Universidade Estadual de Londrina

GUSTAVO FRIGO SCARANTI

SISTEMA IMUNOLÓGICO ARTIFICIAL E LÓGICA FUZZY PARA A DETECÇÃO DE ATAQUES EM REDES DEFINIDAS POR SOFTWARE

Londrina
2021

GUSTAVO FRIGO SCARANTI

SISTEMA IMUNOLÓGICO ARTIFICIAL E LÓGICA FUZZY PARA A DETECÇÃO DE ATAQUES 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 Junior
Coorientador: Prof. Dr. Luiz Fernando Carvalho

Londrina
2021

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

S285 Scaranti, Gustavo Frigo.
SISTEMA IMUNOLÓGICO ARTIFICIAL E LÓGICA FUZZY PARA A DETECÇÃO DE ATAQUES EM REDES DEFINIDAS POR SOFTWARE / Gustavo Frigo Scaranti. - Londrina, 2021.
77 f.

Orientador: Mario Lemes Proença Junior.
Coorientador: Luiz Fernando Carvalho.
Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Ciência da Computação, 2021.
Inclui bibliografia.

1. Seleção Negativa - Tese. 2. Redes Definidas por Software - Tese. 3. Detecção de Anomalias - Tese. 4. Sistema de Detecção de Intrusão - Tese. I. Proença Junior, Mario Lemes. II. Carvalho, Luiz Fernando. III. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. IV. Título.

CDU 519

GUSTAVO FRIGO SCARANTI

SISTEMA IMUNOLÓGICO ARTIFICIAL E LÓGICA FUZZY PARA A DETECÇÃO DE ATAQUES 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 Junior
Universidade Estadual de Londrina – UEL

Prof. Dr. Alan Salvany Felinto
Universidade Estadual de Londrina – UEL

Prof. Dr. Wesley Attrot
Universidade Estadual de Londrina – UEL

Prof. Dr. Marcos Vinicius Oliveira de Assis
Universidade Federal do Paraná – UFPR

Londrina, 08 de junho de 2021.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por ter me dado capacidade de chegar até essa etapa.

Agradeço ao Prof. Dr. Mario Lemes Proença Junior, por ter sempre me apoiado e me orientado, sempre incentivando na busca da excelência e dos melhores resultados. Agradeço também ao Prof. Dr. Luiz Fernando Carvalho, por ter me auxiliado juntamente com o Prof. Mario em diversas etapas do trabalho, até chegarmos nesta dissertação e também ao Prof. Dr. Sylvio Barbon Junior pelas diversas orientações e revisões em várias etapas desta dissertação.

Também gostaria de agradecer meus pais, Carlos e Fátima, que me incentivaram desde o início, até mesmo nos momentos mais difíceis desta caminhada. Agradeço também a minha noiva Isabelle, que sempre me apoiou e motivou desde o início do nosso relacionamento.

"Dez vezes irão aos pobres, dez vezes encontrarão a Deus"
(São Vicente de Paulo)

SCARANTI, G. F. **Sistema Imunológico Artificial e Lógica Fuzzy para a Detecção de Ataques em Redes Definidas por Software.** 2021. 77 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2021.

RESUMO

As redes definidas por software (do inglês, SDN - Software-defined Networks) é uma arquitetura que implementa o desacoplamento dos planos de controle e plano de dados para tornar as redes mais flexíveis, programáveis e centralmente controladas. Apesar desta arquitetura possibilitar vantagens no gerenciamento, as redes ainda são susceptíveis a problemas de segurança, além de que os métodos de detecção tradicionais podem não ser suficientes para conter tais ameaças. Neste trabalho, apresentamos um sistema de detecção de intrusão que utiliza sistemas imunológicos artificiais para realizar a detecção, denominado de AIS-IDS. Esse sistema bioinspirado pode detectar variações no comportamento da rede e identificar ataques sem conhecimento prévio sobre eles. Junto com o AIS, a lógica fuzzy é aplicada na detecção, com o objetivo de minimizar a incerteza quando não há um limiar claro entre o comportamento do tráfego anômalo e legítimo. Para avaliar o AIS-IDS foram realizados testes em uma rede emulada, a qual foi acometida por ataques DDoS e port scan. Os ataques foram gerados com diferentes configurações, modificandose a duração, intensidade e sobreposição. Ainda durante a investigação da eficiência do sistema de detecção proposto, foi utilizada a base de dados pública CICDDoS2019 que contem tráfego coletado a partir da incidência de ataques em uma rede real.

Palavras-chave: seleção negativa; redes definidas por software; detecção de anomalias; sistema de detecção de intrusão.

SCARANTI, G. F. **Artificial Immune Systems and Fuzzy Logic to Detect Attacks in Software-Defined Networks.** 2021. 77 p. Master's Thesis (Master in Science in Computer Science) – State University of Londrina, Londrina, 2021.

ABSTRACT

Software-defined Networks (SDN) is an architecture that implements the decoupling of control plans and data plans to make networks more flexible, programmable, and centrally controlled. Although this architecture allows management advantages, networks are still susceptible to security problems. Besides, traditional detection methods may not be sufficient to contain such threats. In this work, we present an intrusion detection system that uses artificial immune systems to perform the detection, called AIS-IDS. This bioinspired system can detect variations in the network's behavior and identify attacks without prior knowledge about them. Along with AIS, the fuzzy logic is applied in detection to minimize uncertainty when there is no clear threshold between anomalous and legitimate traffic behavior. The AIS-IDS were tested on an emulated network affected by DDoS and port scan attacks. The attacks were generated with different configurations, changing the duration, intensity, and overlap. Also, during the investigation of the proposed detection system's efficiency, the CICDDoS2019 database was used, containing traffic collected from the incidence of attacks on a real network.

Keywords: Negative Selection; Software-defined Networking; Anomaly Detection; Intrusion Detection System.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACC	Accuracy
AIS	Artificial Immune System
AIS-IDS	Intrusion Detection System using Artificial Immune System
CIC	Canadian Institute for Cybersecurity
CNN	Convolutional Neural network
DDoS	Distributed Denial of Service
DoS	Denial of Service
FPR	False Positive Ratio
HIDS	Host-based IDS
Gbps	Gigabits por segundo
HIS	Human Immune System
HS	Half-Space-Trees
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IoT	Internet of Things
IP	Internet Protocol
kNN	K-Nearest Neighbors
LGP	Linear Genetic Programming
LOA	Lion Optimization Algorithm
LOF	Local Outlier Factor
MARS	Multivariate Adaptive Regression Splines
MLP	Multilayer Perceptron
MLR	Multiple Linear Regression
NB	Naive Bayes

NFV	Network Functions Virtualization
NIDS	Network-based IDS
NSA	Negative Selection Algorithm
ONF	Open Network Foundation
PREC	Precision
PSO	Particle Swarm Optimization
RAM	Random Access Memory
REC	Recall
RF	Random Forest
R2L	Remote to User
SDN	Software-defined Networks
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
U2R	User to Root
VLAN	Virtual Local Area Network

LISTA DE SÍMBOLOS

δ	Pontuação de similaridade
Φ	Um detector
Ω	Coleção de detectores
f	Atributo
k	Valor de similaridade mínima
n	Número de detectores desejado
ζ	Valor de saída da função Fuzzy
σ	Desvio padrão da similaridade do atributo f
Γ	Valor de corte
w	Tamanho da janela deslizante

SUMÁRIO

1 INTRODUÇÃO

As redes de computadores estão cada vez mais presentes em nossas vidas. Novas arquiteturas e dispositivos tornaram-se mais comuns, como computação em nuvem, virtualização e Internet das Coisas (do inglês, IoT - *Internet of Things*). A adoção dessas novas tecnologias também proporcionou o surgimento de novos problemas de segurança. Desse modo, faz-se necessária a criação de métodos capazes de lidar com as ameaças emergentes a fim de tornar as redes mais confiáveis e resilientes [2, 3, 4].

Assim que uma nova tecnologia é adotada, inúmeras vulnerabilidades são descobertas e ameaçam a disponibilidade, a confiabilidade e a integridade dos dados trafegados [5, 6]. Os métodos tradicionais tornam o gerenciamento de redes um serviço árduo e pouco viável. Com isso, há a necessidade de meios mais eficazes de gerenciamento das redes, que provenham flexibilidade aos administradores e também tornem a rede resiliente às mudanças que possam ocorrer [7, 8, 9].

As redes definidas por software (do inglês, SDN - *Software-defined Networks*) é um paradigma inovador que possibilita o gerenciamento de diferentes tipos e tamanhos de redes por meio da perspectiva de software [10]. Enquanto no modelo tradicional o plano de controle e o plano de dados são intrinsecamente ligados em cada equipamento, nas redes SDN o plano de controle é desacoplado do plano de dados desses equipamentos. Por consequência, os equipamentos da rede se tornam simples encaminhadores dos pacotes e toda a decisão de encaminhamento é realizada no plano de controle [11].

Em uma rede tradicional, cada equipamento da rede possui um software, regras de segurança, estratégias de quedas de link e também mecanismos de encaminhamento distintos. Desse modo, o gerenciamento, a manutenção e a atualização de cada equipamento acaba acontecendo de maneira individual [12, 13]. As SDN possuem um plano de controle centralizado, chamado de controlador, o qual permite realizar análises de toda a rede e também gerenciá-la de maneira unificada [14, 15, 16].

Com a capacidade de serem gerenciadas via software, as redes SDN viabilizam novas possibilidades, novos serviços e inúmeras customizações, como balanceamento de carga, ajuste da granularidade do monitoramento da rede e agilidade no provisionamento de dispositivos de rede físicos e virtuais [8, 17]. Entretanto, com o surgimento dessas novas tecnologias, também podem ser descobertas novas formas de exploração de vulnerabilidades nas redes. O termo anomalia comumente é utilizado para se referir a um comportamento diferente do habitual que possa oferecer risco ao funcionamento de um serviço como um todo ou impactar sua qualidade [18, 19, 20].

Detectar essas anomalias em redes com mudanças constantes vem ganhando des-

taque nas comunidades científica e industrial [21, 22, 23, 24]. Além disso, é fundamental ter uma rápida identificação e mitigação das ameaças, sempre visando disponibilidade, confiabilidade e resiliência no fornecimento de serviços de rede [25, 26, 27]. O reconhecimento de tais ameaças é um desafio aos administradores de redes, ainda mais com a grande quantidade de dados sensíveis trafegados sem criptografia em equipamentos limitados e utilizando equipamentos e protocolos heterogêneos [28, 29]. Uma das principais características das redes SDN, a sua centralização, também pode ser um alvo conveniente.

Um ataque de Negação de Serviço Distribuído (do inglês, DDoS - *Distributed Denial of Service*), direcionado ao controlador, pode afetar o funcionamento da rede como um todo [11, 30, 31]. Caso seja direcionado a outro servidor, poderá afetar a disponibilidade de serviços aos usuários. Também um ataque de sondagem, como o *port scan*, pode revelar informações sobre um servidor e auxiliar um ataque mais elaborado. Considerando todas essas ameaças, os administradores das redes utilizam mecanismos de segurança, como sistemas de detecção de intrusão (do inglês, IDS - *Intrusion Detection System*), para analisar o tráfego da rede, identificar e mitigar possíveis ameaças [32, 33].

Neste trabalho é apresentado um sistema chamado AIS-IDS (do inglês, *Intrusion Detection System using Artificial Immune System*), que tem o objetivo de identificar as intrusões por meio de anomalias do tipo DDoS e *port scan*. O IDS utiliza o sistema imunológico artificial (do inglês, AIS - *Artificial Immune System*) juntamente com lógica *fuzzy* para automatizar a detecção e mitigação das anomalias localizadas na rede.

O AIS-IDS é executado no controlador da rede SDN e possui três módulos com funções específicas: o primeiro é responsável pela coleta dos fluxos, o segundo pela detecção e o último responsável pela mitigação das anomalias. O primeiro módulo coleta os fluxos IP a cada segundo e extrai os atributos que representam o comportamento da rede naquele momento. O segundo módulo diz respeito à detecção, este que utiliza o AIS, uma classe de algoritmos inspirados no funcionamento do sistema de defesa do corpo humano. Do mesmo modo que o sistema imune humano identifica e reage a organismos desconhecidos dentro do corpo humano, o AIS-IDS também poderá identificar e reagir a padrões que diferem do apresentado na fase de treinamento. O AIS-IDS reconhece as anomalias na rede apenas analisando o comportamento padrão do tráfego. Juntamente com o AIS, o Sistema de inferência *Fuzzy*, é usado para resolver o grau de incerteza no processo de detecção de anomalias, como é o caso da análise do tráfego de rede. Por fim, o módulo mitigador faz-se responsável pelo reconhecimento do tipo da anomalia e identificação dos endereços IP e portas envolvidos na comunicação suspeita. Após a identificação, essas informações são usadas em estratégias de mitigação para bloquear o tráfego malicioso.

As principais contribuições deste trabalho são:

- Desenvolvimento de um IDS bioinspirado para redes definidas por softwares que

dispensa o uso de dados rotulados para o seu treinamento;

- Implementação e uso de um sistema de inferência *fuzzy* para aprimorar a detecção de ataques DDoS e *port scan*;
- Monitoramento da rede e mitigação das anomalias do tráfego em tempo quase real;
- Testes realizados com tráfego real proveniente de base pública e tráfego emulado com variadas configurações de intensidade, duração e sobreposição de ataques.

O restante do trabalho está dividido nos seguintes capítulos: no Capítulo 2 é exposta a fundamentação teórica. No Capítulo 3 são apresentados os trabalhos relacionados. No Capítulo 4 é apresentado o modelo proposto. O Capítulo 5 apresenta os testes executados e seus respectivos resultados e, por fim, o Capítulo 6 apresenta as conclusões.

2 FUNDAMENTAÇÃO TEÓRICA

A evolução tecnológica vem criando novos hábitos, serviços e negócios. Atualmente as redes de computadores são um dos principais meios de comunicação, tornando-se cada vez mais indispensáveis. Com isso, faz-se necessário novos meios de administração, configuração e proteção das redes [11]. Este capítulo apresenta os conceitos que abrangem a detecção de anomalias, bem como os fundamentos de redes SDN e a teoria que embasa o sistema de detecção proposto.

2.1 Detecção de anomalias

Inúmeros autores já definiram o conceito do termo anomalia. Barnett e Lewis [34] a elucidaram como um sub-conjunto de dados que parece estar inconsistente em relação ao restante do conjunto. Chandola et al. [35] explicaram como sendo um padrão de dados que não está coerente com a noção de comportamento normal já definido. Segundo Lakhina et al. [36], anomalias são variações incomuns e significativas nos volumes de tráfego da rede. Por fim, Hoque et al. [37] deliberaram como padrões que não são conformes com a noção de comportamento normal previamente estipulado. De modo geral, os autores apresentaram as anomalias como padrões que se diferenciam dos modelos conhecidos como um comportamento normal. Portanto, definir o que é um comportamento normal é um dos principais pilares para a detecção de anomalias.

O ato de caracterizar um padrão detectado como anomalia ou normal é desafiador. Menor custo computacional, maior velocidade na detecção e menor taxa de falso positivo são alguns dos objetivos desta área [38]. Com isso, pesquisadores estão sempre propondo novas técnicas e algoritmos a fim de melhorar cada vez mais a detecção e minimizar os efeitos negativos das ameaças nas redes.

2.1.1 Tipos de Anomalias

É fundamental conhecer algumas características dos ataques a fim de melhorar a detecção destas ameaças. As anomalias em redes são classificadas basicamente em duas categorias: baseada em sua natureza ou pautada em seu aspecto causal [39].

2.1.1.1 Anomalias categorizadas por sua natureza

Dentro das anomalias categorizadas por sua natureza, podem ser classificadas em três grupos: anomalias pontuais, anomalias contextuais e anomalias coletivas. As anomalias pontuais são eventos isolados que ocorrem uma única vez, como, o acesso de um usuário a um servidor restrito. Se esse evento não estiver vinculado a nenhum outro con-

texto específico ou até a outros acontecimentos atípicos, o mesmo é considerado uma anomalia pontual [40].

Já as anomalias contextuais são eventos que apenas são considerados anômalos dependendo do contexto no qual estão inseridos. Um contexto pode ser interpretado como dois conjuntos de atributos que são definidos durante a formulação do problema. Os atributos subdividem-se em contextuais ou comportamentais. Atributos contextuais informam o contexto inserido, como o horário de um determinado evento ou até a localização por meio das coordenadas geográficas. Por sua vez, os atributos comportamentais são aqueles que indicam se um evento é considerado anômalo ou não dependendo do contexto [35, 41]. Para exemplificar, considere um histórico da taxa de bits média a cada minuto dos últimos dias (atributo contextual) e também que diariamente é realizado o backup de um servidor durante a madrugada (atributo comportamental). Analisando o tráfego de dados durante a madrugada será possível notar um aumento considerável da taxa de bits, porém esse aumento é conhecido pelo fato de estar ocorrendo um backup. Contudo, se um aumento na taxa ocorrer no período da tarde e nenhum outro atributo comportamental for definido, esse evento será considerado uma anomalia contextual.

Por fim, as anomalias coletivas são coleções de amostras que possuem um comportamento anômalo se comparado ao conjunto de dados. Uma anomalia pontual, caso ocorra continuamente por um longo período de tempo, é considerada uma anomalia coletiva. Também é válido considerar a ocorrência de várias ações em uma determinada sequência, o que pode caracterizar anomalia coletiva; entretanto, caso as ações estejam em outra sequência, já não seriam caracterizadas como anomalia [42].

2.1.1.2 Anomalias categorizadas por seu aspecto causal

Outra categoria de anomalias são as baseadas em seu aspecto causal, classificadas em relação a sua motivação ou circunstância em que ocorrem. De acordo com Barford et al. [43] e Marnerides et al. [44], as anomalias nem sempre têm o objetivo de inutilizar sistemas ou roubar informações, elas também podem ser consideradas como eventos de comportamento não usual. Para facilitar, essas anomalias podem ser classificadas em quatro categorias: anomalias operacionais, anomalias geradas pelo uso legítimo dos recursos, anomalias de medição do tráfego e anomalias de abuso de rede.

As anomalias operacionais normalmente estão atreladas à má configuração ou falha dos equipamentos. Alguns exemplos são: travamento de servidores, falta de energia, congestionamento de tráfego da rede, má configuração de serviços e equipamentos, entre outros [45]. Normalmente essas anomalias são detectadas por mudanças repentinas no volume do tráfego da rede [46].

O segundo tipo de anomalias são as geradas pelo uso legítimo dos recursos, mas que distorcem o comportamento padrão. Uma das anomalias mais conhecidas dessa categoria

são os *flash crowd*, eventos nos quais são detectados grandes volumes de tráfego, porém que não se configuram como ataques, mas sim uma grande quantidade de usuários tentando acessar um determinado serviço ou site. Como exemplo, pode-se citar sites de comércio eletrônico em datas especiais como a *black friday* ou a liberação da lista de aprovados em um concurso público, entre outros [47].

A terceira categoria é constituída pelas anomalias de medição de tráfego. Este tipo de anomalia também não tem ligação direta com um evento malicioso. Normalmente estão relacionadas a problemas na infraestrutura de coleta ou o próprio método utilizado para a coleta das informações. Alguns exemplos de anomalias deste tipo são: a sobrecarga dos equipamentos e a consequente perda dos dados coletados ou até a falha na comunicação no momento da coleta dos dados.

A última categoria tem por base as anomalias de abuso de rede, também conhecidas como ataques de rede. Nessa categoria existem vários tipos de ataques cujo o objetivo é espionar, interromper ou até mesmo degradar informações dos serviços ofertados pela rede.

2.2 Ataques de Rede

Dentre os ataques de rede em que o objetivo é a espionagem, os principais são o *port scan* e o *network scan*. O primeiro tem por objetivo localizar portas e serviços ofertados em um host. Já o *network scan* busca localizar dispositivos na rede e coletar informações, como o sistema operacional do equipamento e os serviços que ele está oferecendo à rede [48]. Outro tipo de ataque conhecido são os ataques de negação de serviço. O seu principal objetivo não é o roubo de informações, mas sim degradar os serviços e a infraestrutura da rede [49].

2.2.1 Ataques *Port scan*

Um ataque *port scan* consiste em uma fonte atacante enviar incontáveis requisições para um alvo em diferentes portas de destino com o objetivo de receber respostas a essas requisições. Ao receber a resposta é possível descobrir qual porta está aberta, bem como qual o serviço de rede está sendo ofertado pelo alvo. Geralmente, logo após um *port scan* ocorre um ataque mais complexo que utilizará as informações obtidas da sondagem de portas abertas. A Figura 1 exemplifica o esquema de um ataque *port scan*.

2.2.2 Ataques de Negação de Serviço

Os dois tipos de ataque de negação de serviço mais conhecidos são os ataques *Denial of Service* (DoS) e a sua versão distribuída, o DDoS. Esses ataques visam esgotar todos os recursos computacionais de um servidor e, deste modo, interromper serviços

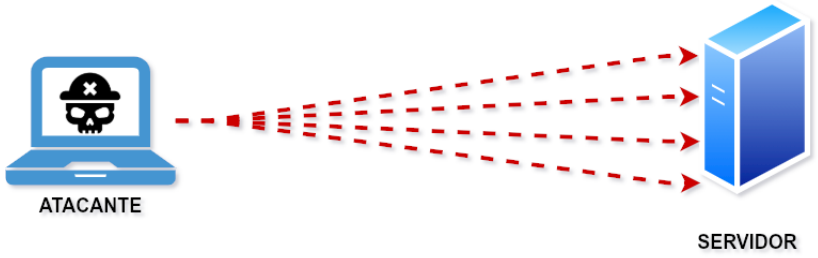


Figura 1 – Esquema de um ataque *port scan*

ofertados, causando principalmente prejuízos financeiros [9, 15, 39]. Normalmente esses ataques são direcionados a empresas que prestam serviços pela internet, contudo, as vítimas normalmente são usuários que utilizam serviços ofertados por bancos, comércios, empresas e governos.

Ataques DoS são realizados por uma única fonte atacante, que enviará inúmeras requisições a um alvo e porta específica. Já os ataques DDoS são de forma distribuída, em que várias fontes atacam um único alvo e porta específica. Para este ataque ser realizado é necessário duas etapas. A primeira é a infecção de inúmeras máquinas da internet para que elas contribuam com o ataque. Para isso são instalados softwares especializados em cada um desses hosts, comumente chamados de zumbis. Na segunda fase o atacante envia uma ordem para que todas as máquinas zumbis executem um ataque simultâneo à maquina alvo. A Figura 2 demonstra o funcionamento de um ataque DDoS.

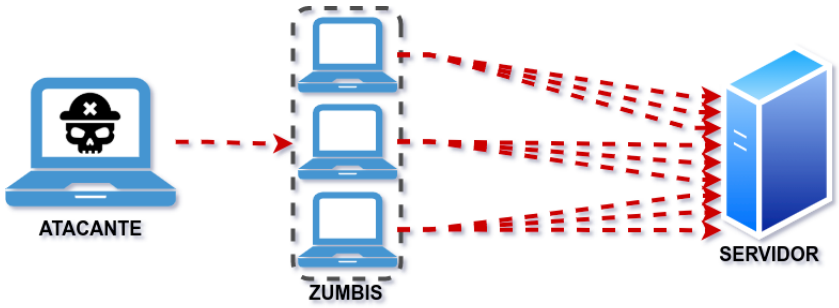


Figura 2 – Esquema de um ataque DDoS

2.3 Sistema de Detecção de Intrusão

Um sistema de detecção de intrusão, ou *Intrusion detection system* (IDS), é uma das ferramentas utilizadas para monitoramento, identificação e proteção de um host ou rede. O objetivo de um IDS é garantir segurança, confiabilidade e disponibilidade do serviço ou rede monitorada. Inúmeras vezes o termo intrusão é atrelado aos IDS, entretanto, nem sempre o IDS é projetado para reconhecer intrusões, mas sim para detectar comportamentos anômalos [35, 50]. Apesar da natureza de uma anomalia, maliciosa ou não, é importante que o IDS tenha informações sobre elas a fim de que seja possível aplicar

técnicas de mitigação eficientes para cada anomalia ou até métodos de detecção mais efetivos [51].

