Universidade Estadual de Londrina

CINARA BRENDA ZERBINI

WAVELETS PARA DETECÇÃO E MITIGAÇÃO DE ANOMALIAS EM AMBIENTE SDN

Londrina
2019

CINARA BRENDA ZERBINI

WAVELETS PARA DETECÇÃO E MITIGAÇÃO DE ANOMALIAS EM AMBIENTE SDN

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

Orientador: Prof. Dr. Mario Lemes Proença Jr

Londrina
2019

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

Zerbini, Cinara Brenda.
Wavelets para detecção e mitigação de anomalias em ambiente sdn / Cinara Brenda Zerbini. - Londrina, 2019.
107 f. : il.

Orientador: Mario Lemes Proença Junior.
Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Ciência da Computação, 2019.
Inclui bibliografia.

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

CINARA BRENDA ZERBINI

WAVELETS PARA DETECÇÃO E MITIGAÇÃO DE ANOMALIAS EM AMBIENTE SDN

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

BANCA EXAMINADORA

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

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

Profa. Dra. Ana Verginia Libos Messetti
Universidade Estadual de Londrina – UEL

Londrina, 28 de fevereiro de 2019.

Este trabalho é dedicado a Deus, que nos dá a cada dia o dom da vida, e nos permite experimentar o melhor de Sua criação, o amor e respeito ao próximo, e a oportunidade de apreciar Sua grandeza. E a minha família e amigos, por vivenciarem essa dádiva comigo.

AGRADECIMENTOS

Primeiramente gostaria de agradecer a Deus, pois não cai uma folha ao chão sem que Ele queira, então o agradeço por sempre me guiar e permitir que coisas boas aconteçam e também por sustentar-me nos momentos difíceis. Gostaria também de agradecer à minha família o apoio nos momentos difíceis e também o incentivo quando me encontrava desanimada, pela confiança em mim depositada e pelo amor e compreensão a mim dedicados.

Ao meu orientador Prof. Dr. Mario Lemes Proença Jr. primeiro por me orientar neste projeto, pela compreensão, incentivo e também pela confiança, contribuindo no meu crescimento pessoal e profissional.

Ao Prof. Dr. Luiz Fernando Carvalho por compartilhar seu conhecimento, por sempre me auxiliar, e também contribuir significativamente no meu crescimento pessoal e profissional, e ao Prof. Ms. Marcos Vinícius de Assis, também compartilhando seus conhecimentos e auxiliando nesse processo de produção.

Mando meus agradecimentos também ao meu companheiro de mestrado Matheus Pereira de Novaes, com o qual troquei experiências e conhecimentos, o apoio recíproco no decorrer do programa, e também a parceria.

A Prof. Dra. Ana Verginia Libos Messetti que sempre contribuiu com sua amizade, bons conselhos, e suas vibrações positivas. Também agradeço à ela por todo o auxílio no período de graduação e pelas sugestões construtivas para melhoria deste trabalho.

Ao Prof. Dr. Elieser Botelho Manhas Jr. pelas sugestões construtivas, que auxiliaram no meu crescimento profissional e pessoal.

A CAPES pela bolsa concedida.

Aos meus amigos, sempre ao meu lado torcendo por mim e me ajudando nos momentos difíceis.

Aos membros do Grupo de Redes do Departamento de Computação da Universidade Estadual de Londrina, pela amizade, pelo convívio e troca de experiências.

Aos que não foram explicitamente citados, sintam-se igualmente agradecidos, pois se em algum momento me ajudaram saibam que com certeza guardo sincera gratidão e respeito.

"Não to mandei eu? Esforça-te, e tem bom ânimo; não temas, nem te espantes; porque o Senhor teu Deus é contigo, por onde quer que andares."

(Bíblia Sagrada, Josué 1, 9)

ZERBINI, C. B. **Wavelets para detecção e mitigação de anomalias em ambiente SDN.** 95 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2019.

RESUMO

Ao longo dos anos a rede de comunicação Internet sofreu um gigantesco aumento, com um salto de quatro dispositivos em 1966 para a previsão de 50 bilhões de dispositivos em 2020, previsão esta que se deve principalmente a inclusão de bilhões de dispositivos IoT nos próximos anos. Ao passo que a rede se dissemina como meio de comunicação de massa, surge um aumento expressivo na quantidade de informações que trafegam na rede bem como a infraestrutura para transportá-las, infraestrutura esta que deve ser capaz de organizar e rotear toda essa troca de informações. Com toda essa estrutura, surgiu a preocupação com a integridade dos dados e também com os serviços disponibilizados na rede mundial, visto que esse aumento também proporcionou a ação de agentes maliciosos dentro desse ambiente. Essa preocupação fez surgir sistemas para auxiliar na detecção de anomalias, a fim de encontrar possíveis ataques dentro da rede. Além da preocupação com a segurança, esse crescimento na estrutura da rede trouxe a necessidade do gerenciamento desses recursos. Esse gerenciamento inicialmente foi concebido com a utilização de um ambiente descentralizado, o ambiente tradicional de gerenciamento, que possui diversas limitações. Para sanar algumas dessas limitações é que surge um novo paradigma de redes, que são as Redes Definidas por Software, ou SDN, que propõem um ambiente de redes centralizado com uma linguagem única (OpenFlow), o que facilita a configuração e o monitoramento da rede. É pautado nessas demandas que este projeto propõe o desenvolvimento de um sistema de detecção e mitigação de anomalias em redes SDN utilizando wavelets, que são ferramentas da área de processamento de sinais, contribuindo para a área de segurança de redes, propondo um sistema eficiente e de baixo custo computacional. Para validar a proposta apresentada, foram utilizados três cenários com diferentes características, a fim de verificar a eficácia da detecção e da mitigação dos ataques. Os testes demonstraram que as abordagens propostas proporcionaram uma boa solução para a detecção de ataques do tipo DDoS e Portscan, com altas taxas de detecção e taxas de falso-positivo abaixo de 1%.

Palavras-chave: Wavelets. Detecção de anomalias em redes. Redes definidas por software.

ZERBINI, C. B. **Wavelets for anomaly detection and Mitigation in SDN environment.** 95 p. Dissertation (Master's Deegre in Science in Computer Science) – Universidade Estadual de Londrina, Londrina, 2019.

ABSTRACT

Throughout the years, Internet communication emerged as a massive communication tool, starting with four hosts in 1966, to a forecast of 20 billions of hosts in 2020. This forecast is mainly affected by the fact that in future years IoT devices will increase its amount in network environment. As the network spreads as a mean of mass communication, there is an expressive increase in the amount of information that travels in the network as well as the infrastructure to transport them, this infrastructure must be able to organize and route all this exchange of information. With all this structure, emerged the concern about the integrity of the data as well as the services provided in the worldwide network, since this increase also provided the action of malicious agents within that environment. This concern has given rise to systems to assist in the detection of anomalies in order to detect possible attacks within the network. In addition to security concerns, this growth in network structure also emerged the need of management in these resources. This management was initially designed with the use of a decentralized environment, the traditional management environment, which has several limitations. To remedy some of these limitations is that a new network paradigm emerges, named Software-Defined Networking, or SDN, that propose a centralized network environment with an unique language of communications (OpenFlow), which facilitates the configuration and monitoring of the network. It is based on these demands that this research proposes the development of a system of intrusion detection and mitigation within SDN environments, using wavelet, a signal-processing based tool, contributing to the area of network security, proposing an efficient and low computational cost system. To validade our proposal, three scenarios were used, to analyze detection and mitigation effectiveness. Our analysis demonstrated that our proposal achieved high detection rates, with false alarm rate under 1%, which implies our proposal proper for DDoS and Portscan detection.

Keywords: Wavelets. Network anomaly detection. Software-defined networking.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

1D	1 dimensão
ABC	*Artificial Bee Colony*
ACO	*Ant-colony Optimization*
AFS	*Artificial Fish Swarm*
ARIMA	*Autoregressive Integrated Moving Average*
ATLANTIC	*Anomaly deTection and machine LeArNing Traffic classifICation for software-defined networking*
CoS	*Class of Service*
CPM	*Correlational Paraconsistent Machine*
CUSUM	*Cumulative Sum*
CWT	*Continuous Wavelet Transform*
DDoS	*Distributed Denial of Service*
DoS	*Denial of Service*
DSNSF	*Digital Signature for Network Segment using Flow Analysis*
DWT	*Discrete Wavelet Transform*
FN	*False Negative* (Falso Negativo)
FN-rate	*False Negative Rate*
FP	*False Positive* (Falso Positivo)
FP-rate	*False Positive Rate*
GA	*Genetic Algorithm*
ICMP	*Internet Control Message Protocol*
IDS	*Intrusion Detection System*
IETF	*Internet Engineering Task Force*
IoT	*Internet of Things*

IP	*Internet Protocol*
IPFIX	*Internet Protocol Flow Information Export*
ISO	*International Organization for Standardization*
kNN	*k Near Neighbors*
LUS	*Local Unimodal Sample*
ONF	*Open Networking Foundation*
PCA	*Principal Component Analysis*
PSO	*Particle Swarm Optimization*
R2L	*Remote to Local*
SDN	*Software-Defined Networking*
sFlow	*Sampling Flow*
SGMP	*Simple Gateway Management Protocol*
SNMP	*Simple Network Management Protocol*
SO	Sistema Operacional
SSL	*Secure Socket Layer*
SVM	*Support Vector Machine*
TCP	*Transmission Control Protocol*
TN	*True Negative* (Verdadeiro Negativo)
TP	*True Positive* (Verdadeiro Positivo)
TM	*Traffic Matrix*
U2R	*User to Root*
UDP	*User Datagram Protocol*
UEL	Universidade Estadual de Londrina
WMA	*Weighted Majority Algorithm*

LISTA DE SÍMBOLOS

α	Nível de significância
$c_{j_0,k}$	Coeficientes de aproximação no último nível j_0 com K elementos
$c_{j,k}$	Coeficientes de aproximação no nível j com K elementos
$d_{j,k}$	Coeficientes de detalhes no nível j com K elementos
ϵ	Multiplicador de σ para gerar o intervalo em torno de μ
$f(W_f)$	Transformada Discreta Wavelet para W_f
$f(W_d)$	Transformada Discreta Wavelet para W_d
$f(X)$	Transformada Discreta Wavelet para X
F	Matriz esparsa contendo os filtros da DWT
g_0	Primeiro filtro passa-baixa da DWT
g_1	Segundo filtro passa-baixa da DWT
g_2	Terceiro filtro passa-baixa da DWT
g_3	Quarto filtro passa-baixa da DWT
h_0	Primeiro filtro passa-alta da DWT
h_1	Segundo filtro passa-alta da DWT
h_2	Terceiro filtro passa-alta da DWT
h_3	Quarto filtro passa-alta da DWT
H_0	Hipótese nula
$H(Y)$	Entropia de Shannon para o atributo Y
IQR	Distância Interquartil
j	Nível arbitrário da DWT
j_0	Último nível de decomposição da DWT
k	Valor entre 1 e K

K	Tamanho do conjunto de coeficientes de detalhes ou aproximação da DWT
l	valor entre 1 e N
μ	Média populacional
M	Tamanho da janela deslizante
N	Tamanho de um sinal ou dado arbitrário de entrada da DWT
$\psi_{j,k}(X)$	função *wavelet* ou *wavelet*-mãe
$Q1$	Primeiro quartil
$Q3$	Terceiro quartil
σ	Desvio padrão populacional
σ^2	Variância populacional
s	Vetor de saída da DWT com coeficientes de detalhes e aproximação
S	Somatório de todas as ocorrências presentes no histograma
t	Posição da amostra em análise
$\varphi_{j_0,k}(X)$	Função escala ou *wavelet*-pai
W_d	Janela deslizante de previsão (DSNSF)
W_f	Janela deslizante de detecção
x_t	Elemento sob análise pertencente ao conjunto de dados X
X	Conjunto arbitrário de dados
y_i	Número de ocorrência da amostra i no intervalo de análise
Y	Atributo arbitrário do tráfego de rede

SUMÁRIO

1 INTRODUÇÃO

A Internet tem passado por inúmeras transformações desde seu surgimento, tornando-se cada vez mais essencial no cotidiano comercial, acadêmico, militar e domiciliar. Isso corroborou para a crescente quantidade de dispositivos conectados à rede. Toda essa necessidade por estar conectado a fim de facilitar processos ou otimizar atividades cotidianas contribui para o surgimento de tecnologias como a Internet das Coisas (IoT) e a computação ubíqua. A Internet das Coisas surgiu com a proposta de incorporar a Internet nos mais diversos objetos cotidianos de forma ubíqua, isto é, sem que notemos sua atuação, como por exemplo carros autônomos, que possuem controle a gerenciamento remoto via conexão com a Internet [3] [4]. Isso certamente causa um aumento exponencial no número de dispositivos conectados, bem como na demanda por qualidade de serviços, visto que atualmente existem mais de cinco bilhões de dispositivos conectados a rede e a previsão, com a incorporação massiva do IoT, seja de 50 bilhões de dispositivos em 2020 [5] [6] [7].

Ao passo que a rede mundial se consolida como meio de comunicação de massa, com sua implantação no ambiente comercial e sua disseminação dentro de áreas domiciliares, a infraestrutura para comportar toda essa gama de usuários tornou-se notável. Com tantos dispositivos conectados, fez-se necessário o gerenciamento e a configuração de todos esses recursos, tal qual o monitoramento das interfaces da rede, com o objetivo de encontrar falhas, melhorar a qualidade dos serviços fornecidos e também proteger as informações e os serviços disponibilizados [8].

Diversas ferramentas foram desenvolvidas para auxiliar o gerente de rede. Estas ferramentas permitem que o gerente configure, gerencie e monitore o tráfego da rede, a fim de otimizar serviços, proteger recursos e identificar possíveis anomalias, sejam elas causadas por falhas mecânicas, de configuração, humanas, ou ainda por intrusos (*malwares*) [9].

As anomalias causadas por falhas mecânicas consistem na queima ou desgaste de algum equipamento da rede. As de configuração ocorrem devido a algum equipamento mal configurado ou modificação na configuração. As falhas humanas consistem em falhas causadas pelo uso inadequado da rede, ou por uma quantidade excessiva de usuários acessando um mesmo serviço na rede (*Flash Crowd*). E, por fim, as falhas causadas por intrusos, que ocorrem quando uma anomalia é causada por um agente malicioso, como *malwares*, ataques DoS ou DDoS, *Portscan*, entre outros.

Quando falhas ocorrem por causa de intrusos, é então necessário que o gerente tome medidas para identificar esta intrusão, e a partir disto minimizar os efeitos deste ataque para que toda a estrutura da rede, bem como os dados e informações que circulam nela,

se mantenham seguros e íntegros. Para tal medidas, o gerente necessita de ferramentas que o auxiliem neste processo, como os sistemas de detecção de anomalias (IDS).

Na área de segurança, os IDS são imprescindíveis. Com o auxílio de protocolos de extração de dados da rede [10], [11], [12], e de técnicas matemáticas e computacionais, os IDS [13], [14] têm acesso a informações sobre o comportamento da rede e com isso são capazes de detectar comportamentos anômalos dentro da rede. Esses sistemas possuem duas vertentes para detecção: os sistemas baseados em assinaturas, que comparam o tráfego com assinaturas de ataques conhecidos, e os sistemas baseados em caracterização, que geram um padrão do tráfego normal [15], [16] e por meio dele identificam se existe ou não comportamento anômalo no tráfego [17].

Estes IDS, como dito, tornaram-se indispensáveis para o ambiente de redes atual. No entanto, é importante destacar também que o paradigma de rede que dispomos atualmente nos fornece uma estrutura de difícil configuração e manuseio, por possuir uma gama de dispositivos com aplicações proprietárias e não padronizadas entre si, e também por não concentrar todo o plano de controle dentro destes dispositivos [18]. Esse ambiente tradicional também dificulta a implantação de IDS e outras ferramentas auxiliares para o gerenciamento e a segurança da rede.

Com o intuito de atender estas e outras demandas um novo paradigma de redes emergiu: as Redes Definidas por *Software* (SDN). As redes SDN consistem em um novo paradigma de rede, tendo como principais características a separação do plano de controle com o plano de ação e a unificação e centralização do gerenciamento e monitoramento da rede [19], [20]. Por meio dos controladores SDN, que são como sistemas operacionais da rede, é possível realizar tarefas e implantar ferramentas auxiliares de maneira mais facilitada que no ambiente tradicional de rede [21], como os supracitados IDS ou ainda ferramentas de gerenciamento e até de controle de acesso.

É notória a importância do desenvolvimento de sistemas de detecção de intrusos (IDS), principalmente dentro do novo paradigma de rede, o SDN. Para tal propósito, diversas técnicas de diferentes áreas podem ser utilizadas, como as teorias estatísticas, teorias matemáticas, os algoritmos de aprendizado de máquina ou de otimização, ou ainda teorias da área de processamento de sinais, como as *wavelets*, que são foco do presente estudo [22].

As *wavelets* são ferramentas que decompõem sinais, ou séries temporais em suas partes espectrais, isto é, em suas diferentes frequências. Por meio da transformada *wavelet*, é possível dividir um sinal em coeficientes de detalhes, que carregam as faixas de frequências mais altas, e em coeficientes de aproximação, que representam as faixas contendo as baixas frequências [23].

Este trabalho tem por objetivo o desenvolvimento de um sistema para detecção e

mitigação de anomalias em redes SDN utilizando *wavelets*. Para a detecção das anomalias duas abordagens utilizando *wavelets* foram propostas, uma nomeada WaveDetect–ND que utiliza *wavelets* com intervalo de confiança baseado em distribuição normal, e a segunda, WaveDetect–IQR, utilizando distância interquartil. Além das propostas para detecção, também foi desenvolvido um sistema de mitigação das anomalias. Para realizar a validação da proposta, foram considerados três cenários, com diferentes topologias e com intervalos de coleta diferentes, sendo estes de trinta segundos, cinco segundos e um segundo.

A seguir são listadas as principais contribuições a partir da realização deste trabalho:

- Desenvolvimento de duas abordagens utilizando *wavelets*, a primeira utilizando conceitos de distribuição normal e a segunda, distância interquartil;
- Testes em cenários com baixa amostragem, permitindo uma reação mais efetiva contra possíveis ataques;
- Implementação do sistema em ambiente de redes SDN;
- Desenvolvimento de um sistema de mitigação para minimizar os impactos dos ataques.

Este trabalho está organizado da seguinte forma: no capítulo 2 é realizada uma revisão dos trabalhos mais relevantes desenvolvidos na área de detecção de anomalias em redes, o capítulo 3 apresenta uma revisão dos conceitos primordiais acerca da área de segurança de redes e detecção de anomalias, o capítulo 4 explana os conceitos de redes definidas por *software*. O capítulo 5 expõe o modelo desenvolvido bem como os conceitos a respeito de *wavelets*, necessários para o seu entendimento, já o capítulo 6 apresenta os resultados das análises realizadas para os três cenários propostos. Por fim, o capítulo 7 explana as considerações finais com respeito às análises realizadas.

2 TRABALHOS RELACIONADOS

A detecção e mitigação de anomalias em redes são campos desafiadores, considerando o fato de que a quantidade e os tipos de ataques aumentam a cada dia. Dentro desta temática, diversas teorias já foram utilizadas na tentativa de fornecer um modelo capaz de detectar ataques em redes. Diversos trabalhos foram desenvolvidos nos últimos anos, alguns utilizando o ambiente de redes tradicional, outros utilizando o paradigma de Redes Definidas por *Software* (SDN). Dentre esses trabalhos, alguns se utilizam da teoria de *wavelets*, como o trabalho de Tian *et al.* [24] que desenvolveram um método de detecção de anomalias utilizando duas ferramentas, uma Matriz de Tráfego (TM) e *wavelets*. Uma TM corresponde a dados de cinco minutos do tráfego da rede, sendo que esta pode conter tráfego normal ou anômalo. Uma transformada *wavelet* é realizada nesta matriz, produzindo coeficientes que fornecem informação histórica do tráfego. Por meio desse histórico, alguns parâmetros são coletados. Uma comparação entre esses coeficientes e os coeficientes considerados anormais é realizada, com o objetivo de identificar ataques do tipo DDoS. Esta técnica mostrou altos índices de detecção, próximos de cem por cento, e uma taxa de alarmes falsos próxima de seis por cento. A desvantagem em utilizar esta abordagem está no fato das matrizes possuírem dados de cinco minutos, o que em uma rede atual com links de 10Gbps, 100Gbps, ou até 400Gbps, significa a troca de grandes quantidades de informações e dados, fazendo com que a detecção ocorra de forma tardia e consequentemente ineficaz.

Outro trabalho relevante na área utilizando *wavelets* foi desenvolvido por Dainotti *et al.* [25] que elaboraram um sistema para a detecção de ataques do tipo DoS à rede. Este sistema é dividido em dois estágios, nomeados: *"Rough Detection"* e *"Fine Detection"*. O primeiro estágio caracteriza o tráfego utilizando algoritmos CUSUM *(Cumulative Sum)* e posteriormente seleciona possíveis pontos de DoS, por meio de um limiar adaptativo. O segundo estágio trata de analisar esses pontos de possíveis ataques por meio de uma Transformada *Wavelet* Contínua (CWT), reduzindo assim as taxas de falso-positivo. Ainda que a CWT auxilie na redução da taxa de falso-positivo, o modelo proposto por Dainotti *et al.* apresentou alta taxa de falso-positivo, mas também teve uma alta taxa de acerto na detecção de intrusos.

Ainda utilizando *wavelets*, temos outros trabalhos desenvolvidos também no ambiente de rede tradicional. Kanarachos *et al.* [26] propuseram um sistema que utiliza uma combinação de três técnicas: *wavelets*, redes neurais e transformada de Hilbert. O modelo se divide em três estágios. No primeiro estágio foi realizada uma transformada *wavelet* de Daubechies com oito níveis de decomposição (DB8), e então foi aplicada uma técnica de remoção de ruídos, por meio da análise dos coeficientes resultantes da transformada

wavelet. No segundo estágio, um subconjunto dos dados sem ruído foi escolhido para treinamento da rede neural, gerando uma previsão do tráfego. No terceiro e último estágio foi utilizada a transformada de Hilbert no erro do sinal, que é o sinal resultante da diferença entre o sinal filtrado e a saída do treinamento da rede neural. Essa solução utilizando três ferramentas diferentes pode apresentar alguns problemas de processamento computacional quando aplicada em sistemas de detecção *fast*, já que as redes neurais podem ter um alto custo computacional. Os métodos apresentados pontuaram taxas de detecção satisfatórias, mas como eles foram desenvolvidos para redes tradicionais, rotinas de mitigação se tornam mais complexas de serem implementadas.

Gao *et al.* [27] apresentaram um modelo de detecção de anomalias utilizando *wavelet packet*, que é uma generalização do algoritmo piramidal da transformada *wavelet* tradicional. Os autores definiram que o método proposto é capaz de detectar anomalias *"long-term"* e médias frequências. O sistema também garante uma melhoria na confiabilidade da detecção utilizando uma reconstrução adaptativa dos coeficientes de detalhe da transformada *wavelet*, incluindo anomalia.

Ainda dentro do ambiente de redes tradicionais, o Grupo de Redes do Departamento de Ciência da Computação da Universidade Estadual de Londrina (UEL) também propôs algumas soluções para detecção e mitigação de anomalias. Os trabalhos têm foco na construção de uma previsão do tráfego da rede, nomeado DSNSF (Assinatura Digital para Segmento de Rede Usando Análise de Fluxo) [28] [29] [30]. Nesta linha, Pena e Carvalho *et al.* [31] propuseram uma solução utilizando duas abordagens diferentes para gerar dois DSNSFs. A primeira utilizou metaheurística de otimização por colônias de formigas (ACO) que consiste em um conjunto de métodos e estratégias computacionais baseados nos comportamentos de colônias de formigas reais. A segunda utilizou a Média de Movimento Integrado Autoregressivo (ARIMA), uma ferramenta capaz de capturar a interdependência linear do futuro em relação aos valores passados. A detecção de anomalia foi realizada utilizando a Máquina Paraconsistente Correlacional (CPM). Os resultados da solução proposta foram comparados com a abordagem de detecção de anomalia tradicional baseada em intervalos de confiança, mostrando que o CPM foi muito eficaz na detecção de anomalias e na diminuição das taxas de falso-positivo.

Hamamoto *et al.* [32] propuseram um esquema combinando Algoritmo Genético (GA) para gerar um DSNSF e Lógica *Fuzzy* para o estágio de detecção de anomalias. A abordagem proposta pelos autores mostrou que o GA forneceu uma boa solução para a caracterização do tráfego e a Lógica *Fuzzy*, quando comparada com outras abordagens, apresentou resultados excelentes com precisão acima de 96% e taxa de falso-positivo abaixo de 1%.

