Bayes é um classificador estatístico supervisionado e, portanto, para a construção do modelo é necessária uma fase de treinamento. E a avaliação do custo computacional do *NB-DS* deve ser avaliado em duas fases: treinamento e operação. Considerando T o tamanho do *dataset* de treinamento, N a dimensionalidade do problema e D o número de classes, temos que o custo computacional do *NB-DS* para a fase de treinamento é $O(T^*N)$. O custo computacional da fase de operação do *NB-DS* para a classificação de uma nova amostra $\mathbf{x}$ do tráfego pode ser definido como $O(N^*D)$ para cada uma das D_i classes presentes no problema. Considerando a fase de treinamento de operação do *NB-DS* podemos concluir que o modelo proposto apresenta um custo computacional linear.

7 CONCLUSÃO

O crescimento da demanda das aplicações rede, os inúmeros dispositivos interconectados e a heterogeneidade do ambiente de rede aumentam a complexidade de configuração e gerência. As redes *SDN* surgiram com o propósito de flexibilizar o controle e gerenciamento, assim como nas redes tradicionais, esse ambiente também está suscetível à ocorrência de ataques. A detecção e contramedidas tomadas contra os ataques é uma tarefa importante para manter o funcionamento da rede e a qualidade dos serviços. Visando propor uma solução para detecção e mitigação de anomalias foi desenvolvido um sistema. A solução que foi desenvolvida nesta dissertação apresenta uma arquitetura modular em que as suas atividades são realizadas de maneira automatizada para facilitar no monitoramento, detecção e nas contramedidas diante de um ataque.

As principais contribuições obtidas com o sistema proposto neste trabalho foram:

- Os resultados demonstraram que o sistema proposto é capaz de detectar ataques de *DDoS* e *Portscan* no ambiente *SDN*;
- A caracterização do tráfego realizada pelo sistema apresentou bons resultados na predição do comportamento normal da rede e dispensa a utilização de uma base histórica de semanas anteriores;
- Redução do intervalo de análise das amostras coletadas do tráfego para diminuir o tempo de resposta aos ataques detectados;
- Aplicação de políticas de mitigação automatizada para minimizar os danos causados por ataques e manter os requisitos de operação da rede;

Para validar o sistema desenvolvido, foram propostos quatro cenários com diferentes características. O primeiro cenário foi utilizado para avaliar a eficiência do módulo de caracterização para gerar as assinaturas do comportamento normal do tráfego. Neste cenário foram explorados os controladores de redes Pox e Floodlight. No total foram avaliados 9 dias de caracterização. A avaliação da qualidade das assinaturas foi realizada por meio da aplicação das métricas de NMSE e Coeficiente de Correlação e os resultados obtidos demonstraram que o módulo de caracterização foi eficiente na geração das assinaturas do tráfego.

O segundo cenário foi empregado para avaliar a detecção de ataques do tipo de DDoS. Neste cenário foi realizada a emulação de dois dias de operação da rede. Durante o processo emulação foram realizados períodos de ataques de DDoS com diferentes intensidades e duração. Os resultados das métricas de Acurácia, AUC e Precisão demonstraram

que o sistema foi capaz de detectar praticamente em sua totalidade os períodos das ocorrências dos ataques. Ambos os dias analisados apresentaram taxa de falso-positivos baixa, ou seja, poucos intervalos legítimos foram detectados como anômalos.

O terceiro cenário foi utilizado para avaliar a detecção e mitigação de ataques de DDoS e Portscan. Neste cenário, o tempo de coleta das amostras do tráfego foi reduzido para cinco segundos. Foram realizados dois dias emulação contendo três períodos para os dois tipos de ataques analisados. Foi avaliado o tempo de processamento de todas as etapas realizadas pelo sistema e o tempo máximo de processamento foi próximo a dois segundos, isso demonstrou que o sistema foi capaz de executar as rotinas dos módulos de caracterização, detecção e mitigação antes que a próxima coleta ocorresse. Foi realizada a comparação da taxa de detecção com outros métodos presentes na literatura e os resultados apresentados demonstram que a abordagem desenvolvida obteve desempenho superior em relação aos métodos de detecção comparados. As contramedidas realizadas por meio do emprego das políticas foram eficientes para mitigar os ataques de DDoS e Portscan e após mitigação os atributos do tráfego retornaram ao seu comportamento esperado.

Os experimentos realizados no quarto cenário tiveram como objetivo reduzir o tempo de coleta do tráfego a um segundo para permitir a detecção dos ataques no estágio inicial. O *PSO-DS* realiza o processo de caracterização aplicando uma janela deslizante e o seu tempo de processamento é superior a um segundo, para atender essa demanda foi proposto o *NB-DS* para a fase caracterização e detecção. Neste cenário foram avaliados ataques de DDoS e Portscan. O resultado das métricas para avaliação do método proposto demonstraram a sua eficiência para a detecção dos ataques. A mitigação desses ataques também foi realizada e grande parte dos períodos anômalos foram mitigados.