Os IDS podem ser classificados em três tipos: baseados em redes, baseados em host e híbridos. O IDS fundamentado em redes é conhecido pela sigla NIDS (do inglês, *Network-based IDS*), sendo responsável por reconhecer atividades anômalas na rede onde está implantado. Em um NIDS são instalados sensores pela rede que coletam informações sobre o comportamento do tráfego da rede para localizar as possíveis anomalias. Uma das principais vantagens deste tipo IDS é a capacidade de monitorar grandes redes apenas com alguns sensores bem posicionados.

Já o segundo tipo de IDS são os baseados em host (do inglês, HIDS - *Host-based IDS*). Este modelo monitora os logs do sistema ou aplicações do host, as tentativas de ações sem privilégios, entre outros [40]. Uma das principais vantagens do HIDS frente ao NIDS é a capacidade de lidar com ataques que utilizam criptografia. Como ele está localizado dentro de um host, consegue monitorar possíveis anomalias após o tráfego já ter sido descriptografado. Entretanto, é mais custosa que a NIDS, pois precisa estar em vários computadores da rede para uma proteção mais robusta.

Uma alternativa equilibrada entre essas duas são os IDS híbridos, que combinam ambas modelos para a proteção da rede. Os IDS híbridos utilizam dados fornecidos pelos hosts e também dados coletados pelos sensores da rede, agregando, deste modo, as vantagens de cada tipo e também minimizando as desvantagens que cada um possui de maneira individual [52].

Os IDS também podem ser classificados em relação ao método de detecção utilizado, são dois tipos: baseado em assinatura ou baseado em anomalias [42, 53]. Os IDS baseados em assinatura possuem grandes bases de dados onde são armazenadas as assinaturas de cada anomalia; caso seja encontrada a mesma assinatura no processo de monitoramento, é apontada a anomalia [54]. Por outro lado, os IDS baseados em anomalias criam um perfil de comportamento sem anomalias, caso esse padrão se altere entre o perfil criado e o monitorado no momento, é detectada a anomalia [55].

Ambos os métodos possuem vantagens e desvantagens. O principal benefício da detecção que utiliza assinaturas é a baixa taxa de falso positivo em ataques já catalogados, contudo, caso o ataque seja desconhecido, pode ser que ele não seja detectado. Já o modelo baseado em anomalias tem a vantagem de conseguir detectar ataques sem a necessidade de conhecê-los, no entanto, manter o perfil de normalidade atualizado é um desafio, pois a rede está em constantes mudanças e qualquer alteração pode afetar seu comportamento da rede.

2.4 Sistemas Imunológico Artificiais

O sistema imunológico humano (do inglês, HIS - *Human Immune System*) é capaz de proteger o corpo humano de inúmeros organismos estranhos e com potencial prejudicial. Desse modo, inúmeros pesquisadores têm apresentado algoritmos que se inspiram no mesmo funcionamento do sistema de defesa do corpo humano para detectar padrões anômalos [56, 57, 58]. Esses algoritmos inspirados no funcionamento do HIS são chamados de sistema imunológico artificial (do inglês, AIS - *Artificial Immune System*).

Realizando um mapeamento entre o HIS e o AIS, enquanto o HIS é um mecanismo de defesa do corpo humano, o AIS pode ser implementado em inúmeros sistemas. Os linfócitos do HIS podem ser entendidos como os detectores gerados pelo AIS. Quando um linfócito é ativado, são criados anticorpos para a defesa do corpo humano. Já no AIS, quando o detector é ativado, produz um alarme a fim de informar a detecção de uma anomalia.

Os antígenos no HIS são substâncias estranhas com capacidade de produzir reações dos linfócitos e a produção dos anticorpos. No AIS, um antígeno é qualquer amostra anômala que tem o potencial de desencadear a ativação de um detector e iniciar o processo de mitigação. Para que ocorra a ativação do linfócito, o antígeno necessita ser compatível com tal linfócito, desencadeando, assim, a reação. Já no AIS, a fim de atestar a compatibilidade entre amostra anômala e o detector, é calculado o seu grau de similaridade. Quanto mais similar uma amostra é de um detector, mais provável dessa amostra ser anômala. O cálculo de afinidade entre detectores e amostras pode ser realizado por inúmeros métodos, por exemplo, distância absoluta, distância euclidiana e distância de Manhattan. A Tabela 1 resume o mapeamento entre o HIS e AIS.

Tabela 1 – Mapeamento entre o HIS e o AIS

HIS	AIS
Linfócitos	Detectores
Antígenos	Amostra anômala
Linfócito Ativado	Alarme Ativado
Produção de Anticorpos	Mitigação da Ameaça
Ligação Antígeno com Linfócito	Alta Similaridade entre amostra e detectores

Os trabalhos apresentados por [59, 60] foram os primeiros a adotar as principais diretrizes do sistema imunológico do corpo humano em algoritmos. O objetivo desses algoritmos é criar uma abstração do sistema imune humano [61]. Dentre os algoritmos clássicos bioinspirados estão: seleção negativa, seleção clonal, redes imunes artificiais e a Teoria do Perigo.

A seleção negativa (do inglês, NSA - *Negative Selection Algorithm*) foi desenvolvida por Forest et al. [56] para abordar o paradigma de próprio e não próprio. Este paradigma consiste em considerar os dados como próprios quando são legítimos, e como não próprios

quando são dados anômalos. Uma das vantagens de um algoritmo que utiliza este paradigma é a capacidade de detectar os momentos anômalos sem a necessidade de dados de amostra para o conjunto de treinamento [62].

O NSA é composto de duas fases: a etapa de treinamento e a etapa de classificação. Na primeira etapa, são gerados detectores aleatoriamente e comparados com dados legítimos. Caso o detector seja semelhante, o mesmo é descartado, caso contrário, ele é mantido até se obter a quantidade de detectores desejada. Já a segunda fase consiste em monitorar os novos dados da rede e compará-los com os detectores existentes. É considerada uma anomalia quando a nova amostra é similar com um ou mais detectores [61].

Outro tipo de HIS é a Seleção Clonal, sendo uma de suas principais diferenças o momento no qual um linfócito é ativado por um antígeno, gerando assim novos linfócitos que também se alteram com o objetivo de atingir melhores taxas de detecção deste antígeno [63]. Já nos AIS, o algoritmo que se inspira no funcionamento da seleção clonal é o CLONALG, introduzido por De Castro [63].

O objetivo da seleção clonal é identificar as anomalias através da criação de detectores que não representem os dados legítimos, mas sim tudo o que for diferente. O CLONALG também possui duas etapas, o treinamento dos detectores e a classificação. Na etapa de treinamento do CLONALG são gerados detectores de forma aleatória, os mesmos são avaliados quanto a sua similaridade e são ranqueados de forma crescente. A afinidade consiste em avaliar o quão parecido é o detector em relação aos dados de treinamento. São selecionados os detectores que possuem o menor valor de afinidade, estes são clonados e sofrem mutações com o objetivo de reduzir ainda mais os valores de similaridade. Os detectores com maiores similaridades são excluídos, dando espaço à criação de novos detectores. O processo se repete até atingir a quantidade de detectores com a similaridade desejada. A segunda etapa é semelhante a da seleção negativa e consiste em avaliar cada nova amostra com os detectores. Caso a similaridade de ambos seja alta, é um indicativo de uma potencial anomalia. Historicamente a seleção clonal foi muito utilizada para otimização e reconhecimento de padrões [64].

As redes imunológicas empregam a seleção clonal e também o conceito de redes. Uma das primeiras aplicações de redes imunológicas é a aiNet, apresentada por de Castro [65]. Esse trabalho aplicou um grafo ponderado para conectar os clusters de detectores. O algoritmo basicamente funciona criando uma rede de detectores espalhados com o objetivo de reconhecer as anomalias. À medida que esses detectores forem sendo ativados ao localizar uma anomalia, sofrem a seleção clonal visando aumentar a quantidade de detectores nesta região anômala. Os detectores que não forem utilizados acabam sendo descartados. Desse modo, nas regiões com alta densidade de detectores são criados *clusters* e esses são conectados entre si para detectarem as regiões anômalas com maior precisão. Cada conjunto de *cluster* conectado revela uma possível região anômala.

A teoria do Perigo é uma alternativa à teoria do próprio e não próprio. Enquanto a teoria do próprio ou não próprio ativa um detector apenas quando um padrão anômalo é reconhecido, na teoria do perigo vários fatores desencadeiam pequenos sinais de alerta que, unidos, acabam detectando uma possível anomalia [66].

2.5 Redes SDN e Openflow

A arquitetura tradicional de redes possui o plano de controle e o plano de dados. O primeiro é o software que controla o encaminhamento de pacotes a ser realizado pelos equipamentos. O segundo é o hardware específico por tais encaminhamentos. Nessa arquitetura, ambos os planos ficam no mesmo equipamento [67], tornando custoso todo e qualquer desenvolvimento ou implantação de novas tecnologias ou recursos, pois para a modificação do plano de controle de toda a rede é necessária a intervenção em cada equipamento de forma individual [1].

O conceito de redes SDN, diferentemente das redes tradicionais, separa os dois planos. O plano de controle é movido para uma entidade central chamada de controlador, já o plano de dados continua embarcado em cada equipamento da rede, que utiliza um hardware específico para realizar o encaminhamento dos pacotes [10], conforme pode ser observado na Figura 3. Essa separação abre possibilidades para programar o plano de dados através do controlador, utilizando um protocolo que envia as instruções e comandos do plano de controle para o plano de dados, onde estão localizados os switches e roteadores [68]. Tal centralização facilita a coleta de dados sobre os fluxos em tempo real e também possibilita a aplicação de medidas de forma automática e mais rápida em comparação com a arquitetura tradicional [13]

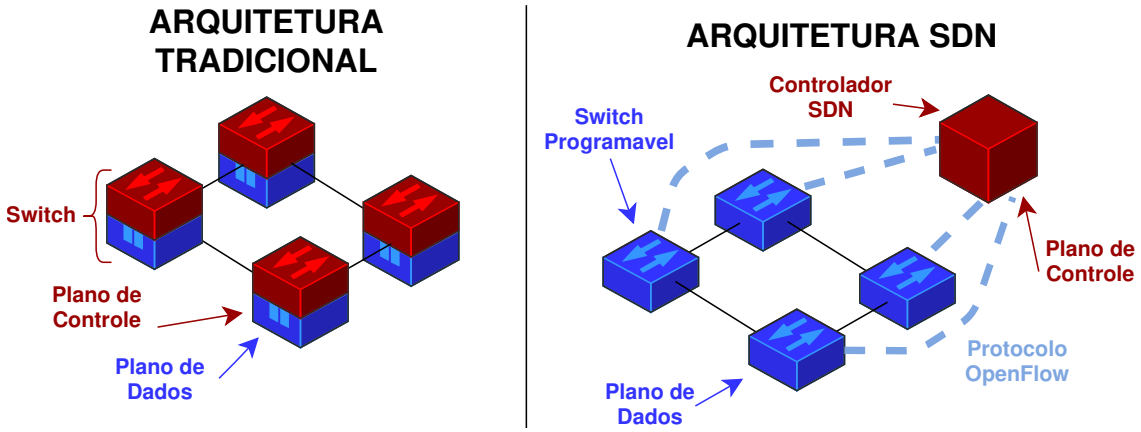


Figura 3 – Comparativo entre Arquitetura tradicional e Definidas por Software. Adaptado de Kreutz, Diego, et al. [1]

No plano de controle está localizada toda a lógica e conhecimento da rede. Acima dele também está a camada de aplicação, onde são implementados os componentes e

softwares que enviam os comandos para a camada de controle. Esta, por sua vez, encarrega-se de realizar as alterações solicitadas no plano de dados. Para um programa na camada de aplicação, a rede SDN é considerada como um único switch. Simplificando, desse modo, a topologia e também alterando a proposta dos equipamentos na arquitetura tradicional, pois antes era necessário interpretar inúmeros protocolos e padrões. Já com a nova arquitetura, os equipamentos precisam apenas receber as instruções do plano de controle e executá-las [10, 69].

As redes SDN têm a capacidade de prover a interoperabilidade entre os equipamentos de marcas distintas. Nas grandes redes tradicionais era necessário possuir equipamentos de uma mesma marca para possibilitar funções de gerência de maneira unificada. Já nas redes SDN são necessários equipamentos que suportem os protocolos das redes SDN para se ter um funcionamento unificado [13].

Um dos protocolos mais conhecidos para redes SDN é o OpenFlow[69], desenvolvido pelas Universidades de Stanford e da Califórnia, em Berkeley. Foi apresentado para a comunidade científica em 2008 e padronizado pela ONF (*Open Network Foundation*) no ano de 2011. Em sua concepção, o Openflow teve o objetivo de possibilizar a realização de testes e experimentos de maneira mais simplificada e também das fabricantes não precisarem abrir as arquiteturas internas dos seus equipamentos [67, 70]. O OpenFlow estabelece uma comunicação criptografada entre o controlador e os equipamentos do plano de dados. Desse modo, é possível que o controlador monitore todo o plano de dados e também gerencie os fluxos de maneira unificada [71].

O principal conceito de abstração presente no OpenFlow são os fluxos que podem ser considerados como as tuplas de uma tabela. A Tabela 2 apresenta um exemplo de tabela de fluxos de um switch SDN, nas quais são usadas as colunas com condições para ativação da regra. Os dados utilizados nas condições são os mais variados, como endereços MAC, IP e portas, portas do switch, informações da camada de enlace, rede, entre outras. Em casos em que um determinado dado é irrelevante na regra, é empregado o uso de asterisco. O campo de ação armazena a ação a ser realizada para um determinado pacote, caso ele seja compatível com esse fluxo. Por fim, a coluna de contador armazena as estatísticas de utilização do fluxo [67]. Dessa maneira, o OpenFlow simplificou o plano de dados para um conjunto de condições e ações. Com isso, abre possibilidades para reutilizar funções de hardware existentes como listas de controle de acessos (do inglês, ACL - *Access Control List*) para elaborar serviços de VLAN, firewall, entre outros [67].

A operação da rede utilizando esse protocolo constitui-se da seguinte maneira: cada pacote que chega ao plano de dados é comparado com a tabela de fluxos, caso algum dos fluxos seja compatível com o pacote, ele será encaminhado de acordo com a regra de tal fluxo e será incrementado ainda o contador desse fluxo. Entretanto, caso nenhuma regra se aplique a este pacote, o mesmo é encaminhado ao controlador para que seja analisada

Tabela 2 – Exemplo de uma tabela de fluxos

Porta Switch	MAC Origem	IP Origem	Porta Origem	IP Destino	...	Ação	Contador
*	64:5A:*	*	*	*	*	Porta 1	25
*	*	192.168.1.*	*	*	*	Porta 2	32
2	*	*	*	*	*	Porta 3	30
1	*	192.168.1.15	3366	*	*	Bloquear	520
*	*	*	22	*	*	Controlador	132

qual a ação a ser realizada com o mesmo. Normalmente o controlador recebe os pacotes para a criação de um novo fluxo ou por alguma regra de encaminhamento de pacote ao controlador. Também é possível realizar outras operações com o pacote, como enviar a outros controladores da rede, bloqueá-lo, enviá-lo através de determinada porta física do switch, entre outras ações [69].

3 TRABALHOS RELACIONADOS

A detecção de anomalias é um dos principais mecanismos de mineração de dados capaz de analisar amostras inconsistentes ou suspeitas. Muitos pesquisadores vêm estudando a sua aplicabilidade em várias áreas, como detecção de fraudes, análise financeira e até em novos ambientes como IoT [72] [73]. Além dessas aplicações, a comunidade científica têm apresentado inúmeros modelos de detecção de anomalias para proteger as redes de usuários maliciosos [39][10] [74].

Dentre os algoritmos e técnicas utilizadas na detecção de anomalias, pode-se citar: algoritmos estatísticos [75], de clusterização [32], de classificação [76] e evolucionários [66]. O IDS pode estar localizado dentro ou fora da rede para protegê-la de invasores que tentam obter acesso indevido à rede [77].

Sistemas de detecção de intrusão podem ser classificados em dois tipos: os baseados em assinatura e os baseados em detecção de anomalia. O primeiro utiliza as assinaturas, que podem ser definidas como uma sequência de padrões que caracterizam um ataque [78]. Este modelo tem a capacidade de atingir altas taxas de detecção com ataques já conhecidos, entretanto tem baixa performance em relação a ataques novos ou desconhecidos. Já o baseado em detecção de anomalias cria um perfil que representa o comportamento padrão, qualquer variação no comportamento atual, se comparado ao esperado, pode ser considerado uma anomalia [79] [80].

O IDS apresentado em [81] utiliza Rede Neural Convolucional (do inglês, CNN - *Convolutional Neural network*) e o algoritmo de otimização do Leão (do inglês, LOA - *Lion Optimization Algorithm*). A CNN é um modelo para reconhecimento visual, utilizado para transformar os atributos coletados do tráfego em atributos de alto nível. Já o LOA é um algoritmo de otimização bioinspirado que simula o comportamento dos leões e realiza uma seleção de atributos dentre esses atributos.

A base de dados empregada para avaliação foi a do NSL-KDD [1]. Também foram realizados outros experimentos alterando os algoritmos de seleção de atributos por outros bioinspirados, como a otimização por colônia de formigas e também a otimização por colônia de abelhas. A CNN e o LOA foram os que obtiveram melhor desempenho dentre os experimentos, atingindo uma acurácia de 98.2%. Embora o IDS seja leve, ele se concentra em ataques DDoS e não usa estratégias de mitigação para as ameaças detectadas.

Avaliando os IDS baseados em assinatura em redes SDN, Lai et al. [76] propôs um sistema de detecção baseado em uma Rede Neural Perceptron Multicamadas (do inglês, MLP - *Multilayer Perceptron*). Foram extraídos seis atributos dos fluxos IP coletados

[1] Disponível em https://www.unb.ca/cic/datasets/nsl.html

e aplicadas tais informações como dados de entrada para a MLP. Os autores também desenvolveram um detector baseado em pacotes e ambos os modelos foram comparados utilizando a base de dados NSL-KDD. Os resultados obtidos demonstraram que os dois foram capazes de detectar as ameaças. Entretanto, a técnica baseada em fluxos apresentou melhores resultados, pois causou menos sobrecarga na detecção e ainda obteve uma performance satisfatória se comparada com a baseada em pacotes. Como o modelo usa dados anômalos para treinamento, pode sofrer erros de classificação quando novos tipos de ataques ocorrerem. O esquema apresentado pelos autores também não possui política de mitigação de ataques.

Mansour et al. [68] apresentou um IDS utilizando algoritmos genéticos. O método seleciona os melhores atributos da base de dados do KDD Cup [2] e também calcula a função *fitness* para identificar as anomalias. A função *fitness* foi definida para cada protocolo, como TCP, UDP e ICMP visando atingir os melhores resultados. Quando uma anomalia é localizada, a proposta divide a análise em quatro tipos de ataques: DoS, *probe*, R2L, e U2R. Ataques *probe*, também conhecidos como ataques de sondagem, são responsáveis por investigar e coletar o comportamento da rede. O ataque (do inglês, U2R - *User to Root*) tem por objetivo o atacante obter acesso privilegiado a uma rede ou sistema. Por fim, o ataque (do inglês, R2L - *Remote to User Attacks*) tem o objetivo de um atacante obter acesso a uma rede através de uma conexão remota [74].

Depois de classificar o ataque, o valor da função *fitness* é recalculado para reconhecer outras instâncias de ataque. De modo geral o IDS obteve uma taxa de verdadeiro positivo próximo a 80%. No entanto, utilizou tráfego anômalo para treinamento, o que pode prejudicar a detecção de anomalias novas ou desconhecidas. Além disso, a estratégia de mitigação utilizada bloqueia toda a comunicação com o dispositivo atacado, possibilitando o bloqueio de tráfego legítimo.

O trabalho apresentado em [82] utilizou o conceito de entropia para detectar ataques DDoS. A solução contém quatro módulos: coletor de fluxos, sensor baseado em entropia, sensor de confirmação de ataque e mitigador. O processo se inicia com a coleta dos fluxos IP e em seguida é calculada a entropia do endereço IP de destino para localizar as possíveis anomalias. Assim que o comportamento anômalo for identificado, o módulo de confirmação de ataques entra em execução e analisa todos os fluxos relevantes para encontrar a origem do atacante. Feito isso, o módulo mitigador modifica os fluxos no controlador a fim de bloquear o ataque. A avaliação da proposta foi utilizando o emulador de redes Mininet. Os autores relataram que a solução detectou os ataques em seus estágios iniciais, mas apenas DDoS foi explorado neste trabalho.

Vidal et al. [66] utilizou a Virtualização das Funções da Rede (do inglês, NFV - *Network Functions Virtualization*) no intuito de elaborar uma solução descentralizada

[2] Disponível em http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

para a detecção de anomalias. Os NFV é um conceito que, com o uso da virtualização, oferece diversas funções dos nós de uma rede e com isso cria serviços de forma mais dinâmica. Desse modo, através do NFV, vários agentes foram espalhados por toda a rede, apropriando-se da teoria de sistema imune e sendo classificados em dois tipos: sistema inato e sistema adaptativo. Para a avaliação foram utilizadas as bases de dados do KDD Cup e Caida 2007. Além dessas bases, também foi colocada em curso uma sub-rede da Faculdade de Ciência da Computação da Universidade Complutense de Madrid e nela foi utilizada a ferramenta DDoSIM para gerar ataques por inundação. Nos resultados apresentados, os autores relataram que as respostas pelos agentes adaptativos foram mais efetivas que as dos agentes inatos e também podem ser utilizadas como alternativa aos tradicionais métodos de mitigação. Já os agentes inatos se comportam semelhante a um sistema de prevenção de intrusos tradicional. De modo geral, a solução foi eficaz no bloqueio dos ataques, todavia tem a necessidade de ser espalhada por toda a rede para obter um resultado adequado. A detecção proposta é especializada apenas em ataques DDoS, portanto, ataques de outros tipos podem não ser detectados.

O trabalho de Rathore et al. [83] apresentou um algoritmo bioinspirado para a detecção de ataques de negação de serviço (do inglês, DoS - *Denial of Service*) em redes SDN. O sistema inato foi utilizado para mitigar os ataques DoS realizados por atacantes, já o sistema adaptativo foi aplicado para mitigar os ataques DoS causados por switches. Quando um ataque é detectado, o módulo de mitigação bloqueia o atacante de se comunicar com a rede, contudo, quando o ataque é oriundo de switches, o mitigador reduz a largura de banda deste switch a fim de proteger o controlador da rede SDN. O emulador de rede Mininet foi utilizado para avaliar o algoritmo apresentado. Os autores informaram que os resultados superaram outras técnicas em relação ao tempo necessário para reagir às ameaças e à sobrecarga do esquema de detecção. No entanto, o algoritmo é especializado em ataques DoS, e a rede usada para as comparações é pequena, contendo apenas cinco hosts.

Dentre os trabalhos que utilizam AIS em redes tradicionais, Aziz et al. [78] propôs um IDS de duas camadas. A primeira camada emprega a seleção negativa para criar os detectores com os valores dos atributos da rede que caracterizam um comportamento normal. Quando forem detectadas variações neste comportamento normal, a segunda camada entra em operação e aplica cinco classificadores para identificar os ataques. Algoritmos genéticos, AIS, árvores de decisão, MLP e *Naive Bayes* foram ranqueados a fim de escolher o melhor classificador para a segunda camada. O IDS apresentado utilizou a base de dados NSL-KDD [3] a fim de avaliar a performance. Nos testes, foi constatado que nenhum dos classificadores testados obteve um resultado superior na detecção de todos os ataques. Os autores relataram que o valor de *f-measure* entre todos os classificadores foi de 78 %.