Fernandes *et al.* [33] também propuseram uma solução, que consiste em um método de detecção de anomalias utilizando dois níveis. O primeiro nível realizou uma caracteri-

zação do tráfego dos atributos da rede, como bits e pacotes transmitidos em um intervalo de tempo, utilizando o método estatístico de Análise de Componentes Principais (PCA). O segundo nível executou uma rotina de detecção de anomalias usando o autovalor PCA como um parâmetro para definir o intervalo considerado normal com base na média do DSNSF. Na avaliação da detecção, os autores testaram a abordagem baseada em PCA com diferentes limiares, gerados por um autovalor variável de 2% a 4% para mais ou para menos. Os resultados mostraram que o DSNSF projetou uma previsão adequada do tráfego, enquanto a detecção de anomalia foi operativa, com 94% de acerto em um modelo de baixa complexidade computacional. Esta abordagem de detecção de anomalias em dois níveis é semelhante ao proposto nesta dissertação, salvo o fato de que o trabalho de Fernandes *et al.* é voltado para o ambiente de rede tradicional ao invés de redes SDN.

Amaral *et al.* [34] propuseram um sistema de detecção de anomalias para identificar e classificar anomalias captando características importantes sobre o uso da rede por meio de fluxos IP.

Outros trabalhos alheios ao grupo e também relevantes foram desenvolvidos, como o trabalho de Hajisalem e Babaie [35] que propuseram um sistema híbrido de detecção de anomalias, combinando dois algoritmos metaheurísticos denominados *Artificial Bee Colony* (ABC) e *Artificial Fish Swarm* (AFS).

Aburomman *et al.* [36] desenvolveram um sistema utilizando a combinação de algumas técnicas: Máquina de Vetor de Suporte (SVM), *k-Near Neighbors (kNN)*, Otimização por Enxame de Partículas (PSO) e o Algoritmo de Maioria Ponderada (WMA). Primeiro, um estágio de treinamento é realizado utilizando seis especialistas em kNN e seis na SVM, que são coleções de cinco classificadores binários para o comportamento da rede. Com a combinação de todos os atributos dos doze especialistas, foram obtidos três conjuntos, o primeiro por meio do PSO, o segundo utilizando Amostragem Local Unimodal (LUS) juntamente com o PSO e o terceiro utilizando WMA. Os autores analisaram a detecção de quatro tipos de anomalias: DoS, Probe, U2L (*User to Root*) e R2L (*Remote to Local*). As taxas de detecção de Probe, DoS e U2L foram superiores a 95% e para R2L foi de cerca de 85%, com a abordagem LUS fornecendo taxas de detecção mais altas. Por outro lado, a LUS obteve o maior tempo de execução, com valores quase um milhão de vezes maior que os valores do WMA, a abordagem mais rápida. Devido ao alto custo computacional de uma combinação de quatro técnicas de complexidade moderada e à necessidade de uma rotina de treinamento, que exige tráfego rotulado anteriormente, este método não é uma solução adequada para sistemas de detecção de anomalia em tempo real, visto que nestes sistemas é necessário um processamento ágil o qual é proporcionado por técnicas de baixo custo computacional.

Todos esses trabalhos foram desenvolvidos dentro do ambiente de redes tradicional. Porém, com o surgimento do novo paradigma de redes, as redes SDN (*Software-defined*

Networking), surgiu também a necessidade do desenvolvimento e do estudo de sistemas de detecção de anomalias dentro deste novo ambiente. Dentre alguns dos principais trabalhos desenvolvidos, temos o estudo de Silva *et al.* [37] e Sahay *et al.* [38]. Silva *et al.* desenvolveram um *framework* chamado Atlantic, para detecção, classificação e mitigação de anomalias no ambiente SDN. O *framework* proposto combina duas fases; a primeira, de baixo custo computacional, detecta de forma rápida fluxos potencialmente maliciosos; e a segunda, com custo computacional mais elevado, utiliza um algoritmo de aprendizado de máquina baseado em SVM para classificação dos fluxos do tráfego. Ambas as fases resultaram em baixo tempo de execução para detecção e classificação, apresentando melhor desempenho em ataques DDoS do que em anomalias do tipo *Portscan*. O sistema apresentado permite uma detecção rápida e eficiente e mostra como o SDN fornece um ambiente propício para implementação de sofisticados sistemas de detecção de anomalias em redes. Sahay *et al.* propuseram uma estrutura para detecção e mitigação de DDoS chamada ArOMA. Os autores construíram um *framework* onde a detecção foi realizada comparando fluxos do tráfego com uma base de dados de tráfego mantido por uma rede de clientes. Esta abordagem pode dificultar a detecção de novas ameaças DDoS, uma vez que a base de dados pode sofrer com a falta de informações fornecidas pelo cliente da rede ou a não atualização da base de dados. Os resultados numéricos demonstraram que o bom rendimento médio é maior nos ataques DDoS UDP do que nos ataques DDoS ICMP e TCP, mas de maneira geral o sistema foi capaz de fornecer uma resposta rápida na mitigação de ataques do tipo DDoS independente do protocolo.

Ainda dentro do ambiente de redes SDN temos o trabalho de Peng *et al.* [39] em que foi apresentado um modelo para detecção de ataques do tipo DDoS em ambiente SDN, utilizando uma teoria de p-valor associado ao algoritmo de classificação *k-Nearest Neighbors* (kNN). Com dados coletados a cada dez segundos, o método proposto apresentou uma *performance* com taxas de detecção de aproximadamente 95% e *Area Under Curve* (AUC) igual a 97.88%. Prasath e Perumal [40] também desenvolveram um IDS utilizando uma técnica de metaheurística Bayesiana para classificação de anomalias no ambiente de redes SDN. Primeiramente os autores realizaram uma etapa de treinamento que consiste em alguns estágios: pré-processamento, normalização dos dados, extração de características dos dados, otimização desses descritores, e por fim uma avaliação destes descritores. Após isto é realizada uma classificação destes dados entre normal ou anômalo.

O Grupo de Redes do Departamento de Computação da UEL também desenvolveu trabalhos utilizando o paradigma SDN. Entre os principais podemos citar o trabalho de Assis e Hamamoto *et al.* [41] que produziram dois modelos para detecção e mitigação de anomalias do tipo DoS/DDoS, sendo o primeiro uma combinação de teoria de jogos e *Holt-Winters* (GT-HWDS) e o segundo uma combinação de Algoritmo Genético (GA) para caracterização e Lógica *Fuzzy* para detecção (GT-*Fuzzy*-GADS) ou ainda o trabalho de Carvalho *et al.* [42] que desenvolveram um ecossistema para detecção e mitigação de

anomalias em ambiente SDN.

Apesar das diferenças e particularidades com relação às técnicas utilizadas, a maioria dos trabalhos apresentados nesta seção partem da obtenção de um comportamento normal do tráfego da rede. Com isso, é possível observar que as detecções são realizadas por meio da análise comparativa do tráfego com o *baseline* da rede, o que caracteriza um modelo de sistema de detecção de anomalias baseado em assinaturas, e que devido à alta gama de novos intrusos, torna-se o método mais adequado para a detecção de anormalidades no tráfego da rede.

No entanto, diversos estudos apresentados anteriormente não se utilizam do novo paradigma SDN. Também é possível verificar que algumas das soluções não possuem um mecanismo de detecção adequado para detecção em tempo real e também não proporcionam cenários com baixos intervalos de coleta de informações do tráfego. Pensando nisto, nossa proposta apresenta uma abordagem que utiliza o novo paradigma de redes (SDN) propondo um modelo de detecção baseado em assinaturas utilizando uma ferramenta de baixa complexidade computacional. Tudo isto aliado a baixos intervalos de coleta (um e cinco segundos), torna a detecção mais eficiente no combate de possíveis ataques, proporcionando uma mitigação mais rápida desses intrusos e desta forma proporcionando maior segurança no ambiente de redes.

A Tabela 1 apresenta um resumo dos trabalhos apresentados nesta seção e as técnicas utilizadas para a detecção de anomalias e intrusos.

Tabela 1 – Descrição das principais características dos trabalhos relacionados. TD = Taxa de Detecção e TFP = Taxa de Falso Positivo.

Trabalho	**Técnica(s) Utilizada(s)**	**Fonte de Dados**	**TD (%)**	**TFP**
Tian *et al.* [24]	*Wavelets*	Abilene	100	5, 5
Dainotti *et al.* [25]	*Wavelets*/CUSUM	Darpa (1, 2, 3)/UCLA/UNINA	86, 3	27, 2
Kanarachos *et al.* [26]	*Wavelets*/RNA/Transf. Hilbert	Ma/keogh	-	-
Gao *et al.* [27]	*Wavelets*	*Internet Traffic Archive*	-	-
Pena e Carvalho *et al.* [31]	ACO/ARIMA/CPM	Fluxos IP (UEL)	95	4
Hamamoto et al. [32]	GA/Lógica *Fuzzy*	Fluxos IP (UEL)	96, 53	0, 56
Fernandes *et al.* [33]	PCA/Limiar	Fluxos IP (UEL)	94	23
Amaral *et al.* [34]	Extração de características dos fluxos	Fluxos IP (UTFPR/GW)	99, 4	0, 12
Hajisalem e Babaie [35]	ABC/AFS	UNSW-NB15/NSL-KDD	83, 3/100	0, 01
Aburomman *et al.* [36]	SVM/kNN/PSO/WMA	KDD99	98, 8534	-
Silva *et al.* [37]	SVM	SDN	88, 7	-
Sahay *et al.* [38]	Comparação com base de dados	SDN	-	-
Peng *et al.* [39]	p-valor/kNN	SDN	95%	$< 5\%$
Prasath e Perumal [40]	Metaheurística Bayesiana	SDN	77, 02	0, 12
Assis e Hamamoto *et al.* [41]	Teoria de Jogos/*Holt-Winters* e GA/Lógica *Fuzzy*	SDN	> 98	-
Carvalho *et al.* [42]	ACO/Regressão Logística	SDN	95, 1	0

3 GERÊNCIA E SEGURANÇA DE REDES

Com a disseminação das redes de computadores como meio de comunicação de massa e também como ferramenta para a realização de diversas atividades, houve um crescimento na quantidade de usuários e por consequência nos equipamentos de rede. Todos esses equipamentos necessitam de configuração e também de controle e monitoramento, ou seja, precisam ser gerenciados.

Esta necessidade fez com que, ao longo do tempo, diversos avanços na área de gerência de redes ocorressem, como o desenvolvimento de equipamentos com recursos para seu gerenciamento e ferramentas para auxiliar esse processo [43], [44]. Foi pensando na formalização desses processos que a ISO [45] acabou então por formalizar as áreas da gerência de redes, dividindo em cinco área funcionais denominadas FCAPS, que são:

- *Fault Management* ou Gerência de Falhas (F): área responsável pela detecção, localização e tratamento de problemas de *hardware* ou *software*;
- *Configuration Management* ou Gerência de Configuração (C): área que trata dos registros de inventário de *hardware* e *software*, histórico de modificação dos dispositivos, e também auxilia na permissão da inicialização dos sistemas que compõem a rede (como o sistema operacional e a configuração de um roteador);
- *Accounting Management* ou Gerência de Registros (A): registra a utilização da rede a fim de contabilizar a utilização dos recursos da mesma;
- *Performance Management* ou Gerência de Performance (P): esta área permite aos administradores monitorar algumas variáveis chave da rede, como: *throughput*, tempo de resposta e disponibilidade, auxiliando-os em uma análise de como e em qual aspecto o desempenho da rede pode ser melhorado;
- *Security Management* ou Gerência de Segurança (S): fiscaliza o acesso a informações e recursos da rede, através de tarefas como: verificação do privilégio de acesso à rede dos usuários, detecção e registro das tentativas de acesso não autorizado.

Dentro da área de gerência existem algumas ferramentas e conceitos relevantes, os quais trataremos a seguir.

3.1 Coleta de Dados

A coleta de dados consiste na forma como os dados da rede são coletados. Para realizar essa coleta algumas ferramentas foram desenvolvidas ao longo do tempo. A primeira ferramenta de destaque desenvolvida com este intuito foi o protocolo SNMP (*Simple Network Management Protocol*).

O protocolo SNMP surgiu no final da década de 80, desenvolvido pela IETF (*Internet Engineering Task Force*). A base para o desenvolvimento do SNMP foi um protocolo de monitoramento de *gateways* IP, o SGMP (*Simple Gateway Management Protocol*). O SNMP surgiu com um conjunto de operações simples que permitem que dispositivos da rede sejam gerenciados de maneira remota. O SNMP utiliza o paradigma de funcionamento gerente/agente de forma que a comunicação gerente/agente é feita por meio do protocolo UDP (*User Datagram Protocol*). O uso do protocolo UDP torna a comunicação gerente/agente não confiável, portanto cabe à aplicação SNMP determinar se os pacotes foram ou não perdidos e retransmiti-los se assim o desejar [46]. O protocolo SNMP armazena informações do agente em uma lista de variáveis definida como MIB *(Management Information Base)*. A primeira versão do protocolo (SNMPv1) era mais simples. A versão 2 (SNMPv2) é uma evolução da primeira versão, com mais operações e melhorias na MIB. A terceira versão do protocolo SNMP (SNMPv3) introduziu mecanismos de segurança e autenticação na comunicação gerente/agente. Com o aumento da complexidade das redes também surgiu a demanda por tecnologias que fossem capazes de realizar a coleta de informações não contidas nos objetos SNMP. Neste contexto surgiu a gerência baseada na análise de fluxos IP [8], [47], [48].

Um fluxo IP é, de forma básica, uma sequência unilateral de vários pacotes IP com atributos semelhantes, entre os quais temos: endereço IP de origem e destino, porta de origem e destino, tipo de protocolo de transporte, tipo de serviço (CoS) e interface do roteador/*switch* [34]. A primeira ferramenta formalizada neste paradigma de coleta foi o *NetFlow*, lançado em 1996 pela *Cisco Systems*. Mais tarde, em 2001, a InMom lançou o protocolo *sFlow* (*Sampling Flow*), que diferencia-se do *NetFlow* principalmente pela ideia da amostragem de pacotes. O *sFlow* foi desenvolvido para atuar de maneira simples em redes com taxas de transferências de 10 e 100 Gb/s [8]. Por fim, em 2004 a IETF (*Internet Engineering Task Force*) lançou o protocolo de extração de fluxos chamado IPFIX *(Internet Protocol Flow Information Export)*, que trouxe uma padronização na sua utilização, já que as ferramentas *NetFlow* e *sFlow* eram proprietárias [11]. A Figura 1 apresenta um exemplo de fluxos coletados por meio do *sFlow*. Cada fluxo possui informações como: data, horário, protocolo utilizado, IP de origem e destino, Porta de origem e destino, quantidade de pacotes e *bytes*, entre outras informações.

Data	Início do fluxo	Duração	Protocolo	Origem IP:Porta		Destino IP:Porta	Pacotes	*Bytes*	Fluxos
2015-10-06	00:00:02.895	0.000	TCP	216.58.222.14:443	->	189.90.65.127:37552	512	34816	1
2015-10-06	00:00:02.895	0.000	TCP	54.231.9.3:443	->	10.65.22.195:57055	512	34816	1
2015-10-06	00:00:02.895	0.000	TCP	64.233.186.95:443	->	10.65.20.107:48998	512	734208	1
2015-10-06	00:00:02.895	0.000	TCP	54.231.9.3:443	->	10.65.22.195:57055	512	50176	1
2015-10-06	00:00:02.895	0.000	GRE	187.18.100.230:443	->	189.90.65.25:57055	512	774144	1
2015-10-06	00:00:02.895	0.000	TCP	187.0.89.220:9050	->	189.90.64.132:61573	512	728064	1
2015-10-06	00:00:02.895	0.000	ICMP	198.27.81.92:3	->	189.90.71.178:0.3	512	43520	1
2015-10-06	00:00:02.895	0.000	GRE	187.18.97.19:3	->	189.90.65.25:3	512	774144	1
2015-10-06	00:00:10.885	0.000	TCP	54.231.9.3:443	->	10.65.22.195:57054	512	34816	1
2015-10-06	00:00:10.885	0.000	TCP	54.231.9.3:443	->	10.65.22.195:57055	512	34816	1
2015-10-06	00:00:10.885	0.000	UDP	109.236.92.157:32906	->	189.90.65.248:53	512	43520	1
2015-10-06	00:00:10.885	0.000	TCP	187.0.89.220:9050	->	189.90.64.132:61573	512	728064	1
2015-10-06	00:00:10.885	0.000	TCP	187.18.102.60:53470	->	189.90.67.84:22	512	37888	1
2015-10-06	00:00:10.885	0.000	TCP	189.14.52.80:443	->	10.90.31.61:62461	512	779264	1
2015-10-06	00:00:10.885	0.000	GRE	187.18.100.230:443	->	189.90.65.25:62461	512	774144	1

Figura 1 – Exemplo de um fluxo IP coletado com *sFlow*.

3.2 Detecção de Anomalias e Intrusos

Como citado anteriormente, a área de segurança auxilia na proteção das informações, dispositivos e serviços presentes na rede. Algumas ferramentas são imprescindíveis, como os sistemas de detecção de anomalias. Esses sistemas têm o objetivo de identificar quando e qual anomalia está presente no tráfego da rede, para posteriormente realizar uma mitigação dos efeitos dessas anomalias [49]. Ainda podemos citar, de uma maneira mais restrita, os sistemas de detecção de intrusos (IDS – *Intrusion Detection Systens*), que são sistemas que detectam e tratam apenas anomalias causadas por agentes maliciosos. Uma anomalia é qualquer trecho de tráfego que foge de um padrão esperado, enquanto intrusos são anomalias causadas por agentes maliciosos. De forma sucinta, todo intruso é uma anomalia, mas nem toda anomalia é um intruso. Deste ponto em diante, nos referiremos a esses sistemas como IDS, visto que o foco deste estudo é exatamente a detecção de anomalias ocasionadas de forma maliciosa.

Os IDS podem seguir duas vertentes: baseados em assinaturas, ou baseados em caracterização. Os sistemas baseados em assinaturas comparam uma base de dados de assinaturas de ataques conhecidos e, quando há semelhança entre alguma assinatura e o dado coletado do tráfego da rede, um alarme é disparado. Como principal vantagem dessa abordagem, temos as baixas taxas de alarmes falsos e também uma alta taxa de acerto. Por outro lado, esse tipo de sistema de detecção possui a desvantagem de necessitar de atualizações constantes, devido ao surgimento contínuo de novas anomalias. Considerando que diariamente surgem diversos tipos de anomalias, é necessário manter uma base de dados com assinaturas dos ataques, o que torna a implantação deste tipo de IDS dificultada, devido a grande demanda por armazenamento e processamento [50].

Já a detecção de intrusos baseada em caracterização considera um intruso como uma alteração do tráfego padrão da rede. Para detectar o que está ou não neste padrão, o sistema gera uma previsão do comportamento normal da rede, chamado de perfil ou

baseline. Para geração deste perfil ou assinatura digital ou ainda *baseline* os sistemas são construídos utilizando ferramentas que podem ser, entre outras, matemáticas, estatísticas, ou ainda algoritmos para classificação e otimização [51]. Este tipo de sistema para detecção de anomalias possui a vantagem de não depender das assinaturas dos ataques, mas sim de uma assinatura do padrão da rede, o que torna a detecção mais ampla. Contudo, devido à falta de detalhes sobre um determinado tipo de anomalia, a detecção pode sofrer com alguns alarmes falsos bem como perder um pouco de sua eficiência no que tange à detecção de intervalos anômalos dentro do tráfego de rede. Este tipo de sistema é o foco da proposta aqui apresentada.

Os IDS baseados em caracterização possuem uma arquitetura geral, que pode ser visualizada na Figura 2.

Figura 2 – Arquitetura geral de um IDS.

Primeiramente os dados da rede são coletados na etapa ”Entrada de Dados da Rede“. Esses dados contêm informações sobre o tráfego da rede, como a quantidade de *bytes* ou pacotes por determinado intervalo de tempo, ou IP e porta de origem e destino. Tudo isso pode ser obtido por meio dos protocolos baseados em análise de fluxos IP (*NetFlow*, *sFlow* ou IPFIX).

Após a coleta, os fluxos coletados são processados para que possa ser feita uma análise dos atributos da rede.

Na etapa ”Caracterização do Tráfego“ é aplicada uma técnica para a geração de um perfil normal da rede, utilizando os dados processados. Feito isto, tem-se um padrão que descreve o tráfego esperado para uma determinada interface de rede.

Por fim, utilizando este padrão, a etapa de ”Detecção de intrusos“ analisa se existe ou não um intruso no tráfego da rede, gerando a ”Saída do Sistema“.

Essas técnicas utilizadas tanto na caracterização do tráfego quanto na detecção de intrusos podem ser divididas em algumas áreas principais, como apresentado na Figura 3.

Os algoritmos de classificação consistem em ferramentas que têm como objetivo classificar um grupo de dados, de forma supervisionada ou não. Essas técnicas são amplamente utilizadas em IDS baseados em caracterização. Os algoritmos de otimização são técnicas que visam buscar o melhor resultado para um determinado problema. Alguns

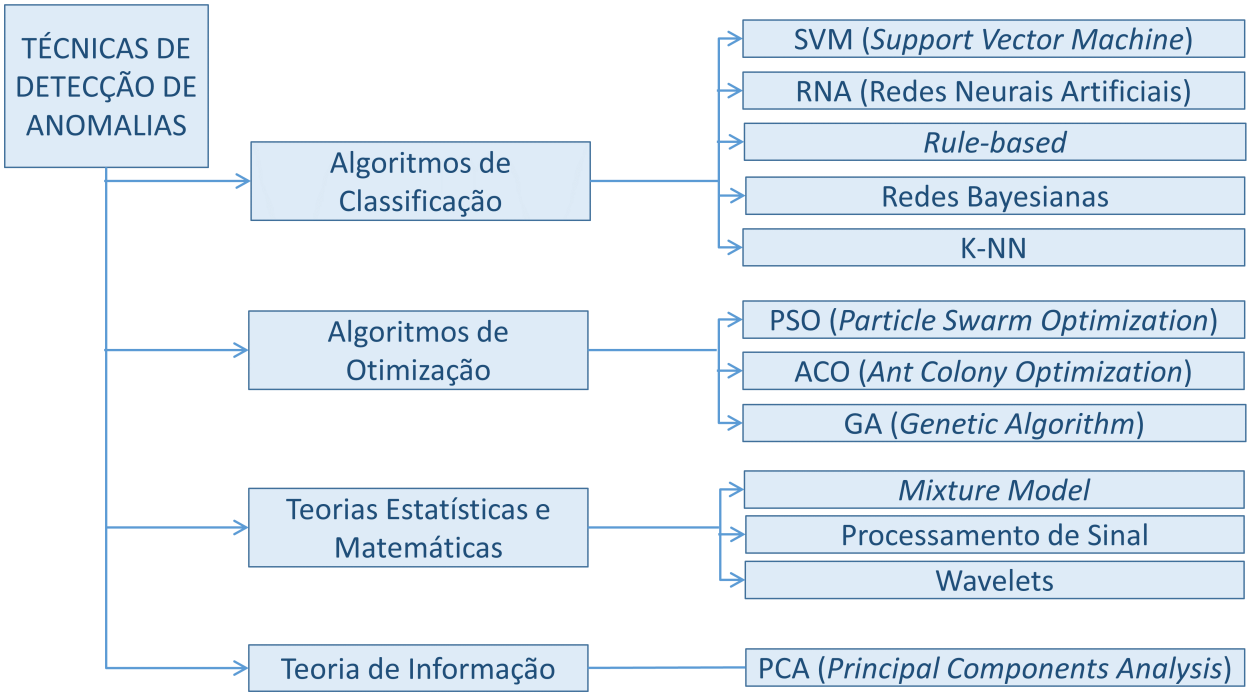


Figura 3 – Algumas técnicas de detecção de anomalia em redes.

estudiosos também desenvolveram trabalhos utilizando teorias estatísticas, ou de processamento de sinais ou análise de séries temporais. E por fim, alguns trabalhos da área apresentam propostas de IDS utilizando teoria da informação. Todas essas teorias e ferramentas que compõem um IDS são utilizadas para a detecção de intrusos. A seção a seguir apresentará alguns exemplos de intrusos e também de anomalias [52].

3.2.1 Tipos de Anomalias e Intrusos

As anomalias podem ou não ser causadas por agentes maliciosos. A seguir serão apresentados os principais tipos de anomalias causadas ou não por esses agentes.

Anomalias que não possuem agentes maliciosos, isto é, anomalias que não são causadas por uma pessoa voluntariamente [53]:

- *Flash Crowd*: quando, de forma súbita, uma grande quantidade de usuários passa a enviar (não maliciosamente) requisições para um determinado servidor podendo fazer com que este interrompa suas operações;
- *Babbling Node*: um nó da rede entra em falha por tempo indeterminado enviando pacotes de maneira indiscriminada para vários pontos da rede;
- Tempestade de *Broadcasts*: ocorre quando uma grande quantidade de pacotes de *broadcast* começa a trafegar pela rede, causando congestionamentos que podem

levar à interrupção das operações da rede;

- Congestionamentos: aumento do tráfego de forma brusca, causando atraso e até descarte de pacotes;
- *Bugs* nos *softwares* de roteamento: esses *bugs* ocorrem quando os *softwares* contidos nos equipamentos de roteamento não foram preparados para tratar pacotes corrompidos, ou com mal formações;
- Erros em configurações: um *firewall* mal configurado pode causar a perda de diversos pacotes por meio do bloqueio desses pacotes. Um servidor mal configurado pode não conseguir responder a todas as requisições, causando congestionamentos.