Os resultados obtidos demonstram que os módulos que compõem o sistema proposto foram eficientes, cumprindo com os objetivos designados a cada um deles. A execução das atividades realizadas pelo sistema é autônoma, isto é, o processo de monitoramento, identificação de eventos adversos e as contramedidas tomadas são realizadas sem a necessidade da interferência humana. O monitoramento e gerenciamento da rede é uma atividade complexa, a aplicação de um sistema autônomo auxilia nas tarefas designadas ao administrador para manter e garantir o funcionamento da rede em sua totalidade. Portanto, o sistema desenvolvido neste trabalho pode ser aplicado para colaborar e facilitar nos procedimentos de gerenciamento e garantir a disponibilidade dos serviços oferecidos.

A arquitetura modular do sistema permite a manutenção e adaptação de outras técnicas para caracterização do tráfego, detecção e mitigação de anomalias em ambientes *SDN*. Essa característica permite adaptar o sistema conforme a dinâmica da rede se altera e surgem novas demandas de segurança. Desse modo, os trabalhos futuros podem explorar outras vulnerabilidades e incorporar políticas de mitigação para suprir as novas demandas quem possam surgir nos ambientes de redes *SDN*. Outro ponto que pode ser estendido

é a exploração de técnicas que permite a detecção de eventos anômalos que ocorrem simultaneamente.

REFERÊNCIAS

[1] Xia, W.; Wen, Y.; Foh, C. H.; Niyato, D.; Xie, H. A survey on software-defined networking. *IEEE Communications Surveys Tutorials*, v. 17, n. 1, p. 27–51, Firstquarter 2015. ISSN 1553-877X.

[2] KREUTZ, D.; RAMOS, F. M. V.; VERISSIMO, P. E.; ROTHENBERG, C. E.; AZODOLMOLKY, S.; UHLIG, S. Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, IEEE, v. 103, n. 1, p. 14–76, jan 2015. ISSN 0018-9219. Disponível em: <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6994333>.

[3] HAKIRI, A.; GOKHALE, A.; BERTHOU, P.; SCHMIDT, D. C.; GAYRAUD, T. Software-defined networking: Challenges and research opportunities for future internet. *Computer Networks*, Elsevier B.V., v. 75, n. PartA, p. 453–471, 2014. ISSN 13891286. Disponível em: <http://dx.doi.org/10.1016/j.comnet.2014.10.015>.

[4] KALKAN, K.; GUR, G.; ALAGOZ, F. Defense Mechanisms against DDoS Attacks in SDN Environment. *IEEE Communications Magazine*, v. 55, n. 9, p. 175–179, 2017. ISSN 01636804.

[5] JONKER, M.; KING, A.; KRUPP, J.; ROSSOW, C.; SPEROTTO, A.; DAINOTTI, A. Millions of targets under attack: A macroscopic characterization of the dos ecosystem. In: *Proceedings of the 2017 Internet Measurement Conference*. New York, NY, USA: ACM, 2017. (IMC '17), p. 100–113. ISBN 978-1-4503-5118-8. Disponível em: <http://doi.acm.org/10.1145/3131365.3131383>.

[6] CARVALHO, L. F.; ABRãO, T.; MENDES, L. de S.; PROENçA, M. L. An ecosystem for anomaly detection and mitigation in software-defined networking. *Expert Systems with Applications*, v. 104, p. 121 – 133, 2018. ISSN 0957-4174. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417418301726>.

[7] LIAO, H.-J.; LIN, C.-H. R.; LIN, Y.-C.; TUNG, K.-Y. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, v. 36, n. 1, p. 16 – 24, 2013. ISSN 1084-8045. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1084804512001944>.

[8] AHMED, M.; Naser Mahmood, A.; HU, J. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, Elsevier, v. 60, p. 19–31, 2016. ISSN 10958592. Disponível em: <http://dx.doi.org/10.1016/j.jnca.2015.11.016>.

[9] 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.

[10] GARG, S.; BATRA, S. A novel ensembled technique for anomaly detection. *International Journal of Communication Systems*, v. 30, n. 11, p. 1–16, 2017. ISSN 10991131.

[11] GARG, S.; BATRA, S. Fuzzified Cuckoo based Clustering Technique for Network Anomaly Detection. *Computers and Electrical Engineering*, Elsevier Ltd, v. 0, p. 1–20, 2017. ISSN 00457906. Disponível em: <http://dx.doi.org/10.1016/j.compeleceng.2017.07.008>.

[12] ADANIYA, M. H. A. C.; ABRÃO, T.; PROENçA, M. L. Anomaly detection using metaheuristic firefly harmonic clustering. *Jornal of Netwoks*, v. 8, p. 82–91, 2013.

[13] ZHANG, P.; SUN, S. Decentralized Network Anomaly Detection via a Riemannian Cluster Approach. *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, p. 1–6, 2017. Disponível em: <http://ieeexplore.ieee.org/document/8254537/>.

[14] FERNANDES, G.; PENA, E. H. M.; CARVALHO, L. F.; RODRIGUES, J. J. P. C.; PROENcA JR., M. L. Statistical, forecasting and metaheuristic techniques for network anomaly detection. In: *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2015. (SAC '15), p. 701–707. ISBN 978-1-4503-3196-8. Disponível em: <http://doi.acm.org/10.1145/2695664.2695852>.