[3] Disponível em https://www.unb.ca/cic/datasets/nsl.html

O IDS utilizou o AIS para detectar momentos de anomalia, no entanto, foram aplicados classificadores que precisam de treinamento com as amostras anômalas, dificultando a detecção de ataques que não estão presentes no conjunto de dados de treinamento.

Hooks et al. [84] apresentou um comparativo entre duas técnicas de AIS, comparando seleção negativa e seleção clonal na detecção de anomalias. No comparativo foi utilizada a base de dados NSL-KDD, onde foram criados vários cenários alterando hiperparâmetros como: número de atributos, instâncias e detectores. Os autores concluíram que as duas propostas sofreram com os cenários que possuíam grandes quantidades de amostras e atributos, e também que a seleção negativa entregou resultados mais rapidamente do que a seleção clonal. Os algoritmos necessitam de ao menos 22 atributos para precisão maior que 80 %, o que gera um alto custo computacional para grandes redes. Além do mais, o trabalho usa um conjunto de dados de treinamento com ataques, com isso os ataques desconhecidos podem não ser detectados.

Outro trabalho que aplicou o AIS foi exposto por [85], que propôs um IDS baseado em seleção negativa. Na tentativa de localizar o melhor algoritmo de seleção de atributos, foram comparadas a teoria de conjuntos aproximados, a programação genética linear (do inglês, LGP - *Linear Genetic Programming*) e a *splines* de regressão adaptativa multivariada (do inglês, MARS - *Multivariate Adaptive Regression Splines*). Cada algoritmo escolheu seis atributos dos quarenta e um atributos disponíveis na base de dados da KDD Cup 99. Com relação aos resultados, os autores informaram que o MARS foi o melhor na seleção dos atributos, atingindo resultados de detecção similares ao de outros IDS da literatura. De modo geral, o IDS obteve bons resultados, sendo implantado em uma rede tradicional, contudo não possui nenhuma estratégia de mitigação para conter os ataques.

Tabatabaefar et al. [86] apresentou uma proposta que aborda duas técnicas de AIS: a seleção negativa e a seleção positiva. Desse modo, os detectores conseguem reconhecer o comportamento normal da rede e também o comportamento anômalo. A Optimização por enxame de partículas (do inglês PSO - *Particle Swarm Optimization*) foi utilizada para a fase de treinamento dos detectores. Os testes realizados adotaram a base de dados da KDD Cup 99 e os resultados demonstraram que a proposta apresentou uma taxa de detecção superior a 99%. Também recorrendo à mesma base de dados, Suliman et al. [33] desenvolveu um IDS com base na técnica de seleção clonal. Os autores relataram que os resultados obtidos foram comparáveis a outras propostas encontradas na literatura. A proposta foi baseada na detecção de ataques DoS e ataques probe. A proposta é implementada em redes tradicionais e não apresentou estratégias para mitigação de ataques.

O trabalho desenvolvido por Zhou et al. [61] demonstra uma proposta semelhante à apresentada por esta dissertação. Eles apresentaram um IDS utilizando a seleção negativa em redes SDN. O IDS cria e treina os detectores nos switches da rede. Após o período de treinamento, os detectores são enviados ao controlador SDN, que, por sua vez,

Tabela 3 – Relação dos trabalhos relacionados apresentados

Trabalho	Técnica	Mitig.	Ataques Variados	Rede	Base de dados
Arivudainambi et al. [81]	Variadas	✓	✓	SDN	NSL-KDD
Lai et al. [76]	MLP	✗	✓	SDN	NSL-KDD
Mansour et al. [68]	GA	✓	✓	SDN	KDD
Vidal et al. [66]	AIS	✓	✗	SDN	KDD, CAIDA e Gerado (DDoSIM)
Rathore et al. [83]	AIS	✓	✗	SDN	Emulado(Mininet)
Aziz et al. [78]	Variadas	✗	✓	Trad.	NSL-KDD
Duy et al. [82]	Entropia	✓	✗	SDN	Emulado (Mininet)
Hooks et al. [84]	AIS	✗	✓	Trad.	NSL-KDD
Shen et al. [85]	AIS	✗	✓	Trad.	KDD
Tabatabaefar et al. [86]	AIS	✗	✓	Trad.	KDD
Suliman et al. [33]	AIS	✗	✓	Trad.	KDD
Zhou et al. [61]	AIS	✗	✓	SDN	KDD e NSL-KDD

selecionará os melhores detectores. Logo em seguida esta coleção de detectores é enviada a todos os switches da rede para localizar potenciais anomalias. O IDS proposto foi avaliado com bases de dados, o KDD Cup e o NSL-KDD. Segundo os autores, o resultado alcançado mostrou-se mais eficiente na detecção das anomalias em comparação com os IDS anteriores que utilizavam AIS. Também a proposta se mostrou mais leve para outros métodos avaliados. Contudo a solução não propôs estratégia de mitigação e utilizou os switches da rede para realizar a detecção da anomalia, o que pode levar a sobrecarga do plano de dados.

Os trabalhos [78, 82] utilizam técnicas para criar um perfil do comportamento normal da rede. Entretanto, caso esse comportamento se altere, poderão ocorrer erros na classificação. Do mesmo modo, os trabalhos apresentados em [76, 68] utilizam dados de anomalias para o treinamento, causando potenciais erros na classificação quando surgirem ataques novos ou desconhecidos. O primeiro trabalho não apresentou estratégia para a mitigação de ataques e o segundo bloqueia toda a comunicação com o dispositivo vítima, prejudicando os pacotes legítimos. Alguns trabalhos focaram-se na detecção de apenas um tipo de ataque [66], ou foram implementados em arquitetura de rede tradicional [85], ou até mesmo não atuam contra a propagação do ataque [61]. A proposta deste trabalho é apresentar um IDS baseado em AIS, que possa detectar as anomalias com um baixo tempo de resposta. O presente trabalho foi desenvolvido tendo como foco as redes SDN, objetivando uma possível vantagem das possibilidades que este tipo de rede oferece, fornecendo assim contra-medidas para ameaças detectadas a fim de prover a resiliência da rede. A Tabela 3 apresenta um comparativo entre os trabalhos relacionados elencados nesta seção.

4 MODELO PROPOSTO

O sistema proposto AIS-IDS é composto por três módulos acoplados ao controlador SDN: coletor de fluxo, módulo de detecção AIS e o módulo mitigador. O primeiro módulo solicita periodicamente os fluxos IP do plano de dados utilizando o protocolo OpenFlow [69]. Assim que os fluxos IP forem recebidos pelo módulo, é realizado um pré-processamento nos fluxos e estes são encaminhados ao segundo módulo.

O módulo de detecção é o responsável por classificar o comportamento da rede como anômalo ou legítimo. Essa classificação é realizada através do AIS. Caso um comportamento for legítimo, regras de encaminhamento são criadas para direcionar os pacotes ao destino normalmente. Entretanto, quando for detectado um comportamento anômalo, os fluxos são remetidos para o módulo de mitigação.

O módulo de mitigação é o encarregado por bloquear a ameaça detectada. Primeiramente é identificada a origem e o tipo do ataque, após essa primeira análise, o módulo define qual a melhor estratégia para o bloqueio da ameaça. Todas essas regras de encaminhamento são enviadas por meio do protocolo OpenFlow para switches localizados no plano de dados.

A Figura 4 apresenta uma visão geral de todo o funcionamento do AIS-IDS. À medida que novos pacotes são recebidos, fluxos são criados e atualizados para encaminhar esses pacotes ao seu destino. Constantemente o controlador irá requisitar os fluxos ativos de todos os switches no plano de dados para o plano de controle, através do protocolo OpenFlow. O sistema AIS-IDS, localizado na camada de aplicação, utiliza os fluxos requisitados a fim de realizar toda classificação das amostras e também na tomada de decisão quanto à mitigação. A estratégia para a mitigação é enviada da camada de aplicação para o plano de controle e o controlador, por meio do protocolo OpenFlow, enviando as ações de mitigação ao plano de dados para que os switches implementem a ação de contramedida. Caso um pacote seja compatível com um fluxo mitigador, o mesmo é bloqueado e a ameaça é contida.

4.1 Módulo coletor de fluxos

Visando a coleta dos fluxos do plano de dados é utilizado o protocolo OpenFlow, que estabelece uma interface comum adotada na instrução de switches sobre regras de encaminhamento direcionadas a pacotes de entrada, bem como fornece acesso a dados estatísticos e configuração de plano de controle [87].

A cada instante, muitos fluxos são criados e expirados em redes SDN. Quanto mais frequente for a coleta de informações dos fluxos, mais precisa será a detecção de

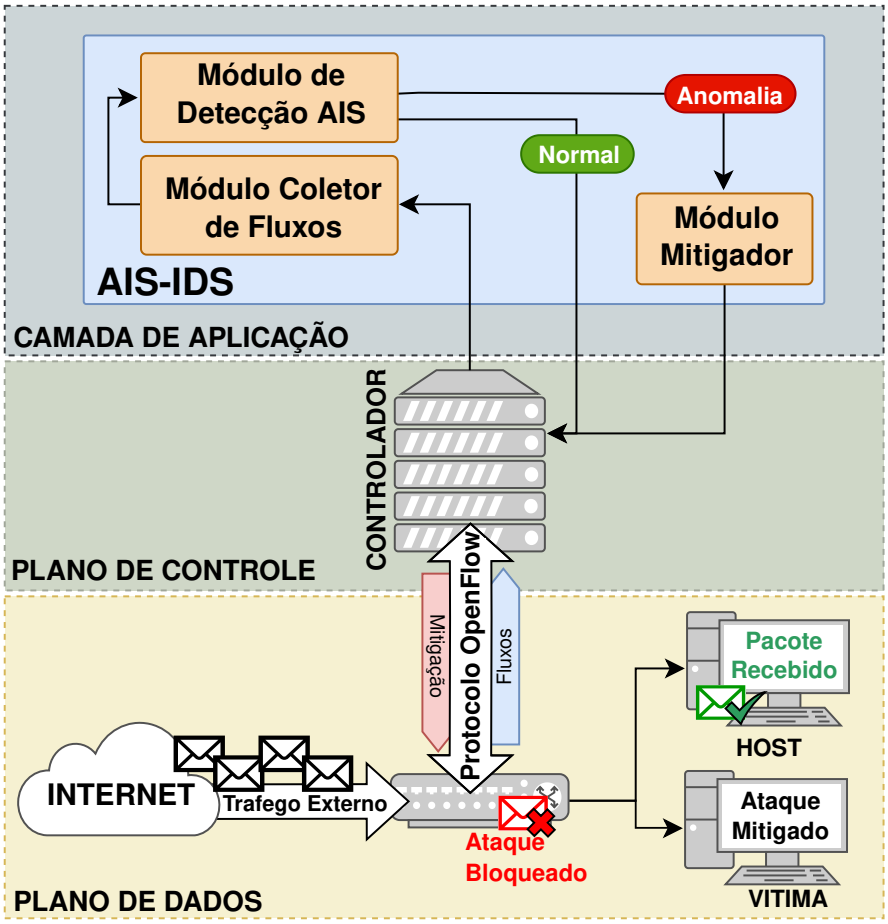


Figura 4 – Visão geral do AIS-IDS

anomalias e, consequentemente, mais rápidas serão tomadas as ações a fim de se impedir a propagação de um ataque.

Para uma ação ser considerada tempo real ela deve ser tomada em menos de 10 milissegundos. Desse modo, a comunidade científica utiliza o termo "quase tempo real", do inglês *near real time*, que define ações que serão tomadas na escala de segundos, mas que não podem ser consideradas tempo real [11, 20]. Portanto, para detectar ataques em quase tempo real, as estatísticas de tráfego devem ser coletadas em intervalos curtos de tempo. Entretanto, uma desvantagem da coleta frequente é o volume de fluxos a serem analisados. Portanto, criar um IDS que reaja com precisão e rapidamente ao detectar ameaças é uma questão desafiadora. Nesse sentido, cada módulo da proposta foi desenvolvido com vista ao equilíbrio entre eficiência computacional e eficácia na detecção de anomalias.

Vários trabalhos que abordam a detecção de anomalias utilizaram intervalos fixos de coleta de 5 minutos [88, 89], intervalos de 30 segundos [11], ou até intervalos mais curtos, como 2 segundos [90, 91]. Entretanto, com o aumento do volume de dados transferidos pelas redes, esses intervalos de coleta acabam se tornando cada vez mais inviáveis. Exemplificando, uma rede de 10 Gbps pode ter Terabits de informações afetadas durante um intervalo de cinco minutos, ou aproximadamente 2,5 Gigabytes de informações em

apenas dois segundos. Desse modo, o primeiro módulo coleta estatísticas dos fluxos IP nos switches e extrai os atributos do tráfego a cada segundo.

Inúmeros atributos podem ser utilizados para caracterizar o comportamento de uma rede, como endereços IP, portas, protocolos, entre outros. Os atributos que descrevem o volume do tráfego, como, por exemplo, bytes e pacotes transmitidos, comumente são utilizados [39, 11]. No entanto, esses atributos podem causar erros e dificuldades na caracterização do comportamento da rede, pois poucos *hosts* utilizando massivamente a rede podem gerar grandes picos nestes atributos e, consequentemente, distorcer o real comportamento da rede [11].

Desse modo, o AIS-IDS utiliza apenas quatro informações qualitativas para definir o comportamento da rede: endereços IP de origem e destino e portas de origem e destino. A vantagem de empregar essas informações do tráfego é dupla. Em primeiro lugar, os endereços IP e portas tornam-se sensíveis às mudanças no comportamento do tráfego quando convertidos para atributos quantitativos, levando à identificação dos momentos em que o tráfego se comporta de forma anômala. Em segundo lugar, é possível localizar *hosts* e serviços envolvidos no evento anômalo e mitigá-los [11]. Também a utilização de um conjunto reduzido de atributos permite a criação de um sistema de detecção parcimonioso, utilizando menos recursos computacionais desde a fase de aquisição de dados até a mitigação dos ataques identificados.

A entropia de Shannon [92], usada neste trabalho, quantifica o nível de concentração da informação de acordo com a distribuição de um conjunto de amostras [93, 94]. Ela é calculada com base em um histograma para cada característica qualitativa do tráfego e usa esta informação na função (4.1), na qual o conjunto $X = \{a_1, ..., a_m\}$ representa uma das características qualitativas, n_i é o número de vezes que o i -ésimo endereço IP ou porta foi observado no intervalo de análise e $s = \sum_{i=1} n_i$ é o total de todas as ocorrências no histograma.

$$H(X) = -\sum_{i=1}^{m} \left(\frac{n_i}{s}\right) \log_2 \left(\frac{n_i}{s}\right) \tag{4.1}$$

A entropia é eficiente em demonstrar os momentos que contém anomalias, pois quaisquer variações nos valores de entropias das características podem indicar a ocorrência de uma anomalia. Além do que, a alteração dos valores das entropias das características é capaz de identificar até o tipo de ataque. Por exemplo, um ataque DDoS causa mudança na concentração de portas de destino, fazendo que o valor da entropia diminua; já a entropia dos endereços IP de origem tende a aumentar, visto que inúmeras máquinas zumbis irão atacar em conjunto. Por outro lado, em um ataque *port scan*, a tendência é que a entropia da porta de destino aumente pela execução do escaneamento das portas do alvo. Desse modo, a utilização da entropia auxilia na sumarização dos dados e também na detecção

dos momentos anômalos e até no tipo de ataque que está ocorrendo.

Assim, o módulo coletor de fluxos é responsável por coletar os fluxos de entrada, processar, resumir e encaminhá-los ao módulo de detecção AIS para caracterização do padrão. O procedimento interno deste módulo é apresentado na Figura 5.

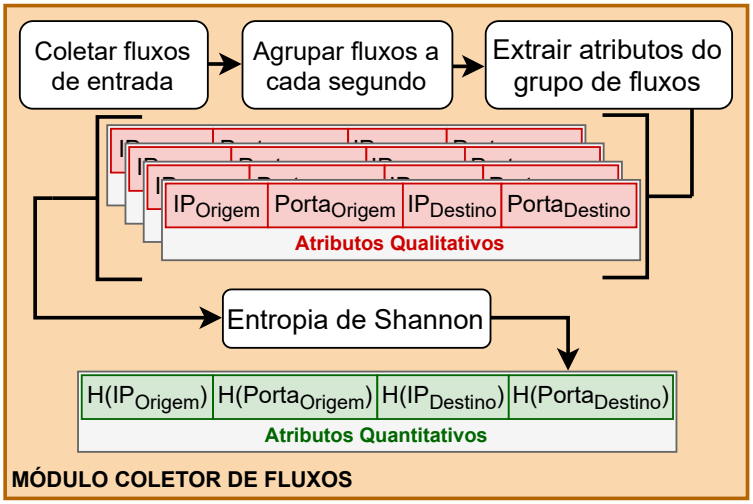


Figura 5 – Esquema do módulo coletor de fluxos: dos fluxos de entrada para os atributos quantitativos

4.2 Módulo de Detecção AIS

O módulo de detecção AIS utiliza um dos algoritmos inspirados em sistemas imunológicos artificiais, chamado de *Negative Selection Algorithm (NSA)*, o qual é um modelo computacional gerador de detectores imunológicos. O algoritmo simula a teoria do próprio e não próprio presente no sistema imune humano [95]. O NSA classifica os dados em próprios quando são legítimos e não próprios quando são anômalos. Uma vantagem dessa classificação é a capacidade de detectar ameaças sem exigir exemplos anômalos no conjunto de dados de treinamento [62].

A Figura 6 ilustra de modo geral o funcionamento da Seleção Negativa. A Figura 6(a) representa o comportamento do atributo $H(IP_{Origem})$ da rede sem anomalias. As colunas ilustram intervalos de um segundo de análise do tráfego. Cada círculo verde retrata o valor da entropia do atributo analisado após ser pré-processado pelo módulo Coletor de Fluxos.

A Figura 6(b) exibe a fase de treinamento, na qual os detectores, representados pelos círculos roxos, são criados. Os detectores são criados de forma aleatória com o objetivo de preencher o espaço de busca que não corresponda ao comportamento legítimo. Ocupar todo o espaço de busca com detectores é uma tarefa computacionalmente custosa. Desse modo, a utilização da similaridade miníma visa possibilitar a detecção de anomalias mesmo em locais onde não se possui um detector. Ao final desta fase, o objetivo é que

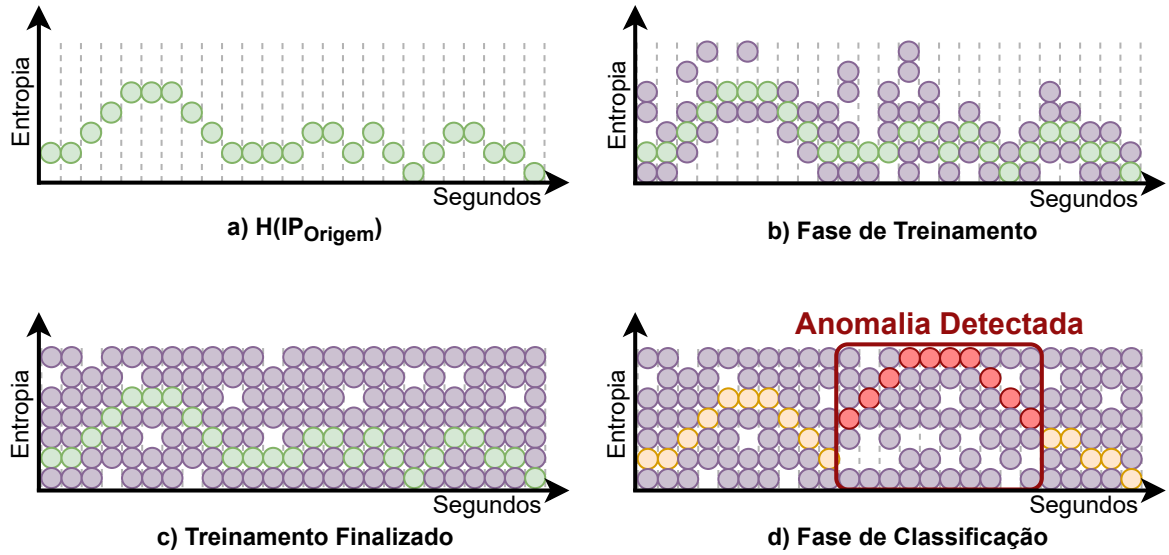


Figura 6 – Exemplo de funcionamento do NSA para geração dos detectores do atributo $H(IP_{Origem})$

os detectores gerados possam detectar o máximo de comportamentos anômalos possíveis, conforme Figura 6(c).

A fase de classificação é demonstrada na Figura 6(d), em que cada círculo amarelo retrata o atributo $H(IP_{Origem})$ da rede coletado e preprocessado para a detecção de anomalias. Caso o comportamento atual seja diferente do esperado, os detectores presentes no intervalo analisado serão acionados e informam sobre a anomalia detectada neste atributo. Na Figura, os círculos vermelhos representam os detectores ativados.

O módulo de detecção do AIS divide a NSA em duas fases. A fase de treinamento cria detectores que são atribuídos na classificação com o objetivo de reconhecer comportamentos anômalos. Em ambas as etapas, é utilizado o cálculo de similaridade. Na fase de treinamento, a similaridade visa aceitar ou descartar os detectores criados. Na fase de aplicação, a similaridade detecta intervalos que contêm anomalias.

A técnica de Distância Absoluta (do inglês, AD - *Absolute Distance*) foi escolhida para calcular a similaridade, sendo empregada com sucesso em outros trabalhos [96] [97]. A AD é expressa pela equação (4.2), em que p e q são os valores que estão sendo comparados para o cálculo da similaridade.

$$\delta = \sqrt{(p-q)^2} = |p-q| \tag{4.2}$$

4.2.1 Criação dos detectores

O objetivo de um detector é mensurar a similaridade entre o tráfego anômalo e o comportamento normal da rede. Dessa maneira, um detector engloba os valores das

quatro entropias (IP de origem, IP de destino, porta de origem e porta de destino) e pode ser representado por um conjunto $\Phi_i^t = \{H(srcIP), H(dstIP), H(srcPort), H(dstPort)\}$, em que $1 \leq i \leq n$ é o seu índice e t indica o intervalo de análise (em segundos) ao qual esse conjunto pertence.

Uma coleção de detectores $\Omega_t = \{\Phi_1^t, \Phi_2^t, \ldots, \Phi_n^t\}$ contém todos os detectores que foram criados para o instante t. A fase de treinamento necessita de um conjunto de dados de treinamento sem anomalias para gerar os detectores. Assim, é comumente gerado em um ambiente controlado para fornecer um conjunto de dados sem qualquer tipo de ataque.

A fase de treinamento começa estabelecendo um valor aleatório para cada atributo f presente no conjunto de dados de treinamento. Esse valor aleatório é elaborado a partir da distribuição uniforme no intervalo zero e o do valor mais alto desse atributo nos dados de treinamento.

Depois disso, a pontuação de similaridade δ_f é calculada entre o valor aleatório e todas as amostras do atributo f, de modo que a pontuação deva ser menor que o hiperparâmetro de similaridade mínima k; caso contrário, um novo valor aleatório é definido para esse atributo e o processo é reiniciado.

Um detector está completo apenas quando seus atributos têm valores de similaridade maiores que o mínimo de similaridade com o conjunto de dados de treinamento. O novo detector é adicionado à coleção de detectores e o processo de geração termina quando esta coleção atinge n detectores, ou seja, $|\Omega_t| = n$.

À medida que os detectores são gerados, também é realizado o cálculo da similaridade entre os detectores, visando evitar a criação de detectores semelhantes e assim não cobrir todo o espaço de busca de forma otimizada. Quanto menor a pontuação de similaridade, mais semelhante o detector é às instâncias de treinamento. O objetivo é gerar detectores diferentes dos dados de treinamento e também dos detectores anteriormente criados. Portanto, um detector precisa representar um possível evento de tráfego anômalo. Consequentemente, a coleção de detectores visa representar possíveis comportamentos incomuns.