Anomalias causadas por agentes maliciosos, ou seja, anomalias causadas por pessoas com intenções levianas [53]:

- DoS e DDoS: a anomalia do tipo DDoS é um tipo de ataque realizado por um computador atacante, que escraviza até milhões de computadores (chamados *slaves*), utilizando-os para "derrubar" um determinado alvo (vítima) e torná-lo indisponível ou inacessível por um determinado tempo, por meio do envio de milhões de requisições, como pode ser visualizado na Figura 4.

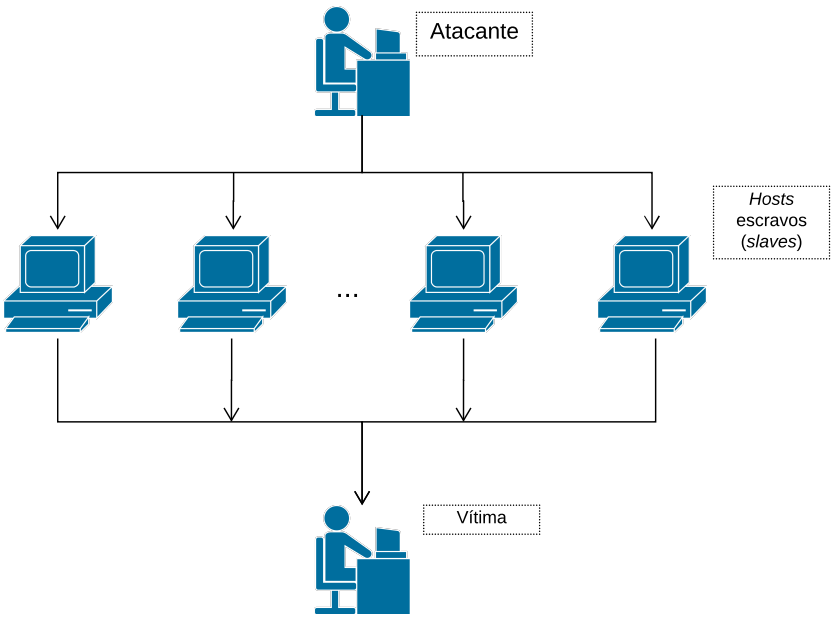


Figura 4 – Descrição de um ataque DDoS.

Os alvos mais comuns são os servidores web. O objetivo destes "ataques" não é invadir ou coletar informações de um sistema, mas sim exaurir e causar indisponibilidades ao alvo. Muitas vezes, os donos de computadores escravizados

nem têm conhecimento disto. Segundo o Comitê Gestor da Internet no Brasil [54], os ataques do tipo DDoS podem ser realizados por diversos meios, como:

  - pelo envio de grandes quantidades de requisições para um serviço, consumindo os recursos necessários ao seu funcionamento (processamento, número de conexões simultâneas, memória e espaço em disco, por exemplo) e impedindo que as requisições dos demais usuários sejam atendidas;
  - pela geração de grande tráfego de dados para uma rede, ocupando toda a banda disponível e tornando indisponível qualquer acesso a computadores ou serviços desta rede;
  - pela exploração de vulnerabilidades existentes em programas, que podem fazer com que um determinado serviço fique inacessível.

  A diferença entre ataques DDoS e DoS está na quantidade de computadores atacantes e de computadores *slaves*. Os ataques do tipo DoS são feitos apenas por um atacante, já o DDoS além de poder ter mais de um atacante, ainda contam com os *hosts* chamados *slaves*.

- *Worms*: são *softwares* capazes de se espalhar e executar cópias em máquinas remotas ao longo da rede, infectando grandes conjuntos de alvos;
- Escaneamento de Portas (*Portscan*): esta técnica é utilizada por administradores de rede para testar a vulnerabilidade da mesma, porém, agentes maliciosos se utilizam desta técnica com o objetivo de encontrar essas vulnerabilidades para realizar ataques.

3.3 Considerações Sobre o Capítulo

Uma rede deve prover ferramentas para sua configuração, gerenciamento, monitoramento e proteção. Alguns protocolos e tecnologias são essenciais nessas demandas. Neste capítulo foram abordadas algumas dessas ferramentas, como os protocolos de extração de dados da rede, que inicialmente foi representado pelo protocolo SNMP, hoje em desuso, e mais tarde pelos protocolos de extração de fluxos de pacotes. Também tornaram-se indispensáveis ferramentas que tratem da segurança das informações da rede, como os antivírus, *firewalls* e os sistemas de detecção de intrusos. Esses sistemas, que podem ser baseados em assinaturas ou ainda baseados em caracterização, proporcionam uma solução para detectar e posteriormente tratar possíveis intrusos na rede.

Recentemente, outras tecnologias têm se apresentado como indispensáveis para o gerenciamento da rede e a implantação de mecanismos de segurança de forma facilitada,

como as Redes Definidas por *Software* (SDN), que se trata de um novo paradigma de rede e que serão abordadas no próximo capítulo.

4 REDES DEFINIDAS POR *SOFTWARE*

Como dito anteriormente, com o crescimento das redes, uma grande quantidade de componentes (equipamentos) é necessária para manter a organização e o funcionamento adequado da mesma.

A configuração desses dispositivos é feita por meio de *softwares* proprietários, em que cada empresa que fabrica um determinado *hardware* (dispositivo) fabrica também o *software* embarcado, para a configuração deste equipamento. Cada um desses *softwares* embarcados possuem uma linguagem própria e também suas particularidades. Essa especificidade na hora de configurar componentes da rede com camadas de *software* e *hardware* proprietários não padronizados, e ainda sua arquitetura com plano de controle distribuído, acabam por tornar as redes atuais limitadas.

Junto a isso, os altos custos em se manter o modelo de arquitetura vigente também atrapalha a evolução das redes.

Para sanar essas deficiências no ambiente de redes atual um novo paradigma de gerenciamento de rede foi desenvolvido, desvinculando o plano de controle (*software*) do plano de ação da rede, facilitando a escalabilidade e a configuração dos componentes da rede. Tudo isso com um plano de controle centralizado [55], [56].

4.1 Visão Geral de SDN

Software-Defined Networking (SDN) têm se mostrado importante no contexto de administração de redes [57], [58], e adiante será explicado o contexto geral dessa tecnologia, sua arquitetura, seus componentes e os principais tipos de controladores.

O SDN nasceu nas redes de campus universitários. Os pesquisadores estavam frustrados por terem que a cada nova abordagem experimentada na administração de redes, ter que modificar o *software* nos dispositivos de rede. Daí surgiu a ideia de tornar o comportamento dos dispositivos de rede programável, permitindo que eles fossem controlados por um elemento central. Isso levou a uma formalização dos principais elementos que definem a SDN hoje [56]:

- Separação das funções de controle e encaminhamento;
- Centralização do controle;
- Capacidade de programar o comportamento da rede usando interfaces bem definidas.

A próxima área de sucesso para SDN foram os *data centers* em nuvem. À medida que o tamanho e o escopo desses *data centers* se expandiam, tornou-se claro que era necessária uma maneira melhor de conectar e controlar a explosão de máquinas virtuais. Os princípios da SDN logo mostraram-se promissores em melhorar a forma como os centros de dados poderiam ser controlados.

De acordo com a ONF (*Open Networking Foundation*), SDN pode ser definida como: "uma arquitetura emergente, que é dinâmica, manejável, economicamente viável, e adaptável, fazendo com que esta seja ideal para alta largura de banda, e para a natureza dinâmica das aplicações atuais. Esta arquitetura desacopla o controle e encaminhamento da rede, tornando o controle da rede diretamente programável e a infraestrutura subjacente a ser abstrata para aplicações e serviços de rede. O protocolo *OpenFlow* é um elemento base para a construção de soluções utilizando a tecnologia SDN" [59]. A ONF determina que a arquitetura SDN seja [60]:

- Diretamente programável: a camada de controle da rede é desacoplada das funções de encaminhamento. Isso torna a arquitetura SDN diretamente controlável;
- Ágil: O controle de encaminhamento de fluxos abstrato permite que os administradores ajustem dinamicamente o fluxo de tráfego em toda a rede para atender às necessidades da mesma;
- Gestão centralizada: a inteligência da rede é centralizada nos controladores SDN, que possuem visão global da rede. Isso permite que as outras camadas da rede identifiquem esta camada como um único *switch* lógico;
- Configuração programável: as redes SDN permitem que os administradores de rede configurem, protejam e otimizem os recursos da rede de forma rápida utilizando aplicações dinâmicas e autônomas do ambiente SDN, que podem ser desenvolvidas pelos próprios administradores, pois estes não dependem de *software* proprietário;
- Padrões abertos e neutros para fornecedores: quando as redes SDN são desenvolvidas por meio de padrões abertos, seu projeto e operação se tornam simplificados, pois as instruções são fornecidas por meio dos controladores SDN, e não mais por diversos dispositivos diferentes com protocolos específicos de suas fabricantes.

A Tabela 2 apresenta de forma resumida as principais características das redes SDN.

Tabela 2 – Resumo das principais características de redes SDN.

Característica	Observação
Motivo pelo qual foi criada	Separação de controle e dados; centralização de controle, e; programabilidade da rede
Local aplicação-alvo	Campus; data centers; nuvem
Dispositivos alvos	Servidores e switches comerciais
Aplicações iniciais	Administração da nuvem e da rede
Novos protocolos	*OpenFlow*
Formalização	ONF

4.2 Arquitetura

Uma rede SDN está pautada na ideia de camadas de abstração. Estas camadas podem ser observadas na Figura 5, onde é possível visualizar as três camadas e também os principais componentes contidos em cada uma delas. A primeira camada, chamada camada de infraestrutura, corresponde ao plano de ação e comporta os dispositivos da rede como roteadores, *switches* e pontos de acesso. A principal função desta camada é o encaminhamento dos pacotes de acordo com as regras definidas. A segunda camada, nomeada camada ou plano de controle, consiste em um controlador que controla as principais funções SDN. Essa camada é o ponto de comunicação entre as camadas de infraestrutura e de aplicação. Esse controlador é responsável pelo gerenciamento do tráfego da rede, e com isso é capaz de tomar decisões a respeito do roteamento, encaminhamento a eliminação de pacotes. Os controladores nesta camada se comunicam com o plano de ação por meio do protocolo *OpenFlow*.

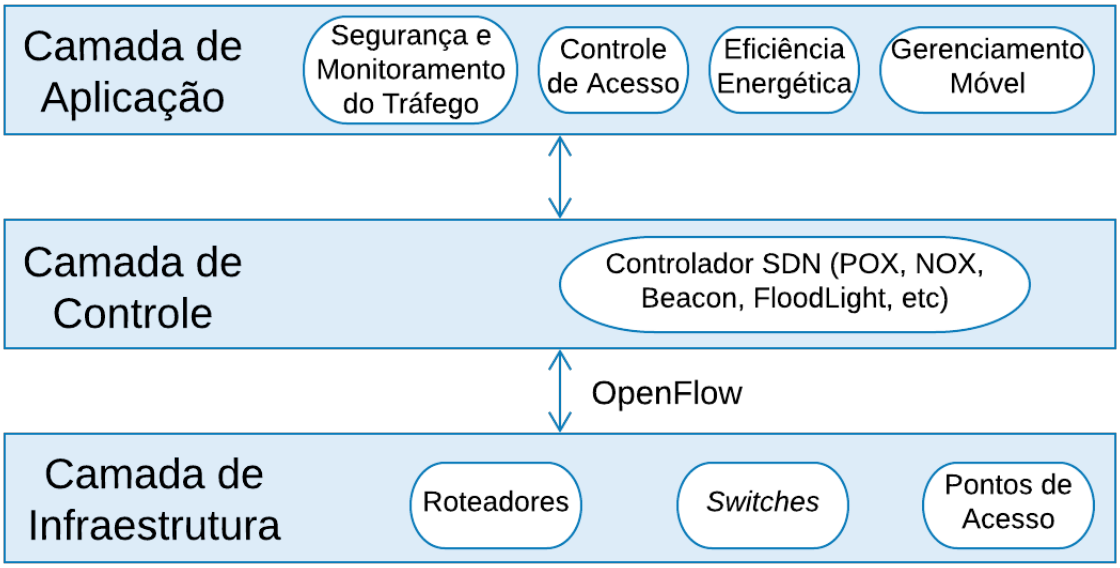


Figura 5 – Arquitetura geral de uma rede SDN. (Adaptado de: [1])

A terceira e última camada, a camada de aplicação, trata das aplicações de segurança, da virtualização da rede, dos IDS, IPS (*Intrusion Prevention System*), *firewalls* e gerenciamento de mobilidade. Toda essa estrutura abstrata pode ser melhor compreendida por meio da Figura 6 que apresenta um exemplo de estrutura de rede sob o paradigma SDN [1] [61].

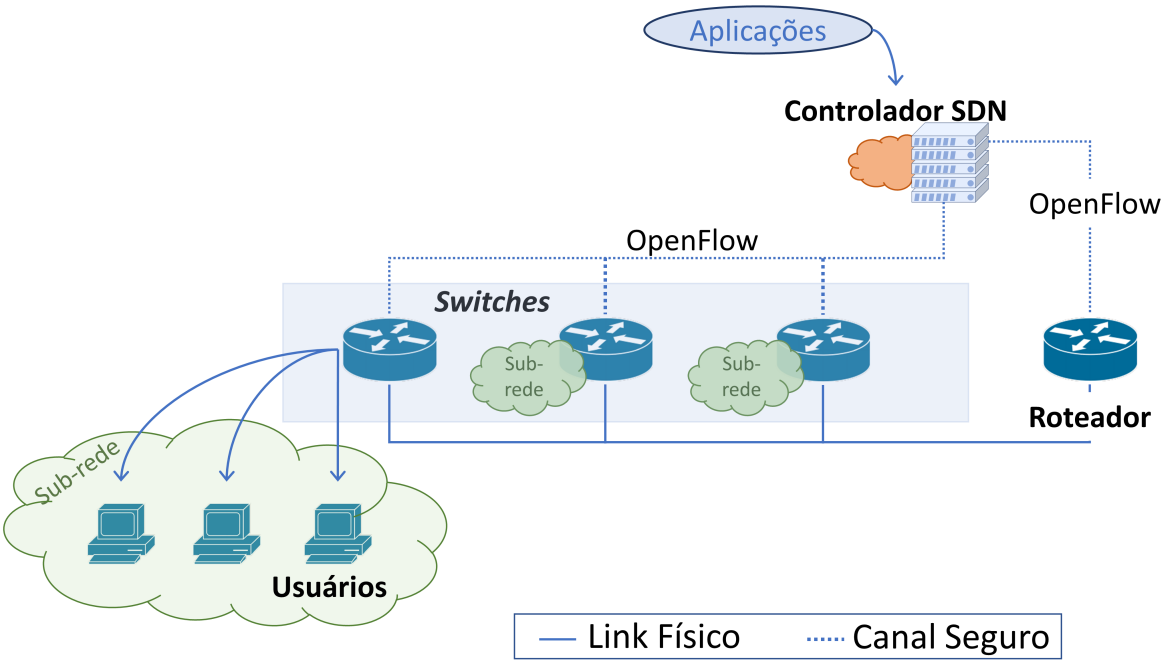


Figura 6 – Exemplo de uma rede SDN.

Uma rede SDN deve possuir basicamente componentes de *hardware* habilitados para o protocolo *OpenFlow*. Entre os principais componentes SDN temos o protocolo *OpenFlow*, a tabela de fluxos, o canal seguro e os controladores SDN [55].

4.2.1 *OpenFlow*

OpenFlow é um protocolo aberto utilizado para a comunicação entre os equipamentos da rede e os controladores. O *OpenFlow* utiliza uma interface externa para esta troca de mensagens.

4.2.2 Canal Seguro

O canal seguro nada mais é que o canal por meio do qual é realizada a comunicação interna entre os dispositivos da rede e os controladores SDN. Este canal é utilizado a fim de evitar ataques de agentes maliciosos. O canal seguro utiliza SSL (*Secure Socket Layer*) como interface de acesso, que é a interface mais recomendada.

4.2.3 Tabela de Fluxos

Uma tabela de fluxos consiste em entradas de fluxos que por sua vez consistem em regras, ações e contadores. Cada regra consiste de um valor de um ou mais campos do cabeçalho de um pacote. As ações definem como os pacotes devem ser roteados. Os contadores mantêm as informações estatísticas a respeito do uso dos fluxos de pacotes.

4.2.4 Controlador

É um *software* que tem a função de tomar decisões além de adicionar e remover entradas na tabela de fluxos, conforme a necessidade. O controlador exerce a função de uma camada de abstração da camada física, o que facilita o desenvolvimento de aplicações e serviços que manipulem as entradas da tabela de fluxos. Pode-se então dizer que o controlador *OpenFlow* atua como um sistema operacional (SO) para o gerenciamento e controle das redes. A SDN também oferece uma plataforma que tem como base a reutilização de componentes e a ideia de níveis de abstração. O controlador SDN permite ainda a evolução em paralelo de tecnologias no plano de dados bem como inovações na lógica das aplicações no plano de controle. Entre essas aplicações podemos ter um IDS.

Ao longo de sua história, foram criados alguns controladores SDN, como o controlador NOX que é o controlador original *OpenFlow*, desenvolvido em C++; o controlador POX, construído em *Python* e baseado no NOX; o Beacon, que é um controlador multiplataforma baseado em Java, e o controlador FloodLight, que é um controlador SDN gratuito desenvolvido em Java, com foco em redes comerciais [62].

4.3 Considerações Sobre o Capítulo

As redes tradicionais, com sua estrutura descentralizada e com dispositivos de configuração diferenciada entre si, dificultam a escalabilidade e a configuração das redes. Com isso, a SDN surge com a proposta de fornecer um ambiente de rede centralizado, com planos de controle e ação desvinculados entre si, e com uma linguagem única para a configuração dos dispositivos da rede, que é o protocolo *OpenFlow*. Essa tecnologia se mostra promissora, pois permite uma comunicação facilitada e única, além de contribuir para melhora do desempenho da rede, desvinculando tarefas de controle do plano de ação. Além dessas vantagens, sua estrutura centralizada permite uma maior facilidade na implementação de sistemas embarcados na rede, como os sistemas de detecção de intrusos, que será tema do próximo capítulo.

5 SISTEMA DE DETECÇÃO DE ANOMALIAS *WAVEDETECT*

A caracterização e a detecção de intrusos em redes é uma área importante para auxiliar seu gerenciamento e monitoramento [10]. O tráfego de uma rede geralmente contém um comportamento cíclico com relação aos diferentes períodos do dia. Um dos principais objetivos deste trabalho é apresentar um modelo que seja capaz de representar esse comportamento normal da rede e, a partir deste ponto, realizar a detecção de intrusos.

Este capítulo descreve as principais características, conceitos e as fases do modelo de detecção de intrusos nomeado *WaveDetect*. Para o entendimento dos processos que constituem esse sistema, inicialmente será realizada um contextualização e conceitualização sobre *wavelets* e posteriormente serão explicadas as fases do processo de caracterização e detecção do sistema *WaveDetect*.

Para as fases de caracterização e detecção, duas abordagens foram propostas, uma nomeada *WaveDetect–ND* e outra *WaveDetect–IQR*.

Além das fases de caracterização e detecção também foi desenvolvido um mecanismo de mitigação, com o intuito de amenizar os efeitos de cada uma das anomalias aqui estudadas. Essa fase também será explicada com detalhes mais adiante neste capítulo.

5.1 *Wavelets*

As *wavelets* são utilizadas em diversas áreas, como no processamento digital de sinais, no processamento de imagens, entre outras. As transformadas *wavelets* surgiram da necessidade de uma ferramenta matemática que manipulasse dados e trouxesse mais precisão na análise tempo-frequência, já que as teorias anteriores não sanavam essa necessidade de forma eficiente, como é o caso das teorias acerca da Transformada de Fourier, que acabou por contribuir para o surgimento das *wavelets*.

A Transformada de Fourier tem sua primeira menção em meados de 1829 na publicação da obra de Jean Baptiste Joseph Fourier (1768-1830) *"Mémoire sur la théorie de la chaleur"*, no livro *"Extrait du mémoire lu à l'Académie des sciences"* [63].

Depois desta publicação, ao longo da história, diversos matemáticos, como o alemão Wierstrass (1815-1897), contribuíram para o passo da matemática discreta à contínua.

Todo esse esforço por parte dos matemáticos trouxe uma ferramenta poderosa e de ampla aplicação, sendo, principalmente na física, de notável indispensabilidade.

A teoria da Transformada de Fourier possui aplicações nas áreas mais diversas,

como criptografia, processamento de imagens, oceanografia, física e química quântica, teoria dos números, teoria das probabilidades, das distribuições, análise matemática, gerência de redes e, principalmente, no processamento digital de sinais [64].

Porém, há uma desvantagem no uso desta teoria, que pode ser explicada por meio de uma analogia com o princípio da incerteza de Heisenberg, que afirma ser possível identificar onde uma partícula está, ou o quão rápido está, mas nunca as duas coisas. A Transformada de Fourier possui o mesmo princípio, ou asseguramos a frequência de um sinal, ou asseguramos sua posição no tempo, e nunca os dois, ou seja, não há precisão exata dos dois parâmetros (frequência e tempo) [2]. Essa deficiência contida na Teoria de Fourier deu margem a estudos que levaram ao surgimento das teorias de *wavelets*.

As primeiras teorias sobre *wavelets* foram desenvolvidas por Alfred Haar no ano de 1909, no entanto apenas em meados de 1985 ocorreu um novo avanço significativo nesta temática, com a Escola Francesa. Os principais estudiosos a contribuírem com esse avanço foram, primeiramente Stéphane Mallat, na área de processamento digital de imagens, depois Yves Meyer, que contribuiu com os seus estudos para a construção da primeira *wavelet* não-trivial. E por fim, Ingrid Daubechies, que com base nos trabalhos anteriores, construiu um conjunto de bases ortonormais de *wavelets* suaves (não-triviais), com suportes compactos [65]. O modelo desenvolvido por Daubechies, juntamente com o modelo de Alfred Haar, é o mais utilizado atualmente.

5.1.1 Definição de *Wavelets*

Segundo afirma Chui [66], as *wavelets* são ferramentas versáteis com um rico conteúdo matemático e grande potencial de aplicação.

Lima [65] diz que as transformadas *wavelets* podem ser vistas como mecanismos para decompor ou quebrar sinais nas suas partes constituintes, permitindo analisar os dados em diferentes domínios de frequências com a resolução de cada componente amarrada à sua escala [23].

Daubechies define *wavelets* como uma ferramenta que fatia dados ou funções ou operadores em componentes com frequências diferentes [67].

Além dos trabalhos desenvolvidos na área de detecção de anomalias em redes (vide Capítulo 2), as *wavelets* possuem aplicações em diversas áreas relevantes, entre as quais podemos citar o processamento e compressão de imagens [68], a análise da temperatura do ar [69], e até a detecção precoce de surtos de doenças [70].

As transformadas *wavelets* podem ser contínuas (CWT) ou discretas (DWT). O sistema *WaveDetect* utiliza um tipo de DWT. Uma DWT divide os dados de entrada em coeficientes de detalhes ou *wavelet*, representados por $d_{j,k}$, que trazem as frequências mais altas dos dados de entrada, e em coeficientes de aproximação ou escala ($c_{j,k}$), que

carregam a informação mais ”grosseira“ do sinal de entrada, que consiste nas frequências mais baixas.

Matematicamente, uma DWT pode ser representada por (5.1) [71], onde $\varphi_{j_0,k}(X)$ é a função escala, também chamada de *wavelet*-pai, gerando os coeficientes de aproximação $c_{j_0,k}$ do nível j_0 e $\psi_{j,k}(X)$, que nada mais é que a função *wavelet*, também chamada de *wavelet*-mãe, que define os filtros da DWT e é responsável pela geração dos coeficientes de detalhes $d_{j,k}$ ao longo dos níveis de decomposição. Ambos coeficientes (detalhes e aproximação) são compostos por K elementos, onde K depende do tamanho do sinal de entrada do nível anterior da DWT.

$$f(X) = \sum_{k=1}^{K} c_{j_0,k}(k)\,\varphi_{j_0,k}(X) + \sum_{j=j_0}^{\infty}\sum_{k=1}^{K} d_{j,k}\psi_{j,k}(X) \tag{5.1}$$

A DWT pode ainda ser interpretada como a aplicação de filtros a um determinado dado de entrada, que consiste em uma multiplicação de duas matrizes, descrito na equação (5.2), em que F é a matriz com os filtros *wavelet* de passa-alta (h) e passa-baixa (g), $\mathbf{X}$ contém os dados de entrada, e $\vec{s}$ consiste na saída da multiplicação de F por $\mathbf{X}$, que são os coeficientes de detalhes e aproximação. Os filtros *wavelet* são definidos por meio da função *wavelet* utilizada [72]. Nessa proposta as funções *wavelet* utilizadas são a de Haar e a de Daubechies, já que na literatura Haar é dita computacionalmente tratável e de baixa complexidade computacional e Daubechies consiste no filtro *wavelet* mais utilizado [73] [70].