[15] ASSIS, M. V. O. D.; HAMAMOTO, A. H.; ABRãO, T.; PROENçA, M. L. A game theoretical based system using holt-winters and genetic algorithm with fuzzy logic for dos/ddos mitigation on sdn networks. *IEEE Access*, v. 5, p. 9485–9496, 2017. ISSN 2169-3536.

[16] PROENÇA, M. L.; COPPELMANS, C.; BOTTOLI, M.; ALBERTI, A.; MENDES, L. S. The hurst parameter for digital signature of network segment. Springer Berlin Heidelberg, Berlin, Heidelberg, p. 772–781, 2004.

[17] PROENÇA Jr, M. L. *Baseline Aplicado a gerência de redes*. Tese (Doutorado) — Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, 2005.

[18] NUNES, I.; SCHARDONG, F.; SCHAEFFER-FILHO, A. Bdi2dos: An application using collaborating bdi agents to combat ddos attacks. *Journal of Network and Computer Applications*, v. 84, p. 14 – 24, 2017. ISSN 1084-8045. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1084804517300577>.

[19] YOON, C.; PARK, T.; LEE, S.; KANG, H.; SHIN, S.; ZHANG, Z. Enabling security functions with SDN: A feasibility study. *Computer Networks*, Elsevier Ltd., v. 85, n. 2015, p. 19–35, 2015. ISSN 13891286. Disponível em: <http://dx.doi.org/10.1016/j.comnet.2015.05.005>.

[20] ALEROUD, A.; ALSMADI, I. Identifying cyber-attacks on software defined networks: An inference-based intrusion detection approach. *Journal of Network and Computer Applications*, Elsevier, v. 80, n. November 2016, p. 152–164, 2017. ISSN 10958592. Disponível em: <http://dx.doi.org/10.1016/j.jnca.2016.12.024>.

[21] Santos Da Silva, A.; WICKBOLDT, J. A.; GRANVILLE, L. Z.; SCHAEFFER-FILHO, A. ATLANTIC: A framework for anomaly traffic detection, classification, and mitigation in SDN. *Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, n. Noms, p. 27–35, 2016.

[22] GARG, G.; GARG, R. Detecting anomalies efficiently in SDN using adaptive mechanism. *International Conference on Advanced Computing and Communication Technologies, ACCT*, v. 2015-April, p. 367–370, 2015. ISSN 23270659.

[23] MOUSAVI, S. M.; ST-HILAIRE, M. Early detection of DDoS attacks against SDN controllers. *2015 International Conference on Computing, Networking and Communications, ICNC 2015*, p. 77–81, 2015. ISSN 9781479969593.

[24] CARVALHO, L. F.; FERNANDES, G.; RODRIGUES, J. J.; MENDES, L. S.; PROENCA, M. L. A novel anomaly detection system to assist network management in SDN environment. *IEEE International Conference on Communications*, 2017. ISSN 15503607.

[25] LI, C.; WU, Y.; YUAN, X.; SUN, Z.; WANG, W.; LI, X.; GONG, L. Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN. *International Journal of Communication Systems*, v. 31, n. 5, p. 1–15, 2018. ISSN 10991131.

[26] UNB. *ISCX intrusion detection evaluation DataSet*. 2018. Acessado em: https://www.unb.ca/cic/datasets/index.html. Disponível em: <https://www.unb.ca/cic/datasets/index.html>.

[27] EDDY, W.; CLARK, G.; DAILEY, J. *Customer-Controlled Filtering Using SDN*. [S.l.], 2015. Work in Progress. Disponível em: <https://datatracker.ietf.org/doc/html/draft-eddy-sdnrg-customer-filters-01>.

[28] SAHAY, R.; BLANC, G.; ZHANG, Z.; DEBAR, H. ArOMA: An SDN based autonomic DDoS mitigation framework. *Computers and Security*, Elsevier Ltd, v. 70, p. 482–499, 2017. ISSN 01674048. Disponível em: <https://doi.org/10.1016/j.cose.2017.07.008>.

[29] SHENFIELD, A.; DAY, D.; AYESH, A. Intelligent intrusion detection systems using artificial neural networks. *ICT Express*, Elsevier B.V., v. 4, n. 2, p. 95–99, 2018. ISSN 24059595. Disponível em: <https://doi.org/10.1016/j.icte.2018.04.003>.

[30] HAMAMOTO, A. H.; CARVALHO, L. F.; SAMPAIO, L. D. H.; ABRÃO, T.; PROENÇA, M. L. Network Anomaly Detection System using Genetic Algorithm and Fuzzy Logic. *Expert Systems with Applications*, Elsevier Ltd, v. 92, p. 390–402, 2017. ISSN 09574174. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S095741741730619X>.

[31] FERNANDES, G.; CARVALHO, L. F.; RODRIGUES, J. J.; PROENÇA, M. L. Network anomaly detection using IP flows with Principal Component Analysis and Ant Colony Optimization. *Journal of Network and Computer Applications*, Elsevier, v. 64, p. 1–11, 2016. ISSN 10848045. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1084804516000618>.