O cálculo da similaridade atributo por atributo, ao invés de um detector completo, acelera o processo de geração porque o detector desenvolvido não é totalmente descartado, mas apenas valores de atributos que não foram aceitos pela similaridade mínima. Além do mais, na fase de classificação, é possível avaliar cada atributo separadamente e assim detectar ataques com comportamentos específicos. Ao final da primeira fase, a coleção de detectores gerada é enviada para a segunda fase do módulo AIS. O Algoritmo 1 descreve a primeira fase da elaboração dos detectores.

Algorithm 1: Pseudo-código da primeira etapa do módulo de detecção AIS.

Entrada: Dados de Treinamento (tData), número de detectores (n) e valor de similaridade mínima (k)
Saída : Coleção de Detectores (Ω_t)

```
1  Ω_t ← ∅
2  numDetectores ← 0
3  while numDetectores ≤ n do
4      foreach atributo ∈ tData do
5          colAtributos ← tData[atributo]
6          colAtributosDetectores ← Ω_t[atributo]
7          foiusado ← não
8          while foiusado = não do
9              atributoAleatorio ← gerarAtributoAleatorio()
10             if calcSimilaridade(atributoAleatorio, colAtributos) ≥ k e
                calcSimilaridade(atributoAleatorio, colAtributosDetectores) ≥ k
                then
11                 novoDetector[atributo] ← atributoAleatorio
12                 foiusado ← sim
13             end
14         end
15     end
16     Ω_t ← Ω_t ∪ novoDetector
17     numDetectores ← numDetectores + 1
18 end
19 return Ω_t
```

4.2.2 Classificação do tráfego baseado nos detectores gerados

A segunda fase do módulo de detecção AIS classifica o tráfego corrente em normal ou anômalo. As amostras coletadas e pré-processadas pelo módulo Coletor são recebidas para cálculo da similaridade com a coleção de detectores gerada anteriormente.

Usar dados coletados ou extraídos por meios estatísticos para detectar anomalias pode levar a erros significativos. Além disso, não há um limiar claro entre o que é um comportamento anormal ou legítimo [38] [98]. Nesse sentido, em vez de usar um limite rígido (*hard thresholding*) como a classificação clássica, a lógica *fuzzy* [99] foi empregada para o reconhecimento de ameaças de rede.

Um Sistema de Inferência Fuzzy visa produzir um valor de saída suportado pela lógica *fuzzy* em uma determinada entrada. Os valores de saída variam de 0 a 1 [99] para fornecer uma análise racional em um ambiente que pode possuir informações imprecisas ou incompletas [38], como a análise de tráfego de rede. A primeira etapa é difundir a entrada por meio de uma função de pertinência, criando um conjunto *fuzzy*. Diversas regras são aplicadas ao conjunto *fuzzy*, preparando-o para o processo de defuzzificação, elaborando

um conjunto nítido que leva em consideração todas as regras aplicadas anteriormente.

Para usar um Sistema de Inferência *Fuzzy* no cálculo do valor de similaridade δ_f, calculado com um dado atributo f a partir dos dados de entrada e da coleção de detectores (Ω_t), pode-se usar a função de associação *fuzzy*. Neste trabalho é aplicada a pertinência gaussiana[99], expressa pela função (4.3),

$$\zeta_f = \begin{cases} e^{\left(\frac{-\left(\delta_f - k\right)^2}{2\sigma_f^2}\right)}, & \text{se } \delta_f > k \\ 1, & \text{caso contrário} \end{cases}, \tag{4.3}$$

em que δ_f é o valor que representa a similaridade, k é o valor da similaridade mínima e σ_f é o desvio padrão da similaridade do atributo f. Se δ_f é menor que k, o resultado da função é 1. Esta regra foi criada para evitar que o valor de similaridade diminua quando o δ_f é menor que k.

Depois de aplicar a função de associação *fuzzy*, cada amostra tem quatro valores difusos ζ_f, ou seja, um valor para cada atributo do tráfego (entropia de IP de origem e destino e entropia de portas de origem e destino), definidos por $f(0)$, $f(1)$, $f(2)$ e $f(3)$. Para calcular a pontuação geral da amostra é preciso somar as pontuações de similaridade. No entanto, cada atributo é afetado de maneira diversa por inúmeros tipos de ataques. O cálculo de um coeficiente de importância para cada atributo fornece uma detecção mais precisa. Ao ponderar a importância de cada atributo para detecção de anomalias no tráfego, a identificação de ataques pode ser facilitada no caso de i) ataques que não afetam significativamente o comportamento normal da rede, como os ataques *port scan*; ii) ou ataques que possuem comportamentos específicos.

Dessa forma, para obter o valor dos quatro coeficientes de importância definidos por $c(0)$, $c(1)$, $c(2)$ e $c(3)$ foi aplicada a regressão logística multinomial (do inglês, MLR - *Multinomial Logistic Regression*). A MLR é um método que emprega a regressão logística em dados em que as variáveis dependentes são desordenadas e os valores independentes são contínuos ou categóricos [100] [101].

De modo geral, a MLR induz um classificador capaz de diferenciar entre as anomalias utilizando amostras rotuladas [11]. No AIS-IDS, a MLR utiliza um conjunto de dados de um dia de tráfego, com amostras normais e anômalas, representadas pelos quatro atributos já utilizados $H(srcIP), H(dstIP), H(srcPort), H(dstPort)$ e também o rótulo de cada amostra, categorizados em três grupos: normal, DDoS e *port scan*. O resultado desse modelo são quatro valores que representam os coeficientes de importância de cada atributo. A MLR foi utilizada uma vez apenas para a obtenção de tais coeficientes, não sendo necessário recalcular ou recalibrar para outros dias ou redes, pois o objetivo desses coeficientes é mensurar o quão impactante é cada atributo no comportamento de um

ataque.

Assim, para a obtenção do valor da similaridade geral de uma dada amostra, multiplica-se o valor *fuzzy* de cada atributo pelo seu respectivo valor de coeficiente, o resultado dessas multiplicações são somadas compondo o valor da similaridade da amostra. Quando essa pontuação é maior do que o limite de corte Γ, tais dados são considerados anômalos; caso contrário, são legítimos. A avaliação dos hiper-parâmetros é apresentada na seção 5.1.1 juntamente com a apresentação do cenário 1, em que serão apresentados valores e resultados numéricos que demonstram a escolha.

Existem propostas que mitigam ataques através da limitação de conexões ativas entre os *hosts* [83, 38]. A ideia central desse tipo de proposta é monitorar a quantidade de conexões entre os *hosts* e, se o valor limite for alcançado, a comunicação é bloqueada. Essa técnica é eficiente para ataques de força bruta ou de inundação, porém, é necessário definir o valor limite de conexões. Quando o número for mal dimensionado, poderá ocorrer bloqueios indesejados ou ataques não mitigados. Desse modo, técnicas como o AIS-IDS, que utilizam o comportamento normal da rede para definir quando um ataque está ocorrendo, são mais robustas e capazes de detectar ataques nas mais variadas situações.

O Algoritmo 2 apresenta o pseudo-código para a segunda fase e a Figura 7 apresenta a visão geral de ambas as fases do módulo de detecção de AIS.

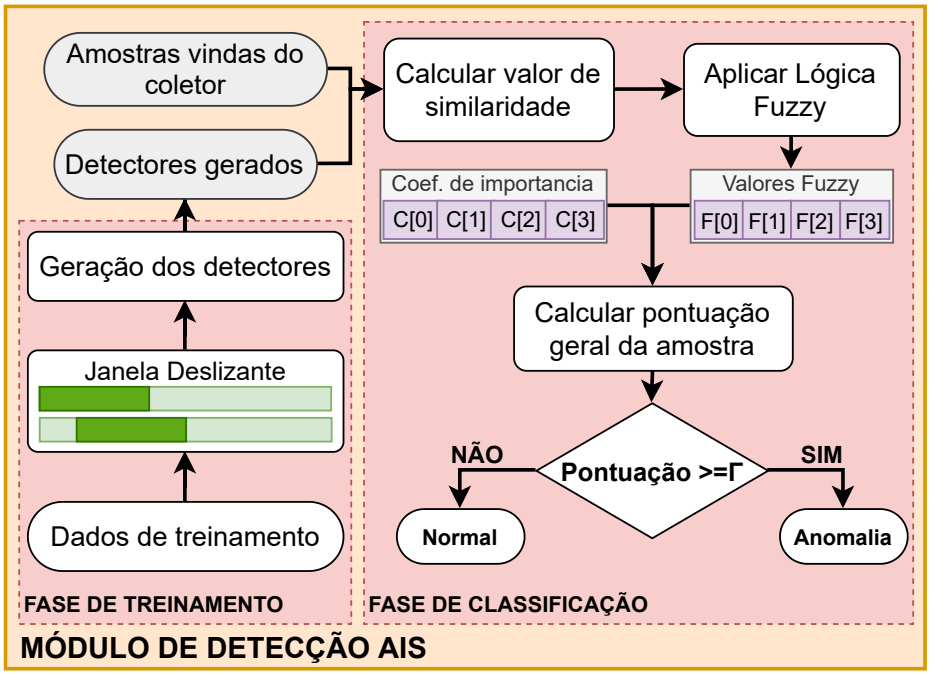


Figura 7 – Funcionamento do módulo de detecção AIS

Algorithm 2: Pseudo-código para a classificação das dados

Entrada: Amostra a ser classificada (nAmostra), coleção de detectores (Ω_t), coeficiente de importância (coefImportancia) e valor da similaridade mínima (k)
Saída : Resultado da Classificação

```
1  pontuação ← 0
2  foreach atributo ∈ nAmostra do
3  |  colAtributos ← Ω_t[atributo]
4  |  valorSimilaridade ← calcSimilaridade(atributo, colAtributos, k)
5  |  valoresFuzzy ← calcularFuzzy(valorSimilaridade, k, sigma)
6  |  pontuação ← pontuação + (valoresFuzzy × coefImportancia[atributo])
7  end
8  if pontuação > Γ then
9  |  return anomalia
10 end
11 else
12 |  return normal
13 end
```

4.2.2.1 Janela Deslizante

Os IDSs tradicionais normalmente usam vários dias como um conjunto de dados de treinamento. Assim, um novo comportamento pode demorar para ser reconhecido. Como resultado, novos padrões nos dados da rede podem não ser detectados rapidamente [102]. Para resolver esse problema, foi utilizada a janela deslizante. Essa técnica garante que o algoritmo AIS gere os detectores no tempo de convergência definido para a janela deslizante e possibilite a análise de novos fluxos o mais rápido possível. Outra vantagem é que apenas dados recentes são utilizados para a criação dos detectores, melhorando assim a capacidade de detecção, uma vez que, durante o dia, as flutuações normais do próprio comportamento da rede podem levar a detecções errôneas.

A Figura 8 exemplifica o funcionamento da janela deslizante utilizando o tamanho de janela w. Como pode ser visto, são empregados dois conjuntos de dados, um deles contendo o tráfego de um dia sem anomalias (d) e o outro com as medições realizadas periodicamente no dia atual (dA). O primeiro no processo de geração do detector e o último, na classificação. Por exemplo, as últimas w amostras do conjunto de dados d são aplicadas para gerar detectores e classificar a amostra $dA_{(w)}$.

Após a classificação da amostra em normal ou anormal, ambas as janelas deslizantes avançam uma posição para classificar a próxima amostra de entrada. Durante o início do processo, se as últimas w amostras não estiverem disponíveis para treinamento, todas as amostras acessíveis serão utilizadas até que o número seja atingido. Assim, o hiper-parâmetro w define o número de amostras utilizadas na fase de geração de detectores. Variações nesse valor afetam diretamente a qualidade da classificação e o tempo

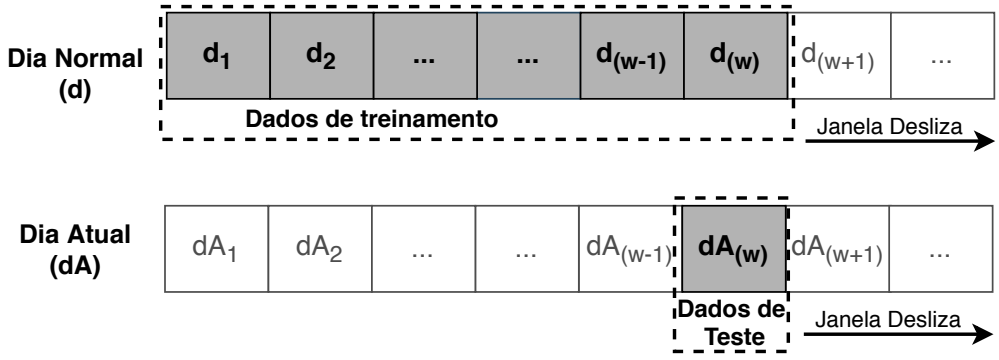


Figura 8 – Funcionamento da janela deslizante

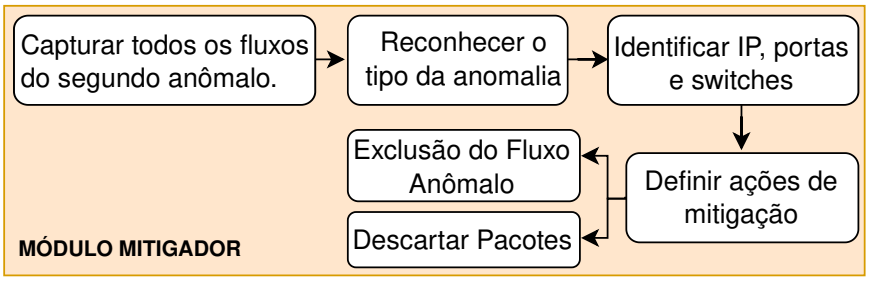


Figura 9 – Funcionamento do módulo mitigador

necessário para realizar tal tarefa.

Caso o comportamento da rede atual sofra alterações expressivas, como mudança na topologia, aumento significativo de computadores ou no uso dos recursos da rede, se faz necessário a atualização do dia sem anomalias (d) para que os detectores conheçam o novo comportamento comum.

4.3 Módulo Mitigador

Quando o módulo de detecção de AIS identifica a anomalia, encaminha todos os fluxos do intervalo anômalo ao módulo de mitigação a fim de aplicar políticas de descarte de pacotes e conter o ataque. DDoS e *port scan* podem alterar o comportamento dos atributos de tráfego, tornando possível descobrir endereços IP e portas em comunicações maliciosas.

Duas estratégias são definidas e usadas de acordo com o tipo de anomalia, ataques DDoS e *port scan*. O DDoS inunda um servidor com o objetivo de tornar um serviço online indisponível por meio de inúmeras solicitações enviadas de várias fontes. A primeira etapa desencadeada a fim de interromper o ataque é descobrir o endereço IP do host que está sendo atacado. Em seguida, são identificados todos os hosts que estão inundando a porta específica associada a um serviço oferecido pelo host atacado.

Após a identificação dos endereços IP maliciosos, o módulo de mitigação AIS-IDS

cria um fluxo com a ação de descarte visando bloquear pacotes do invasor. Esse fluxo é enviado do controlador que, por sua vez, encaminha os fluxos para switches no plano de dados via protocolo OpenFlow. Os pacotes de entrada correspondentes a esses fluxos de mitigação são descartados a fim de evitar a sobrecarga o alvo.

Os ataques *port scan* tentam descobrir serviços ativos, enviando mensagens para portas diferentes de um mesmo *host*. Quando um ataque *port scan* é detectado, tem-se início a estratégia de mitigação. Inicialmente é localizado o endereço do atacante através de inúmeros fluxos associados ao mesmo endereço IP de origem e destino, entretanto designado para diferentes portas de destino. A varredura de portas pode ser realizada de uma única fonte ou coordenada por vários adversários. O ataque de forma distribuída não altera significativamente o comportamento normal da rede, tornando assim, mais difícil sua detecção [103] [104].

Quando o endereço IP do invasor é identificado, a política permite a estratégia de descarte de todos os pacotes do *host* de origem mal-intencionada direcionados à vítima. O fluxo bloqueador criado tem a prioridade mais alta possível (65535) entre todos os fluxos, por isso é executado antes de todos os fluxos regulares de encaminhamento. O Algoritmo 3 apresenta o pseudo-código para o módulo de mitigação e a Figura 9 apresenta o esquema do módulo.

Algorithm 3: Pseudo-código para o módulo mitigador

Entrada: Amostras do momento anômalo (amostrasAnômalas)
Saída : Fluxo Mitigador

1 Identificar o endereço IP do alvo e portas através das amostrasAnômalas
2 Criar um fluxo mitigador
3 Definir endereço IP e portas de destino do fluxo como o endereço IP e portas da vítima
4 Definir ação de descarte dos fluxos no fluxo bloqueador
5 Definir a maior prioridade possível (65,535) para esse fluxo
6 Enviar o fluxo mitigador para o plano de dados através do protocolo OpenFlow

5 TESTES E RESULTADOS

Com o objetivo de avaliar o sistema proposto, foram elaborados três cenários, sendo os dois primeiros em ambiente emulado e o último utilizando uma base de dados pública. O primeiro é uma análise do AIS-IDS com relação a sua detecção e também da mitigação dos ataques. Por sua vez, o segundo cenário apresenta um comparativo da performance de detecção do AIS-IDS com outros algoritmos conhecidos no meio acadêmico, como Naive Bayes (NB), k-Nearest Neighbors (kNN), Random Forest (RF), Local Outlier Factor (LOF) e o Half-Space-Trees (HS) em um cenário com vários ataques DDoS e *port scan* de diferentes intensidades e durações. Por fim, o terceiro cenário avalia o AIS-IDS e os outros algoritmos já utilizados na base de dados pública CiCDDoS2019, a fim de validar a proposta apresentada. A Tabela 4 sintetiza todos os cenários elaborados e o objetivo de cada um.

Tabela 4 – Descrição dos cenários

Cenário	Objetivo
1	Avaliação da performance de detecção e mitigação do AIS-IDS, em ambiente emulado com ataques DDoS e *port scan*
2	Comparativo do AIS-IDS com os algoritmos NB, kNN, RF, LOF e HS, em ambiente emulado com ataques DDoS e *port scan* em várias configurações de duração, intensidade e sobreposição
3	Comparativo do AIS-IDS com os algoritmos NB, kNN, RF, LOF e HS, em base de dados pública CiCDDoS2019

Para a elaboração dos dois primeiros cenários, uma rede SDN foi emulada no Mininet [105], ferramenta que permite a emulação de uma estrutura SDN com switches, controladores, hosts e links em um único kernel Linux. As aplicações desenvolvidas no ambiente emulado podem ser implantadas em cenários reais com mudanças mínimas. O Mininet também oferece suporte ao Open vSwitch[1], um switch virtual multicamadas capaz de lidar com a comutação baseada em software e hardware e ainda oferecer suporte ao protocolo OpenFlow.

Todas as funções necessárias para monitoramento do tráfego, extração de atributos, detecção e mitigação de anomalias foram realizadas por aplicações incorporadas ao Floodlight[2], um controlador SDN de código aberto baseado em Java. Toda a topologia foi testada em um computador com um processador de seis núcleos, com frequência de 2.6 GHz, 32 GB de memória RAM, executando o sistema operacional Ubuntu 18.04. A versão do Mininet utilizada foi a 2.2.2 e o Floodlight na versão 1.2.

[1] https://www.openvswitch.org/
[2] https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/

A topologia emulada é composta por seis switches Open vSwitch em árvore, sessenta hosts e um controlador SDN, como pode ser visto na Figura 10. O equipamento rotulado como s1 é o switch raiz e todos os outros switches foram conectados a ele. Em cada switch do segundo nível da árvore foram conectados doze hosts. O controlador Floodlight comunica-se diretamente com cada switch por meio do protocolo OpenFlow.

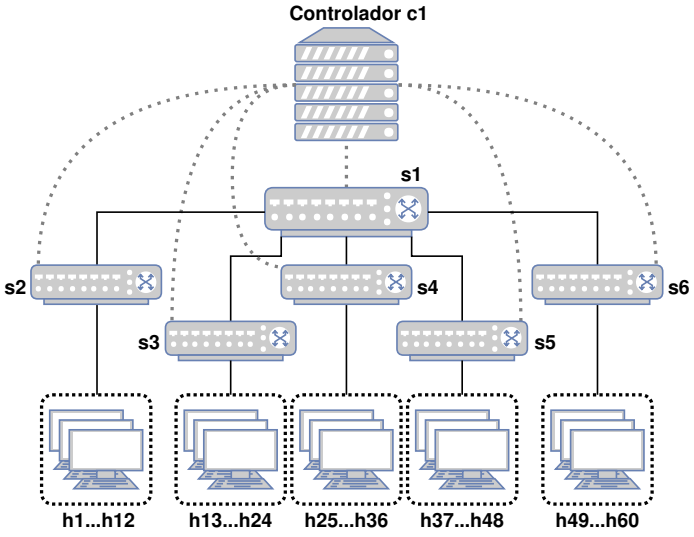


Figura 10 – Topologia Emulada no Mininet

A ferramenta Scapy[3], na versão 2.4.0, foi utilizada para gerar o tráfego normal na rede emulada. Scapy é uma ferramenta poderosa para ser empregada em um ambiente de teste, forjando pacotes e enviando-os por meio da interface de rede. Os fluxos que compunham o tráfego da rede foram produzidos aleatoriamente e o volume de dados foi alterado ao longo do dia para simular variações no uso legítimo da rede.

Ataques DDoS foram realizados usando o software hping3 [4]. Várias fontes (endereços IP de origem) foram configuradas com o intuito de criarem diversas solicitações direcionadas a um único destino na rede emulada. Um host na rede foi vítima de ataques *port scan*, conduzidos por um host utilizando Scapy. Nesse caso, o invasor enviou pacotes com o *flag SYN* habilitado em direção a diferentes portas do host de destino, tentando receber a confirmação das portas que estavam em operação.

Para verificar a eficiência do modelo proposto nos cenários em estudo, foram utilizadas cinco métricas para avaliação: acurácia, precisão, recall, taxa de falsos positivos e *f-measure*. Acurácia (ACC) avalia a proporção dos intervalos corretamente classificados. A segunda métrica, precisão (PREC), enfatiza a detecção de intervalos anômalos e penaliza os intervalos legítimos classificados erroneamente. Portanto essa métrica complementa as

[3] https://scapy.net/
[4] https://github.com/antirez/hping

informações fornecidas pela acurácia, apresentando resultados adequados quando as classes não são representadas igualmente (desbalanceamento). O recall (REC), ou revocação, indica a proporção de amostras anômalas classificadas corretamente de todas as amostras anômalas. A taxa de Falso Positivo (FPR) indica a proporção de amostras normais classificadas incorretamente de todas as amostras normais. Finalmente, o *f-measure* consiste em uma pontuação geral dada ao classificador. Essa pontuação é obtida pela média harmônica entre precisão e revocação. O resultado das métricas usadas para avaliar o esquema de detecção varia de 0 a 1, em que o primeiro é o pior cenário e o último representa o valor ideal.

5.1 Cenário 1: Avaliação do AIS-IDS

Para o primeiro cenário foram gerados e coletados dois dias de dados em um ambiente emulado. O primeiro correspondeu a um dia com tráfego sem anomalias e o segundo foi realizado um ataque DDoS e um ataque de *port scan*. O dia sem ataques foi utilizado como conjunto de dados de treinamento para o AIS-IDS, já o segundo dia foi o conjunto de dados de teste. Os conjuntos de dados gerados foram disponibilizados para o uso público[5]. As informações completas dos dias coletados são apresentadas na Tabela 5.

Tabela 5 – Descrição dos dias coletados para o Cenário 1

Dia	Primeiro Ataque	Segundo Ataque
1	Sem Ataque	Sem Ataque
2	Tipo do ataque: DDoS IPs Atacantes: 10.0.0.3-10.0.0.11 IP e porta do alvo: 10.0.0.42:443 Duração: 10:15:00 - 11:30:00	Tipo do ataque: *Port scan* IP Atacante: 10.0.0.48 IP e portas do alvo: 10.0.0.8:1-30000 Intervalo entre pacotes: 0.25 Duração: 13:25:00 - 14:35:00

A avaliação do AIS-IDS é dividida em duas etapas. A primeira trata da definição dos valores dos hiper-parâmetros usados pelo IDS. Após essa definição, o AIS-IDS é testado quanto a sua capacidade de detecção e mitigação dos eventos anômalos.