Para obter os níveis subsequentes da DWT, conforme apresentado na Figura 7 os coeficientes de aproximação do nível anterior serão a nova entrada para a nova multiplicação de matrizes.

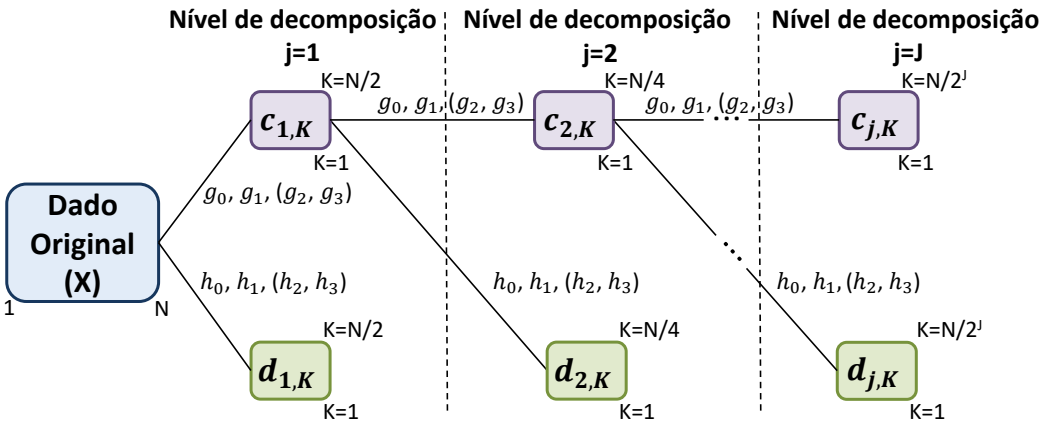


Figura 7 – Estrutura dos níveis de decomposição da DWT.

$$F\mathbf{X} = \vec{s} \tag{5.2}$$

5.1.2 Principais Tipos de *Wavelets*

Como dito anteriormente, existem as transformadas *wavelets* contínuas (CWT), e as transformadas *wavelets* discretas (DWT). Existem diversos tipos de CWT e DWT, como pode ser visualizado na Figura 8. Entre os filtros contínuos podemos citar Morlet e *wavelet* Chapéu Mexicano (*Mexican Hat*), e entre os discretos os filtros de Haar, Daubechies, Coiflets, Symlets e Discrete Meyer. Os dois principais e mais utilizados filtros *wavelets*, são os filtros de Haar e de Daubechies.

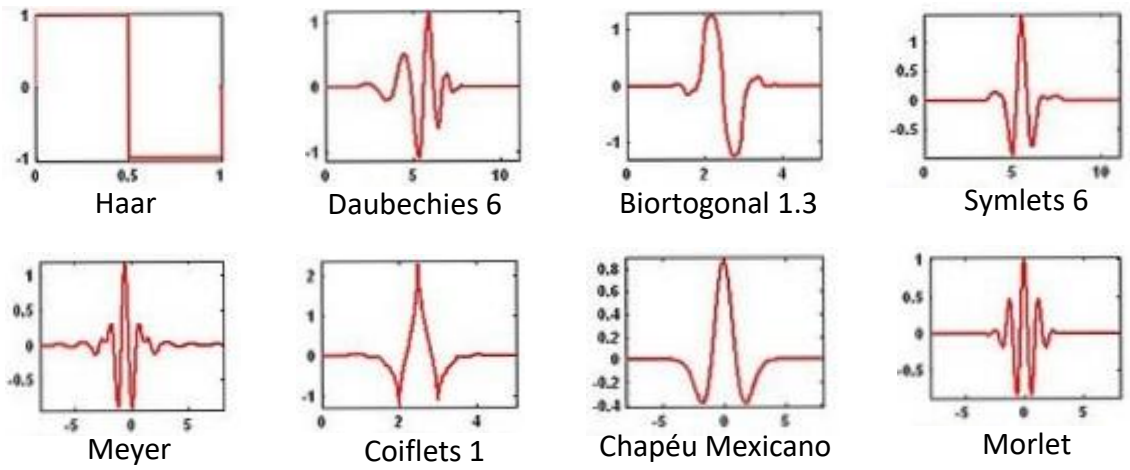


Figura 8 – Principais tipos de *wavelets* (filtros). (Adaptado de: [2])

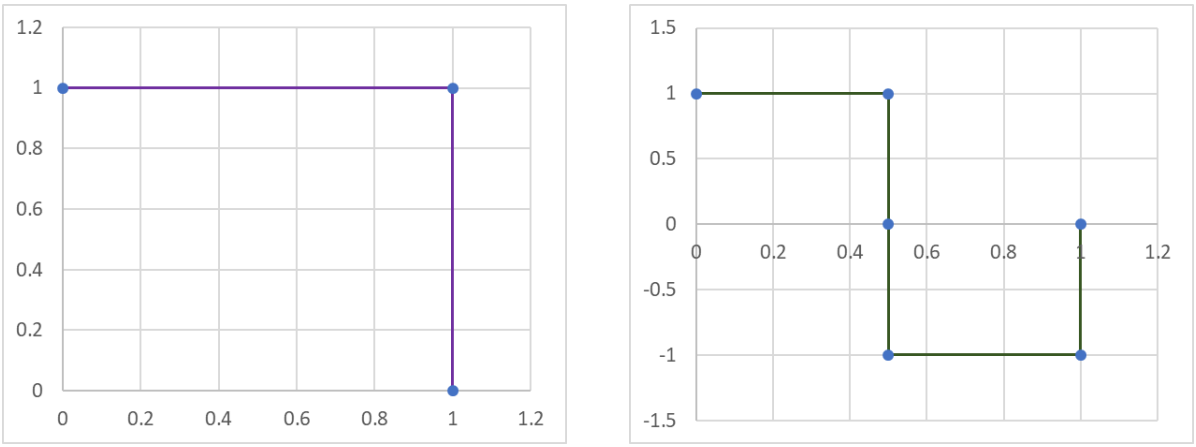
5.1.2.1 *Wavelets* de Haar

De conceito mais simples, as *wavelets* de Haar são as primeiras a surgirem, através do matemático húngaro Alfred Haar. As *wavelets* de Haar são obtidas através da Transformada de Haar, que nada mais é que uma transformação discreta de um conjunto de dados (sinais, funções). Suas funções *wavelets* e escala, são apresentadas em (5.3) e (5.4) [65] [66].

$$\varphi(x) = \begin{cases} 1, \text{ se } X \in [0,1) \\ 0, \text{ caso contrário} \end{cases} \tag{5.3}$$

$$\psi(x) = \begin{cases} 1, \text{ se } X \in [0,\frac{1}{2}) \\ -1, \text{ se } X \in [\frac{1}{2},1) \\ 0, \text{ caso contrário} \end{cases} \tag{5.4}$$

Graficamente, as definições das funções escala e *wavelet* de Haar podem ser visualizadas nas Figuras 9.



(a) Função escala $\varphi_{j_0,k}(\mathbf{X})$ (b) Função *wavelet* $\psi_{j,k}(\mathbf{X})$

Figura 9 – Gráfico com a representação da função escala e *wavelet* de Haar.

Como já explicado, a DWT pode ser representada por uma multiplicação de matrizes. Para a *wavelet* de Haar, essa multiplicação está descrita com mais detalhes na equação (5.5), onde h representa os filtros de passa-alta e g os de passa-baixa. Os coeficientes de detalhes ($d_{j,K}$) e os coeficientes de aproximação ($c_{j,k}$) são obtidos por meio da multiplicação produzindo um vetor de saída $\mathbf{s}$ com K elementos no nível de decomposição j, onde $K = N/(j^2)$ e N é o tamanho dos dados de entrada.

$$\begin{pmatrix} g_0 & g_1 & 0 & \dots & 0 \\ & & \ddots & & \\ 0 & 0 & \dots & g_0 & g_1 \\ h_0 & h_1 & 0 & \dots & 0 \\ & & \ddots & & \\ 0 & 0 & \dots & h_0 & h_1 \end{pmatrix}_{N\times N} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{N-2} \\ x_{N-1} \\ x_N \end{pmatrix}_{N\times 1} = \begin{pmatrix} c_{j,1} \\ c_{j,2} \\ \vdots \\ c_{j,K} \\ d_{j,1} \\ d_{j,2} \\ \vdots \\ d_{j,K} \end{pmatrix}_{N\times 1} \tag{5.5}$$

A transformada de Haar pode ser considerada como um caso particular das *wavelets* de Daubechies, sendo chamadas também de *wavelets* de Daubechies D2.

As *wavelets* de Haar, segundo coloca Gauciniski [74], estão relacionadas com operações matemáticas chamadas Transformadas de Haar. Essas transformadas decompõem um sinal em dois sub-sinais discretos, um de média ou tendência (aproximação) e o outro uma diferença ponderada de dois impulsos vizinhos (detalhes). Esses dois sub-sinais são obtidos por meio dos filtros de passa-baixa (g) e passa-alta (h) de Haar, definidos em (5.6)

e (5.7).

$$g_0 = \frac{1}{\sqrt{2}} \ , \ g_1 = \frac{1}{\sqrt{2}} \tag{5.6}$$

$$h_0 = -\frac{1}{\sqrt{2}} \ , \ h_1 = \frac{1}{\sqrt{2}} \tag{5.7}$$

5.1.2.2 *Wavelets* de Daubechies

As *wavelets* de Daubechies foram propostas por Ingrid Daubechies. Existem várias transformações de Daubechies, porém a mais comum é a *wavelet* Daubechies 4 (DB4). As *wavelets* DB4 são definidas pela integral da equação (5.8), que define que a *wavelet* possui todos os momentos até ordem $N-1$ nulos [65].

$$\int_{-\infty}^{+\infty} x^l N\psi(x)dx = 0, \ l = 0, ..., N-1. \tag{5.8}$$

Para $N \gg 1$, onde N é o tamanho do sinal de entrada, $\psi(x)$ é a função *wavelet*-mãe (filtro) e x são os valores do sinal de entrada.

A *wavelet* DB4 possui quatro filtros de passa-baixa (g) e quatro filtros de passa-alta (h), obtidos a partir de g. Os filtros DB4 de passa-alta e passa-baixa estão definidos nas equações (5.9) a (5.16).

$$g_0 = \frac{1+\sqrt{3}}{4\sqrt{2}} \tag{5.9}$$

$$g_1 = \frac{3+\sqrt{3}}{4\sqrt{2}} \tag{5.10}$$

$$g_2 = \frac{3-\sqrt{3}}{4\sqrt{2}} \tag{5.11}$$

$$g_3 = \frac{1-\sqrt{3}}{4\sqrt{2}} \tag{5.12}$$

$$h_0 = g_3 \tag{5.13}$$

$$h_1 = -g_2 \tag{5.14}$$

$$h_2 = g_1 \tag{5.15}$$

$$h_3 = -g_0 \tag{5.16}$$

Assim como a *wavelet* de Haar, DB4 também pode ser representada por uma multiplicação de matrizes, conforme apresentado na equação (5.17).

$$\begin{pmatrix} g_0 & g_1 & g_2 & g_3 & \dots & 0 \\ & & \ddots & & & \\ 0 & 0 & \dots & g_0 & g_1 & g_2 & h_3 \\ h_0 & h_1 & h_2 & h_3 & \dots & 0 \\ & & \ddots & & & \\ 0 & 0 & \dots & h_0 & h_1 & h_2 & h_3 \end{pmatrix}_{N\times N} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{N-2} \\ x_{N-1} \\ x_N \end{pmatrix}_{N\times 1} = \begin{pmatrix} c_{j,1} \\ c_{j,2} \\ \vdots \\ c_{j,K} \\ d_{j,1} \\ d_{j,2} \\ \vdots \\ d_{j,K} \end{pmatrix}_{N\times 1} \tag{5.17}$$

No entanto há que se considerar que no caso de DB4, o último produto interno da multiplicação de F por $\mathbf{x}$ necessitará dos elementos de entrada x_{N+1} e x_{N+2} que são elementos inválidos. Este problema não ocorre para a transformada de Haar pois os produtos internos da multiplicação de matrizes são realizados com apenas dois elementos da entrada.

Para resolver essa falta de elementos, é possível a utilização de quatro alternativas [75]:

1. Considera-se o sinal de entrada como cíclico, isto é, ao chegar no último elemento de $\mathbf{X}$ o sinal de entrada é sucedido pelo início do mesmo sinal $\mathbf{X}$;
2. Consideram-se as extremidades espelhadas, com o último elemento duplicado no começo e o primeiro elemento de $\mathbf{X}$ duplicado no final;
3. Completa-se o vetor de entrada com valores nulos. Esta abordagem não é muito utilizada pois interfere na precisão da transformada;
4. Ortogonalização de *Gram-Schmidt*. Esta técnica calcula funções especiais de escala e wavelet que são utilizadas no início e no final do conjunto de dados.

5.2 Arquitetura Geral do Sistema

Esta seção apresenta a arquitetura geral do sistema *WaveDetect*. Como mostrado na Figura 10 o sistema *WaveDetect* possui a etapa de coleta dos dados SDN, por meio de um controlador SDN, possui também as fases de Caracterização e de Detecção de intrusos e ainda a fase de Mitigação. A cada intervalo de tempo (amostra do tráfego) essas fases Caracterização e Detecção são executadas, identificando se existe ou não um determinado

intruso dentro deste período do tráfego. Para as etapas de caracterização e detecção foram propostas duas abordagens, ambas utilizando DWT (*Discrete Wavelet Transform*) porém a primeira utiliza um intervalo de confiança baseado na teoria de distribuição normal e a segunda utiliza distância interquartil (IQR).

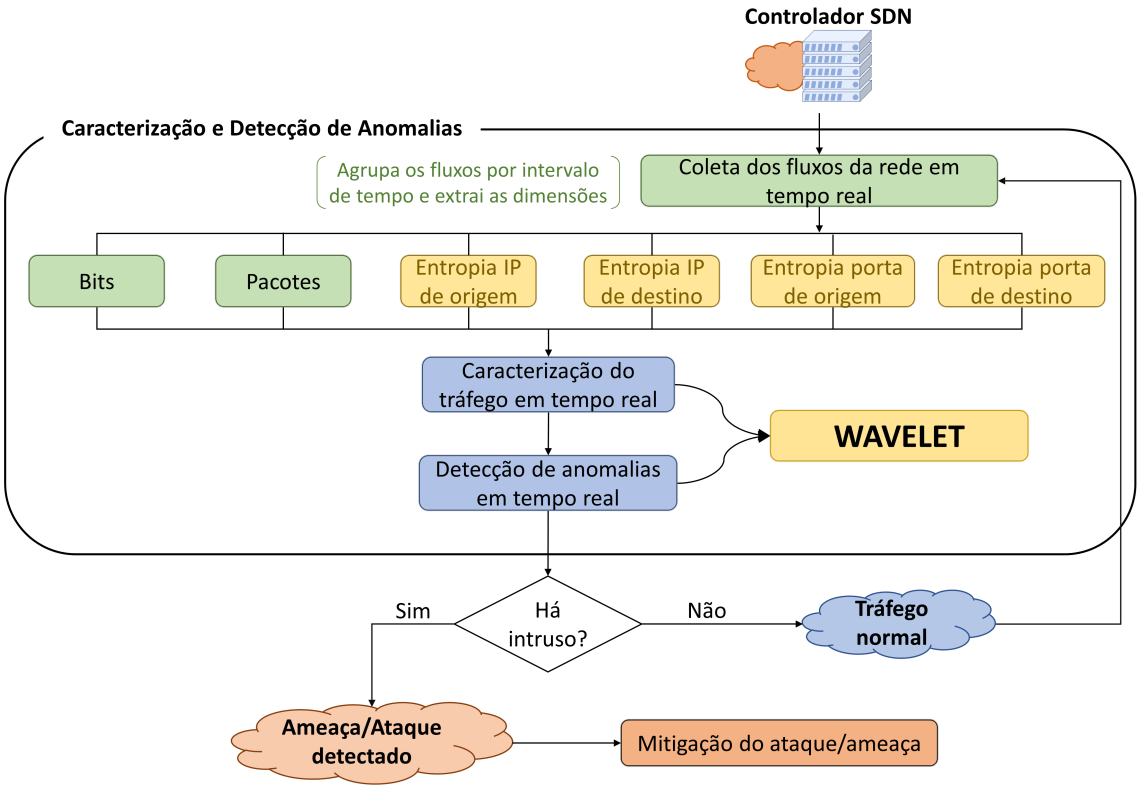


Figura 10 – Arquitetura geral do sistema.

O desenvolvimento das fases de caracterização, detecção e mitigação foi realizado por meio da linguagem R, que é uma linguagem de alto nível e também um ambiente utilizado na análise de dados e na geração de gráficos [76]. Esta linguagem é manipulada através do *software R Studio* [77]. Para a implementação das transformadas *wavelets*, foi utilizado o pacote "*wavelets*", disponível na linguagem R [78].

Cada uma das fases do sistema de detecção, bem como as fases de Caracterização e Detecção de cada uma das abordagens serão apresentadas de forma mais detalhada nas seções seguintes.

5.2.1 Aquisição dos Dados e Pré-processamento

A coleta dos dados é realizada por meio do protocolo *OpenFlow* [79], que provê uma comunicação unificada permitindo ao controlador acessar informações sobre tabelas de fluxos, configuração e estatísticas do tráfego. O controlador envia requisições, em intervalos de um, cinco ou trinta segundos, de todas as entradas de fluxos nos *switches*

por meio do protocolo *OpenFlow*. Os *switches* retornam a requisição enviando trechos do conteúdo das tabelas de fluxos.

A partir desta coleta obtêm-se os fluxos brutos do tráfego, ou seja, fluxos contendo informações como *bytes*, pacotes, endereço IP de origem, endereço IP de destino, porta de origem e porta de destino. Para a etapa de processamento, as quantias de *bytes* e pacotes são agrupadas para um intervalo de tempo. Já endereços IP e porta de origem e destino, por serem atributos nominais, necessitam de um processamento para que sejam representadas de forma numérica permitindo assim uma análise matemática e também a aplicação de métodos computacionais na sua análise. Para isto, é utilizada a entropia de Shannon que permite extrair informações da concentração e dispersão desses atributos. Para o cálculo da entropia, dado um atributo $Y = \{y_1, y_2, \ldots, y_N\}$ em que y_i representa o número de ocorrência da amostra i no intervalo de análise, a entropia H para Y é definida conforme apresentado em (5.18), em que $S = \sum_{i=1}^{N} y_i$ é o somatório de todas as ocorrências presentes no histograma.

$$H(Y) = -\sum_{i=1}^{N} \left(\frac{y_i}{S}\right) log_2 \left(\frac{y_i}{S}\right), \tag{5.18}$$

A entropia de Shannon torna-se essencial para a análise dos atributos nominais, visto que para a aplicação da DWT é necessário que os dados sejam grandezas numéricas.

5.2.2 Mecanismo de Janelas Deslizantes

Antes de explicar cada uma das abordagens utilizadas para caracterização e detecção é importante apresentar o conceito de janelas deslizantes. Uma janela deslizante consiste em um conjunto de valores de um determinado atributo da rede em um intervalo de tempo. Para as abordagens propostas foram utilizadas duas janelas deslizantes, uma nomeada W_f, que consiste em uma janela de tamanho M com as últimas amostras do tráfego classificadas como normais. A segunda janela, W_d consiste em uma janela de mesmo tamanho contendo as últimas $M-1$ amostras do tráfego e a amostra do intervalo t, que é a amostra que será analisada. A Figura 11 apresenta as duas janelas visualmente, e (5.19)–(5.21) trazem uma explicação matemática de W_f e W_d para as amostras de $\mathbf{X}$.

$$|W_f| = |W_d| = M \tag{5.19}$$

$$W_f = \left\{x_{(t-1)-M}, x_{(t-M)}, \ldots, x_{t-1}\right\} \tag{5.20}$$

$$W_d = \left\{x_{(t-M)}, x_{(t-M)+1}, \ldots, x_t\right\} \tag{5.21}$$

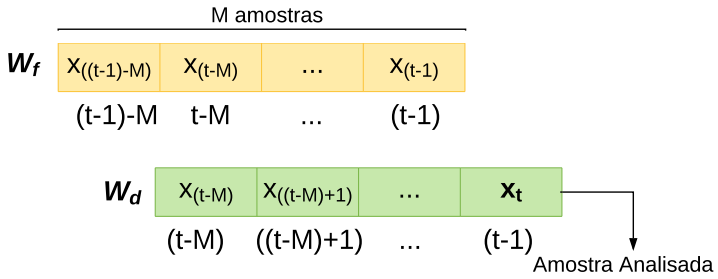


Figura 11 – Janelas Deslizantes W_f e W_d.

onde M é o tamanho da janela deslizante e $x_{(\cdots,(t-1),t,\cdots)}$ é uma amostra de uma determinada dimensão do tráfego.

5.2.3 Primeira Abordagem: WaveDetect–ND

A primeira abordagem, nomeada WaveDetect–ND, utiliza uma DWT unidimensional (1D) associada com um intervalo de confiança obtido com base em propriedades da teoria de distribuições normais. Esta abordagem possui duas fases, uma de caracterização do tráfego e outra de detecção de intrusos. A Figura 12 apresenta de maneira mais detalhada essas duas fases.

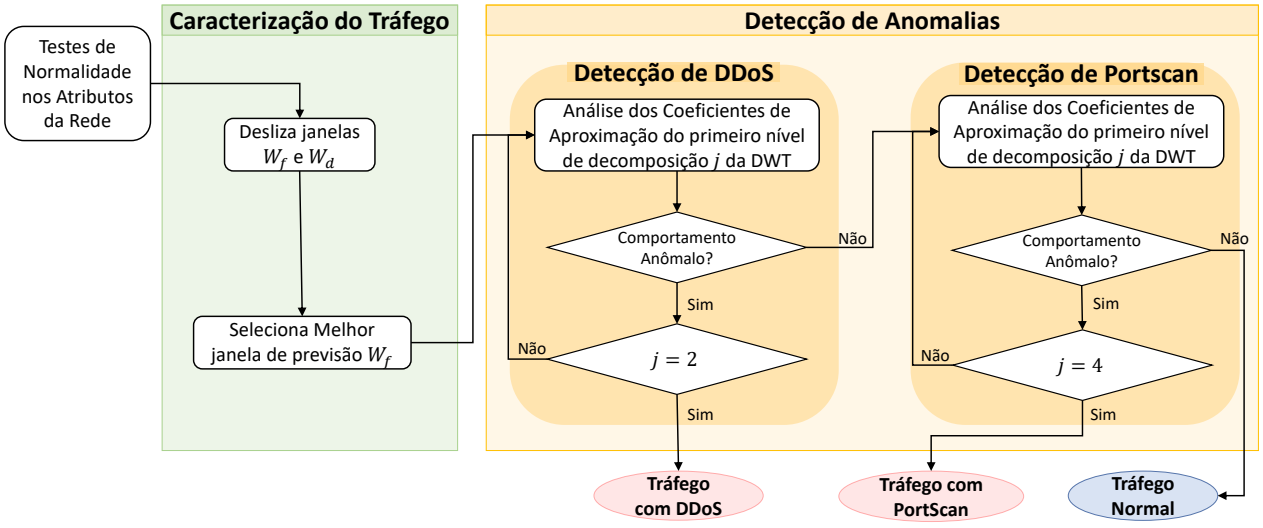


Figura 12 – Fluxograma com as etapas de caracterização e detecção.

Cada uma das fases descritas na Figura 12 será descrita a seguir.