[32] PENA, E. H. M.; BARBON, S.; RODRIGUES, J. J. P. C.; PROENçA, M. L. Anomaly detection using digital signature of network segment with adaptive arima model and paraconsistent logic. In: *2014 IEEE Symposium on Computers and Communications (ISCC)*. [S.l.: s.n.], 2014. p. 1–6. ISSN 1530-1346.

[33] CARVALHO, L. F.; BARBON, S.; MENDES, L. D. S.; PROENÇA, M. L. Unsupervised learning clustering and self-organized agents applied to help network management. *Expert Systems with Applications*, v. 54, p. 29–47, 2016. ISSN 09574174.

[34] KARAMI, A.; GUERRERO-ZAPATA, M. A fuzzy anomaly detection system based on hybrid PSO-Kmeans algorithm in content-centric networks. *Neurocomputing*, Elsevier, v. 149, n. PC, p. 1253–1269, 2015. ISSN 18728286. Disponível em: <http://dx.doi.org/10.1016/j.neucom.2014.08.070>.

[35] ABUROMMAN, A. A.; Ibne Reaz, M. B. A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing Journal*, Elsevier B.V., v. 38, p. 360–372, 2016. ISSN 15684946.

[36] NUNES, B. A. A.; MENDONCA, M.; NGUYEN, X. N.; OBRACZKA, K.; TURLETTI, T. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys Tutorials*, v. 16, n. 3, p. 1617–1634, Third 2014. ISSN 1553-877X.

[37] FEAMSTER, N.; REXFORD, J.; ZEGURA, E. The road to sdn: An intellectual history of programmable networks. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 44, n. 2, p. 87–98, abr. 2014. ISSN 0146-4833. Disponível em: <http://doi.acm.org/10.1145/2602204.2602219>.

[38] TENNENHOUSE, D. L.; WETHERALL, D. J. Towards an active network architecture. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 37, n. 5, p. 81–94, out. 2007. ISSN 0146-4833. Disponível em: <http://doi.acm.org/10.1145/1290168.1290180>.

[39] CAMPBELL, A. T.; MEER, H. G. D.; KOUNAVIS, M. E.; MIKI, K.; VICENTE, J. B.; VILLELA, D. A survey of programmable networks. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 29, n. 2, p. 7–23, abr. 1999. ISSN 0146-4833. Disponível em: <http://doi.acm.org/10.1145/505733.505735>.

[40] Smith, J. M.; Nettles, S. M. Active networking: one view of the past, present, and future. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, v. 34, n. 1, p. 4–18, Feb 2004. ISSN 1094-6977.

[41] CALVERT, K. Reflections on network architecture: An active networking perspective. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 36, n. 2, p. 27–30, abr. 2006. ISSN 0146-4833. Disponível em: <http://doi.acm.org/10.1145/1129582.1129590>.

[42] LAKSHMAN, T.; NANDAGOPAL, T.; RAMJEE, R.; SABNANI, K.; WOO, T. The softrouter architecture. In: CITESEER. *Proc. ACM SIGCOMM Workshop on Hot Topics in Networking*. [S.l.], 2004. v. 2004.

[43] OKTIAN, Y. E.; LEE, S.; LEE, H.; LAM, J. Distributed sdn controller system: A survey on design choice. *Computer Networks*, v. 121, p. 100 – 111, 2017. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128617301706>.

[44] JAMMAL, M.; SINGH, T.; SHAMI, A.; ASAL, R.; LI, Y. Software defined networking: State of the art and research challenges. *Computer Networks*, v. 72, p. 74 – 98, 2014. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128614002588>.

[45] SHENKER, S. The Future of Networking , and the Past of Protocols Software-Defined Networking.

[46] MASOUDI, R.; GHAFFARI, A. Software defined networks: A survey. *Journal of Network and Computer Applications*, v. 67, p. 1 – 25, 2016. ISSN 1084-8045. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1084804516300297>.

[47] DIERKS, E. R. T. The transport layer security (tls) protocol version 1.1. 2006. Disponível em: <https://www.ietf.org/rfc/rfc4346.txt>.

[48] FREIER P. KARLTON, P. K. A. The secure sockets layer (ssl) protocol version 3.0. 2011. Disponível em: <https://tools.ietf.org/html/rfc6101>.

[49] HALEPLIDIS K. PENTIKOUSIS, S. D. J. H. S. D. M. O. K. E. Software-defined networking (sdn): Layers and architecture terminology. 2015. Disponível em: <https://tools.ietf.org/html/rfc7426>.

[50] SCHALLER, S.; HOOD, D. Software defined networking architecture standardization. *Computer Standards and Interfaces*, v. 54, n. January, p. 197–202, 2017. ISSN 09205489.

[51] ROTHENBERG, C. E.; NASCIMENTO, M. R.; SALVADOR, M. R.; MAGALHÃES, M. F. OpenFlow e redes definidas por software: um novo paradigma de controle e inovação em redes de pacotes. *Cadernos CPqD Tecnologia*, v. 7, n. 1, p. 65–75, 2011.

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

[53] OU, C. M. Host-based intrusion detection systems adapted from agent-based artificial immune systems. *Neurocomputing*, Elsevier, v. 88, p. 78–86, 2012. ISSN 09252312. Disponível em: <http://dx.doi.org/10.1016/j.neucom.2011.07.031>.