5.1.1 Avaliação dos valores dos hiper-parâmetros

O AIS-IDS possui alguns hiper-parâmetros que precisam ser definidos para uma boa performance. O tamanho da janela deslizante é representado por w, k é o valor de similaridade mínima, n é o número de detectores desejado e Γ é o valor limite de corte para se considerar uma amostra legítima ou não.

Para encontrar o melhor valor para w, foram realizados testes variando este hiper-parâmetro e avaliados o valor de *f-measure* obtido. Os valores são mostrados na Figura 11.

[5] http://www.uel.br/grupos/orion/datasets.html

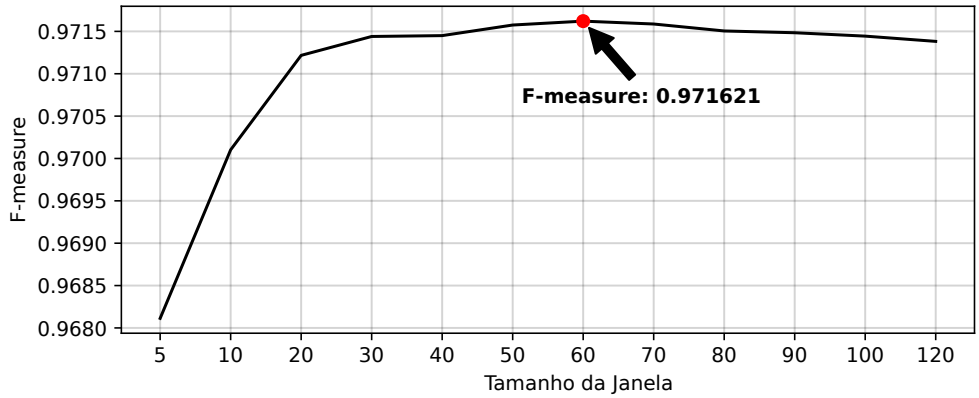


Figura 11 – Testes sobre o tamanho da janela deslizante

As melhores performances foram obtidas quando o tamanho da janela foi ajustado para 60 segundos. Portanto, o hiper-parâmetro w será considerado 60 para os experimentos.

Uma pesquisa em grade (*grid search*) foi conduzida com o intuito de definir os melhores valores dos hiper-parâmetros n e k. Nos testes realizados, os valores de n variaram de 30 até 140, e o k de 0,1 até 0,4. O valor de *f-measure* foi utilizado como métrica para comparar a performance geral de cada teste.

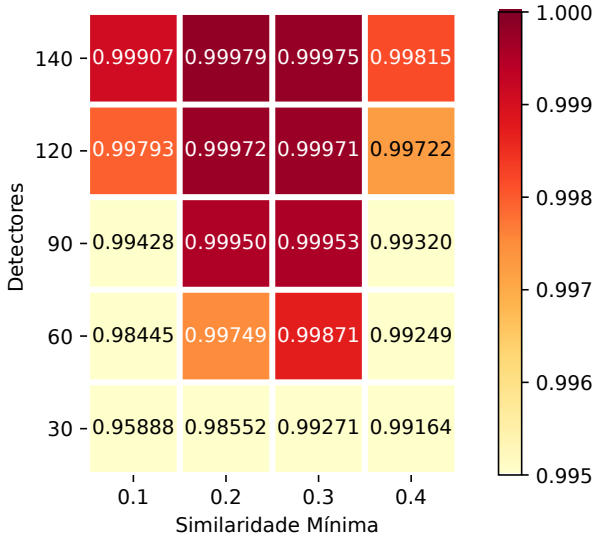


Figura 12 – Mapa de calor da pesquisa em grade

Todos os resultados estão na Figura 12. O gráfico mapa de calor apresenta os resultados de cada teste. Com este gráfico, foi possível verificar que a partir de 90 detectores, os resultados obtidos pela similaridade mínima 0,2 e 0,3 são muito próximos. O melhor resultado foi obtido usando 140 detectores e a similaridade mínima de 0,2. Assim, os valores dos hiper-parâmetros de k e n para as próximas avaliações foram definidos como 0,2 e 140, respectivamente.

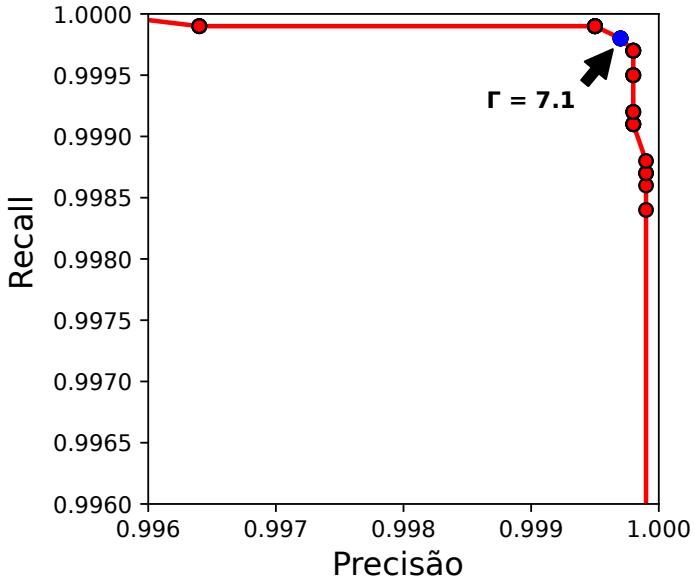


Figura 13 – Curva ROC para estimar valor de Γ.

O último hiper-parâmetro, Γ, corresponde ao limite de corte e foi definido usando uma curva ROC, conforme ilustrado na Figura 13.

A curva ROC (do inglês, ROC - *Receiver Operating Characteristic Curve*) é também outra métrica para avaliar os resultados de um algoritmo. Essa curva apresenta os valores de precisão e recall de cada valor de corte de 0 a 10,5, variando de 0,1. Assim, quanto mais próximo a 1 em ambos os eixos, melhor é a performance do algoritmo. No teste realizado, o melhor valor alcançado foi 7,1, portanto esse foi o valor atribuído a Γ nos experimentos de avaliação.

5.1.2 Avaliação do desempenho de detecção e mitigação

Após todos os hiper-parâmetros definidos, o AIS-IDS foi executado. A Figura 14 evidencia o comportamento de cada atributo coletado dos fluxos da rede durante o dia analisado. Os intervalos detectados como anômalos pelo AIS-IDS foram destacados em vermelho. Conforme representado na figura, foi possível observar que a maioria dos intervalos em que o tráfego foi afetado pelos ataques foi detectada corretamente. Porém alguns períodos sem ataques foram considerados anômalos (falsos-positivos) e também os intervalos com ataques foram classificados como normais (falsos-negativos). Em geral, o IDS produziu taxas de detecção confiáveis, o valor de *f-measure* foi superior a 0,9997, ou seja, 99,97%.

Outra análise é apresentada no gráfico de dispersão, que demonstra como cada atributo se comportou durante o dia de teste. A Figura 15 demonstra os valores *fuzzy* de cada atributo analisado no dia de teste. Para melhor visualização, cada ponto representa

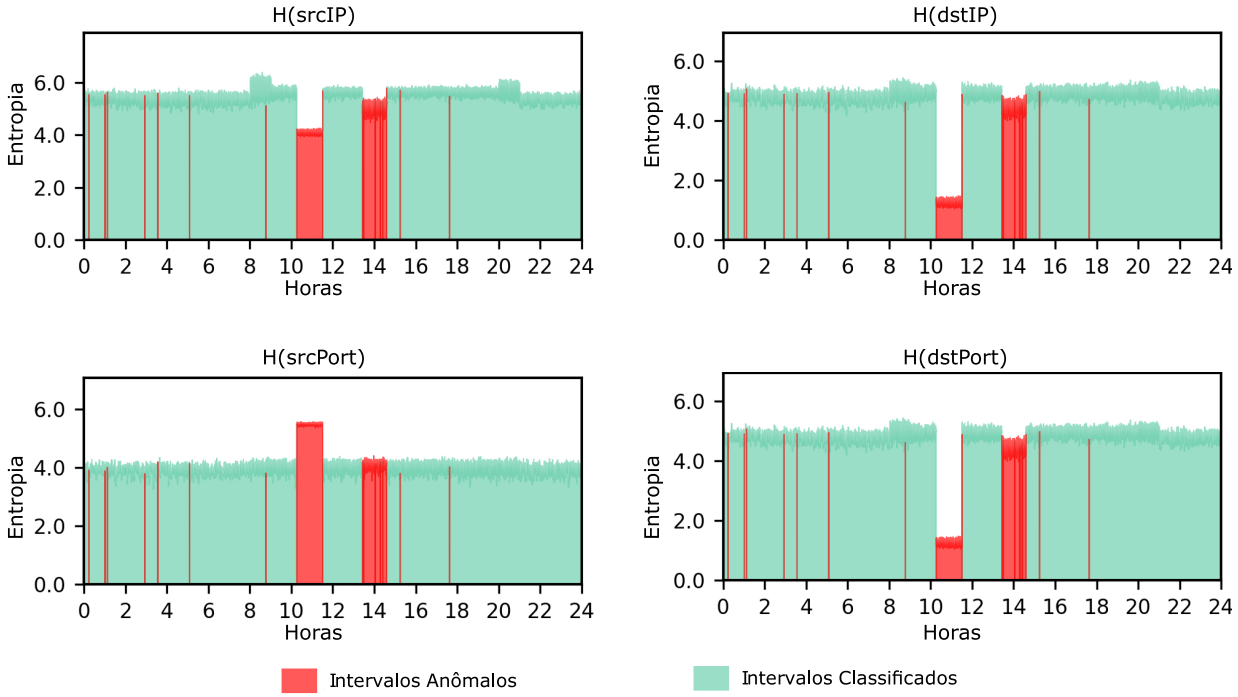


Figura 14 – Classificação do dia de teste.

um agrupamento de 60 segundos. No momento em que ocorrem os ataques, é possível notar que os valores *fuzzy* em todos os atributos são altos, apresentando variação aparente e demonstrando os ataques presentes nesses intervalos.

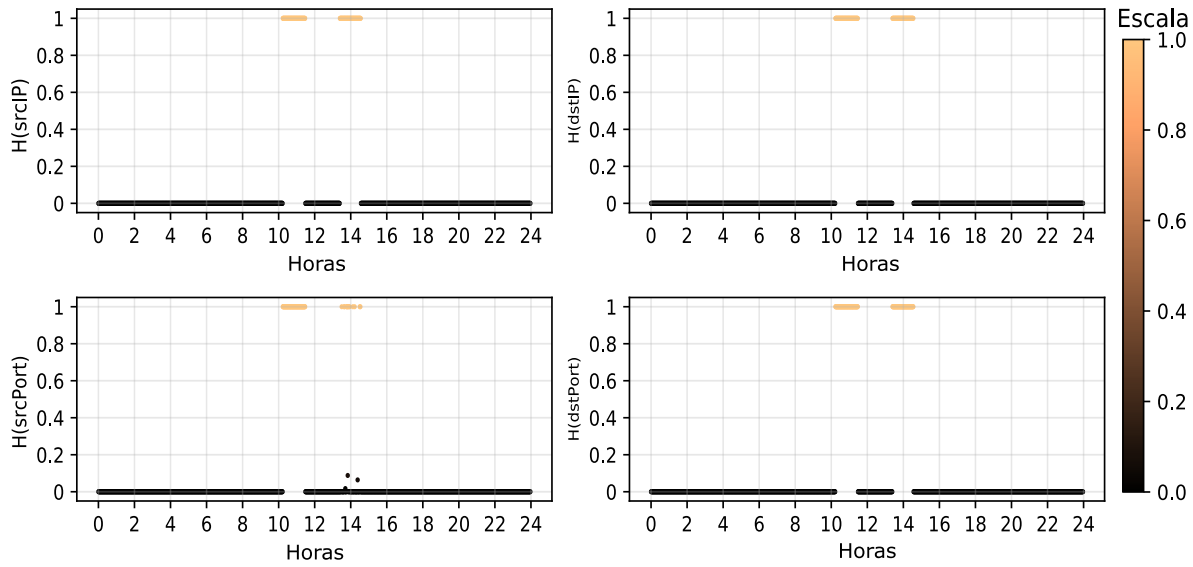


Figura 15 – Gráfico de dispersão dos valores *fuzzy* do dia de teste

Para avaliar o resultado da detecção e mitigação dos ataques por parte do AIS-IDS, o módulo de detecção AIS foi configurado para disparar um alarme e informar ao módulo de mitigação. A Figura 16 apresenta em verde o tráfego após a mitigação e as linhas vermelhas são o tráfego antes de tal processo. Foi possível perceber que variações

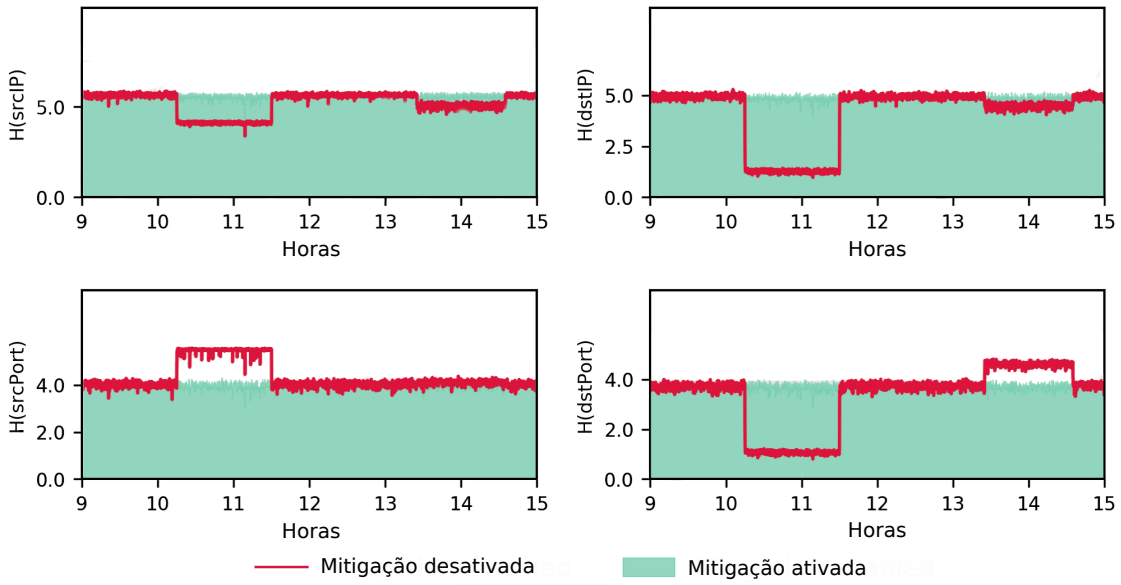


Figura 16 – Resultado do módulo de mitigação

nos atributos foram eliminadas após o processo de mitigação. Esse resultado confirma que os fluxos de ataque foram bloqueados corretamente e que os fluxos normais foram mantidos.

5.2 Cenário 2: Comparativo com outros métodos

Nesta seção, o AIS-IDS foi comparado com os algoritmos Naive Bayes (NB) [106], k-Nearest Neighbors (kNN) [107] e o algoritmo Random Forest (RF) [108] foram selecionados para esse propósito. Esses algoritmos foram empregados pois seguem diferentes paradigmas de aprendizagem com vieses de aprendizagem variados. Além dos algoritmos citados, também foram utilizados os algoritmos Local Outlier Factor (LOF) [109] e o Half-Space-Trees (HS) [110], ambos foram escolhidos por serem semelhantes ao AIS-IDS, ou seja, não utilizam amostras de anomalias para treinamento. Todos os classificadores avaliados na comparação foram implementados usando a linguagem Python, na versão 3.5. Os algoritmos KNN, RF, LOF e NB utilizaram a biblioteca scikit-learn, na versão 0.21.3, já o algoritmo HS utilizou uma implementação própria.

Para a execução deste cenário foram coletados oito dias de tráfego. Os dois primeiros dias foram utilizados como conjunto de treinamento, sendo o primeiro dia sem anomalias e o segundo dia com anomalias. O segundo dia contém ataques DDoS e *port scan* em várias configurações de intensidade e duração. Além disso, possui ataques sobrepostos a fim de mapear as mais variadas possibilidades de ataques possíveis. De modo geral, o primeiro dia foi utilizado pelas propostas que não necessitam de amostras anômalas, já o segundo dia foi utilizado pelos algoritmos que necessitam de tais dados. A descrição desses dois dias está apresentado na Tabela 6. Já cada um dos outros seis dias

Tabela 6 – Descrição dos dias de treinamento coletados para o Cenário 2

Dia	Descrição do dia
1	Sem ataques durante todo o dia
2	**Primeiro Ataque:** DDoS, **Duração:** 09:01:48 - 09:08:01 **IPs Atacantes:** 10.0.0.45 - 10.0.0.54, **IP e porta do alvo:** 10.0.0.15:8000 **Segundo Ataque:** *Port scan*, **Duração:** 10:03:56 - 11:04:56 **IP Atacante:** 10.0.0.30, **IP e portas do alvo:** 10.0.0.8:1-29999 **Intervalo entre pacotes:** 0.05 **Terceiro Ataque:** DDoS, **Duração:** 11:28:18 - 11:45:38 **IPs Atacantes:** 10.0.0.2 - 10.0.0.46, **IP e porta do alvo:** 10.0.0.48:9990 **Quarto Ataque:** *Port scan*,**Duração:** 11:37:51 - 12:35:15 **IP Atacante:** 10.0.0.1, **IP e portas do alvo:** 10.0.0.50:1-8133 **Intervalo entre pacotes:** 0.5 **Quinto Ataque:** DDoS, **Duração:** 16:31:26 - 17:41:57 **IPs Atacantes:** 10.0.0.3 - 10.0.0.47, **IP e porta do alvo:** 10.0.0.4:1180 **Sexto Ataque:** *Port scan*, **Duração:** 16:31:26 - 17:41:57 **IP Atacante:** 10.0.0.60, **IP e portas do alvo:** 10.0.0.1:1-64198 **Intervalo entre pacotes:** 0.05

coletados possuem dois ataques, um *port scan* e outro DDoS, com diferentes configurações de duração e intensidade. Cada um desses seis dias foi considerado um dia de análise e a descrição dos ataques desses dias está exposta na Tabela 7.

Assim como ocorreu com o AIS-IDS, também foi realizada uma pesquisa em grade (*grid search*) para os algoritmos que necessitam de ajustes de hiper-parâmetros. Essa avaliação considerou os melhores hiper-parâmetros para as 6 bases da avaliação. O resultado do kNN foi mensurado variando-se o número de vizinhos mais próximos k de 1 a 40, e a melhor resposta apresentou-se quando $k = 25$. Em relação ao algoritmo Random Forest (RF), foi considerado o número de estimadores (árvores) de 1 a 100, e a melhor performance foi alcançada com o número de árvores definido em 10. Para o LOF, o número de vizinhos de 1 a 60 e os valores de contaminação de 0,05 a 0,4 foram examinados. Os melhores resultados foram obtidos com o número de vizinhos definido em 45 e o valor de contaminação em 0,05. Após a definição dos hiper-parâmetros, os algoritmos foram avaliados em cada uma das 6 bases de avaliação que compõem este cenário.

O primeiro dia de teste possui os ataques de longa duração e alta intensidade e os ataques não estão sobrepostos, portanto, pode ser considerado o dia de mais fácil detecção. Os resultados obtidos por todos os algoritmos neste primeiro dia estão apresentados na Tabela 8.

Tabela 7 – Descrição dos dias de teste coletados para o Cenário 2

Dia	Descrição do dia
1	**Primeiro ataque:** DDoS, **Duração:** 11:03:36 - 12:05:07 **IPs Atacantes:** 10.0.0.3 - 10.0.0.47, **IP e porta do alvo:** 10.0.0.50:2020 **Segundo ataque:** *Port scan*, **Duração:** 17:28:37 - 18:39:22 **IP Atacante:** 10.0.0.15, **IP e portas do alvo:** 10.0.0.29:1-41243 **Intervalo entre pacotes:** 0.05
2	**Primeiro ataque:** DDoS, Duração: 10:31:48 - 10:43:12 **IPs Atacantes:** 10.0.0.45 - 10.0.0.54, **IP e porta do alvo:** 10.0.0.15:8000 **Segundo ataque:** *Port scan*, **Duração:** 13:31:59 - 14:34:00 **IP Atacante:** 10.0.0.33, **IP e portas do alvo:** 10.0.0.55:1-6817 **Intervalo entre pacotes:** 0.5
3	**Primeiro ataque:** *Port scan*, **Duração:** 14:04:23 - 15:20:54 **IP Atacante:** 10.0.0.59, **IP e portas do alvo:** 10.0.0.50:1-55320 **Intervalo entre pacotes:** 0.05 **Segundo ataque:** DDoS, **Duração:** 15:02:59 - 15:55:16 **IPs Atacantes:** 10.0.0.5 - 10.0.0.49, **IP e porta do alvo:** 10.0.0.1:8090
4	**Primeiro ataque:** *Port scan*, **Duração:** 14:15:17 - 15:31:48 **IP Atacante:** 10.0.0.13 , **IP e portas do alvo:** 10.0.0.42:1-55320 **Intervalo entre pacotes:** 0.05 **Segundo ataque:** DDoS, **Duração:** 15:13:53 - 16:14:28 **IPs Atacantes:** 10.0.0.30 - 10.0.0.39, **IP e porta do alvo:** 10.0.0.43:3790
5	**Primeiro ataque:** DDoS, **Duração:** 14:57:25 - 16:00:22 **IPs Atacantes:** 10.0.0.10 - 10.0.0.19, **IP e porta do alvo:** 10.0.0.7:7222 **Segundo ataque:** *Port scan*, **Duração:** 14:57:26 - 15:59:15 **IP Atacante:** 10.0.0.2, **IP e portas do alvo:** 10.0.0.4:1-65534 **Intervalo entre pacotes:** 0.05
6	**Primeiro ataque:** *Port scan*, **Duração:** 14:12:13 - 15:13:46 **IP Atacante:** 10.0.0.2, **IP e portas do alvo:** 10.0.0.10:1-12620 **Intervalo entre pacotes:** 0.5 **Segundo ataque:** DDoS, **Duração:** 14:36:42 14:47:19 **IPs Atacantes:** 10.0.0.16-10.0.0.60, **IP e porta do alvo:** 10.0.0.9:5798

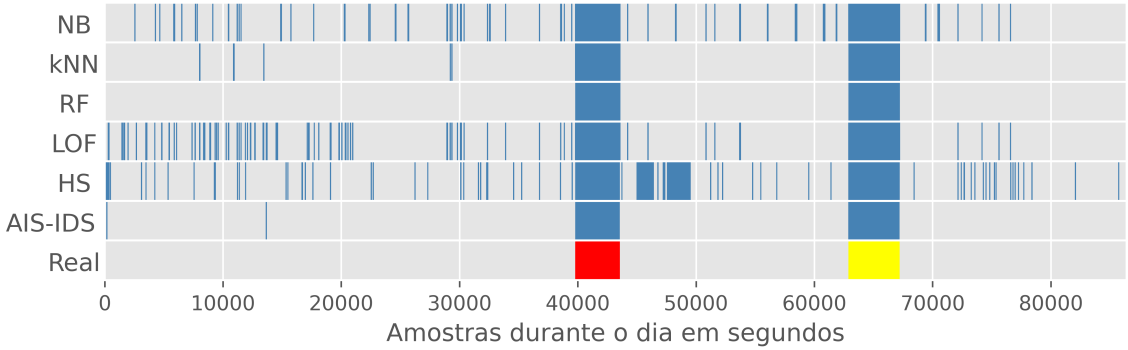


Figura 17 – Resultado da detecção dos algoritmos no primeiro dia

Tabela 8 – Comparativo de performance dos algoritmos no primeiro dia

	AIS-IDS	RF	KNN	NB	LOF	HS
ACURÁCIA	0,9996	0,999	0,998	0,9829	0,99	0,9499
PRECISÃO	0,9975	0,9912	0,981	0,8538	0,9022	0,6793
RECALL	0,9985	0,9979	0,9977	0,9824	0,9995	0,8618
F-MEASURE	0,998	0,9945	0,9893	0,9136	0,9484	0,7598
AUC	0,9991	0,9985	0,9979	0,9827	0,9943	0,9103
FP-rate	0,0003	0,0009	0,0019	0,017	0,011	0,0411

As seis primeiras linhas da Figura 17 apresentam os valores das cinco métricas de avaliação para cada algoritmo comparado. Os momentos anômalos identificados por cada proposta são representados por barras azuis. A última linha, denominada “Real”, revela os momentos em que realmente ocorreram anomalias. Nesta linha, os ataques DDoS foram marcados pela cor vermelha e os ataques *port scan* representados pela cor amarela. O eixo x representa os 86400 intervalos de um segundo analisados durante as 24 horas desse dia da base.