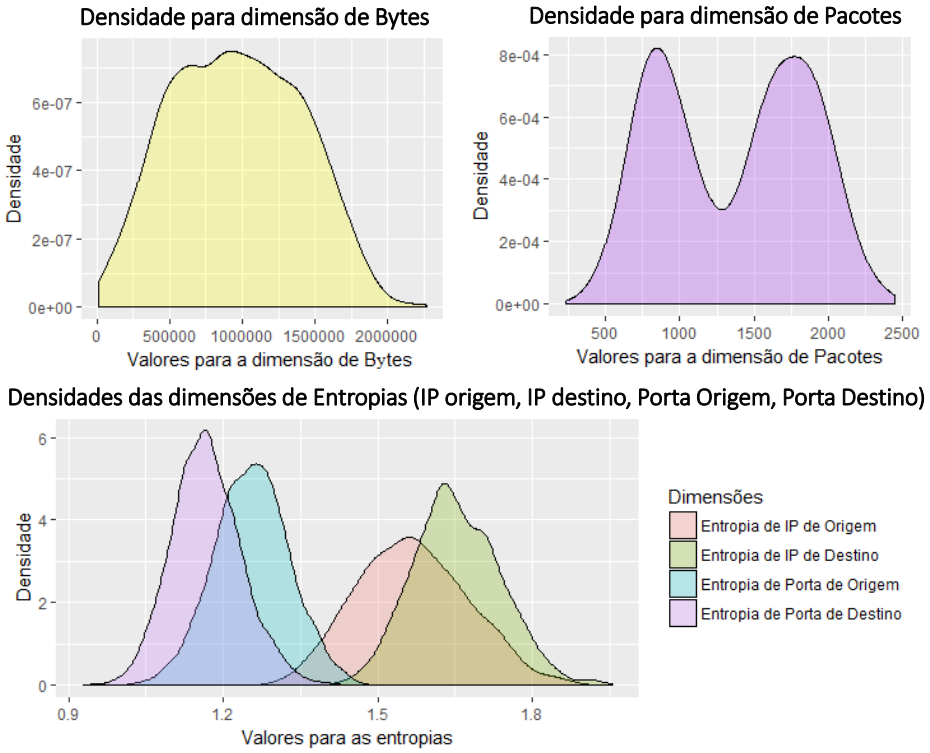


Figura 13 – Densidade dos atributos do tráfego.

5.2.3.1 Primeira Fase: Caracterização do Tráfego

A primeira fase, nomeada como fase de Caracterização se subdivide em duas etapas. No entanto, antes da primeira fase uma etapa inicial foi realizada. Esta etapa ("Teste de Normalidade nos Dados") realiza um teste de normalidade para verificar se as seis variáveis de entrada possuem ou não distribuição normal. Foram realizadas duas análises, uma visual, por meio de histogramas e a segunda consistindo no teste de Shapiro-Wilk, que é um teste de hipótese utilizado para verificar se o comportamento de uma variável aleatória, sob H_0, segue uma distribuição normal, para conjuntos de dados com tamanhos de até 5 mil amostras [80]. No teste de Shapiro-Wilk, com nível de significância $\alpha = 5\%$, se o p-valor do teste for maior que α então a variável possui distribuição normal, caso contrário a variável não provém de uma distribuição normal.

Os resultados obtidos pelos testes de Shapiro-Wilk para dois dias de dados demonstraram que apenas variáveis referentes às dimensões de entropia de IP de origem e porta de origem possuem distribuição similar à normal, como demonstrado na Figura 13 e também nos resultados de p-valor dos testes de Shapiro-Wilk, presentes na Tabela 3.

Apesar de apenas duas das seis dimensões apresentarem distribuição similar à normal, a ideia da abordagem é apenas partir da teoria de distribuição normal e se ajustar à nossa proposta por meio de testes para os parâmetros, os quais serão explicados na seção 6.3.1.

Tabela 3 – Resultados dos testes de Shapiro-Wilk para dois dias de dados dos controladores POX.

Dimensão	p-valor (Dia 1)	p-valor (Dia 2)
Bytes	$< 0,0001$	$< 0,0001$
Pacotes	$< 0,0001$	$< 0,0001$
Entropia IP Origem	$< 0,0001$	$0,683$
Entropia IP Destino	$< 0,0001$	$< 0,0001$
entropia Porta Origem	$0,782$	$0,02$
Entropia Porta Destino	$< 0,0001$	$0,0087$

A primeira etapa da caracterização, nomeada "Desliza Janela de Previsão", trata de deslizar W_f. W_f possui quatro tamanhos com $M = 32$, 64, 128 ou 256.

A janela W_f carrega o tráfego considerado normal, ou seja, possui apenas amostras livres de anomalias, e é a janela que representa o padrão esperado do tráfego. A cada análise de uma amostra x_t, W_f é deslizada a fim de englobar a amostra posterior à última amostra de W_f. Isso ocorre apenas se esta amostra tiver sido classificada como normal. Caso isso aconteça, o ponto mais antigo de W_f é descartado, e o novo ponto (x_t) adicionado a W_f. Este processo é realizado para quatro janelas W_f com tamanhos M especificados anteriormente.

Como a solução utiliza um histórico do tráfego para gerar ambas janelas, ao inicializar o processo de caracterização e detecção, é necessária uma base de dados históricos da rede com uma quantidade de amostras previamente classificadas como normais, para uma caracterização livre de interferências. Esse histórico compõe a primeira janela deslizante do sistema de detecção.

A segunda etapa da primeira fase tem o objetivo de selecionar a melhor janela W_f entre os quatro possíveis tamanhos, para cada uma das seis dimensões. O teste de Shapiro-Wilk é utilizado e a janela que apresentar o p-valor mais baixo será a janela escolhida, isto porque o p-valor indica a probabilidade de acerto em não rejeitar H_0.

5.2.3.2 Segunda Fase: Detecção de Anomalias

Esta fase se subdivide em três partes. Na primeira parte, uma DWT de uma dimensão (1D) é realizada em W_f e W_d para cada uma das seis dimensões. A DWT foi executada utilizando um, dois, três ou quatro níveis de decomposição. A DWT com dois níveis de decomposição foi a abordagem escolhida para DDoS e quatro níveis para *Portscan* por proporcionar melhores resultados (vide seção 6.3.1).

A segunda e terceira partes da segunda etapa realizam a detecção de DDoS e *Portscan*, respectivamente. A segunda e a terceira partes possuem quatro e oito etapas. A diferença entre elas está na quantidade de níveis que serão utilizados e também uma mudança no valor de ϵ, que será explicado mais adiante. Portanto, aqui será explanado o

processo de detecção para DDoS, o qual será o mesmo para a detecção de *Portscan*.

A detecção de DDoS se divide em quatro passos. O primeiro passo é a realização de uma análise dos coeficientes de aproximação do 1º nível de decomposição da DWT. Primeiramente, é extraída a média (μ) e o desvio padrão (σ) dos coeficientes de aproximação ($c_{1,k}$) de W_f. A partir da média e da variância coletada, é gerado um intervalo de confiança desses coeficientes, baseando-se na teoria σ que afirma que $99,7\%$ dos dados com distribuição normal estão situados no intervalo de $\epsilon = 3$, isto é, três vezes o desvio padrão para mais ou para menos com relação a média. Essa regra pode ser melhor compreendida por meio de (5.22) [81].

$$\Pr\left(\mu - 3\sigma \leq X \leq \mu - 3\sigma\right) \approx 0,9973 \tag{5.22}$$

onde a variável aleatória X possui distribuição normal, com $X \sim \mathcal{N}\left(\mu; \sigma^2\right)$.

O segundo passo compara a amostra t, ou seja, o último coeficiente de aproximação do primeiro nível da DWT ($c_{1,M/2}$) de W_d com o intervalo obtido no passo anterior. Se a amostra estiver dentro do intervalo normal o terceiro e o quarto passos da detecção DDoS não são executados e o tráfego é classificado como livre de DDoS na amostra t. Se estiver fora do intervalo, os passos três e quatro serão executados. Essa comparação está explanada em (5.23).

$$\left(\mu_{W_f}\left[c_{(j,k)}\right] - \epsilon \cdot \sigma_{W_f}\left[c_{(j,k)}\right]\right) \leq \left(W_d\left(c_{j,K}\right)\right) \leq \left(\mu_{W_f}\left[c_{(j,k)}\right] + \epsilon \cdot \sigma_{W_f}\left[c_{(j,k)}\right]\right) \tag{5.23}$$

onde $j \in \{1, 2, 3 \text{ ou } 4\}$ e $\epsilon = 3$ ou $0,5$.

O terceiro e quarto passos executam o mesmo processo dos passos um e dois, mas agora para o 2º nível de decomposição da DWT. Portanto, primeiramente calcula-se μ e σ para os coeficientes de aproximação ($c_{2,k}$) de W_f. Com essas medidas estatísticas, o intervalo de confiança dos coeficientes é definido (três vezes o desvio padrão com relação a média). Por fim, o último valor dos coeficientes de aproximação de W_d é comparado com este intervalo, se estiver dentro do intervalo normal o ponto é classificado como livre de DDoS, caso contrário este ponto é classificado como sob ataque DDoS.

Caso o tráfego tenha sido classificado como livre de DDoS, então o processo de detecção de *Portscan* é iniciado. Este processo se assemelha ao executado para a detecção de DDoS, ou seja, primeiro calcula-se o intervalo de confiança para os coeficientes de aproximação de W_f do primeiro nível da DWT. Feito isto, compara-se o último coeficiente de aproximação de W_d com este intervalo, se estiver dentro do intervalo, o ponto é classificado como normal, caso contrário esse processo é reexecutado para o segundo nível da DWT. A diferença na detecção de *Portscan* para detecção DDoS está na quantidade

de níveis de decomposição DWT utilizada, onde o processo de detecção se repete não mais até o segundo nível de decomposição, mas sim até o quarto nível da DWT. Então avaliando o quarto nível da DWT, se o último coeficiente de W_d estiver dentro do intervalo normal será classificado como normal, e caso não, será classificado como sob ataque *Portscan*. Outra diferença entre a detecção de *Portscan* e a detecção de DDoS está no valor do ϵ, que para a detecção de DDoS possui $\epsilon = 3$ e para a detecção de *Portscan* $\epsilon = 0,5$. O valor de três vezes o desvio padrão, obtido por meio da teoria σ se mostrou eficiente para detecção de DDoS, porém apresentou resultados ruins para a detecção de *Portscan*. Isto se deve ao fato de que DDoS constitui-se como um ataque do tipo *flooding*, isto é, um ataque que ocasiona um aumento expressivo do tráfego, alterando as taxas de *bytes* e pacotes e também alterando as entropias de maneira mais expressiva, já o ataque *Portscan* possui um comportamento mais sutil. Com isso, para a detecção de *Portscan* foi definido $\epsilon = 0,5$. Este valor foi obtido por meio de testes que serão apresentados na subseção 6.3.1. A etapa de Detecção também está descrita no Algoritmo 1.

Algorithm 1 Algoritmo da etapa de detecção para WaveDetect–ND

```
Calcula fW_f e fW_d

if (intruso de interesse for DDoS) then
    ε = 3
elseε = 0,5
for (j de 1 até 2) do                                   ▷ Detecção de DDoS
    Calcula μ e σ de c_(j,k) para W_f
    if ((c_j, N/2) de W_d é maior que μ − ε ∗ σ e menor que μ + ε ∗ σ) then
        Classifica tráfego como Normal
    else
        Classifica tráfego como DDoS
for (j de 1 até 4) do                                  ▷ Detecção de Portscan
    Calcula μ e σ de c_(j,k) para W_f
    if ((c_j, N/2) de W_d é maior que μ − ε ∗ σ e menor que μ + ε ∗ σ) then
        Classifica tráfego como Normal
    else
        Classifica tráfego como Portscan
```

5.2.4 Segunda Abordagem: WaveDetect–IQR

A segunda abordagem também adota DWT, mas ao invés de utilizar um intervalo de confiança obtido com base em propriedades de distribuição normal, esta abordagem está associada a uma distância interquartil (IQR), por isso foi nomeada como WaveDetect–IQR.

Assim como a primeira abordagem, WaveDetect–IQR divide-se em duas fases: caracterização do tráfego e detecção de anomalias, conforme apresentado na Figura 14.

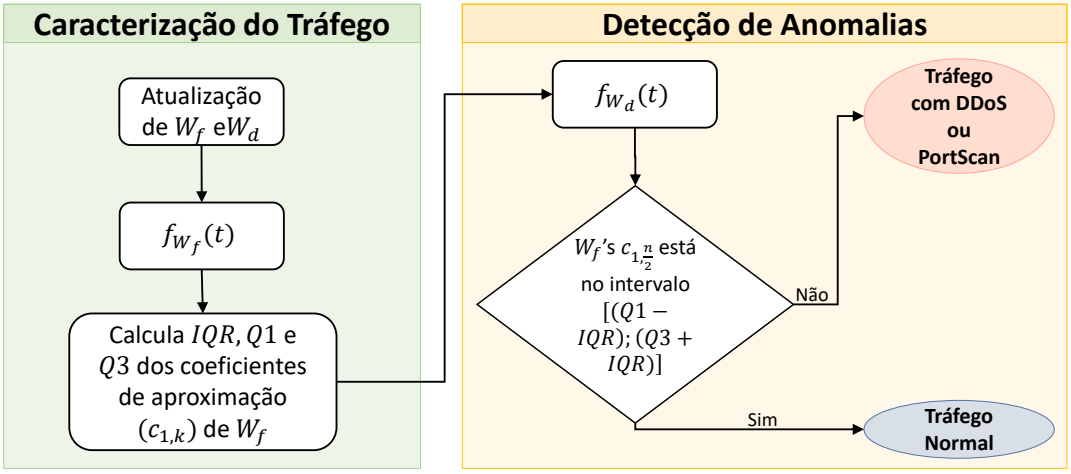


Figura 14 – Arquitetura WaveDetect–IQR.

A seguir, ambas as fases serão apresentadas de maneira mais detalhada.

5.2.4.1 Primeira Fase: Caracterização do Tráfego

A primeira fase é responsável primeiramente pelo processo de seleção de W_f e W_d. W_f e W_d possuem tamanho $M = 1024$. Este tamanho foi selecionado por meio de testes que comprovaram que a janela com tamanho 1024 forneceu melhor detecção com baixa taxa de falsos alarmes. Os resultados destes testes serão melhor discutidos na subseção 6.3.2. Assim como na primeira abordagem, a janela W_f será deslizada para a amostra seguinte toda vez que esta tiver sido classifica como normal. Desta forma, a nova janela eliminará o ponto mais antigo e englobará o novo ponto.

Tal qual explicado em WaveDetect–ND, WaveDetect–IQR também não necessita de uma fase de treinamento, no entanto necessita de um ponto de partida, que consiste em um conjunto de 1024 amostras anteriores ao período analisado com comportamento normal, que compõem as primeiras janelas deslizantes. Após a seleção de janelas, é iniciado o processo de detecção das anomalias.

Realizada a seleção da janela deslizante W_f, é então realizada uma DWT nesta janela, e a partir dos coeficientes de aproximação do primeiro nível é realizada a extração de um intervalo considerado normal (DSNSF) [82], [83], [84] baseando-se na distância interquartil (IQR). O IQR é obtido por meio de (5.24). Essa teoria foi proposta por John W. Tukey [85] baseando-se nos padrões normais de pressões diastólica do sangue.

$$IQR = Q3 - Q1 \tag{5.24}$$

Este valor será utilizado para a geração do intervalo considerado normal para os coeficientes de aproximação. Este intervalo de confiança será obtido por meio de (5.25) e

será nomeado como o DSNSF do tráfego em análise.

$$(Q1 - (\text{IQR } \times 1.5)) \leq (W_d\,(c_{j,K})) \leq (Q3 + (\text{IQR } \times 1.5)) \tag{5.25}$$

onde $(W_d\,(c_{j,K}))$ é o dado analisado com $K = M/(2^j)$; $Q1$ é o primeiro quartil e $Q3$ é o terceiro quartil dos coeficientes de aproximação de W_f.

5.2.4.2 Segunda Fase: Detecção de Anomalias

Nesta fase é realizada a detecção de anomalias do tipo DDoS e *Portscan*. Para isso, primeiramente uma DWT 1D é aplicada em W_d. Feito isto, o próximo passo consiste em detectar possíveis desvios (anomalias) em cada uma das seis dimensões analisadas. Para cada dimensão, o último coeficiente de aproximação de W_d é comparado com o intervalo encontrado em (5.25), se o coeficiente estiver dentro do intervalo de confiança o ponto é classificado como normal para uma dada dimensão, caso contrário, é classificado como anômalo. No caso do tráfego anômalo, também é verificado se o coeficiente ficou abaixo ou acima do intervalo normal, e esta informação é então armazenada para auxiliar na próxima etapa.

A última etapa consiste na classificação do tráfego considerado anômalo, com análise das seis dimensões. Para que seja detectada uma anomalia do tipo DDoS é necessário que tenha sido detectada uma concentração nas entropias de IP de destino e porta de destino e uma dispersão na dimensões de entropia de IP de origem. Já para uma anomalia do tipo *Portscan* ser detectada, é necessário que exista uma concentração nas dimensões de entropia de IP de origem e de destino, e uma dispersão na dimensão de entropia de porta de destino. Por meio dessas informações é possível classificar o tráfego entre essas duas anomalias. Caso nenhum dos dois comportamentos sejam detectados, o tráfego é classificado como normal. O Algoritmo 2 detalha os passos das fases de caracterização e detecção de anomalias.

5.2.5 Mitigação

O processo de mitigação consiste em dois mecanismos de descarte de pacotes [86]. Um para ataques DDoS, e outro para ataques *Portscan*.

Independente do tipo de ataque (DDoS ou *Portscan*) o primeiro passo da mitigação ao se deparar com uma amostra com anomalia é selecionar os fluxos considerados suspeitos. Os fluxos considerados como suspeitos consistem em todos os fluxos que se destinam ao IP que mais recebe fluxos no intervalo da amostra em questão.

Posteriormente, se a detecção foi de um ataque DDoS, será realizado um descarte dos fluxos que possuam o IP de origem mais frequente e simultaneamente a porta de destino mais frequente nos fluxos suspeitos.

Algorithm 2 Algoritmo que descreve a detecção de anomalias para WaveDetect–IQR.

Calcula fW_f e fW_d

Calcula $Q1$ e $Q3$ de $c_{(j,k)}$ para W_f
if (($c_1, N/2$) de W_d é maior que ($Q1 - \text{IQR} \times 1.5$) e menor que ($Q3 + \text{IQR} \times 1.5$)) **then**
    Classifica tráfego como Normal
else
    Classifica tráfego como Anômalo
if (Tráfego possui comportamento DDoS) **then**
    Classifica tráfego anômalo como DDoS
else
    if (Tráfego possui comportamento *Portscan*) **then**
        Classifica tráfego anômalo como *Portscan*
    else
        Classifica tráfego como Normal

Algorithm 3 Processo de Mitigação.

function MITIGAÇÃO(dados do tráfego)
    if Existe anomalia DDoS **then**
        Seleciona os fluxos que se destinam ao IP que mais recebe fluxos;
        Seleciona a Porta de destino e o IP de origem que mais aparece nesses fluxos;
        Elimina os fluxos que contiverem esse endereço IP de origem e essa Porta de destino;
    if Existe anomalia *Portscan* **then**
        Seleciona os fluxos que se destinam ao IP que mais recebe fluxos;
        Contabiliza a quantidade de portas de destino que cada IP de origem presente nesses fluxos acessa;
        O IP de origem que possuir maior diversidade em suas Portas de destino será considerado o IP atacante, logo, seus fluxos serão excluídos;

Para um ataque *Portscan*, é considerada a diversidade das portas de destino para cada IP de origem, isto é, para cada origem dos fluxos suspeitos, verifica-se a quantidade de portas a que esse IP se destina. Os fluxos do IP de origem que possuírem a maior variedade nas portas de destino serão eliminados.

O Algoritmo 3 apresenta a sequência do processo de mitigação para DDoS e *Portscan*.

5.3 Considerações Sobre o Capítulo

Conforme tudo o que já foi explanado ao longo deste trabalho, o desenvolvimento de mecanismos para auxiliar na detecção e minimização dos ataques ao longo do tráfego da rede tornou-se indispensável, tanto quanto ferramentas para auxiliar a configuração e gerenciamento da rede. Pensando nisto este capítulo apresenta uma proposta para detec-

ção e mitigação de ataques utilizando DWT. Com relação à detecção, duas abordagens se fazem presente, ambas utilizando DWT, porém uma utiliza teorias de distribuição normal e a outra utiliza distância interquartil. Para a mitigação, foram implementadas algumas politicas, específicas para cada um dos dois tipos de ataques que são foco deste trabalho, DDoS e *Portscan*. O próximo capítulo apresentará os resultados da análise do sistema em cada um dos diferentes cenários utilizados.

6 RESULTADOS

Este capítulo apresenta os resultados para os testes realizados com diferentes cenários para analisar o desempenho do sistema proposto para a detecção e mitigação dos ataques do tipo DDoS e *Portscan*.

6.1 Cenários

Para avaliar o sistema e comparar as abordagens, foram emulados três cenários em uma rede SDN utilizando o Mininet [87], que cria um ambiente virtual de rede, com *hosts*, *switches* e controladores. Para proporcionar cenários que se aproximem o máximo possível de um ambiente SDN real, foi utilizada uma ferramenta chamada *Scapy* [88] para injetar o tráfego na rede emulada.

Dentro da rede SDN, como já citado, é necessária a existência de um controlador SDN. Os controladores utilizados nos três cenários foram o POX e o Floodlight. O controlador POX foi utilizado no primeiro cenário. Este controlador Foi desenvolvido com base no controlador NOX, que é o controlador original SDN. O POX consiste em uma plataforma desenvolvida em Python projetada para o desenvolvimento e prototipagem de aplicações SDN, com interface mais amigável se comparado ao controlador NOX. O Floodlight foi utilizado nos outros dois cenários. Este controlador consiste em um controlador SDN gratuito desenvolvido em Java pela BigSwitch, com foco em redes comerciais [89]. Este controlador oferece suporte a uma ampla variedade de switches OpenFlow virtuais e físicos e pode lidar com redes mistas OpenFlow e não OpenFlow. O Floodlight ainda possui instalação e utilização relativamente simples [90].

Com o propósito de testar a efetividade do sistema, anomalias sintéticas foram geradas a fim de simular o comportamento de ataques do tipo DDoS e *Portscan*. Esse tráfego foi gerado por meio do *hping3*, um montador e analisador de pacotes TCP/IP usado principalmente na área de segurança de redes. O primeiro tipo de anomalia gerado foi DDoS, que altera quatro das seis dimensões utilizadas, que são: pacotes, entropia de porta de destino e endereço IP de origem e destino.

Para emular os ataques DDoS, o *hping3* enviou pacotes SYN para um *host* a partir de um conjunto aleatório e continuamente alterado de portas e endereços IP de origem. Estes pacotes SYN consistem em pacotes de sincronização, os quais são enviados seguidamente com o objetivo de sobrecarregar o alvo em questão. A análise geral incluiu a mistura de diversos tipos de tráfego legítimo e diferentes parâmetros de tráfego de ataque.

O segundo tipo de ataque injetado nos três cenários foi *Portscan*. Esse ataque altera as dimensões de entropia de porta de destino e endereço IP de origem e destino.

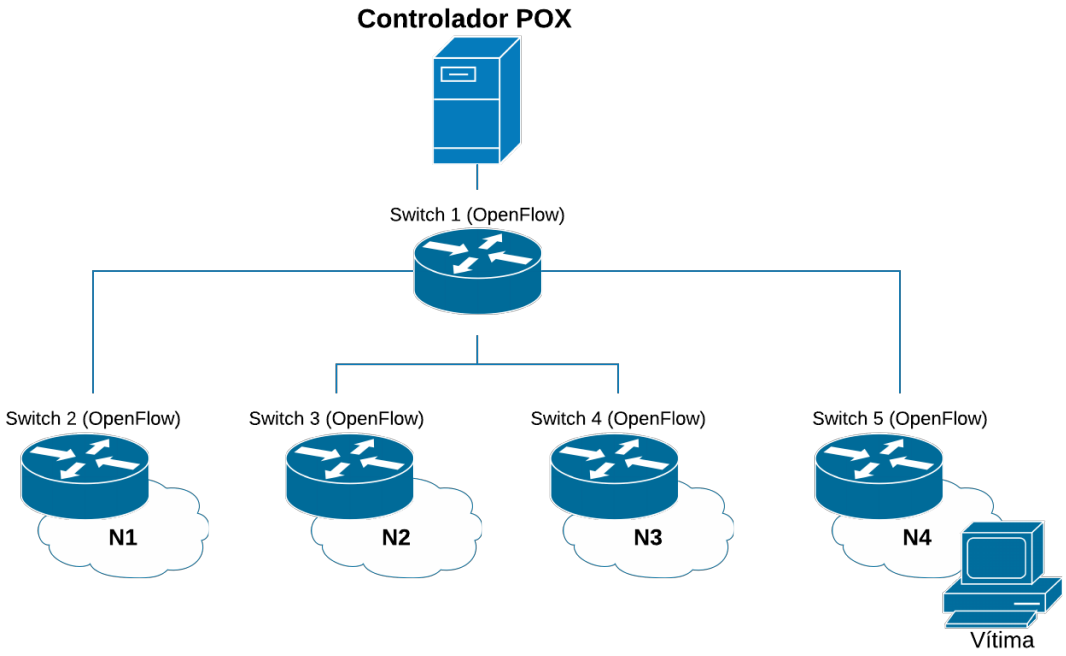


Figura 15 – Topologia emulada no Mininet para o primeiro cenário (CNR1).

6.1.1 Primeiro Cenário

O primeiro cenário (CNR1) emulado possui estrutura em árvore com profundidade de dois níveis. Um *switch* raiz se conecta a outros quatro *switches* e cada uma das sub-redes (N1, N2, N3 e N4) possui vinte *hosts*, conforme apresentado na Figura 15. Este cenário possui um tempo de coleta de fluxos de 30 segundos. Para este cenário, o controlador utilizado foi o POX.