[54] SHIPLEY, G. Intrusion detection, take two. *Netw. Comput.*, CMP Publications, Inc., Manhasset, NY, USA, v. 10, n. 23, p. 44–48, nov. 1999. ISSN 1046-4468. Disponível em: <http://dl.acm.org/citation.cfm?id=354305.354311>.

[55] CASE J. CASE, M. S. J. D. J. A simple network management protocol (snmp). 1990. Disponível em: <https://www.ietf.org/rfc/rfc1157.txt>.

[56] SOFTWARE, C. Introduction to cisco ios netflow - a technical overview. 2012. Disponível em: <http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_white_paper0900aecd80406232.html>.

[57] SFLOW. Traffic monitoring using sflow. 2003. Disponível em: <http://www.sflow.org/sFlowOverview.pdf>.

[58] QUITTEK T. ZSEBY, B. C. S. Z. J. Requirements for ip flow information export (ipfix). 2004. Disponível em: <https://tools.ietf.org/html/rfc3917>.

[59] FERNANDES, G.; RODRIGUES, J. J. P. C.; CARVALHO, L. F.; AL-MUHTADI, J. F.; PROENÇA, M. L. A comprehensive survey on network anomaly detection. *Telecommunication Systems*, Jul 2018. ISSN 1572-9451. Disponível em: <https://doi.org/10.1007/s11235-018-0475-8>.

[60] THOTTAN, M.; JI, C. Anomaly detection in ip networks. *Trans. Sig. Proc.*, IEEE Press, Piscataway, NJ, USA, v. 51, n. 8, p. 2191–2204, ago. 2003. ISSN 1053-587X. Disponível em: <https://doi.org/10.1109/TSP.2003.814797>.

[61] ZARPELAO, B. B. *Detecção de Anomalias em Redes de Computadores*. Tese (Doutorado) — Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, 2010.

[62] STADING, T.; MANIATIS, P.; BAKER, M. Peer-to-peer caching schemes to address flash crowds. In: _____. *Peer-to-Peer Systems: First InternationalWorkshop, IPTPS 2002 Cambridge, MA, USA, March 7–8, 2002 Revised Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. p. 203–213. ISBN 978-3-540-45748-0. Disponível em: <http://dx.doi.org/10.1007/3-540-45748-8_19>.

[63] ARI, I.; HONG, B.; MILLER, E. L.; BRANDT, S. A.; LONG, D. D. E. Managing flash crowds on the internet. In: *Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003. 11th IEEE/ACM International Symposium on*. [S.l.: s.n.], 2003. p. 246–249. ISSN 1526-7539.

[64] TSENG, Y.-C.; NI, S.-Y.; SHIH, E.-Y. Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network. *IEEE Transactions on Computers*, v. 52, n. 5, p. 545–557, May 2003. ISSN 0018-9340.

[65] SINGH, A. K.; MEENU. A survey on congestion control mechanisms in packet switch networks. In: *Computer Engineering and Applications (ICACEA), 2015 International Conference on Advances in*. [S.l.: s.n.], 2015. p. 902–906.

[66] ROUGHAN, M.; GRIFFIN, T.; MAO, Z. M.; GREENBERG, A.; FREEMAN, B. Ip forwarding anomalies and improving their detection using multiple data sources. In: *Proceedings of the ACM SIGCOMM Workshop on Network Troubleshooting: Research, Theory and Operations Practice Meet Malfunctioning Reality*. New York, NY, USA: ACM, 2004. (NetT '04), p. 307–312. ISBN 1-58113-942-X. Disponível em: <http://doi.acm.org/10.1145/1016687.1016703>.

[67] CERT.BR. *Cartilha de Segurança para Internet, versão 4.0*. Comitê Gestor da Internet no Brasil, 2012. ISBN 978-85-60062-54-6. Disponível em: <http://cartilha.cert.br/livro/>.

[68] SRIDHARAN, A.; YE, T.; BHATTACHARYYA, S. Connectionless port scan detection on the backbone. In: *2006 IEEE International Performance Computing and Communications Conference*. [S.l.: s.n.], 2006. p. 10 pp.–576. ISSN 1097-2641.

[69] SINGH, K. J.; DE, T. Mlp-ga based algorithm to detect application layer ddos attack. *Journal of Information Security and Applications*, Elsevier, v. 36, p. 145–153, 2017.

[70] KRUEGEL, C.; MUTZ, D.; ROBERTSON, W.; VALEUR, F. Bayesian event classification for intrusion detection. In: *19th Annual Computer Security Applications Conference, 2003. Proceedings.* [S.l.: s.n.], 2003. p. 14–23.

[71] ESKIN, E.; ARNOLD, A.; PRERAU, M.; PORTNOY, L.; STOLFO, S. A geometric framework for unsupervised anomaly detection. In: _____. *Applications of Data Mining in Computer Security.* Boston, MA: Springer US, 2002. p. 77–101. ISBN 978-1-4615-0953-0. Disponível em: <https://doi.org/10.1007/978-1-4615-0953-0_4>.