Dentre os algoritmos utilizados, todos conseguiram detectar os momentos anômalos, contudo alguns algoritmos obtiveram muitos falsos positivos durante esse primeiro dia de teste, como foi o caso do NB, LOF e o HS. Também foi possível averiguar que alguns algoritmos tiveram dificuldades em detectar o comportamento normal durante as primeiras horas do dia. Nesse quesito, pode-se citar os algoritmos NB e LOF que, somente após o primeiro ataque, reduziram a taxa de falsos positivos. No geral, os algoritmos que obtiveram a melhor performance foram o AIS, kNN e RF, respectivamente.

Considerando o segundo dia de teste, ambos os ataques são de baixa intensidade, sendo que o ataque DDoS possui curta duração e o *port scan* é de longa duração. Os ataques também são totalmente separados entre si. Isso posto, o segundo dia é mais complexo do que o anterior, sobretudo por conta dos ataques serem menos intensos e também pelo fato do ataque DDoS ser de curta duração. A Figura 18 e a Tabela 9 apresentam os resultados referentes ao segundo dia.

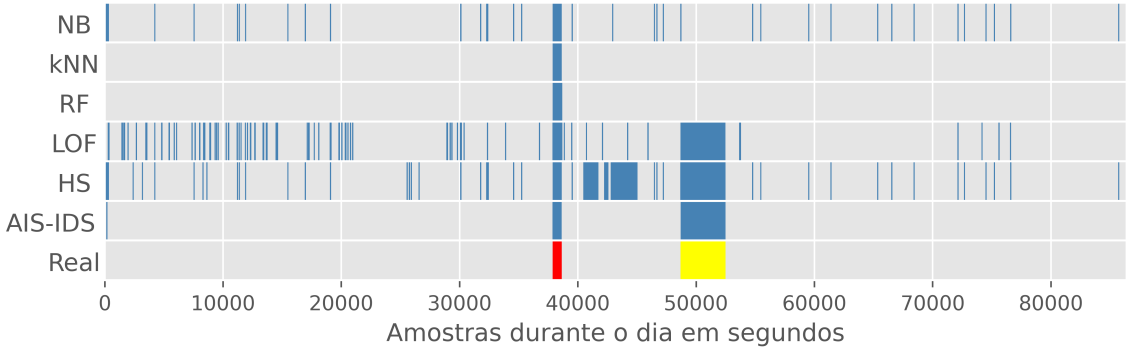


Figura 18 – Resultado da detecção dos algoritmos no segundo dia

Tabela 9 – Comparativo da performance dos algoritmos no segundo dia

	AIS-IDS	RF	KNN	NB	LOF	HS
ACURÁCIA	0,9996	0,9564	0,9569	0,9564	0,9906	0,9439
PRECISÃO	0,9966	0,9392	1	0,9421	0,8486	0,4677
RECALL	0,9957	0,1544	0,1553	0,1551	0,9934	0,7301
F-MEASURE	0,9961	0,2652	0,2688	0,2663	0,9134	0,5702
AUC	0,9978	0,5769	0,5776	0,5773	0,9919	0,8427
FP-rate	0,0002	0,0005	0	0,0005	0,0095	0,0446

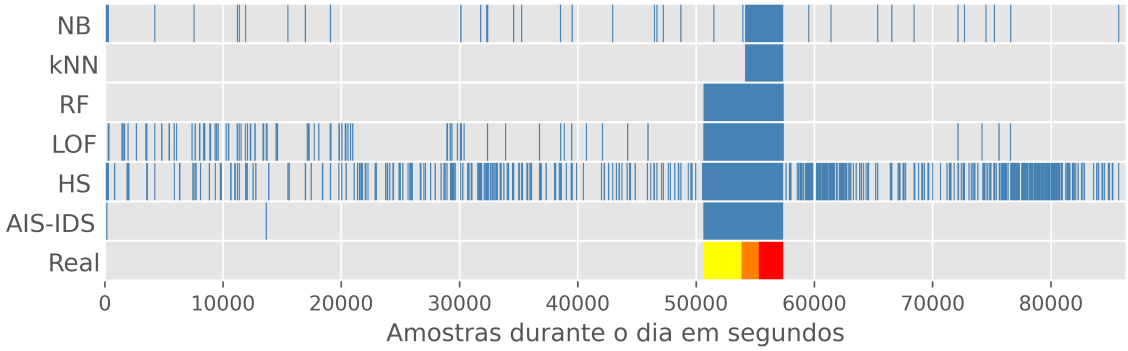


Figura 19 – Resultado da detecção dos algoritmos no terceiro dia

Explorando os resultados obtidos pelos algoritmos, todos detectaram o ataque DDoS corretamente, porém o ataque *port scan*, mesmo que mais longo que o DDoS, não foi detectado pelos algoritmos que necessitam de amostras anômalas para treinamento, como NB, kNN e RF. Desse modo, a vantagem neste cenário foi dos algoritmos que não precisam de amostras anômalas para realizar a detecção. Os algoritmos LOF e HS foram os que obtiveram maiores taxas de falsos positivos. Com relação à performance em geral, o AIS, LOF e HS foram os algoritmos que obtiveram os melhores resultados.

O terceiro dia de teste possui dois ataques de alta intensidade e longa duração, entretanto os ataques agora estão parcialmente sobrepostos. Dessa maneira, a dificuldade é detectar o momento anômalo quando ambos os ataques estão em execução. Esses momentos foram representados pela cor laranja na Figura 19. A Tabela 10 apresenta os resultados referentes ao terceiro dia.

Tabela 10 – Comparativo da performance dos algoritmos no terceiro dia

	AIS-IDS	RF	KNN	NB	LOF	HS
ACURÁCIA	0,9993	0,9994	0,9592	0,9589	0,9909	0,9889
PRECISÃO	0,9961	0,9931	1	0,9871	0,8943	0,9009
RECALL	0,9943	0,9994	0,4702	0,4718	0,9998	0,9617
F-MEASURE	0,9952	0,9963	0,6396	0,6385	0,9441	0,9303
AUC	0,997	0,9994	0,7351	0,7357	0,995	0,9764
FP-rate	0,0003	0,0006	0	0,0005	0,0099	0,0088

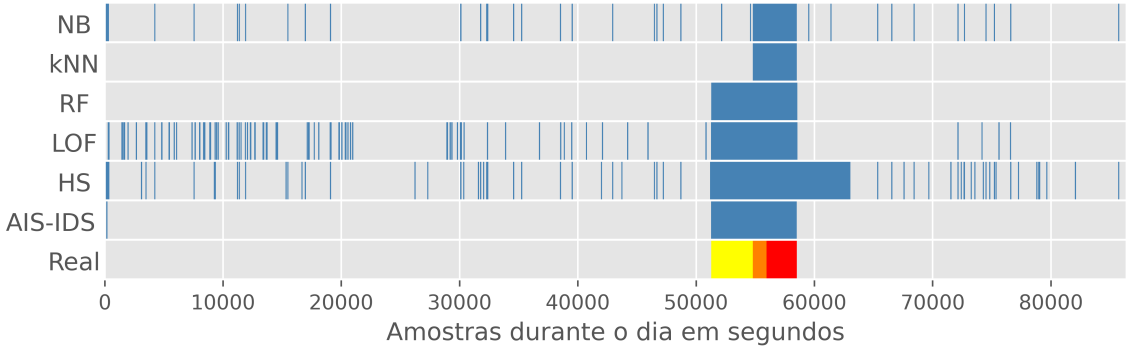


Figura 20 – Resultado da detecção dos algoritmos no quarto dia

Tabela 11 – Comparativo da performance dos algoritmos no quarto dia

	AIS-IDS	RF	KNN	NB	LOF	HS
ACURÁCIA	0,9993	0,9994	0,9592	0,9586	0,9908	0,9445
PRECISÃO	0,9972	0,9939	1	0,9888	0,9005	0,6233
RECALL	0,9948	0,9994	0,5075	0,5057	0,9999	0,8332
F-MEASURE	0,996	0,9967	0,6733	0,6691	0,9476	0,7131
AUC	0,9973	0,9994	0,7537	0,7526	0,9949	0,8939
FP-rate	0,0003	0,0006	0	0,0005	0,01	0,0454

Avaliando o terceiro dia foi possível averiguar que os algoritmos NB e kNN não conseguiram detectar o ataque *port scan*, apenas o ataque DDoS. Revelando, dessa maneira, que quando os ataques de alta intensidade e longa duração são sobrepostos, dificultam a detecção dos ataques por esses algoritmos. Já os algoritmos LOF, RF, AIS-IDS e HS conseguiram detectar ambos os ataques sem problemas e até mesmo no momento em que os dois ataques ocorreram. Em relação à taxa de falsos positivos, os algoritmos que obtiveram os piores resultados foram o LOF e HS. Dos algoritmos avaliados, a RF e o AIS-IDS alcançaram os melhores resultados.

O quarto dia de avaliação é semelhante ao terceiro dia, diferenciando-se apenas em um aspecto: o ataque DDoS foi realizado com baixa intensidade. Em vista disso, o objetivo desse dia é avaliar a diferença que a intensidade do ataque DDoS pode acarretar na detecção feita pelos algoritmos. A Figura 20 e a Tabela 11 apresentam os resultados referentes ao quarto dia.

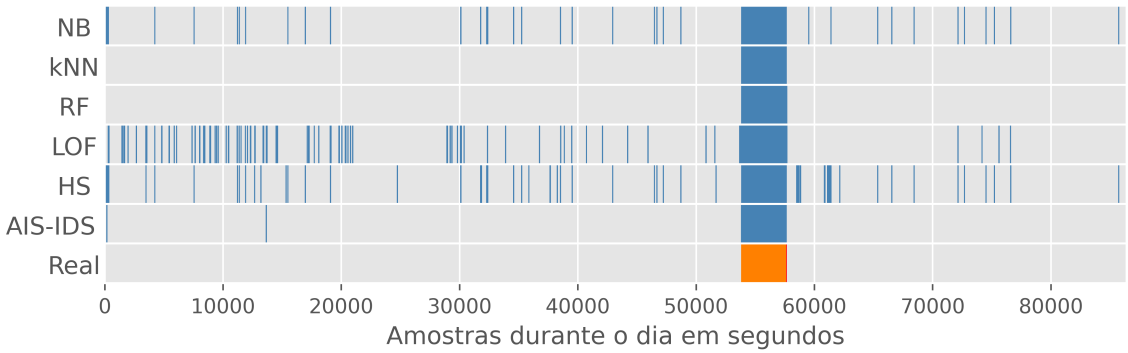


Figura 21 – Resultado da detecção dos algoritmos no quinto dia

Tabela 12 – Comparativo da performance dos algoritmos no quinto dia

	AIS-IDS	RF	KNN	NB	LOF	HS
ACURÁCIA	0,9997	0,9994	0,9999	0,9992	0,9907	0,9982
PRECISÃO	0,9947	0,9885	0,9997	0,9892	0,8248	0,9704
RECALL	0,9989	0,9987	0,9981	0,9929	0,9995	0,9889
F-MEASURE	0,9968	0,9935	0,9989	0,991	0,9038	0,9795
AUC	0,9993	0,9991	0,9991	0,9962	0,9949	0,9938
FP-rate	0,0002	0,0005	0	0,0005	0,0097	0,0014

Examinando os resultados do quarto dia, os valores obtidos pelos algoritmos NB e kNN foram semelhantes ao dia anterior, ou seja, apenas o ataque DDoS foi detectado. Desse modo, foi possível verificar que a diferença na intensidade do ataque DDoS não afetou a detecção dos momentos anômalos. Mesmo o ataque *port scan* sendo mais intenso que o DDoS, tanto o NB quanto o kNN não conseguiram realizar a detecção.

O algoritmo HS obteve uma taxa de falso positivo bem mais elevada se comparado às outras propostas. Isso se deve ao fato que o algoritmo acabou detectando erroneamente o final do período anômalo. Já os outros algoritmos conseguiram detectar os momentos de ataque corretamente. Os algoritmos RF, AIS-IDS e LOF foram os que obtiveram os melhores resultados.

O quinto dia de teste avaliado possui dois ataques de longa duração, sendo que o ataque DDoS possui baixa intensidade e o ataque *port scan* é de alta intensidade. O objetivo deste dia é verificar se a ocorrência simultânea desses ataques dificulta a classificação do período como anômalo. A Figura 21 e a Tabela 12 apresentam as métricas avaliadas para cada algoritmo referentes ao quinto dia.

Verificando os dados obtidos pelos algoritmos, constata-se que todos foram capazes de detectar o período anômalo. A execução de ambos os ataques em simultâneo acabou não prejudicando a detecção em nenhum dos algoritmos. Em relação aos falsos positivos, os algoritmos kNN, AIS, RF e NB obtiveram os melhores desempenhos, sendo que o kNN não classificou nenhuma amostra como falso positivo. No geral, os algoritmos kNN,

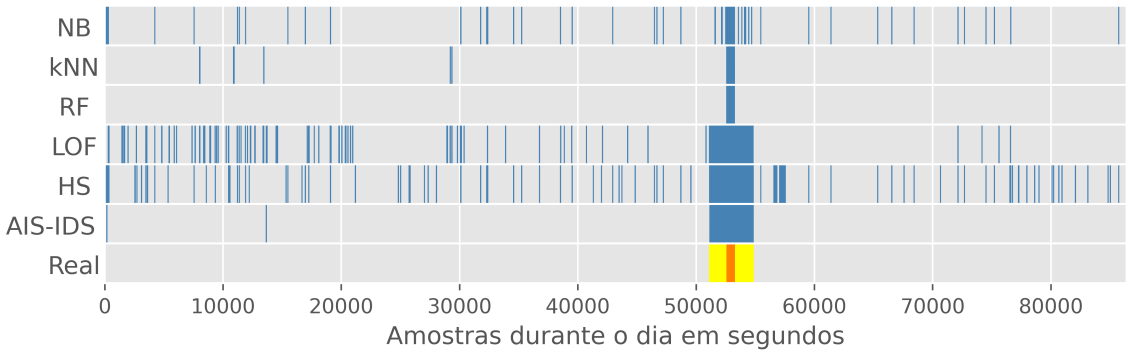


Figura 22 – Resultado da detecção dos algoritmos no sexto dia

Tabela 13 – Comparativo da performance dos algoritmos no sexto dia

	AIS-IDS	RF	KNN	NB	LOF	HS
ACURÁCIA	0,9996	0,9648	0,9642	0,9643	0,9912	0,9872
PRECISÃO	0,997	1	0,9252	0,9394	0,8332	0,8048
RECALL	0,9932	0,1771	0,1776	0,1763	0,9927	0,9242
F-MEASURE	0,9951	0,3009	0,298	0,2969	0,906	0,8603
AUC	0,9965	0,5885	0,5885	0,5879	0,9919	0,9571
FP-RATE	0,0001	0	0,0006	0,0005	0,0089	0,01

RF e AIS-IDS foram os que obtiveram os melhores resultados. Portanto, conclui-se que esses algoritmos foram capazes de classificar tanto os períodos normais quanto os períodos anômalos com uma alta taxa de eficácia.

Por fim, o sexto e último dia de teste possui um ataque *port scan* de longa duração e baixa intensidade e um ataque DDoS de alta intensidade e curta duração. Além disso, o ataque DDoS está totalmente compreendido dentro do período de execução do ataque *port scan*, ou seja, ocorreu sobreposição total dos ataques. O objetivo deste dia é averiguar se um ataque DDoS de alta intensidade irá atrapalhar na detecção do ataque *port scan*, tanto quando o ataque DDoS se iniciar, quanto no seu término. A Figura 22 e a Tabela 13 apresentam os resultados referentes ao sexto dia.

Ponderando sobre os resultados dos algoritmos no sexto dia analisado, novamente os algoritmos NB, kNN e RF conseguiram detectar o ataque DDoS corretamente, mas não o ataque *port scan*. Vale mencionar que o NB ainda foi capaz de identificar alguns momentos do ataque *port scan*, contudo não conseguiu detectar todo o momento do momento do ataque. Já os algoritmos LOF, AIS-IDS e HS classificaram os momentos de ambos os ataques corretamente. Neste dia avaliado, novamente os algoritmos que não utilizam momentos anômalos obtiveram vantagem na detecção dos ataques, mesmo que em baixa intensidade.

Analisando os valores de *f-measure* que cada algoritmo obteve nos seis dias de teste, o AIS-IDS alcançou os melhores resultados, alcançando um valor médio de 99,63%.

Tabela 14 – Taxas de acertos dos algoritmos no segundo cenário em porcentagem (%)

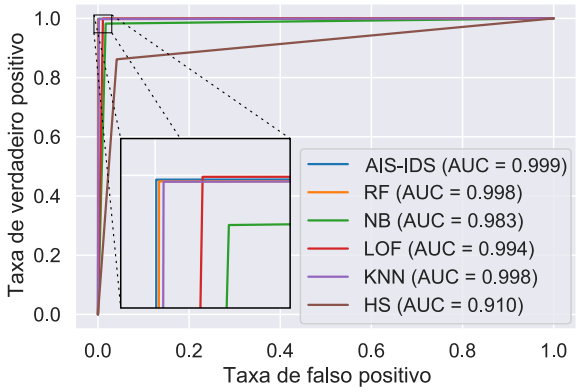
		AIS-IDS	NB	RF	HS	kNN	LOF
DIA 1	NORMAL	**99,97**	98,29	99,91	95,88	99,80	98,90
	DDOS	99,81	**99,97**	99,89	99,91	**99,97**	**99,97**
	PS	99,88	96,72	99,69	74,22	99,60	**99,92**
DIA 2	NORMAL	99,98	99,94	99,94	95,53	**100,00**	99,04
	DDOS	98,97	99,85	99,41	**100,00**	**100,00**	99,70
	PS	**99,67**	0,00	0,00	68,04	0,00	99,27
DIA 3	NORMAL	99,96	99,94	99,94	99,11	**100,00**	99,01
	DDOS	99,75	**100,00**	99,80	99,90	99,85	99,95
	PS	99,13	0,03	**100,00**	94,03	0,00	**100,00**
	AMBOS	99,63	78,48	**100,00**	95,55	77,97	**100,00**
DIA 4	NORMAL	99,97	99,94	99,94	95,45	**100,00**	99,00
	DDOS	99,76	99,49	99,84	99,29	**99,96**	**99,96**
	PS	99,11	0,05	**100,00**	66,90	0,02	**100,00**
	AMBOS	**100,00**	99,34	**100,00**	98,97	99,53	**100,00**
DIA 5	NORMAL	99,97	99,95	99,94	99,86	**99,99**	99,02
	DDOS	94,03	**100,00**	92,53	**100,00**	**100,00**	97,01
	AMBOS	**100,00**	99,27	**100,00**	98,86	99,81	**100,00**
DIA 6	NORMAL	99,98	99,94	**100,00**	98,99	99,93	99,11
	PS	**99,18**	0,45	0,65	90,83	0,65	99,11
	AMBOS	**100,00**	**100,00**	99,52	**100,00**	99,84	**100,00**

O LOF apresentou o segundo melhor resultado com 92,72%, em seguida o HS com 80,22%, RF com 75,78%, kNN com 64,46% e por fim o NB com 62,92%.

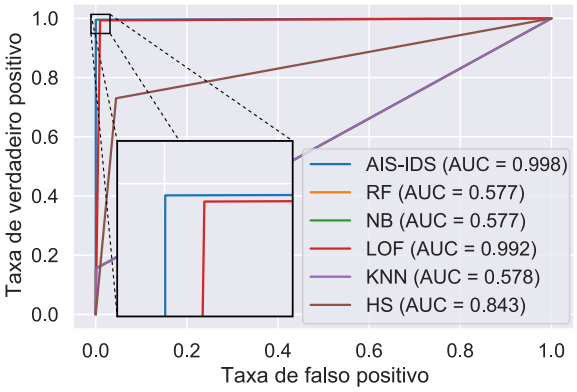
A Tabela 14 foi criada com o objetivo de sintetizar todos os resultados coletados nos seis dias avaliados. Desse modo, ela exibe a taxa de acertos de cada algoritmo durante a classificação das amostras. Os acertos em relação à classificação dos intervalos em normal, ataque DDoS, ataque *port scan* e também momento em que ambos os ataques foram executados estão expressos em porcentagem. Os melhores resultados na detecção dos ataques e do comportamento normal estão destacados para cada dia analisado.

De modo geral, as propostas que não utilizaram as amostras anômalas foram as que conseguiram o melhor resultado médio. Embora outros algoritmos como a RF alcançaram 100% de acerto em alguns casos, em outros momentos obtiveram taxas próximas a zero, como é o caso do terceiro dia.

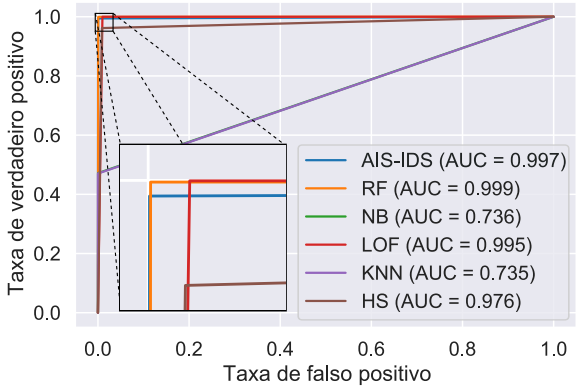
Desse modo, demonstrando que a RF foi super efetivo na classificação de ataques DDoS, contudo, em ataques mais difíceis de detectar, como o *port scan*, não conseguiu detectá-los corretamente. Calculando-se a média de acertos de cada algoritmo, o melhor algoritmo foi o LOF, atingindo 99,45%. Todavia, foi o segundo pior algoritmo em detectar os intervalos normais.