Para testar o sistema, tráfego anômalo foi injetado no tráfego da rede de dois dias. O primeiro tipo de ataque injetado consiste em ataque DDoS. Este ataque foi direcionado para um *host* na sub rede N4 cujo IP é 10.0.0.78. As anomalias foram inseridas nos intervalos entre 14h e 15h e entre 17h e 18h do primeiro dia.

Já os ataques *Portscan* foram inseridos entre os intervalos de 15h a 15h30 e das 17h às 17h30 do segundo dia.

6.1.2 Segundo Cenário

O segundo cenário (CNR2) emulado também possui estrutura em árvore com profundidade de dois níveis. Um *switch* raiz se conecta a outros três *switches* e cada uma das sub-redes (N1, N2 e N3) possui vinte *hosts*, conforme apresentado na Figura 16. Este cenário possui tempo de coleta de 5 segundos. Para este cenário, um controlador diferente foi utilizado, o *Floodlight*.

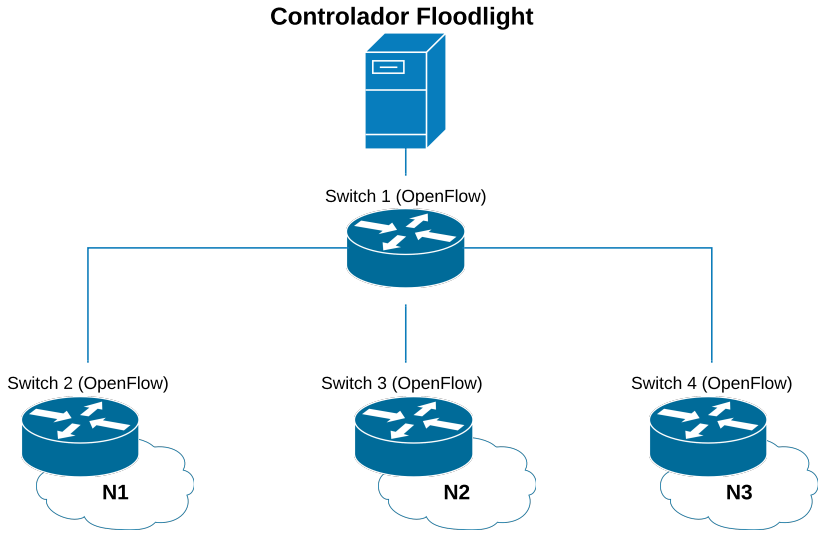


Figura 16 – Topologia emulada no Mininet para o segundo cenário (CNR2).

Para o teste do sistema foram utilizados dois ataques DDoS e dois ataques *Portscan* em um dia de tráfego. Foram injetados dois ataques DDoS, um no intervalo entre 6h e 6h59m20s, destinando o ataque para o *host* de IP 10.0.0.10, e o segundo ataque injetado no intervalo entre 13h e 13h59m10s destinando-se ao *host* 10.0.0.50. O *range* de IPs atacantes do primeiro ataque DDoS situa-se entre 10.0.0.31 e 10.0.0.39, e do segundo ataque entre 10.0.0.11 e 10.0.0.25.

Os dois ataques *Portscan* foram inseridos entre os intervalos de 4h a 4h59m30s para o IP 10.0.0.35 e portas 1-14988, com IP atacante 10.0.0.55. O segundo ataque situa-se entre 10h e 10h39m50s destinado ao IP 10.0.0.25 e as portas 1-19999, com IP atacante 10.0.0.60.

6.1.3 Terceiro Cenário

O último cenário (CNR3) também utiliza o controlador Floodlight. Possui topologia estrela, com um *switch* central conectado a seis *switches* (N1, N2, N3, N4, N5 e N6) que por sua vez conectam-se com vinte *hosts* cada, totalizando 120 *hosts*, conforme apresentado na Figura 17. Este cenário possui tempo de coleta de um segundo.

Neste cenário foram injetados dois ataques, um DDoS e outro *Portscan* em um dia de tráfego. O ataque DDoS foi inserido no intervalo entre 10h32 e 11h45 destinando-se ao IP 10.0.0.50 com IPs atacantes entre 10.0.0.11 e 10.0.0.25. O ataque *Portscan* foi inserido no intervalo de 16h45 à 17h45 a partir do *host* de IP 10.0.0.20 para o *host* de IP 10.0.0.82 e portas entre 1 e 30000.

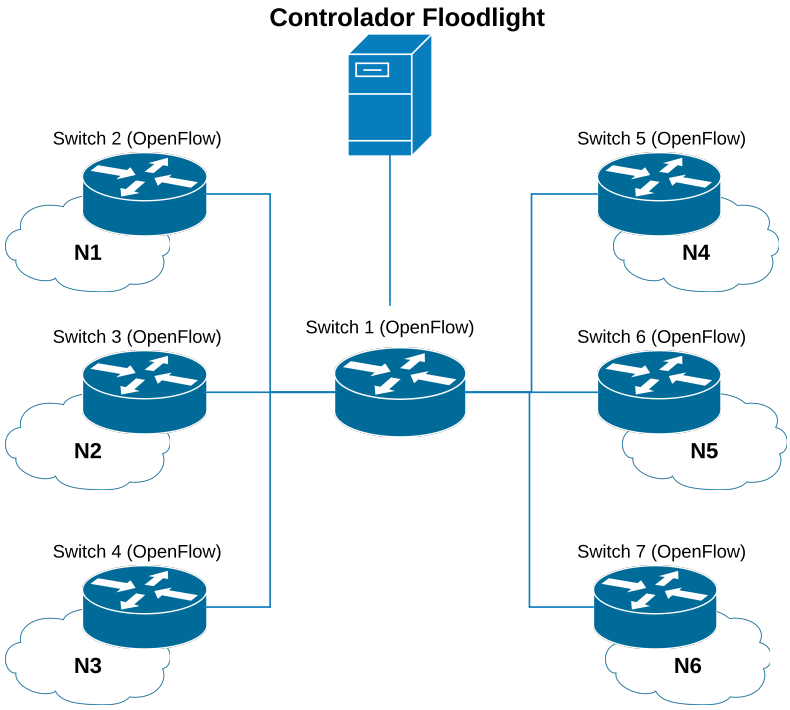


Figura 17 – Topologia emulada no Mininet para o terceiro cenário (CNR3).

6.2 Métricas Utilizadas

Para avaliar o desempenho das abordagens propostas, bem como definir os parâmetros de cada uma delas, algumas métricas foram utilizadas, que são [91]:

- Acurácia: indica o percentual total de acertos;
- Precisão: apresenta o percentual de pontos corretamente classificados como anômalos entre o total de pontos classificados como anômalos;
- Taxa de falso-negativo (*FN-rate* ou Taxa de FN): indica o percentual de pontos com anomalias que foram classificados como normais;
- Taxa de falso-positivo (*FP-rate* ou Taxa de FP): apresenta o percentual de pontos normais que foram indicados como anômalos (alarme falso);
- *Recall* (sensibilidade, Verdadeiro-positivo ou Taxa de TP): é o percentual de acerto para os intervalos com anomalias;
- *F-measure*: pode ser interpretada como a média harmônica entre a Precisão e a Taxa de verdadeiro positivo (*recall*). Consiste em uma métrica que apresenta a acurácia de um determinado teste, considerando a taxa de verdadeiro-positivo e a precisão para tal;

- *Area Under Curve* (AUC): Esta métrica descreve a área sob a curva ROC, que representa o *trade-off* entre as métricas de verdadeiro-positivo e falso-positivo.

Matematicamente, estas métricas estão descritas nas equações (6.1)–(6.6), onde TP = quantidade de amostras de verdadeiro-positivo, TN = quantidade de amostras de verdadeiro-negativo, FP = quantidade de amostras de falso-positivo e FN = quantidade de amostras de falso-negativo.

$$\text{Acurácia} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{6.1}$$

$$\text{Precisão} = \frac{TP}{(TP + FP)} \tag{6.2}$$

$$\text{Taxa de FN} = \frac{FN}{(FN + TP)} \tag{6.3}$$

$$\text{Taxa de FP} = \frac{FP}{(FP + TN)} \tag{6.4}$$

$$\textit{Recall} = \frac{TP}{(TP + FN)} \tag{6.5}$$

$$\textit{F-measure} = 2 \times \frac{(\text{Precisão} \times \textit{recall})}{(\text{Precisão} + \textit{recall})} \tag{6.6}$$

6.3 Análise e Escolha dos Parâmetros das Abordagens

Ambas abordagens – WaveDetect–ND e WaveDetect–IQR – necessitam do ajuste de alguns parâmetros. A primeira abordagem necessita do ajuste do valor de ϵ e também da quantidade de níveis. Esta abordagem ainda realiza a seleção do tamanho da janela deslizante. Já a segunda abordagem necessita do ajuste do parâmetro que define a quantidade de níveis da DWT, e também o tamanho da janela deslizante. Os resultados dos testes e análise destes para a definição destes parâmetros será apresentada a seguir.

6.3.1 Parâmetros da Primeira Abordagem

Para a primeira abordagem primeiramente foram definidos os tamanhos para as janelas deslizantes. Nesta abordagem, a cada ponto analisado, um tamanho de janela é utilizado para cada dimensão. Este tamanho é definido por meio do teste estatístico de Shapiro-Wilk, selecionando o tamanho da janela entre quatro tamanhos pré-definidos, sendo: $32, 64, 128$ e 256. Para os outros dois parâmetros (níveis da DWT e valor de ϵ)

foram realizados testes com uma variação de níveis da DWT entre um e quatro, e valores de ϵ variando de 0 a $3,5$ para a detecção de *Portscan*.

Para a definição dos níveis de decomposição da DWT, a abordagem foi executada para quatro níveis diferentes da DWT. Avaliando as métricas para cada um dos níveis da DWT, foram obtidos os resultados apresentados no radar da Figura 18 e da Tabela 4.

Conforme apresentado na Figura 18 e na Tabela 4, a DWT com apenas um nível de decomposição traz um baixo resultado nas taxas de FN, porém uma taxa de FP mais alta do que os outros níveis. A precisão resulta em valores baixos e o valor da *F-measure* contém apenas um resultado aceitável (em torno de 75%). A DWT com três e quatro níveis de decomposição mostra uma ligeira melhoria na taxa de FP, mas, por outro lado, a taxa de FN sofre um aumento significativo, mostrando perda na detecção de anomalias, uma vez que esses dois níveis só alcançam dois terços dos pontos anômalos. Usando um nível de decomposição, a detecção é alta, mas muitos alarmes falsos são gerados, o que não é um recurso desejável em um sistema de detecção de anomalias reais já que os falsos alarmes geram uma perda de tempo e recursos. Por outro lado, usando três e quatro níveis, os alarmes falsos são quase nulos, no entanto metade dos ataques DDoS não são detectados, o que também não é desejável, pois um sistema que detecta apenas dois terços das anomalias não é um sistema confiável. Portanto, dois níveis foi a melhor quantidade de níveis de decomposição na DWT para DDoS e quatro níveis para *Portscan*, uma vez que a detecção de anomalias está próxima de 100% ($99,62\%$), e a taxa de alarmes falsos está próxima de 0% ($0,296\%$).

Tabela 4 – Métricas para os testes utilizando diferentes níveis de decomposição da DWT.

Quantidade de níveis	**Acurácia**	**Precisão**	**FN-rate**	**FP-rate**	***recall***	***F-measure***
1 nível	$0,965$	$0,642$	$0,00556$	$0,037$	$0,994$	$0,779$
2 níveis	$0,986$	$0,873$	$0,35$	$0,0063$	$0,65$	$0,745$
3 níveis	$0,972$	$0,873$	$0,35$	$0,0063$	$0,65$	$0,745$
4 níveis	$0,975$	$0,936$	$0,35$	$0,0029$	$0,65$	$0,767$
****2 e 4 níveis****	$0,9962$	$0,957$	$0,0167$	$0,0029$	$0,983$	$0,969$

Com relação ao valor de ϵ para detecção de *Portscan*, as métricas indicam que valores abaixo de $0,5$ apresentaram altas taxas de detecção, verificados pelas métricas precisão e *recall*, porém apresentaram alta taxa de falso-positivo indicando que o modelo foi capaz de detectar os intervalos anômalos mas também se tornou sensível à minúsculas alterações no tráfego, gerando muitos alarmes falsos, e desta forma tornando a detecção pouco confiável e eficiente. Para valores de ϵ maiores que $0,5$ o modelo proposto apresentou quase nenhum ou nenhum alarme falso, porém também não foi capaz de detectar praticamente nenhum intervalo com *Portscan*. Isso indica que para valores de ϵ maiores

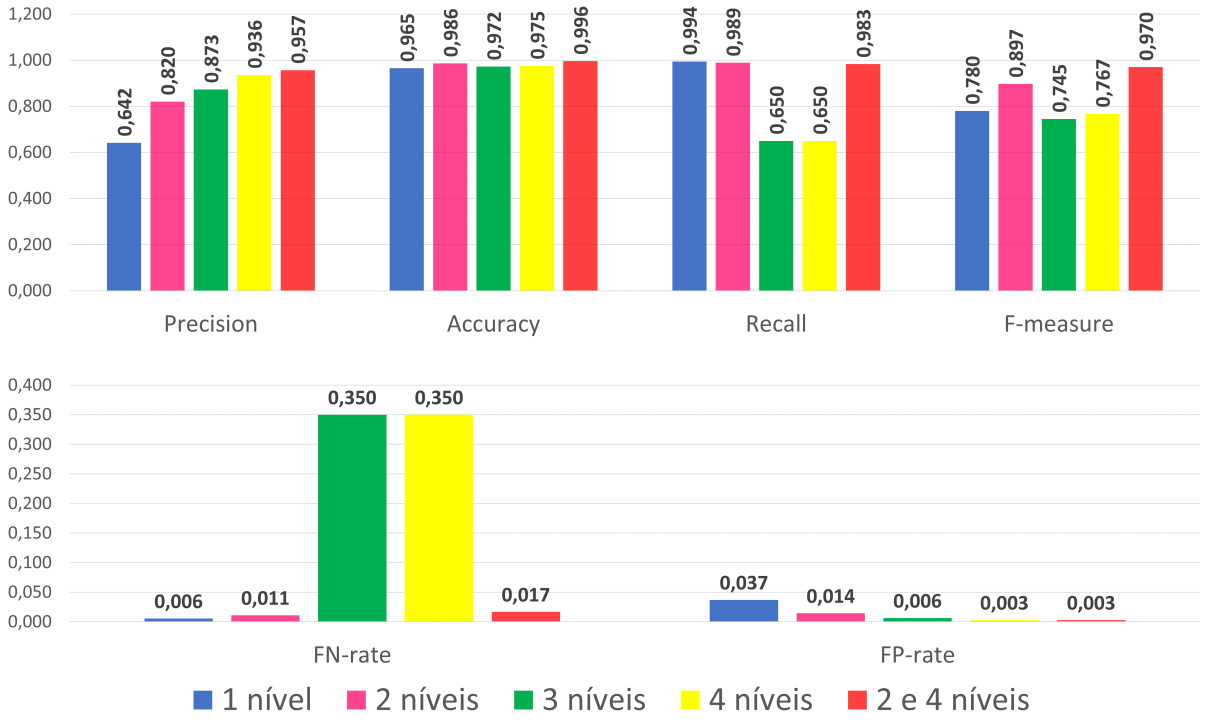


Figura 18 – Resultados dos testes utilizando quatro diferentes níveis da DWT e utilizando 2 e 4 níveis.

que $0,5$ a detecção foi inoperante e pouco confiável. Logo, o valor de ϵ mais adequado para a detecção de *Portscan* foi $\epsilon = 0,5$. Todos os resultado obtidos podem ser melhor visualizados na Figura 19 e na Tabela 5.

Tabela 5 – Resultados das métricas utilizadas na escolha do melhor valor de ϵ para detecção de *Portscan*.

ϵ	**Precisão**	**Acurácia**	**FN-rate**	**FP-rate**	***recall***	**F-measure**	**AUC**
0	$0,613$	$0,973$	$0,05$	$0,026$	$0,95$	$0,745$	$0,962$
$0,25$	$0,77$	$0,986$	$0,05$	$0,012$	$0,95$	$0,851$	$0,969$
$0,5$	$0,905$	$0,994$	$0,05$	$0,004$	$0,95$	$0,927$	$0,973$
$0,75$	$0,952$	$0,985$	$0,333$	$0,001$	$0,667$	$0,784$	$0,833$
1	$0,967$	$0,978$	$0,517$	$0,0007$	$0,483$	$0,644$	$0,741$
$\geq 1,25$	–	$0,958$	1	0	0	–	$0,5$

6.3.2 Parâmetros da Segunda Abordagem

Para a segunda abordagem foram realizados testes combinando dez tamanhos de janelas (16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192 amostras) com quatro profundidades de decomposição da DWT (1, 2, 3 e 4 níveis). A Figura 20 apresenta uma análise de duas métricas (AUC e *recall*) para todos os testes realizados. O tamanho da janela está representado por M e o nível de decomposição da DWT por j.

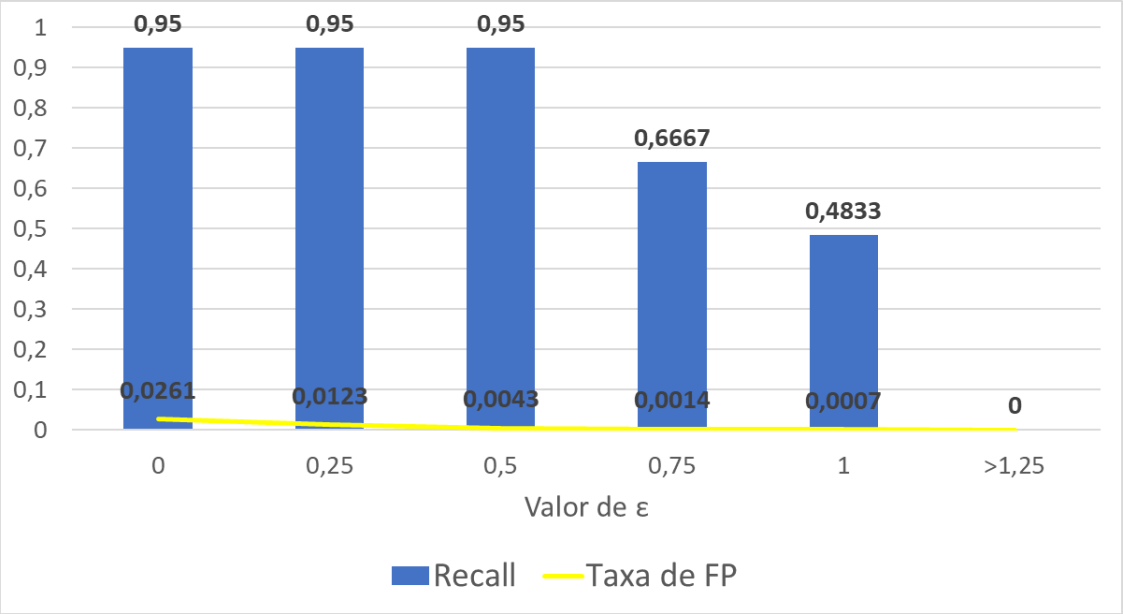


Figura 19 – Taxas de Recall e falso-positivo para os valores de ϵ de 0 até $3,5$, em intervalos de $0,25$.

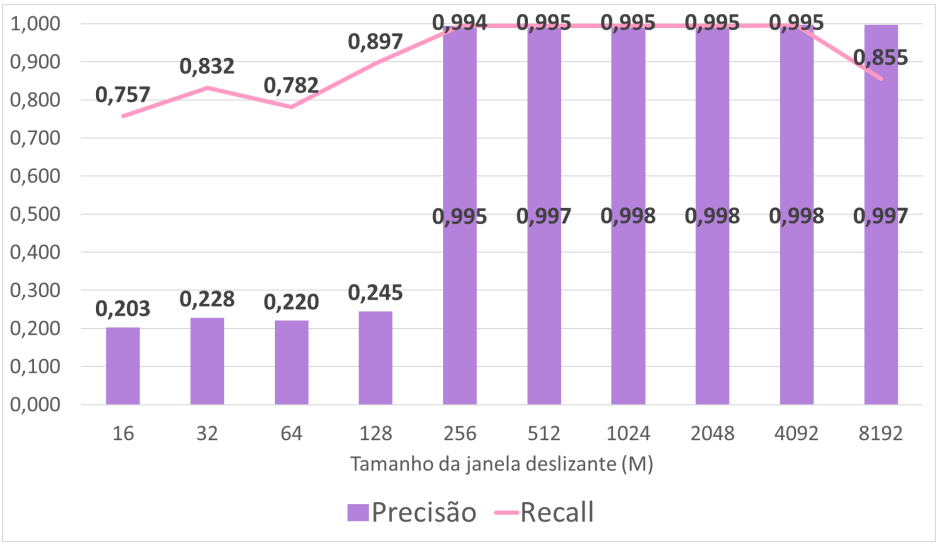


Figura 20 – Análise da combinação de diferentes parâmetros de WaveDetect–IQR.

Por meio da Figure 20 é possível observar que janelas deslizantes de tamanho abaixo de 256 possuem taxas de precisão abaixo de 25% e *recall* abaixo de 90%, isto é, possuem uma detecção menos eficaz se comparada com janelas maiores e também pela precisão é possível observar que existe uma alta taxa de falsos-alarmes, o que não é desejável em um sistema de detecção anomalias. Para os tamanhos de janela deslizante a partir de 256 os resultados começam a tendenciar para uma detecção efetiva, com seu ápice na janela deslizante de tamanho 1024, com excelentes taxas de precisão (99.89%) e *recall* (99.47%) que indicam que houve uma detecção próxima da totalidade. Já utilizando uma janela muito grande, de tamanho 8192, os resultados sofreram um decaimento com

relação às janelas de tamanho menor, com *recall* próximo a 85% o que indica uma detecção menos efetiva que nas janelas imediatamente menores.

Com relação aos níveis da DWT, é possível observar que nas janelas menores de 256, ainda que independente do nível o resultado seja insatisfatório, nos níveis mais profundos as taxas de falso-positivo tendem a diminuir e, em contrapartida, as taxas de falso-negativo tendem a aumentar, exatamente pelo fato de que em níveis mais profundos os valores dos coeficientes de aproximação são cada vez mais "grosseiros", muitas vezes não sendo suficientes para auxiliar em uma detecção eficiente. Esta tendência é percebida independente da janela utilizada, às vezes com diferenças sutis entre os níveis, outras vezes com discrepâncias mais aparentes. Considerando tudo isto, a combinação de parâmetros que apresentou os melhores resultados foi a janela de tamanho 1024 utilizando um nível de decomposição da DWT.

6.4 Avaliação dos Sistemas

Conforme explicado na seção 6.1, para a análise das abordagens foram utilizados três cenários, que serão explicados com maiores detalhes a seguir.

6.4.1 Primeiro Cenário

Para o primeiro cenário foi utilizada a abordagem WaveDetect–ND. Neste cenário, a ideia é avaliar a eficácia da detecção dos ataques para esta abordagem utilizando dois tipos de filtros wavelet, os filtros de Haar e de Daubechies (DB4).

A primeira avaliação realizada foi com respeito à caracterização do tráfego. Apesar da caracterização se apresentar de maneira implícita neste modelo, já que cada janela W_f representa a previsão para um intervalo de trinta segundos é possível realizar a média para cada janela W_f para obter uma representação visual para comparar com os dados originais do tráfego. Essa visualização pode ser observada na Figura 21, para os filtros de Haar (linha laranja) e de Daubechies (linha roxa), respectivamente, considerando os dois dias de dados utilizados, com anomalias DDoS e *Portscan*. É possível observar que a caracterização não acompanha o movimento anômalo para ambos filtros DWT, o que indica que a janela deslizante W_f não engloba esses pontos.

A partir desta caracterização é realizada a detecção dos ataques. As Figuras 22(a) e 22(b) e também a Tabela 6 apresentam os resultados da detecção. Por meio destes resultados pode-se concluir que ambos os filtros foram eficazes na detecção dos intervalos anômalos. No entanto, o filtro de Haar, apesar de possuir taxas de falso-positivo ligeiramente mais altas, mostrou-se melhor para a detecção dos intervalos anômalos, com valores de *recall*, acurácia, F-measure e AUC mais elevados que os valores para DB4.

Tabela 6 – Resultados para as métricas de avaliação da detecção de WaveDetect–ND (Haar) e WaveDetect–ND (DB4).