[72] BAHROLOLUM, M.; KHALEGHI, M. Anomaly intrusion detection system using gaussian mixture model. In: *2008 Third International Conference on Convergence and Hybrid Information Technology.* [S.l.: s.n.], 2008. v. 1, p. 1162–1167.

[73] ANDRYSIAK, T.; SAGANOWSKI, Ł.; MASZEWSKI, M. Network anomaly detection based on signal processing techniques. *Image Processing & Communications*, v. 18, n. 1, 2013.

[74] FERNANDES, G.; RODRIGUES, J. J.; PROENcA, M. L. Autonomous profile-based anomaly detection system using principal component analysis and flow analysis. *Appl. Soft Comput.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 34, n. C, p. 513–525, set. 2015. ISSN 1568-4946. Disponível em: <http://dx.doi.org/10.1016/j.asoc.2015.05.019>.

[75] de Assis, M. V. O.; Rodrigues, J. J. P. C.; Proença, M. L. A novel anomaly detection system based on seven-dimensional flow analysis. In: *2013 IEEE Global Communications Conference (GLOBECOM).* [S.l.: s.n.], 2013. p. 735–740. ISSN 1930-529X.

[76] HERNANDES, P. R. G.; CARVALHO, L. F.; PROENçA, M. L. Digital signature of network segment using flow analysis through genetic algorithm and aco metaheuristics. In: *2014 33rd International Conference of the Chilean Computer Science Society (SCCC).* [S.l.: s.n.], 2014. p. 92–97. ISSN 1522-4902.

[77] LIMA, M. F.; SAMPAIO, L. D. H.; ZARPELAO, B. B.; RODRIGUES, J. J. P. C.; ABRAO, T.; JR., M. L. P. 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.

[78] ASSIS, M. V. O. de; CARVALHO, L. F.; RODRIGUES, J. J. P. C.; PROENçA, M. L. 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.

[79] SALMEN, F.; JR, P. G. H.; CARVALHO, L.; JUNIOR, M. L. P. 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.

[80] JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: A review. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 31, n. 3, p. 264–323, set. 1999. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/331499.331504>.

[81] FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. d. L. F. d. *Inteligência artificial: uma abordagem de aprendizado de máquina*. [S.l.]: LTC, 2011.

[82] XU, R.; WUNSCH, D. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, v. 16, n. 3, p. 645–678, May 2005. ISSN 1045-9227.

[83] AGRAWAL, S.; AGRAWAL, J. Survey on anomaly detection using data mining techniques. *Procedia Computer Science*, v. 60, p. 708 – 713, 2015. ISSN 1877-0509. Knowledge-Based and Intelligent Information Engineering Systems 19th Annual Conference, KES-2015, Singapore, September 2015 Proceedings. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877050915023479>.

[84] HRUSCHKA, E. R.; EBECKEN, N. F. f. A genetic algorithm for cluster analysis. *Intell. Data Anal.*, IOS Press, Amsterdam, The Netherlands, The Netherlands, v. 7, n. 1, p. 15–25, jan. 2003. ISSN 1088-467X. Disponível em: <http://dl.acm.org/citation.cfm?id=1293920.1293922>.

[85] NANDA, S. J.; PANDA, G. A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary Computation*, v. 16, p. 1 – 18, 2014. ISSN 2210-6502. Disponível em: <http://www.sciencedirect.com/science/article/pii/S221065021300076X>.

[86] KENNEDY, J.; EBERHART, R. C. Particle swarm optimization. In: *Proceedings of the 1995 IEEE International Conference on Neural Networks*. Perth, Australia, IEEE Service Center, Piscataway, NJ: [s.n.], 1995. v. 4, p. 1942–1948.

[87] CLERC, M.; KENNEDY, J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 1, p. 58–73, Feb 2002. ISSN 1089-778X.

[88] HU, X.; SHI, Y.; EBERHART, R. Recent advances in particle swarm. In: *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*. [S.l.: s.n.], 2004. v. 1, p. 90–97 Vol.1.

[89] DORIGO, M.; MANIEZZO, V.; COLORNI, A. Ant system: Optimization by a colony of cooperating agents. *Trans. Sys. Man Cyber. Part B*, IEEE Press, Piscataway, NJ, USA, v. 26, n. 1, p. 29–41, fev. 1996. ISSN 1083-4419. Disponível em: <http://dx.doi.org/10.1109/3477.484436>.

[90] YANG, X.-S. A new metaheuristic bat-inspired algorithm. In: _____. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 65–74. ISBN 978-3-642-12538-6. Disponível em: <https://doi.org/10.1007/978-3-642-12538-6_6>.

[91] YANG, X.-S. *Nature-Inspired Metaheuristic Algorithms*. [S.l.]: Luniver Press, 2008. ISBN 1905986106, 9781905986101.

[92] ABDEL-KADER, R.; EL-TARABILY, M.; MARIE, M.; ABDEL-AZEEM, G. A PSO-Based Subtractive Data Clustering Algorithm. *International Journal of Research in Computer Science*, v. 3, p. 1–9, 2013.

[93] JR., M. L. P.; ZARPELãO, B. B.; MENDES, L. de S. Anomaly detection using digital signature of network segment aiming to help network management. *Journal of Communication and Information Systems*, v. 23, n. 1, Jun. 2015. Disponível em: <https://jcis.sbrt.org.br/jcis/article/view/269>.