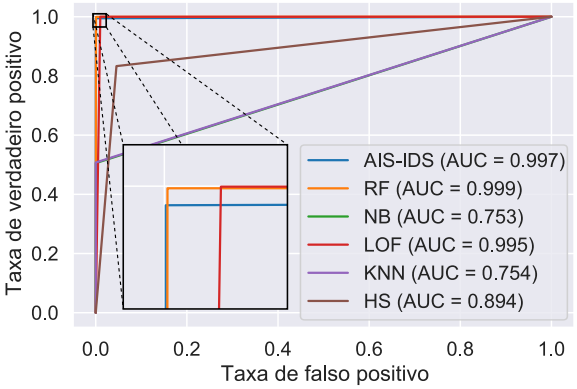
(a) Curva ROC do primeiro dia



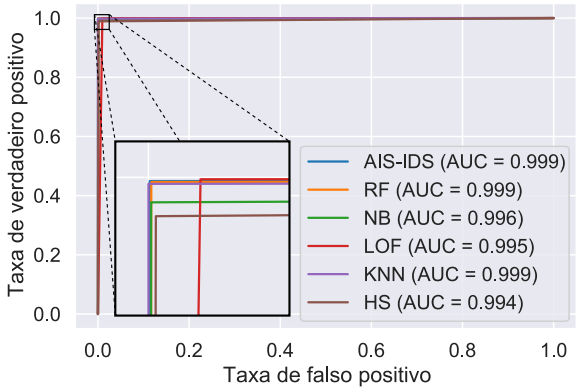
(b) Curva ROC do segundo dia



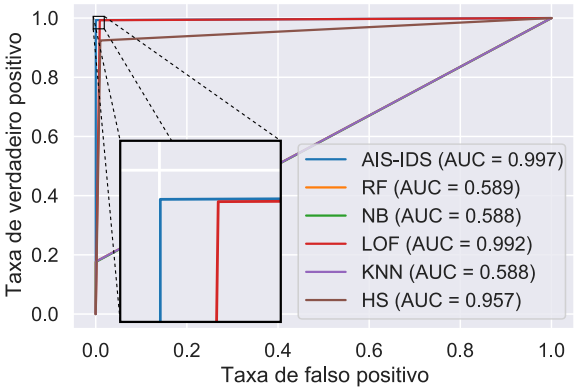
(c) Curva ROC do terceiro dia



(d) Curva ROC do quarto dia



(e) Curva ROC do quinto dia



(f) Curva ROC do sexto dia

Figura 23 – Curvas ROC do segundo cenário

A Figura 23 apresenta as curvas ROC de cada algoritmo para cada dia avaliado. A curva ROC evidencia o desempenho de cada algoritmo em classificar corretamente a taxa de verdadeiros positivos e a taxa de falsos positivos. Assim, quanto mais próximo a 1 no eixo do verdadeiro positivo e mais próximo zero no eixo de falso positivo, melhor é a performance do algoritmo. Além das curvas ROC também são apresentados as áreas sobre as curvas (do inglês, AUC - *Area Under the Curve*), que são uma maneira simplificada da curva ROC, agregando ambos os resultados em um único valor. Quanto mais próximo a 1, melhor é o resultado do algoritmo. Levando em consideração as curvas ROC e os valores de AUC dos algoritmos nos seis dias avaliados, o AIS-IDS obteve os melhores resultados, seguido do LOF, HS, RF, kNN e NB, respectivamente.

5.3 Cenário 3: Comparativo com base pública CiCDDoS2019

Além dos cenários com o trafego emulado, a utilização de bases de dados pública oferece mais credibilidade e veracidade aos resultados obtidos. Desse modo, algumas bases públicas foram avaliadas para este cenário, como a KDD Cup, NSL-KDD e também a CiCDDoS2019.

Dentre as bases citadas, as duas primeira não possuem os atributos necessários para a aplicação do algoritmo, além disso, muitos trabalhos já consideram essas bases com obsoletas e que não retratam um comportamento real da rede [111, 112], portanto a base utilizada para a avaliação do cenário 3 foi a CiCDDoS2019 [6], uma base de dados criada pelo Instituto Canadense de Cibersegurança (do inglês, CIC - *Canadian Institute for Cybersecurity*) com sede na Universidade de New Brunswick. Esse conjunto de dados contém diferentes tipos de ataques DDoS gerados a partir em um ambiente controlado, mas que se assemelham a ataques do mundo real. Os dados gerados simulam o comportamento de 25 usuários na rede, utilizando vários protocolos, como HTTP, FTP e SSH. O CiCDDoS2019 possui dois dias. O primeiro é um dia de treinamento e contempla variações de ataques DDoS, como DNS, TFTP, Syn, NTP, LDAP, SNMP, NetBIOS, UDP-Lag, SSDP, Web-DDoS, MSSQL e UDP, totalizando 12 tipos. O outro, o dia do teste, possui 6 ataques DDoS, que são MSSQL, LDAP, UDP-Lag, UDP, NetBIOS e Syn.

Analisando o conjunto de dados, verificou-se que fluxos anômalos e normais estavam desequilibrados devido aos fluxos de ataques DDoS, que geralmente são numerosos para inundar o alvo. Assim, foi realizado um processo de *downsampling* para balancear e normalizar o conjunto de dados referente a cada tipo de ataque. Além disso, para cada tipo de ataque, cinco fluxos normais disponíveis na base foram coletados aleatoriamente. Essa proporção foi adotada observando-se a frequência dos ataques. O objetivo desse preprocessamento foi manter os dados representativos de todos os tipos de ataques DDoS e evitar o desbalanceamento dentro do conjunto de treinamento.

[6] Disponível em https://www.unb.ca/cic/datasets/ddos-2019.html

Tabela 15 – Comparativo da performance dos algoritmos no conjunto de dados CICDDoS2019

	AIS-IDS	RF	KNN	NB	LOF	HS
ACC	0,890	0,909	0,554	0,450	0,897	0,875
PREC	0,886	0,772	0,284	0,160	0,869	0,806
REC	0,264	0,934	0,255	0,169	0,801	0,806
F-MEASURE	0,922	0,845	0,269	0,164	0,833	0,806
AUC	0,880	0,873	0,476	0,376	0,872	0,858
FP-RATE	0,264	0,099	0,304	0,417	0,056	0,091

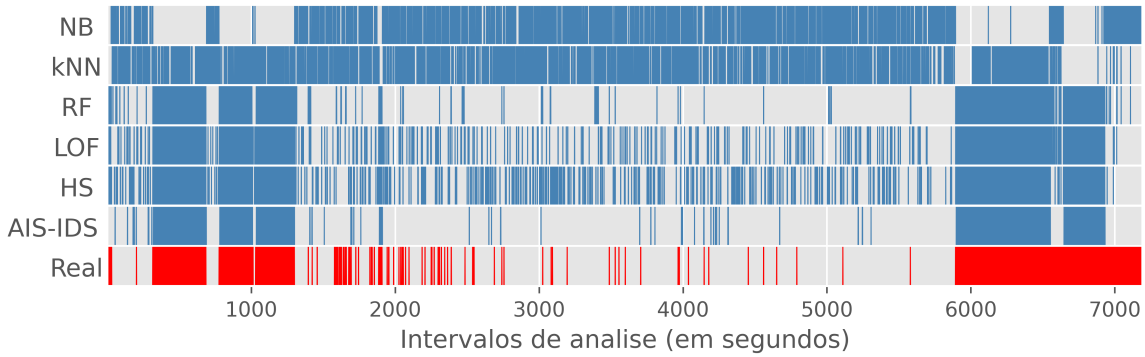


Figura 24 – Comparativo entre as propostas no conjunto de dados público CiCDDoS2019

Após o processo de *downsampling*, os fluxos foram encaminhados ao módulo coletor de fluxos. Neste momento foi verificado que, ao agrupar os fluxos a cada segundo, todos os grupos do conjunto de ataques continham fluxos anômalos, portanto nenhum intervalo do conjunto de treinamento seria considerado livre de anomalias. Nesta perspectiva, a avaliação de algoritmos que necessitam de amostras anômalas e normais para o treinamento, como o NB, kNN e RF, seria comprometida. Desta forma, os fluxos anômalos e normais foram reagrupados para evitar este problema e reenviados ao módulo coletor de fluxos. Ao analisar o conjunto de treinamento, constatou-se também a necessidade de realizar o reagrupamento para que o conjunto de dados contenha momentos considerados normais e anômalos. Para propostas que não usam amostras anômalas no conjunto de treinamento, como AIS-IDS, LOF e o HS, todas as amostras anômalas foram removidas.

Na comparação do conjunto de dados CiCDDoS2019, foram utilizados os mesmos atributos e algoritmos do segundo cenário. Confrontando os resultados dos algoritmos apresentados na Tabela 15, o AIS alcançou o melhor resultado, atingindo um valor de *f-measure* de 92,2%, o algoritmo RF obteve a segunda colocação, com um resultado de 84,5%. As técnicas semelhantes aos AIS, ou seja, LOF e HS, alcançaram 83,3% e 80,6%, respectivamente. Já os algoritmos KNN e NB foram os algoritmos que obtiveram os piores resultados dentre todos avaliados.

A Figura 24 apresenta um comparativo da detecção dos momentos anômalos identificados pelos algoritmos testados. Cada linha representa os momentos considerados anô-

malos por cada algoritmo, destacados na cor azul. Na última linha foram sinalizados em vermelho os momentos que realmente possuem anomalias no conjunto de testes. Para uma melhor análise, a avaliação foi dividida em três partes: do intervalo 1 até intervalo 1200, do 1201 ao 5800 e do 5801 até o final do conjunto de testes. Desse maneira, o primeiro momento contemplou um longo período anômalo, já o segundo momento tem ataques mais espaçados e o terceiro e último momento contém outro longo período com amostras anômalas.

Avaliando o primeiro momento, as propostas como AIS-IDS e RF foram as que melhor detectaram os momentos anômalos. LOF e HS também foram capazes de detectar os momentos anômalos, mas a quantidade de falsos positivos foi bem superior. Por fim, kNN e o NB foram os piores em detectar os ataques no primeiro momento.

Em relação ao segundo momento, como os ataques são mais espaçados, a maior dificuldade é reconhecer corretamente o início e o término de cada evento anômalo, minimizando a taxa de falsos positivos. Desse modo, AIS-IDS e RF obtiveram os melhores resultados nesse quesito.

O terceiro e último momento contém um longo e contínuo período anômalo e, dentre as propostas avaliadas, nenhuma conseguiu identificar todo esse período com exatidão. Os melhores resultados neste momento foram os do LOF e da RF. O NB apresentou o pior resultado, detectando somente intervalos anômalos esparsos.

De modo geral, AIS-IDS e RF foram as melhores, pois detectaram os longos momentos anômalos e obtiveram menos falsos positivos nos momentos de ataques mais espaçados, se comparadas aos outros algoritmos. O LOF também foi capaz de detectar ataques longos, no entanto, com muitos falsos positivos em comparação ao AIS-IDS e RF. De acordo com a análise também foi possível constatar que NB e kNN apresentaram grande quantidade de falsos positivos, prejudicando significativamente a qualidade da detecção.

A Figura 25 apresenta a curva ROC e também o valor AUC de cada algoritmo no conjunto de dados CiCDDoS2019. O AIS-IDS foi o algoritmo que obteve o melhor resultado, a RF foi o segundo melhor algoritmo. O LOF e o HS, foram o terceiro e quarto colocados, respectivamente. Os algoritmos com pior performance foram KNN e o NB.

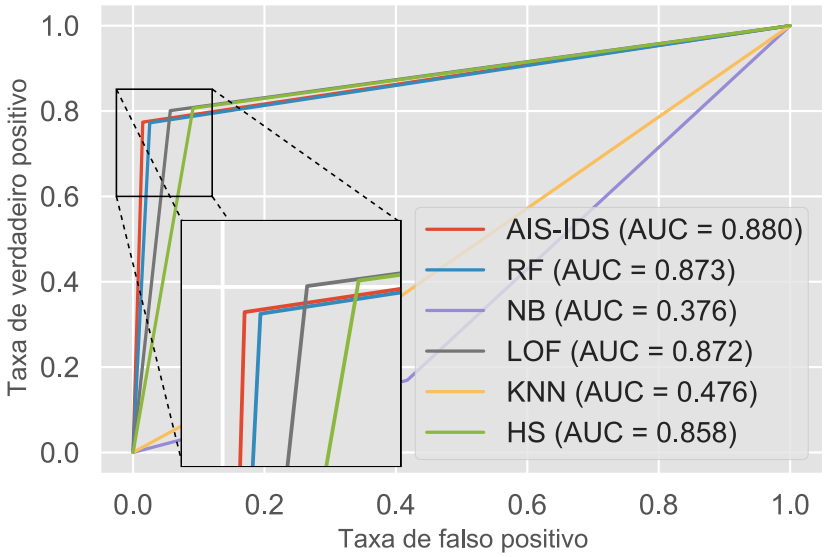


Figura 25 – Curva ROC dos algoritmos no conjunto de dados público CiCDDoS2019

6 CONCLUSÃO

A utilização de novas arquiteturas de redes e tecnologias de comunicação vem se tornando cada vez mais presente na indústria e cotidiano do usuário. Desse modo, toda a inovação acaba possibilitando a exploração de vulnerabilidades e tornando a rede insegura. Com isso, a necessidade de prover segurança é indispensável. Os IDS são ferramentas que visam proteger a rede de tais ameaças, detectando-as e as bloqueando. Este trabalho apresentou um IDS bioinspirado que empregou a seleção negativa, um algoritmo da classe dos AIS para detecção de ataques do tipo DDoS e *port scan*.

O sistema é composto por três módulos. O primeiro módulo é o responsável por coletar, extrair e preprocessar os fluxos provenientes do plano dados. Neste módulo é utilizado o conceito de entropia para sumarizar os dados e auxiliar na detecção dos momentos em que o comportamento da rede sofre distorções. O segundo módulo é responsável pela criação dos detectores e também pela detecção das anomalias. Na primeira etapa, os detectores são criados por meio da seleção negativa, a qual necessita apenas do tráfego usual e legítimo para aprender o comportamento padrão da rede. Desse modo, a seleção negativa propicia gerar detectores que têm por objetivo reconhecer comportamentos anômalos, e por consequência, reconhecer ataques na rede.

Durante a segunda etapa do segundo módulo, denominada classificação, a similaridade entre os detectores anteriormente gerados e as amostras vindas do coletor é calculada. Como a análise do tráfego da rede pode acarretar em imprecisões provenientes da coleta, do preprocessamento e da própria noção subjacente de normalidade, a lógica *Fuzzy* foi aplicada para aumentar o grau de certeza na classificação das amostras entre legítimas ou anômalas. Além disso, a MLR foi utilizada para potencializar a detecção de momentos anômalos usando coeficientes de importância, desse modo, ponderando quais atributos são mais significativos no reconhecimento de momentos em que as anomalias estão ocorrendo. Após a pontuação geral da amostra ser estabelecida pela lógica *Fuzzy*, ela é classificada como normal ou anômala. Caso a amostra seja normal, ela é encaminhada normalmente ao plano de dados para ser enviada ao seu destino. No entanto, se a amostra for considerada anômala, medidas protetivas são acionadas pelo módulo mitigador.

O último módulo, o módulo mitigador, é o responsável por bloquear o ataque e manter a rede segura. Assim que a amostra é considerada anômala, todos os fluxos deste momento anômalo são encaminhados para este módulo com o objetivo que seja reconhecido o tipo da anomalia e também os endereços IP e portas envolvidos no ataque. Nesse momento, o controlador SDN cria um fluxo contendo a instrução de descarte de pacotes que correspondam aos endereços IP e portas anteriormente descobertos e envia-o para plano de dados, onde os equipamentos implementarão essa contramedida o mais

rápido possível para contenção da ameaça.

O IDS proposto foi avaliado em três cenários. O primeiro avaliou extensivamente a capacidade de detecção e mitigação do IDS apresentado. O segundo e terceiro cenários compararam o AIS-IDS a outros métodos da literatura, também utilizados na detecção de comportamentos anômalos no tráfego da rede. O segundo cenário utilizou seis dias de tráfego coletados de uma rede emulada e o terceiro cenário usou a base de dados pública CICDDoS2019.

Durante a avaliação dos resultados, no primeiro cenário, o AIS-IDS foi capaz de detectar os momentos anômalos e bloquear os ataques, de modo que após o bloqueio, o comportamento da rede voltou à normalidade. Os atributos utilizados para a detecção e mitigação mostraram-se eficazes, pois através da entropia foi possível detectar os momentos em que os eventos anômalos estavam ocorrendo. Através dos endereços IP e portas coletados dos fluxos, foi possível localizar o atacante e cessar a sua comunicação com o alvo.

Considerando o segundo cenário e seus seis dias avaliados, cada dia utilizou configurações de intensidade e duração distintas e também a possibilidade da sobreposição de ataques, de modo que dificultasse a detecção das anomalias nesses momentos. Analisando os seis dias, foi constatado que os algoritmos que utilizavam amostras anômalas para o treinamento tiveram maior dificuldade para detectar ataques menos intensos como o *port scan*. Alguns algoritmos atingiram altas taxas de falsos positivos como o HS e o LOF, desse modo, demonstrando que inúmeros pacotes legítimos seriam bloqueados se fossem utilizados para a proteção da rede. De modo geral, o AIS-IDS obteve os melhores resultados frente aos outros algoritmos comparados, também demonstrou que foi capaz de detectar os ataques nas mais variadas configurações de intensidade, duração e também sobreposição.

Os resultados apurados foram semelhantes no terceiro cenário, em que a base de dados pública foi usada. Novamente o AIS-IDS obteve o melhor resultado, onde foi possível constatar a sua performance de detecção. Nos momentos de maior intensidade dos ataques, o AIS-IDS mostrou sua capacidade de reconhecimento dos intervalos onde ocorriam anomalias no tráfego. Essa eficiência também pôde ser comprovada em momentos de ataques mais espaçados, quando o IDS proposto mostrou uma baixa taxa de falsos positivos. Vale ressaltar que a base de dados pública tem um grau de complexidade superior se comparada aos conjuntos de dados extraídos do ambiente emulado. Portanto, esse cenário continha variados tipos de ataques DDoS de características distintas e, mesmo dessa maneira, a proposta se mostrou eficaz em detectar grande parte deles.

Avaliando os resultados obtidos nos dois últimos cenários, foi possível observar que os algoritmos que não utilizam dados anômalos para treinamento, como AIS, LOF e o HS, também obtiveram resultados superiores aos demais algoritmos comparados. Por

não exigirem amostras anômalos para treinamento, esses algoritmos trazem benefícios ao processo de monitoramento e gerência da rede. A rotulação do tráfego como anômalo ou não, mesmo que deva ser realizada por um especialista, é onerosa e propensa a erros devido às pequenas flutuações no tráfego decorrente do uso cotidiano da rede. Além do mais, esses algoritmos, por lidarem somente com o comportamento normal do tráfego, possuem a capacidade de detecção de ataques novos ou desconhecidos.

O AIS-IDS se mostrou o melhor IDS dentre os testados, até mesmo se comparado aos algoritmos que não utilizam amostras anômalas para o treinamento. Tal resultado foi obtido por conta do uso da janela deslizante, a qual possibilitou a análise mais detalhada de cada momento do dia, tornando o AIS-IDS mais eficiente e propiciando respostas aos eventos anômalos em tempo quase real. Aliado à janela deslizante, o cálculo da similaridade tornou possível o reconhecimento do padrão de normalidade do tráfego, resultando na detecção acurada dos ataques avaliados. Além disso, a utilização da lógica *fuzzy* juntamente com a MLR fez com que o AIS-IDS pudesse tratar as incertezas durante a detecção dos ataques melhor do que os métodos comparados, sobretudo nos momentos em que tais anomalias eram menos intensas e não causavam alterações significativas no tráfego.

Para os trabalhos futuros, outros atributos serão estudados para aprimorar a detecção de novos tipos de ataques. Também será estudada a utilização de outros algoritmos para uso em conjunto com a seleção negativa com o objetivo de aumentar a performance de detecção.

REFERÊNCIAS

[1] KREUTZ, D. et al. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, IEEE, v. 103, n. 1, p. 14–76, 2014. <https://doi.org/10.1109/JPROC.2014.2371999>.

[2] GAMAGE, S.; SAMARABANDU, J. Deep learning methods in network intrusion detection: A survey and an objective comparison. *Journal of Network and Computer Applications*, Elsevier, p. 102767, 2020. <https://doi.org/10.1016/j.jnca.2020.102767>.

[3] ALDRIBI, A. et al. Hypervisor-based cloud intrusion detection through online multivariate statistical change tracking. *Computers & Security*, Elsevier, v. 88, p. 101646, 2020. <https://doi.org/10.1016/j.cose.2019.101646>.

[4] CAUTERUCCIO, F. et al. A framework for anomaly detection and classification in multiple iot scenarios. *Future Generation Computer Systems*, Elsevier, v. 114, p. 322–335, 2020. <https://doi.org/10.1016/j.future.2020.08.010>.

[5] ZHOU, Y. et al. Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Computer Networks*, Elsevier, p. 107247, 2020. <https://doi.org/10.1016/j.comnet.2020.107247>.

[6] NOVAES, M. P. et al. Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment. *IEEE Access*, IEEE, v. 8, p. 83765–83781, 2020. <https://doi.org/10.1109/ACCESS.2020.2992044>.

[7] SALMAN, O. et al. Iot survey: An sdn and fog computing perspective. *Computer Networks*, Elsevier, v. 143, p. 221–246, 2018. <https://doi.org/10.1016/j.comnet.2018.07.020>.

[8] XU, Y. et al. Efficient ddos detection based on k-fknn in software defined networks. *IEEE Access*, IEEE, v. 7, p. 160536–160545, 2019. <https://doi.org/10.1109/ACCESS.2019.2950945>.

[9] BARKI, L. et al. Detection of distributed denial of service attacks in software defined networks. In: IEEE. *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. [S.l.], 2016. p. 2576–2581. <https://doi.org/10.1109/ICACCI.2016.7732445>.

[10] SAHAY, R.; MENG, W.; JENSEN, C. D. The application of software defined networking on securing computer networks: A survey. *Journal of Network and Computer Applications*, Elsevier, v. 131, p. 89–108, 2019. <https://doi.org/10.1016/j.jnca.2019.01.019>.

[11] CARVALHO, L. F. et al. An ecosystem for anomaly detection and mitigation in software-defined networking. *Expert Systems with Applications*, Elsevier, v. 104, p. 121–133, 2018. <https://doi.org/10.1016/j.eswa.2018.03.027>.

[12] MOUSAVI, S. M.; ST-HILAIRE, M. Early detection of ddos attacks against software defined network controllers. *Journal of Network and Systems Management*, Springer, v. 26, n. 3, p. 573–591, 2018. <https://doi.org/10.1007/s10922-017-9432-1>.

[13] CORDEIRO, W. L. da C.; MARQUES, J. A.; GASPARY, L. P. Data plane programmability beyond openflow: Opportunities and challenges for network and service operations and management. *Journal of Network and Systems Management*, Springer, v. 25, n. 4, p. 784–818, 2017. <https://doi.org/10.1007/s10922-017-9423-2>.

[14] ALSAEEDI, M.; MOHAMAD, M. M.; AL-ROUBAIEY, A. A. Toward adaptive and scalable openflow-sdn flow control: A survey. *IEEE Access*, IEEE, v. 7, p. 107346–107379, 2019. <https://doi.org/10.1109/ACCESS.2019.2932422>.

[15] YU, Y. et al. Fault management in software-defined networking: A survey. *IEEE Communications Surveys & Tutorials*, IEEE, v. 21, n. 1, p. 349–392, 2018. <https://doi.org/10.1109/COMST.2018.2868922>.

[16] METI, N.; NARAYAN, D.; BALIGAR, V. P. Detection of distributed denial of service attacks using machine learning algorithms in software defined networks. In: IEEE. *2017 international conference on advances in computing, communications and informatics (ICACCI)*. [S.l.], 2017. p. 1366–1371. <https://doi.org/10.1109/ICACCI.2017.8126031>.

[17] PENG, H. et al. A detection method for anomaly flow in software defined network. *IEEE Access*, IEEE, v. 6, p. 27809–27817, 2018. <https://doi.org/10.1109/ACCESS.2018.2839684>.

[18] ZHONG, Y. et al. Helad: A novel network anomaly detection model based on heterogeneous ensemble learning. *Computer Networks*, Elsevier, v. 169, p. 107049, 2020. <https://doi.org/10.1016/j.comnet.2019.107049>.

[19] PENA, E. H. et al. Anomaly detection using digital signature of network segment with adaptive arima model and paraconsistent logic. In: IEEE. *2014 IEEE Symposium on Computers and Communications (ISCC)*. [S.l.], 2014. p. 1–6. <https://doi.org/10.1109/ISCC.2014.6912503>.

[20] ASSIS, M. V. et al. A gru deep learning system against attacks in software defined networks. *Journal of Network and Computer Applications*, Elsevier, v. 177, p. 102942, 2021. <https://doi.org/10.1016/j.jnca.2020.102942>.