Métrica	WaveDetect–ND (Haar)	WaveDetect–ND (DB4)
Precisão	0,957	0,960
Acurácia	0,996	0,993
Recall	0,983	0,933
F-measure	0,970	0,946
AUC	0,997	0,990
Taxa de FP	0,003	0,0026

Embora estes resultados mostrem os pontos de acerto e erro com relação à existência ou não de anomalias, eles não informam erros com relação à classificação entre os dois tipos de anomalias. Um ponto pode ser classificado como anômalo porém ser classificado na categoria errada de ataque. Então as matrizes de confusão apresentadas nas Figuras 23(a) e 23(b) expõem de maneira simplificada qual a quantidade de intervalos classificados como normais, como DDoS e como *Portscan*, comparado com os rótulos reais, para os dois dias analisados. Os elementos da diagonal principal da matriz representam os intervalos em que houve uma classificação correta, sendo que com o filtro de Haar houve acerto de 5738 das 5760 amostras e para Daubechies 5722 em 5760. Os outros valores das matrizes apresentam os valores para os possíveis erros (FP *Portscan*, FP DDoS, FN *Portscan*, FN DDoS, DDoS classificado como *Portscan* e *Portscan* classificado como DDoS). É possível verificar que o filtro de Daubechies foi ligeiramente melhor quanto à taxa de falsos alar-

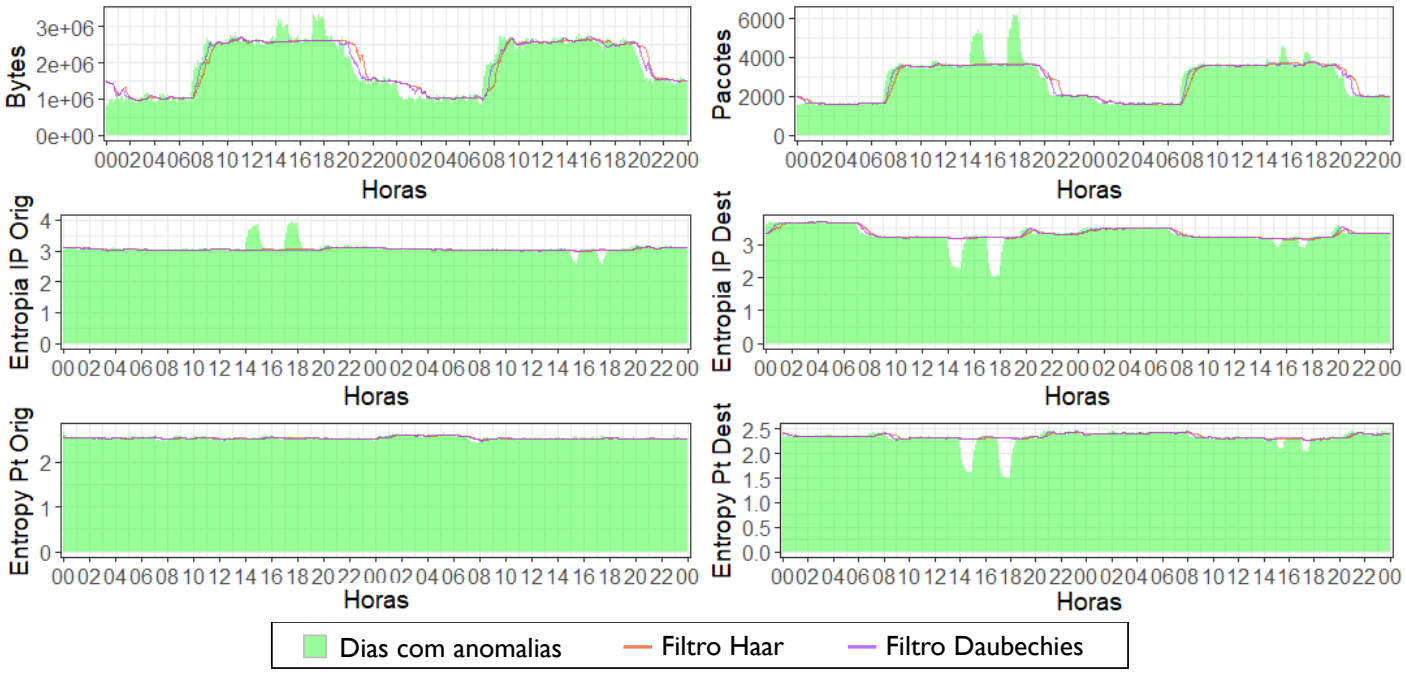
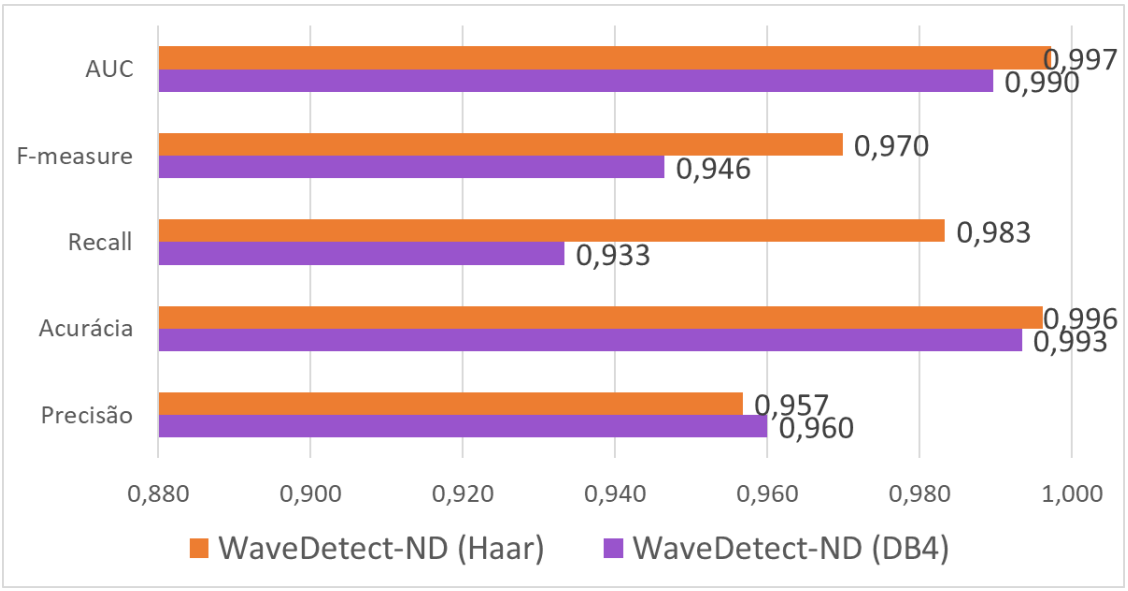
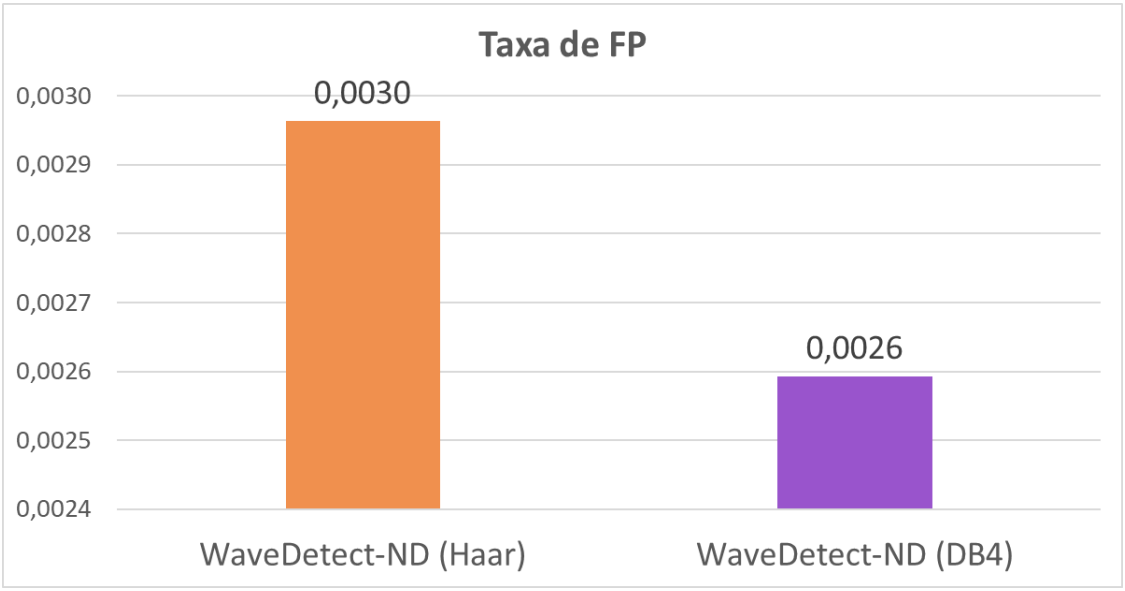


Figura 21 – Tráfego original emulado comparado com o DSNSF gerado pelo sistema.



(a)

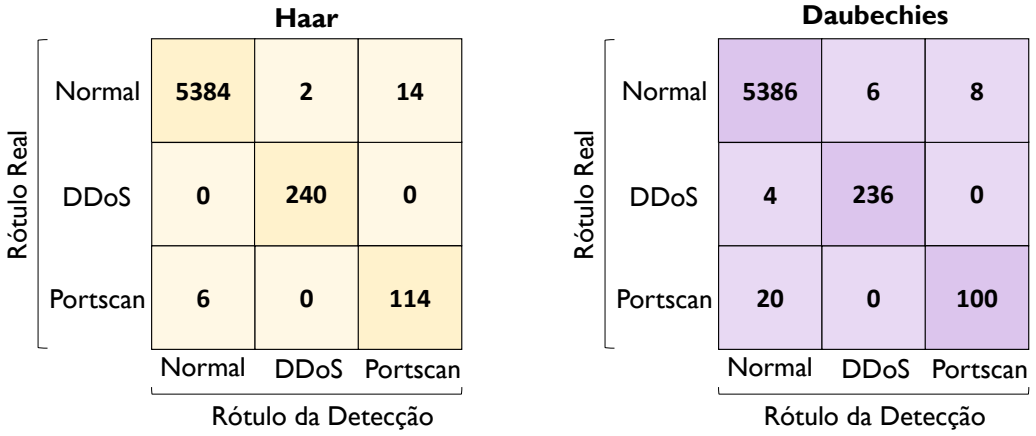


(b)

Figura 22 – Métricas para os testes realizados com os filtros de Haar e Daubechies.

mes, detectando duas amostras a menos que Haar, o que indica menos alarmes falsos no sistema. No entanto, quando tratamos das detecções dos intervalos com anomalias, Haar se mostrou mais eficiente para os dois tipos de ataques, com uma detecção de todas as 240 amostras de DDoS contra 236 de Daubechies, e detecção de 114 amostras contra apenas 100 do total de 120 amostras de *Portscan*.

Com respeito aos filtros é necessário analisar a natureza de cada um deles. Como as formas matriciais do algoritmo DB4 apresentam sobreposição entre iterações na etapa de transformação, o DB4 captura melhor detalhes que são perdidos pelo algoritmo de *wavelet* Haar. No entanto, o fato de possuir aproximações obtidas por meio do produto



(a) (b)

Figura 23 – Matrizes de confusão para os filtros de Haar (a) e Daubechies (b).

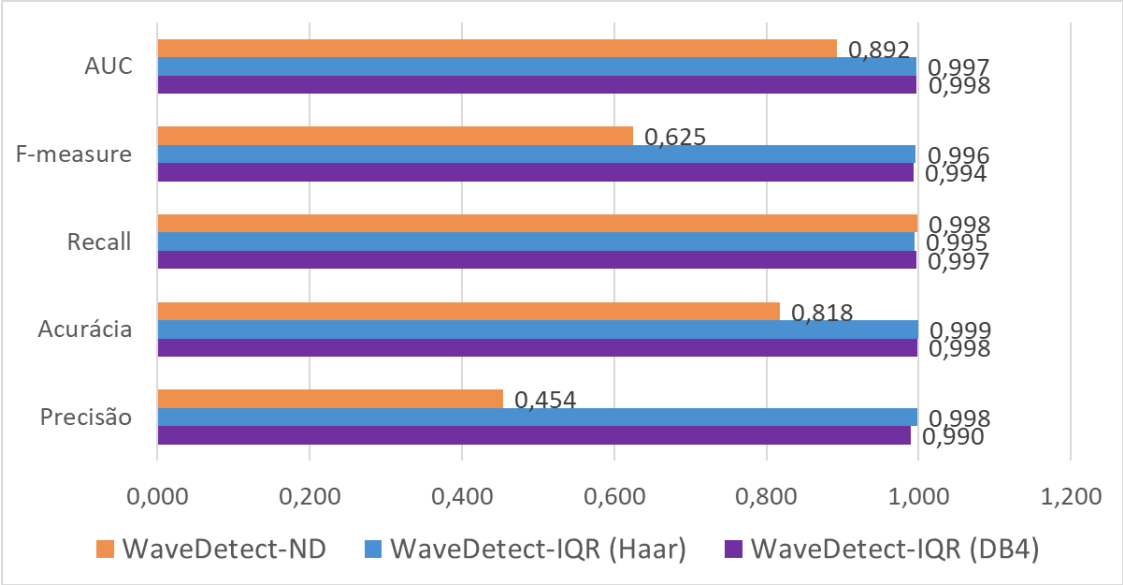
interno entre o dobro de filtros e consequentemente o dobro de amostras de entrada que Haar, DB4 fornece coeficientes de aproximação mais distantes que o dado original para um mesmo nível [67], se comparado ao filtro de Haar. Estas diferenças entre os filtros de Haar e DB4 explicam o fato da detecção com Haar ser mais precisa que a detecção realizada utilizando o filtro DB4. Esta análise se fortalece se considerarmos a diferença entre ambos filtros para a detecção de DDoS e *Portscan*. Para DDoS, que utiliza dois níveis para detecção a diferença de detecção entre os filtros foi de apenas $1,67\%$. Já para a detecção de *Portscan*, que utiliza quatro níveis de decomposição a diferença foi de $11,67\%$.

6.4.2 Segundo Cenário

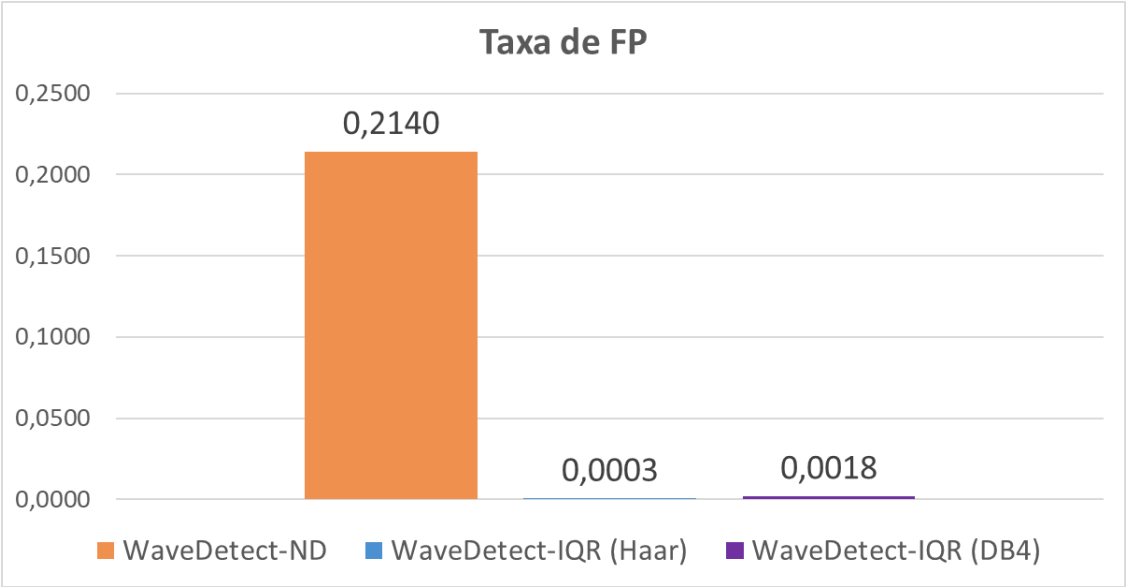
O segundo cenário utiliza coleta de cinco segundos. Para este cenário foram realizados testes com ambas abordagens, WaveDetect–ND com filtro Haar e WaveDetect–IQR, a fim de confrontá-las. Para o modelo WaveDetect–IQR também foram realizados testes com os dois filtros utilizados no primeiro cenário, Haar e Daubechies.

Os gráficos da Figura 24(a) e Figura 24(b) e a Tabela 7 apresentam os resultados da detecção para o segundo cenário.

Por meio da análise destas métricas, é possível verificar que a taxa de detecção dos intervalos anômalos foi alta para os três modelos, com valores de *recall* de $99,8\%$ para WaveDetect-ND, $99,5\%$ para WaveDetect-IQR com o filtro de Haar e $99,7\%$ para WaveDetect-IQR com o filtro de Daubechies. No entanto, a maior discrepância entre as abordagens está exatamente na quantidade de alarmes falsos que cada uma gerou. A abordagem WaveDetect–ND produziu grandes quantidades de alarmes falsos, como pode ser verificado por meio da sua taxa de falso-positivo, que é de $21,4\%$ uma taxa de quase um quarto dos pontos normais assinalados como anômalos. Já para a abordagem WaveDetec–IQR utilizando os dois filtros a discrepância foi menor, com taxas de falso-



(a)



(b)

Figura 24 – Resultados da detecção no segundo cenário.

positivo de $0,03\%$ para Haar e $0,18\%$ para Daubechies. De forma geral a abordagem WaveDetect–ND se mostrou menos eficaz que a segunda abordagem com Haar e com Daubechies, e entre os dois filtros para a segunda abordagem a diferença é mínima. Isso pode ser observado pelo valor de AUC para cada um dos modelos, com AUC de $89,2\%$ para WaveDetect–ND, $99,7\%$ para WaveDetect–IQR com Haar e $99,8\%$ para WaveDetect–IQR com Daubechies, conforme apresentado nas Figuras 24(a) e 24(b).

Com relação ao tipo de filtro utilizado, neste cenário não houve uma diferença marcante entre os filtros. Isso pode ser explicado pela taxa de amostragem, que sendo menor provê uma diferença menor entre as amostras subsequentes ou ainda, e mais pos-

Tabela 7 – Resultados para as métricas de avaliação da detecção de WaveDetect–ND (Haar), WaveDetect–IQR (Haar) e WaveDetect–IQR (DB4).

Métrica	WaveDetect–ND (Haar)	WaveDetect–IQR (Haar)	WaveDetect–IQR (DB4)
Precisão	0, 545	0, 998	0, 99
Acurácia	0, 818	0, 999	0, 998
Recall	0, 998	0, 995	0, 997
F-measure	0, 625	0, 996	0, 994
AUC	0, 892	0, 997	0, 998
Taxa de FP	0, 214	0, 0003	0, 0018

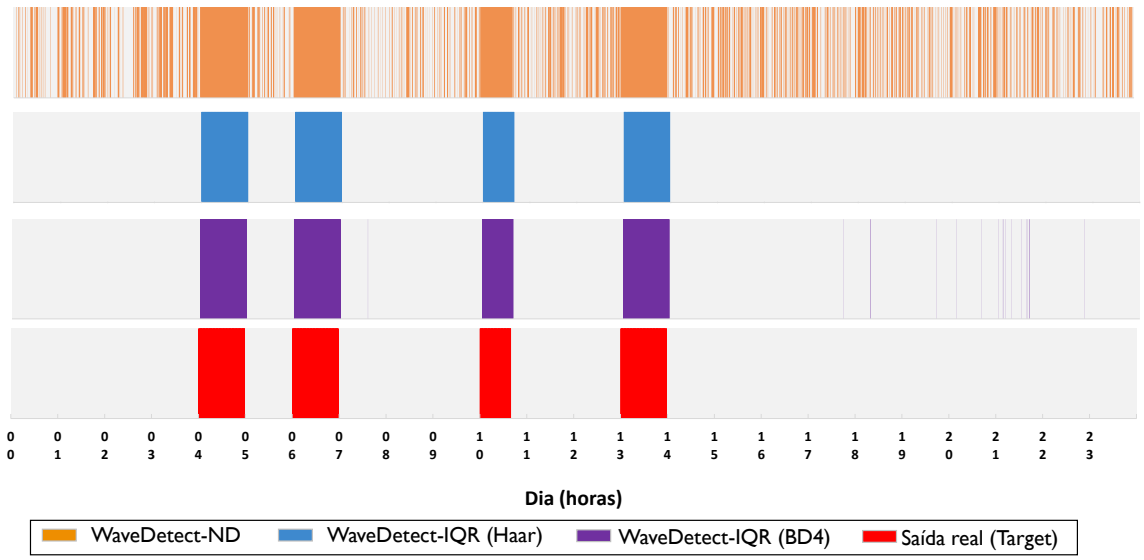


Figura 25 – Alarmes de WaveDetect–ND, WaveDetect–IQR com Haar, WaveDetect–IQR com DB4 e da saída real.

sivelmente, pelo nível da DWT, já que WaveDetect–IQR utiliza apenas um nível, o que carrega menos aproximações sucessivas do dado original. Desta forma, mesmo que o filtro de Daubechies forneça coeficientes de aproximação mais distantes do dado original que Haar, ao utilizar profundidade DWT de apenas um nível, esta diferença de precisão na representação do dado original não se apresenta de maneira tão significativa quanto na abordagem WaveDetect-ND.

Os alarmes gerados por cada uma das duas abordagens pode ser visualizado na Figura 25. Para cada um dos modelos temos no eixo X o intervalo em horas durante o dia. Quando uma anomalia é detectada, um alarme é gerado, que são representadas pelas linhas coloridas dentro do gráfico de alarmes. Comparando o *Target* com as outras imagens, é possível verificar o que as métricas apresentadas anteriormente indicam, que WaveDetect–ND proporciona uma detecção alta, porém com muitos alarmes falsos, enquanto que WaveDetect–IQR com Haar e Daubechies possuem menos alarmes falsos e detecção similar.

Toda a análise feita até o momento nos mostra que WaveDetect–IQR com ambos filtros obteve maiores taxas de detecção e uma baixa taxa de falso-positivo, se comparada à abordagem WaveDetect–ND. No entanto, não deixa claro se a classificação quanto ao tipo de intrusão detectada é feita de maneira correta. Tal qual no primeiro cenário, neste segundo cenário realizamos a análise das matrizes de confusão.

As matrizes de confusão na Figura 26(a), Figura 26(b) e 26(c) apresentam os resultados da detecção para WaveDetect–ND, WaveDetect–IQR com filtro de Haar e WaveDetect–IQR com filtro de Daubechies, respectivamente. Para cada uma das 17280 amostras presentes em um dia de tráfego, é possível verificar a classificação feita pelo modelo, comparando com a real classificação de cada amostra. Na diagonal principal da matriz é possível verificar os acertos na detecção. Analisando estas matrizes, é possível concluir que não houve erro quanto a classificação entre DDoS e *Portscan*, ou seja, DDoS detectado como *Portscan* ou vice versa. Também é possível visualizar que das 1424 amostras com DDoS, WaveDetect–ND e WaveDetect–IQR (DB4) detectaram 1421 e WaveDetect–IQR (Haar) detectou 1412. Para as 1194 amostras com *Portscan*, WaveDetect–ND detectou 1193, WaveDetect–IQR (Haar), 1192, e WaveDetect–IQR (DB4) detectou 1189 amostras. Já para os alarmes falsos comprova-se o que já foi apresentado pela taxa de falso-positivo, em que WaveDetect–ND obteve 3138 amostras de falso-positivo, contra 5 de WaveDetect–IQR (Haar) e 26 de WaveDetect–IQR (DB4).

Após a detecção de um ataque, é necessário que medidas sejam tomadas para combater essas ameaças. Então, a partir dos alarmes gerados, a etapa de mitigação foi executada, com o objetivo de minimizar os efeitos dos processos anômalos ao longo do tráfego.

Os resultados da mitigação podem ser visualizados nas Figuras 27, 28 e 29. As figuras foram aproximadas nos intervalos que contêm as anomalias para facilitar a visualização e a análise. A partir delas é possível concluir que as detecções fornecidas por WaveDetect–IQR com ambos filtros foram mais adequadas para a mitigação das ameaças, visto que WaveDetect–ND, apesar de detectar os intervalos anômalos com alta acurácia forneceu grandes quantidades de alarmes falsos, fazendo com que a mitigação atuasse em diversos intervalos de tráfego legítimo.