[94] XU, H.; YU, Z.; QIAN, C.; LI, X.; LIU, Z. Minimizing flow statistics collection cost of sdn using wildcard requests. In: *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*. [S.l.: s.n.], 2017. p. 1–9.

[95] SU, Z.; WANG, T.; XIA, Y.; HAMDI, M. Cemon: A cost-effective flow monitoring system in software defined networks. *Computer Networks*, v. 92, p. 101 – 115, 2015. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128615003291>.

[96] SU, Z.; WANG, T.; XIA, Y.; HAMDI, M. Flowcover: Low-cost flow monitoring scheme in software defined networks. In: *2014 IEEE Global Communications Conference*. [S.l.: s.n.], 2014. p. 1956–1961. ISSN 1930-529X.

[97] ONF. "openflow switch specification 1.5.1. 2015. Disponível em: <https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>.

[98] GIOTIS, K.; ARGYROPOULOS, C.; ANDROULIDAKIS, G.; KALOGERAS, D.; MAGLARIS, V. Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on sdn environments. *Computer Networks*, v. 62, p. 122 – 136, 2014. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128613004003>.

[99] PENA, E. H.; CARVALHO, L. F.; JR., S. B.; RODRIGUES, J. J.; JR., M. L. P. Anomaly detection using the correlational paraconsistent machine with digital signatures of network segment. *Information Sciences*, v. 420, p. 313 – 328, 2017. ISSN 0020-0255. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0020025517309131>.

[100]ASSIS, M. V. de; RODRIGUES, J. J.; PROENçA, M. L. A seven-dimensional flow analysis to help autonomous network management. *Information Sciences*, v. 278, p. 900 – 913, 2014. ISSN 0020-0255. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0020025514003995>.

[101]ASSIS, M. V. O. D.; NOVAES, M. P.; ZERBINI, C. B.; CARVALHO, L. F.; ABRãAO, T.; PROENçA, M. L. Fast defense system against attacks in software defined networks. *IEEE Access*, v. 6, p. 69620–69639, 2018. ISSN 2169-3536. DOI=10.1109/ACCESS.2018.2878576.

[102]SHANNON, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, v. 27, n. 3, p. 379–423, July 1948. ISSN 0005-8580.

[103]AMARAL, A. A.; MENDES, L. de S.; ZARPELãO, B. B.; JUNIOR, M. L. P. Deep ip flow inspection to detect beyond network anomalies. *Computer Communications*, v. 98, p. 80 – 96, 2017. ISSN 0140-3664. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0140366416306612>.

[104]ZHOU, Y.; WANG, N.; XIANG, W. Clustering hierarchy protocol in wireless sensor networks using an improved pso algorithm. *IEEE Access*, v. 5, p. 2241–2253, 2017. ISSN 2169-3536.

[105]REJINAPARVIN, J.; VASANTHANAYAKI, C. Particle swarm optimization-based clustering by preventing residual nodes in wireless sensor networks. *IEEE Sensors Journal*, v. 15, n. 8, p. 4264–4274, Aug 2015. ISSN 1530-437X.

[106]YANG, S.; LI, C. A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments. *IEEE Transactions on Evolutionary Computation*, v. 14, n. 6, p. 959–974, Dec 2010. ISSN 1089-778X.

[107]SHISHIRA, S. R.; KANDASAMY, A.; CHANDRASEKARAN, K. Survey on meta heuristic optimization techniques in cloud computing. In: *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. [S.l.: s.n.], 2016. p. 1434–1440.

[108]ABDMOULEH, Z.; GASTLI, A.; BEN-BRAHIM, L.; HAOUARI, M.; AL-EMADI, N. A. Review of optimization techniques applied for the integration of distributed generation from renewable energy sources. *Renewable Energy*, v. 113, p. 266 – 280, 2017. ISSN 0960-1481. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0960148117304822>.

[109]BRATTON, D.; KENNEDY, J. Defining a standard for particle swarm optimization Citado por mí. *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, n. Sis, p. 120–127, 2007.

[110]ROUSSEEUW, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, v. 20, p. 53–65, 1987. ISSN 0377-0427. Disponível em: <http://www.sciencedirect.com/science/article/pii/0377042787901257>.

[111]AMIDAN, B. G.; FERRYMAN, T. A.; COOLEY, S. K. Data outlier detection using the chebyshev theorem. *IEEE Aerospace Conference Proceedings*, v. 2005, p. 3–8, 2005. ISSN 1095323X.

[112]TAYLOR, C.; ALVES-FOSS, J. An empirical analysis of nate: Network analysis of anomalous traffic events. In: *Proceedings of the 2002 Workshop on New Security Paradigms*. New York, NY, USA: ACM, 2002. (NSPW '02), p. 18–26. ISBN 1-58113-598-X. Disponível em: <http://doi.acm.org/10.1145/844102.844106>.

[113]KRUEGEL, C.; VIGNA, G. Anomaly detection of web-based attacks. In: *Proceedings of the 10th ACM Conference on Computer and Communications Security*. New York, NY, USA: ACM, 2003. (CCS '03), p. 251–261. ISBN 1-58113-738-9. Disponível em: <http://doi.acm.org/10.1145/948109.948144>.