[21] CALIKUS, E. et al. No free lunch but a cheaper supper: A general framework for streaming anomaly detection. *Expert Systems with Applications*, Elsevier, p. 113453, 2020. <https://doi.org/10.1016/j.eswa.2020.113453>.

[22] KOPP, M.; PEVNỲ, T.; HOLEŇA, M. Anomaly explanation with random forests. *Expert Systems with Applications*, Elsevier, v. 149, p. 113187, 2020. <https://doi.org/10.1016/j.eswa.2020.113187>.

[23] RASOOL, R. u. et al. A survey of link flooding attacks in software defined network ecosystems. *Journal of Network and Computer Applications*, Elsevier, p. 102803, 2020. <https://doi.org/10.1016/j.jnca.2020.102803>.

[24] NISAR, K. et al. A survey on the architecture, application, and security of software defined networking. *Internet of Things*, Elsevier, p. 100289, 2020. <https://doi.org/10.1016/j.iot.2020.100289>.

[25] THAKKAR, A.; LOHIYA, R. Role of swarm and evolutionary algorithms for intrusion detection system: A survey. *Swarm and Evolutionary Computation*, Elsevier, v. 53, p. 100631, 2020. <https://doi.org/10.1016/j.swevo.2019.100631>.

[26] SHARMA, A. et al. Towards trustworthy internet of things: A survey on trust management applications and schemes. *Computer Communications*, Elsevier, 2020. <https://doi.org/10.1016/j.comcom.2020.06.030>.

[27] YAMANSAVASCILAR, B. et al. Fault tolerance in sdn data plane considering network and application based metrics. *Journal of Network and Computer Applications*, Elsevier, v. 170, p. 102780, 2020. <https://doi.org/10.1016/j.jnca.2020.102780>.

[28] LIU, G. et al. Efficient ddos attacks mitigation for stateful forwarding in internet of things. *Journal of Network and Computer Applications*, Elsevier, v. 130, p. 1–13, 2019. <https://doi.org/10.1016/j.jnca.2019.01.006>.

[29] ALDWEESH, A.; DERHAB, A.; EMAM, A. Z. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowledge-Based Systems*, Elsevier, v. 189, p. 105124, 2020. <https://doi.org/10.1016/j.knosys.2019.105124>.

[30] UJJAN, R. M. A. et al. Towards sflow and adaptive polling sampling for deep learning based ddos detection in sdn. *Future Generation Computer Systems*, Elsevier, v. 111, p. 763–779, 2020. <https://doi.org/10.1016/j.future.2019.10.015>.

[31] SINGH, M. P.; BHANDARI, A. New-flow based ddos attacks in sdn: Taxonomy, rationales, and research challenges. *Computer Communications*, Elsevier, 2020. <https://doi.org/10.1016/j.comcom.2020.02.085>.

[32] HACHMI, F.; BOUJENFA, K.; LIMAM, M. Enhancing the accuracy of intrusion detection systems by reducing the rates of false positives and false negatives through multi-objective optimization. *Journal of Network and Systems Management*, Springer, v. 27, n. 1, p. 93–120, 2019. <https://doi.org/10.1007/s10922-018-9459-y>.

[33] SULIMAN, S. I. et al. Network intrusion detection system using artificial immune system (ais). In: IEEE. *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*. [S.l.], 2018. p. 178–182. <https://doi.org/10.1109/CCOMS.2018.8463274>.

[34] BARNETT, V.; LEWIS, T. Outliers in statistical data john wiley and sons. *New York*, 1994. <https://doi.org/10.1016/0169-2070(95)00625-7>.

[35] CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 41, n. 3, p. 1–58, 2009. <https://doi.org/10.1145/1541880.1541882>.

[36] LAKHINA, A.; CROVELLA, M.; DIOT, C. Diagnosing network-wide traffic anomalies. *ACM SIGCOMM computer communication review*, ACM New York, NY, USA, v. 34, n. 4, p. 219–230, 2004. <https://doi.org/10.1145/1030194.1015492>.

[37] HOQUE, N. et al. Network attacks: Taxonomy, tools and systems. *Journal of Network and Computer Applications*, Elsevier, v. 40, p. 307–324, 2014. <https://doi.org/10.1016/j.jnca.2013.08.001>.

[38] MASDARI, M.; KHEZRI, H. A survey and taxonomy of the fuzzy signature-based intrusion detection systems. *Applied Soft Computing*, Elsevier, p. 106301, 2020. <https://doi.org/10.1016/j.asoc.2020.106301>.

[39] FERNANDES, G. et al. A comprehensive survey on network anomaly detection. *Telecommunication Systems*, Springer, v. 70, n. 3, p. 447–489, 2019. <https://doi.org/10.1007/s11235-018-0475-8>.

[40] BADDAR, S. W. A.; MERLO, A.; MIGLIARDI, M. Anomaly detection in computer networks: A state-of-the-art review. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, v. 5, n. 4, p. 29–64, 2014. <https://doi.org/10.22667/JOWUA.2014.12.31.029>.

[41] SONG, X. et al. Conditional anomaly detection. *IEEE Transactions on knowledge and Data Engineering*, IEEE, v. 19, n. 5, p. 631–645, 2007. <https://doi.org/10.1109/TKDE.2007.1009>.

[42] AHMED, M.; MAHMOOD, A. N.; HU, J. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, Elsevier, v. 60, p. 19–31, 2016. <https://doi.org/10.1016/j.jnca.2015.11.016>.

[43] BARFORD, P. et al. A signal analysis of network traffic anomalies. In: *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. [S.l.: s.n.], 2002. p. 71–82. <https://doi.org/10.1145/637201.637210>.

[44] MARNERIDES, A. K.; SCHAEFFER-FILHO, A.; MAUTHE, A. Traffic anomaly diagnosis in internet backbone networks: A survey. *Computer Networks*, Elsevier, v. 73, p. 224–243, 2014. <https://doi.org/10.1016/j.comnet.2014.08.007>.

[45] LÖF, A.; NELSON, R. Annotating network trace data for anomaly detection research. In: IEEE. *39th Annual IEEE Conference on Local Computer Networks Workshops*. [S.l.], 2014. p. 679–684. <https://doi.org/10.1109/LCNW.2014.6927720>.

[46] BARFORD, P.; PLONKA, D. Characteristics of network traffic flow anomalies. In: *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*. [S.l.: s.n.], 2001. p. 69–73. <https://doi.org/10.1145/505202.505211>.

[47] BEHAL, S.; KUMAR, K.; SACHDEVA, M. Characterizing ddos attacks and flash events: Review, research gaps and future directions. *Computer Science Review*, Elsevier, v. 25, p. 101–114, 2017. <https://doi.org/10.1016/j.cosrev.2017.07.003>.

[48] MURALEEDHARAN, N.; PARMAR, A.; KUMAR, M. A flow based anomaly detection system using chi-square technique. In: IEEE. *2010 IEEE 2nd international Advance computing conference (IACC)*. [S.l.], 2010. p. 285–289. <https://doi.org/10.1109/IADCC.2010.5422996>.

[49] SCARANTI, G. F. et al. Artificial immune systems and fuzzy logic to detect flooding attacks in software-defined networks. *IEEE Access*, IEEE, v. 8, p. 100172–100184, 2020. <https://doi.org/10.1109/ACCESS.2020.2997939>.

[50] KEMMERER, R. A.; VIGNA, G. Intrusion detection: a brief history and overview. *Computer*, IEEE, v. 35, n. 4, p. supl27–supl30, 2002. <https://doi.org/10.1109/MC.2002.1012428>.

[51] CAMPBELL, P. L. The denial-of-service dance. *IEEE security & privacy*, IEEE, v. 3, n. 6, p. 34–40, 2005. <https://doi.org/10.1109/MSP.2005.162>.

[52] BOSTANI, H.; SHEIKHAN, M. Hybrid of anomaly-based and specification-based ids for internet of things using unsupervised opf based on mapreduce approach. *Computer Communications*, Elsevier, v. 98, p. 52–71, 2017. <https://doi.org/10.1016/j.comcom.2016.12.001>.

[53] CARVALHO, L. F. et al. Unsupervised learning clustering and self-organized agents applied to help network management. *Expert Systems with Applications*, Elsevier, v. 54, p. 29–47, 2016. <https://doi.org/10.1016/j.eswa.2016.01.032>.

[54] BHUYAN, M. H.; BHATTACHARYYA, D. K.; KALITA, J. K. Network anomaly detection: methods, systems and tools. *IEEE communications surveys & tutorials*, IEEE, v. 16, n. 1, p. 303–336, 2013. <https://doi.org/10.1109/SURV.2013.052213.00046>.

[55] PENA, E. H. et al. Anomaly detection using the correlational paraconsistent machine with digital signatures of network segment. *Information Sciences*, Elsevier, v. 420, p. 313–328, 2017. <https://doi.org/10.1016/j.ins.2017.08.074>.

[56] FORREST, S. et al. Self-nonself discrimination in a computer. In: IEEE. *Proceedings of 1994 IEEE computer society symposium on research in security and privacy*. [S.l.], 1994. p. 202–212. <https://doi.org/10.1109/RISP.1994.296580>.

[57] HOFMEYR, S. A.; FORREST, S. Immunity by design: An artificial immune system. In: CITESEER. *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 2*. [S.l.], 1999. p. 1289–1296. <https://dl.acm.org/doi/epdf/10.5555/2934046.2934107>.

[58] JI, Z.; DASGUPTA, D. Revisiting negative selection algorithms. *Evolutionary Computation*, MIT Press, v. 15, n. 2, p. 223–251, 2007. <https://doi.org/10.1162/evco.2007.15.2.223>.

[59] KING, R. L. et al. An artificial immune system model for intelligent agents. *Future Generation Computer Systems*, Elsevier, v. 17, n. 4, p. 335–343, 2001. <https://doi.org/10.1016/S0167-739X(99)00115-6>.

[60] HARMER, P. K. et al. An artificial immune system architecture for computer security applications. *IEEE transactions on evolutionary computation*, IEEE, v. 6, n. 3, p. 252–280, 2002. <https://doi.org/10.1109/TEVC.2002.1011540>.

[61] ZHOU, Q.; PEZAROS, D. P. Bids: Bio-inspired, collaborative intrusion detection for software defined networks. In: IEEE. *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. [S.l.], 2019. p. 1–6. <https://doi.org/10.1109/ICC.2019.8761410>.

[62] SONG, K. et al. Artificial immune system (ais) based intrusion detection system (ids) for smart grid advanced metering infrastructure (ami) networks. Virginia Tech, 2018. <https://doi.org/10919/83203>.

[63] CASTRO, L. N. D.; ZUBEN, F. J. V. The clonal selection algorithm with engineering applications. In: *Proceedings of GECCO*. [S.l.: s.n.], 2000. v. 2000, p. 36–39. <https://www.researchgate.net/publication/2468677_The_Clonal_Selection_Algorithm_with_Engineering_Applications>.

[64] GARRETT, S. M. How do we evaluate artificial immune systems? *Evolutionary computation*, MIT Press, v. 13, n. 2, p. 145–177, 2005. <https://doi.org/10.1162/1063656054088512>.

[65] CASTRO, L. N. de; ZUBEN, F. J. V. ainet: an artificial immune network for data analysis. In: *Data mining: a heuristic approach*. [S.l.]: IGI Global, 2002. p. 231–260. <https://doi.org/10.4018/978-1-930708-25-9.ch012>.

[66] VIDAL, J. M.; OROZCO, A. L. S.; VILLALBA, L. J. G. Adaptive artificial immune networks for mitigating dos flooding attacks. *Swarm and Evolutionary Computation*, Elsevier, v. 38, p. 94–108, 2018. <https://doi.org/10.1016/j.swevo.2017.07.002>.

[67] MCKEOWN, N. et al. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, ACM New York, NY, USA, v. 38, n. 2, p. 69–74, 2008. <https://doi.org/10.1145/1355734.1355746>.

[68] MANSOUR, A. et al. Biologically-inspired sdn-based intrusion detection and prevention mechanism for heterogeneous iot networks. In: IEEE. *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. [S.l.], 2018. p. 1120–1125. <https://doi.org/10.1109/IEMCON.2018.8614759>.

[69] Open Networking Foundation. Software-defined networking: the new norm for networks. *ONF white paper*, 2012. <https://pdfs.semanticscholar.org/a3f6/9f6181a0b4d481073a21eafbcc434a800db6.pdf>.

[70] LATIF, Z. et al. A comprehensive survey of interface protocols for software defined networks. *Journal of Network and Computer Applications*, Elsevier, v. 156, p. 102563, 2020. <https://doi.org/10.1016/j.jnca.2020.102563>.

[71] CARVALHO, L. F. et al. A novel anomaly detection system to assist network management in sdn environment. In: IEEE. *2017 IEEE International Conference on Communications (ICC)*. [S.l.], 2017. p. 1–6. <https://doi.org/10.1109/ICC.2017.7997214>.

[72] JIN, L.; CHEN, J.; ZHANG, X. An outlier fuzzy detection method using fuzzy set theory. *IEEE Access*, IEEE, v. 7, p. 59321–59332, 2019. <https://doi.org/10.1109/ACCESS.2019.2914605>.

[73] FAHIM, M.; SILLITTI, A. Anomaly detection, analysis and prediction techniques in iot environment: A systematic literature review. *IEEE Access*, IEEE, v. 7, p. 81664–81681, 2019. <https://doi.org/10.1109/ACCESS.2019.2921912>.

[74] GARG, S. et al. Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in sdn: A social multimedia perspective. *IEEE Transactions on Multimedia*, IEEE, v. 21, n. 3, p. 566–578, 2019. <https://doi.org/10.1109/TMM.2019.2893549>.

[75] PROENÇA, M. L. et al. The hurst parameter for digital signature of network segment. In: SPRINGER. *International Conference on Telecommunications*. [S.l.], 2004. p. 772–781. <https://doi.org/10.1007/978-3-540-27824-5_103>.

[76] LAI, Y.-C. et al. F1ow-based anomaly detection using multilayer perceptron in software defined networks. In: IEEE. *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. [S.l.], 2019. p. 1154–1158. <https://doi.org/10.23919/MIPRO.2019.8757199>.

[77] HAIDER, S. et al. Deep learning based ensemble convolutional neural network solution for distributed denial of service detection in sdns. In: IEEE. *2019 UK/China Emerging Technologies (UCET)*. [S.l.], 2019. p. 1–4. <https://doi.org/10.1109/UCET.2019.8881856>.

[78] A. Aziz, A. S.; HANAFI, S. E.; HASSANIEN, A. E. Comparison of classification techniques applied for network intrusion detection and classification. *Journal of Applied Logic*, Elsevier, v. 24, p. 109–118, 2017. <https://doi.org/10.1016/j.jal.2016.11.018>.

[79] HOSSEINZADEH, S.; AMIRMAZLAGHANI, M.; SHAJARI, M. An aggregated statistical approach for network flood detection using gamma-normal mixture modeling. *Computer Communications*, Elsevier, v. 152, p. 137–148, 2020. <https://doi.org/10.1016/j.comcom.2020.01.028>.

[80] QIN, Y.; WEI, J.; YANG, W. Deep learning based anomaly detection scheme in software-defined networking. In: IEEE. *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. [S.l.], 2019. p. 1–4. <https://doi.org/10.23919/APNOMS.2019.8892873>.

[81] ARIVUDAINAMBI, D.; KA, V. K.; CHAKKARAVARTHY, S. S. Lion ids: A meta-heuristics approach to detect ddos attacks against software-defined networks. *Neural Computing and Applications*, Springer, v. 31, n. 5, p. 1491–1501, 2019. <https://doi.org/10.1007/s00521-018-3383-7>.

[82] DUY, P. T.; PHAM, V.-H. A role-based statistical mechanism for ddos attack detection in sdn. In: IEEE. *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)*. [S.l.], 2018. p. 177–182. <https://doi.org/10.1109/NICS.2018.8606851>.

[83] RATHORE, H.; SAMANT, A.; GUIZANI, M. A bio-inspired framework to mitigate dos attacks in software defined networking. In: IEEE. *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. [S.l.], 2019. p. 1–5. <https://doi.org/10.1109/NTMS.2019.8763818>.

[84] HOOKS, D. et al. Applying artificial immune system for intrusion detection. In: IEEE. *2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService)*. [S.l.], 2018. p. 287–292. <https://doi.org/10.1109/BigDataService.2018.00051>.

[85] SHEN, J.; WANG, J. Network intrusion detection by artificial immune system. In: IEEE. *IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society*. [S.l.], 2011. p. 4716–4720. <https://doi.org/10.1109/IECON.2011.6119993>.

[86] TABATABAEFAR, M.; MIRIESTAHBANATI, M.; GRÉGOIRE, J.-C. Network intrusion detection through artificial immune system. In: IEEE. *Systems Conference (SysCon), 2017 Annual IEEE International*. [S.l.], 2017. p. 1–6. <https://doi.org/10.1109/SYSCON.2017.7934751>.

[87] XU, T. et al. Defending against new-flow attack in sdn-based internet of things. *IEEE Access*, IEEE, v. 5, p. 3431–3443, 2017. <https://doi.org/10.1109/ACCESS.2017.2666270>.

[88] AMARAL, A. A. et al. Deep ip flow inspection to detect beyond network anomalies. *Computer Communications*, Elsevier, v. 98, p. 80–96, 2017. <https://doi.org/10.1016/j.comcom.2016.12.007>.

[89] LIU, D. et al. Network traffic anomaly detection using adaptive density-based fuzzy clustering. In: IEEE. *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*. [S.l.], 2014. p. 823–830. <https://doi.org/10.1109/TrustCom.2014.109>.

[90] ASSIS, M. V. D. et al. Fast defense system against attacks in software defined networks. *IEEE Access*, IEEE, v. 6, p. 69620–69639, 2018. <https://doi.org/10.1109/ACCESS.2018.2878576>.

[91] JONGSUEBSUK, P.; WATTANAPONGSAKORN, N.; CHARNSRIPINYO, C. Network intrusion detection with fuzzy genetic algorithm for unknown attacks. In: IEEE. *The International Conference on Information Networking 2013 (ICOIN)*. [S.l.], 2013. p. 1–5. <https://doi.org/10.1109/ICOIN.2013.6496342>.

[92] SHANNON, C. E. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, Association for Computing Machinery, New York, NY, USA, v. 5, n. 1, p. 3–55, jan. 2001. ISSN 1559-1662. <https://doi.org/10.1145/584091.584093>.

[93] KRISHNAVENI, S. et al. Anomaly-based intrusion detection system using support vector machine. In: *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. [S.l.]: Springer, 2020. p. 723–731. <https://doi.org/10.1007/978-981-15-0199-9_62>.

[94] ASSIS, M. V. D.; RODRIGUES, J. J.; PROENÇA, M. L. A novel anomaly detection system based on seven-dimensional flow analysis. In: IEEE. *2013 IEEE Global Communications Conference (GLOBECOM)*. [S.l.], 2013. p. 735–740. <https://doi.org/10.1109/GLOCOM.2013.6831160>.

[95] FAN, Z. et al. An antigen space triangulation coverage based real-value negative selection algorithm. *IEEE Access*, IEEE, v. 7, p. 51886–51898, 2019. <https://doi.org/10.1109/ACCESS.2019.2911660>.

[96] AGGARWAL, C. C. Outlier analysis. In: SPRINGER. *Data mining*. [S.l.], 2015. p. 237–263. <https://doi.org/10.1007/978-3-319-14142-8_8>.

[97] BRAHMA, A.; PANIGRAHI, S. Role of soft outlier analysis in database intrusion detection. In: *Advanced Computing and Intelligent Engineering*. [S.l.]: Springer, 2020. p. 479–489. <https://doi.org/10.1007/978-981-15-1081-6_41>.

[98] HAMAMOTO, A. H. et al. Network anomaly detection system using genetic algorithm and fuzzy logic. *Expert Systems with Applications*, Elsevier, v. 92, p. 390–402, 2018. <https://doi.org/10.1016/j.eswa.2017.09.013>.

[99] TSOUKALAS, L. H.; UHRIG, R. E. *Fuzzy and neural approaches in engineering*. [S.l.]: John Wiley & Sons, Inc., 1996. <https://dl.acm.org/doi/10.5555/548356>. ISBN 978-0-471-16003-8.

[100] WANG, Y. A multinomial logistic regression modeling approach for anomaly intrusion detection. *Computers & Security*, Elsevier, v. 24, n. 8, p. 662–674, 2005. <https://doi.org/10.1016/j.cose.2005.05.003>.

[101] KHAMMASSI, C.; KRICHEN, S. A nsga2-lr wrapper approach for feature selection in network intrusion detection. *Computer Networks*, Elsevier, p. 107183, 2020. <https://doi.org/10.1016/j.comnet.2020.107183>.

[102] JR, M. L. P. et al. Digital signature to help network management using flow analysis. *International Journal of Network Management*, Wiley Online Library, v. 26, n. 2, p. 76–94, 2016. <https://doi.org/10.1002/nem.1892>.

[103] LI, Z. et al. Supereye: A distributed port scanning system. In: SPRINGER. *International Conference on Artificial Intelligence and Security*. [S.l.], 2019. p. 46–56. <https://doi.org/10.1007/978-3-030-24268-8_5>.

[104] KHAN, M. A. A survey of security issues for cloud computing. *Journal of network and computer applications*, Elsevier, v. 71, p. 11–29, 2016. <https://doi.org/10.1016/j.jnca.2016.05.010>.

[105] LANTZ, B.; HELLER, B.; MCKEOWN, N. A network in a laptop: rapid prototyping for software-defined networks. In: ACM. *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. [S.l.], 2010. p. 19. <https://doi.org/10.1145/1868447.1868466>.

[106] LEWIS, D. D. Naive (bayes) at forty: The independence assumption in information retrieval. In: SPRINGER. *European conference on machine learning*. [S.l.], 1998. p. 4–15. <https://doi.org/10.1007/BFb0026666>.

[107] HAUTAMAKI, V.; KARKKAINEN, I.; FRANTI, P. Outlier detection using k-nearest neighbour graph. In: IEEE. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* [S.l.], 2004. v. 3, p. 430–433. <https://doi.org/10.1109/ICPR.2004.1334558>.

[108] BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001. <https://doi.org/10.1023/A:1010933404324>.

[109] CHIU, A. L.; FU, A. W. Enhancements on local outlier detection. In: IEEE. *Seventh International Database Engineering and Applications Symposium, 2003. Proceedings.* [S.l.], 2003. p. 298–307. <https://doi.org/10.1109/IDEAS.2003.1214939>.

[110] TAN, S. C.; TING, K. M.; LIU, T. F. Fast anomaly detection for streaming data. In: *Twenty-Second International Joint Conference on Artificial Intelligence*. [S.l.: s.n.], 2011. <https://doi.org/10.5591/978-1-57735-516-8%2FIJCAI11-254>.

[111] HOODA, M. et al. An improved intrusion detection system based on kdd dataset using feature ranking and data sampling. In: IEEE. *2020 International Conference on Communication and Signal Processing (ICCSP)*. [S.l.], 2020. p. 1128–1132.

[112] KUMAR, V. et al. An integrated rule based intrusion detection system: analysis on unsw-nb15 data set and the real time online dataset. *Cluster Computing*, Springer, v. 23, n. 2, p. 1397–1418, 2020.

TRABALHOS PUBLICADOS PELO AUTOR

Trabalhos publicados pelo autor durante o programa.

1. G. F. Scaranti, L. F. Carvalho, S. Barbon and M. L. Proença, **Artificial Immune Systems and Fuzzy Logic to Detect Flooding Attacks in Software-Defined Networks**, IEEE Access, vol. 8, pp. 100172-100184, 2020, doi: 10.1109/ACCESS.2020.2997939. (Qualis CC A1)

Artigos submetidos que estão em revisão R1

1. G. F. Scaranti, L. F. Carvalho, S. Barbon and M. L. Proença, **Unsupervised online anomaly detection in Software-defined Network environments**, Expert Systems with Application. (Qualis CC A1)