[114]VILLAMARIN-SALOMON, R.; BRUSTOLONI, J. C. Identifying botnets using anomaly detection techniques applied to dns traffic. In: *2008 5th IEEE Consumer Communications and Networking Conference*. [S.l.: s.n.], 2008. p. 476–481. ISSN 2331-9852.

[115]SAKIB, M. N.; HUANG, C. Using anomaly detection based techniques to detect http-based botnet c amp;c traffic. In: *2016 IEEE International Conference on Communications (ICC)*. [S.l.: s.n.], 2016. p. 1–6. ISSN 1938-1883.

[116]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>.

[117]CHANG, S.; QIU, X.; GAO, Z.; QI, F.; LIU, K. A flow-based anomaly detection method using entropy and multiple traffic features. In: *2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*. [S.l.: s.n.], 2010. p. 223–227.

[118]CHANG, S.; QIU, X.; GAO, Z.; LIU, K.; QI, F. A flow-based anomaly detection method using sketch and combinations of traffic features. In: *2010 International Conference on Network and Service Management*. [S.l.: s.n.], 2010. p. 302–305. ISSN 2165-9605.

[119]SAHAY, R.; BLANC, G.; ZHANG, Z.; TOUMI, K.; DEBAR, H. Adaptive policy-driven attack mitigation in sdn. In: *Proceedings of the 1st International Workshop on Security and Dependability of Multi-Domain Infrastructures*. New York, NY, USA: ACM, 2017. (XDOMO'17), p. 4:1–4:6. ISBN 978-1-4503-4937-6. Disponível em: <http://doi.acm.org/10.1145/3071064.3071068>.

[120]TEAM, M. *Mininet Overview*. 2018. Acessado em: http://mininet.org/overview/. Disponível em: <http://mininet.org/overview/>.

[121]BIONDI, P.; COMMUNITY the S. *Scapy*. 2018. Acessado em: http://www.secdev.org/projects/scapy/. Disponível em: <http://www.secdev.org/projects/scapy/>.

[122]SANFILIPPO, S. *hping3*. 2018. Acessado em: http://www.hping.org/. Disponível em: <http://www.hping.org/>.

[123]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>.

[124]FAWCETT, T. An introduction to roc analysis. *Pattern Recogn. Lett.*, Elsevier Science Inc., New York, NY, USA, v. 27, n. 8, p. 861–874, jun. 2006. ISSN 0167-8655. Disponível em: <http://dx.doi.org/10.1016/j.patrec.2005.10.010>.

[125]FAWCETT, T. An introduction to roc analysis. *Pattern Recognition Letters*, v. 27, n. 8, p. 861 – 874, 2006. ISSN 0167-8655. ROC Analysis in Pattern Recognition. Disponível em: <http://www.sciencedirect.com/science/article/pii/S016786550500303X>.

[126]SUN, X.; YANG, Z. Generalized mcnemar's test for homogeneity of the marginal distributions. In: *SAS Global forum*. [s.n.], 2008. v. 382, p. 1–10. Disponível em: <http://www2.sas.com/proceedings/forum2008/382-2008.pdf>.

[127]HAUTAMAKI, V.; KARKKAINEN, I.; FRANTI, P. Outlier detection using k-nearest neighbour graph. In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* [S.l.: s.n.], 2004. v. 3, p. 430–433 Vol.3. ISSN 1051-4651.

[128]MüNZ, G.; LI, S.; CARLE, G. Traffic anomaly detection using kmeans clustering. In: *In GI/ITG Workshop MMBnet.* [S.l.: s.n.], 2007.

[129]VIJAYASARATHY, R.; RAGHAVAN, S. V.; RAVINDRAN, B. A system approach to network modeling for ddos detection using a naìve bayesian classifier. In: *2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011).* [S.l.: s.n.], 2011. p. 1–10. ISSN 2155-2487.

[130]OKE, G.; LOUKAS, G.; GELENBE, E. Detecting denial of service attacks with bayesian classifiers and the random neural network. In: *2007 IEEE International Fuzzy Systems Conference.* [S.l.: s.n.], 2007. p. 1–6. ISSN 1098-7584.

[131]NANDA, S.; ZAFARI, F.; DECUSATIS, C.; WEDAA, E.; YANG, B. Predicting network attack patterns in sdn using machine learning approach. *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, p. 167–172, 2016.

[132]CUI, H.; ZHU, Y.; YAO, Y.; YUFENG, L.; LIU, Y. Design of intelligent capabilities in sdn. In: *2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace Electronic Systems (VITAE).* [S.l.: s.n.], 2014. p. 1–5.

TRABALHOS PUBLICADOS PELO AUTOR

Trabalhos publicados:

1. ASSIS, M. V.; NOVAES, M. P.; ZERBINI, C. B.; CARVALHO, L. F.; ABRÃO, T.; PROENÇA, M. L. Fast defense system against attacks in software defined networks.IEEE Access, vol. 6, pp. 69620-69639, 2018. DOI: 10.1109/ACCESS.2018.2878576. (**Qualis B3**).