Uma análise mais aprofundada desta mitigação é feita por meio da aplicação de um teste estatístico chamado Teste de McNemar e também das taxas de pacotes anômalos descartados. O teste de McNemar é um teste não paramétrico aplicado em amostras pareadas de dados nominais. Ele é aplicado em tabelas de contingência (2x2) para dados com categorias dicotômicas a fim de determinar se as frequências marginais são ou não iguais [92]. Este teste fornece uma análise matemática que juntamente com a análise visual comprova a eficácia da mitigação. Vale destacar que este teste comprova a eficácia da mitigação nos casos onde os alarmes foram adequados, o que não ocorre com o modelo

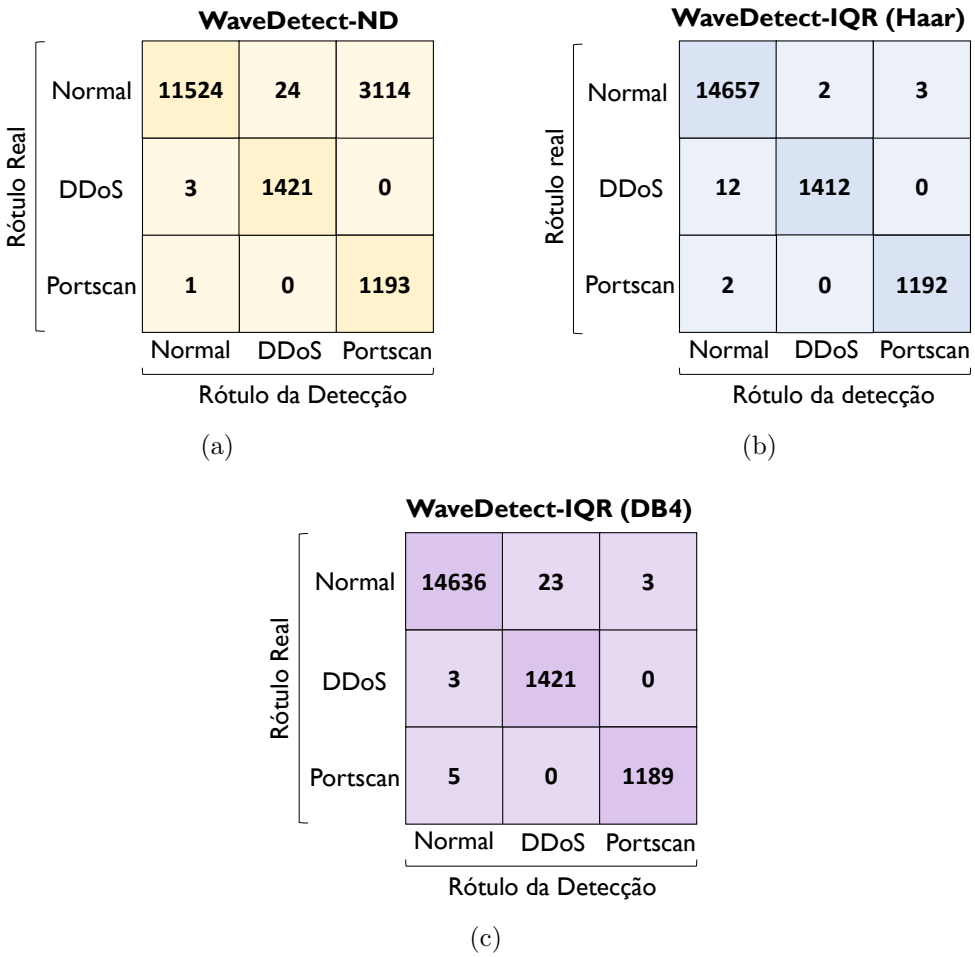


Figura 26 – Matriz de confusão para WaveDetect–ND (a), WaveDetect–IQR (Haar) (b) e WaveDetect–IQR (DB4) (c).

WaveDetect–ND. O teste foi aplicado em dois conjuntos de alarmes pareados, antes e após a mitigação, que informam onde há ou não ataques. A partir daí, uma tabela de contingência é formada para as abordagens, que está descrita na Tabela 8. Esta tabela tem o objetivo de apresentar o que foi detectado como normal e anômalo antes da mitigação – normal (antes) e anômalo (antes) e o que foi detectado como normal e anômalo depois da mitigação – normal (depois) e anômalo (depois). Com base nesta tabela podemos definir quatro situações: *i)* pontos detectados como normais antes e após a mitigação; *ii)* pontos detectados como anômalos antes e após a mitigação; *iii)* pontos detectados como normais antes da mitigação e após a mitigação detectados como anômalos, e por fim; *iv)* pontos detectados como anômalos antes da mitigação e após esta, detectados como normais. Os resultados para cada uma das quatro situações serão utilizadas para a execução do teste de McNemar.

O teste de McNemar utilizado possui nível de significância $\alpha = 5\%$, e conta com a hipótese nula (H_0) de que as frequências marginais são iguais, o que indicaria que a mitigação não foi eficaz. Aplicando o teste na tabela, os resultados para as três aborda-

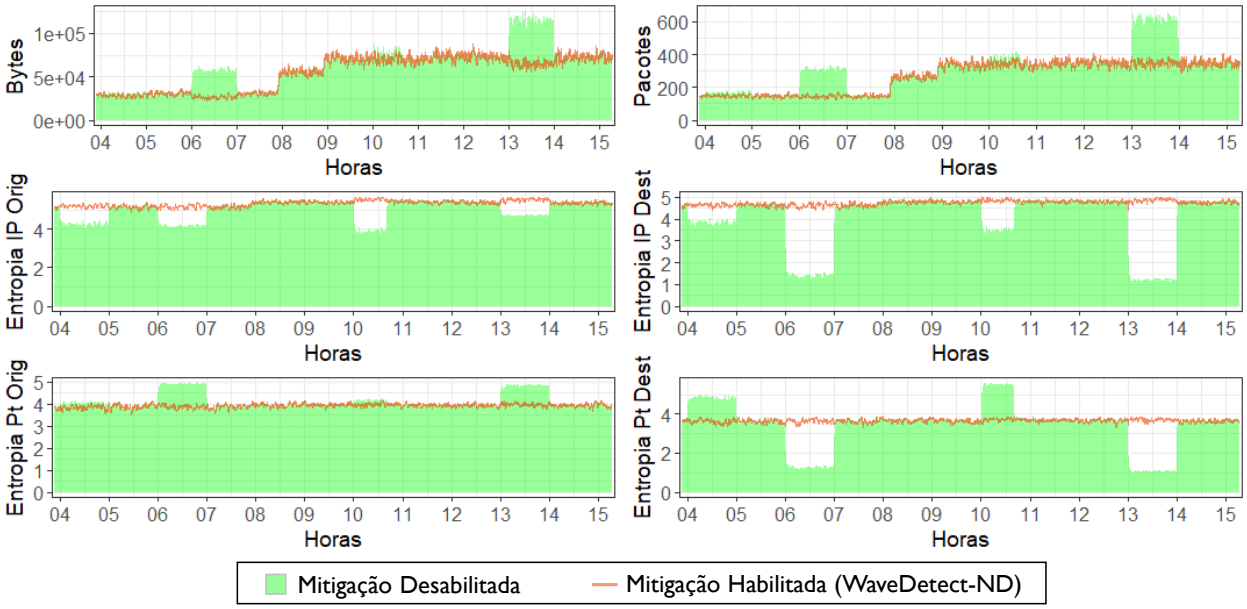


Figura 27 – Gráfico com o tráfego sem mitigação e o tráfego mitigado utilizando WaveDetect–ND.

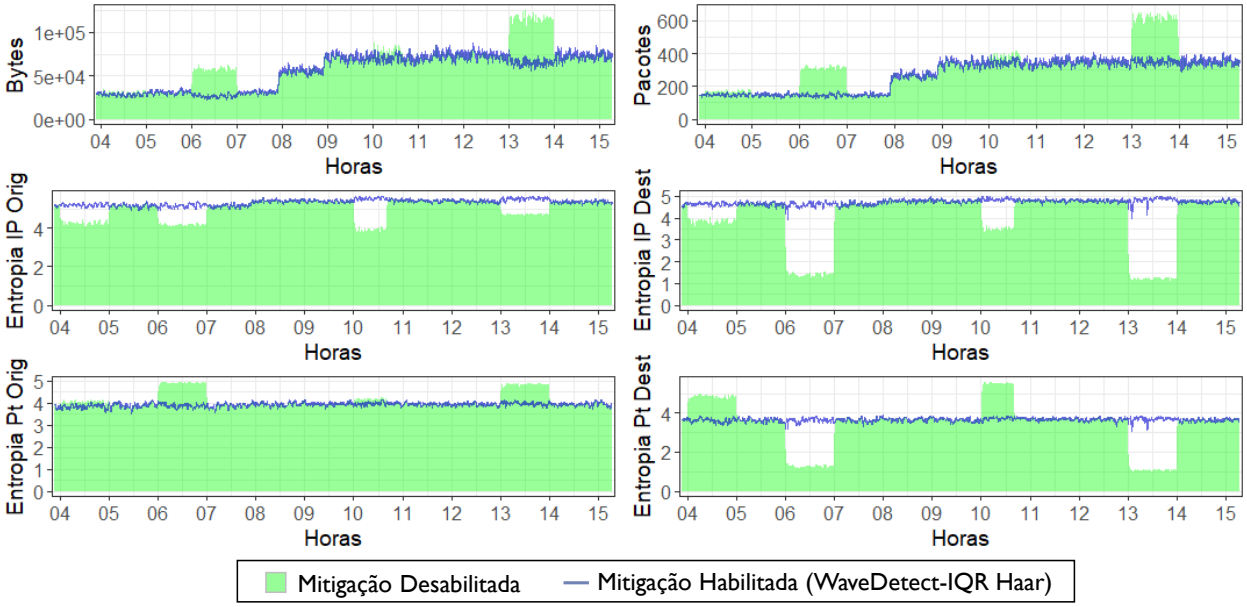


Figura 28 – Gráfico com o tráfego sem mitigação e o tráfego mitigado utilizando WaveDetect–IQR (Haar).

gens foram de $p - valor < 0,0001$ que é menor que o valor de α, logo, a hipótese nula é rejeitada, o que indica que houve sim diferença nas frequências e que então, a mitigação foi eficaz em minimizar os efeitos das ameaças. Isso pode ser claramente verificado na Tabela 8 observando que antes da mitigação as propostas que utilizam IQR detectaram aproximadamente 2600 amostras como anômalas, no entanto, depois da mitigação,

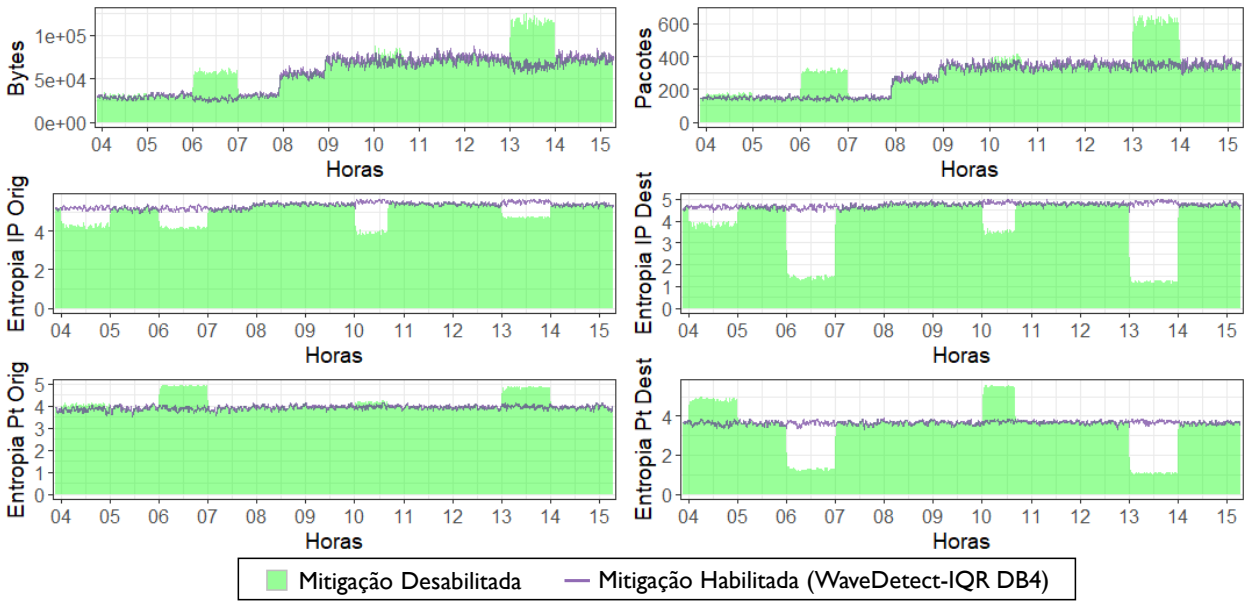


Figura 29 – Gráfico com o tráfego sem mitigação e o tráfego mitigado utilizando WaveDetect–IQR (DB4).

Tabela 8 – Tabela de contingência para WaveDetect–ND, WaveDetect–IQR (Haar) e WaveDetect–IQR (DB4) (Teste de McNemar - se $p - valor < 0,0001$ então rejeita H_0.

Abordagem	normal (antes)	anômalo (antes)	normal (depois)	anômalo (depois)	p-valor
WaveDetect–ND	11528	5752	13702	3578	$< 0,0001$
WaveDetect–IQR (Haar)	14671	2609	17277	3	$< 0,0001$
WaveDetect–IQR (DB4)	14644	2636	17257	23	$< 0,0001$

WaveDetect–IQR (Haar) detectou como anômalo apenas 3 amostras do total de 17280 enquanto WaveDetect–IQR (DB4) detectou 23, o que mostra que os comportamentos anômalos foram retirados do tráfego, fazendo com que não fosse detectada praticamente nenhuma amostra como anômala.

Como dito anteriormente, o modelo WavedDetect–ND não produziu alarmes adequados, visto que possui uma taxa de falso-positivo de 21%. Isto implica que, independente de mitigar anomalias ou não, a abordagem ainda realizará a detecção de diversos pontos como anômalos, pois seu processo de detecção considera diversos pontos normais como anômalos.

Ainda com relação à mitigação, a Tabela 9 apresenta a taxa de pacotes anômalos e normais descartados por cada abordagem. É possível verificar que neste cenário as três propostas foram capazes de eliminar praticamente todos os pacotes anômalos presentes no cenário, com taxas finais de remoção de pacotes de $99,18\%$ para WaveDetect–ND,

99% para WaveDetect–IQR (Haar) e 99,1% para WaveDetect–IQR (DB4). No entanto, WaveDetect–ND, como já foi comprovado pela taxa de alarmes falsos, realizou o descarte de cerca de 0,5% dos pacotes normais, enquanto a abordagem com IQR não chegou a 0,03% de descarte.

Tabela 9 – Taxa de pacotes anômalos e normais descartados para as três abordagens utilizadas no segundo cenário.

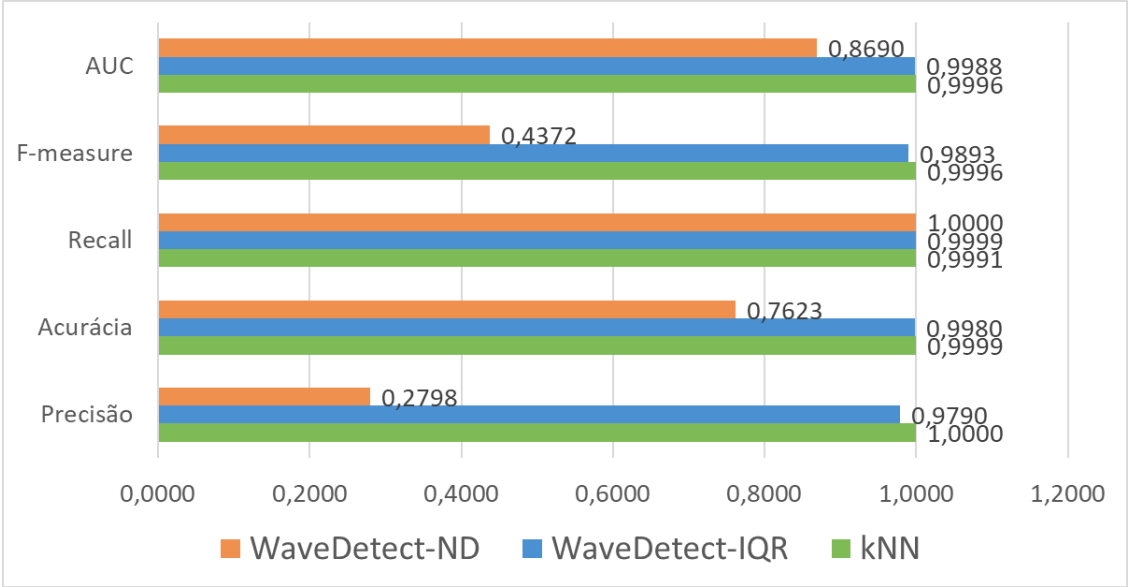
Ataque	Pacotes removidos (WaveDetect–ND)	Pacotes removidos (WaveDetect–IQR Haar)	Pacotes removidos (WaveDetect–IQR DB4)
Ataque 1	100%	100%	99,73%
Ataque 2	98,62%	98,35%	98,62%
Ataque 3	100%	100%	99,8%
Ataque 4	98,08%	97,65%	98,08%
Intervalo Normal	0,56%	0,021%	0,024%

6.4.3 Terceiro Cenário

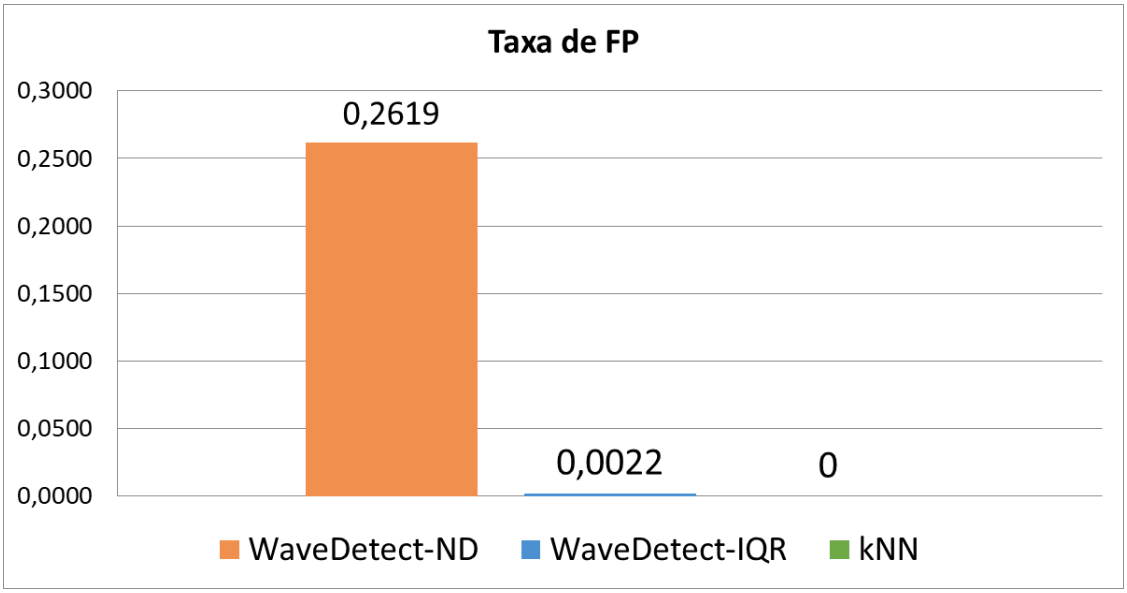
O terceiro cenário é composto por dois intervalos anômalos, um com ataque DDoS e outro com ataque *Portscan*, com amostragem de um segundo. Para a análise dos modelos neste cenário, a ideia é comparar a abordagem WaveDetect–ND com a abordagem WaveDetect–IQR, ambas com filtro de Haar, e ainda compará-las ao modelo proposto por [93], que utiliza kNN, uma técnica supervisionada de classificação utilizada na detecção de anomalias. Para tal análise, as Figuras 30(a) e 30(b) e a Tabela 10 apresentam os resultados da detecção para os modelos WaveDetect–ND, WaveDetect–IQR e kNN. Por meio das Figuras mencionadas, é possível concluir que as três abordagens possuem uma detecção eficaz para os dois tipos de ataques, como pode ser verificado nos valores de *recall* de cada uma. No entanto, WaveDetect–ND forneceu uma quantidade de alarmes falsos muito superior ao fornecido pelas outras duas abordagens, com valores acima dos 26%, contra taxas abaixo de 0,5% para WaveDetect–IQR e kNN.

Tabela 10 – Resultados para as métricas de avaliação da detecção de WaveDetect–ND (Haar), WaveDetect–IQR (Haar) e kNN.

Métrica	WaveDetect–ND (Haar)	WaveDetect–IQR (Haar)	kNN
Precisão	0,2798	0,979	1
Acurácia	0,7623	0,998	0,999
Recall	1	0,9999	0,9991
F-measure	0,4372	0,9893	0,9996
AUC	0,869	0,9988	0,9996
Taxa de FP	0,2619	0,0022	0



(a)



(b)

Figura 30 – Resultados da detecção no terceiro cenário.

Para facilitar a visualização dos alarmes gerados por cada uma das propostas deste cenário, a Figura 31 apresenta os alarmes gerados para todas as propostas e também o alarme real.

Para avaliar a qualidade da detecção, isto é, se a classificação das anomalias detectadas foi realizada de forma correta, as Figuras 32(a) a 32(c) apresentam as matrizes de confusão para os três modelos, indicando quantas amostras (segundos) cada classe possui de fato em comparação com as amostras que foram detectadas para cada categoria (saída do modelo proposto).

É possível verificar que as três abordagens detectaram a totalidade dos intervalos

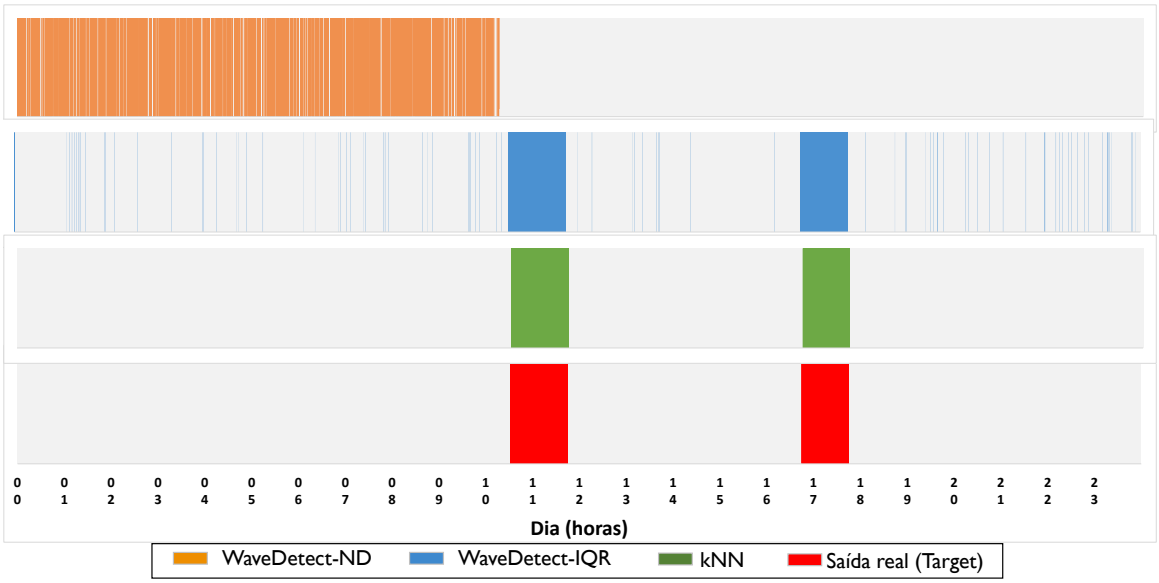


Figura 31 – Alarmes de WaveDetect–ND, WaveDetect–IQR, kNN e da real saída (Target).

com DDoS, detectando todos os 4380 segundos com esta anomalia. Já para a detecção de *Portscan*, kNN foi ligeiramente menos eficaz que as abordagens com DWT, detectando 3593 segundos contra 3596 de WaveDetect–ND e 3595 de WaveDetect–IQR, do total de 3600 segundos. WaveDetect–IQR produziu uma quantidade maior que kNN em intervalos de falso-positivo, 171 pontos contra 0 alarmes falsos. No entanto, WaveDetect–ND forneceu a maior taxa entre as três, com 20276 amostras. Por fim, verificando o acerto com relação à classificação do tipo de anomalia é possível verificar que kNN foi ligeiramente melhor que DWT, visto que não proporcionou uma classificação incorreta entre os dois tipos de anomalias. Se comparado ao kNN, WaveDetect–ND e WaveDetect–IQR obtiveram detecção equivocada nos intervalos com *Portscan*, com 4 segundos detectados como DDoS.

O processo de mitigação também foi realizado para este cenário para as três abordagens. A partir das indicações dos modelos foi realizado o processo de mitigação. As Figuras 33, 34 e 35 apresentam as saídas da mitigação (linha laranja/azul/verde) em comparação com os fluxos antes de sofrerem a mitigação (área em verde). É possível verificar as taxas de detecção explicadas anteriormente se apresentaram coerentes com esta etapa, já que na mitigação tanto utilizando os alarmes fornecidos pelas abordagens DWT quanto kNN pode-se perceber os intervalos de DDoS e *Portscan* serem minimizados, eliminando os pontos de picos e declínios nos intervalos anômalos. Algo importante a ser destacado e que já foi mencionado no cenário anterior é com relação aos falsos alarmes de WaveDetect-ND. Com uma alta taxa de falso-positivo, no processo de mitigação diversos intervalos normais serão mitigados. No entanto, a natureza da política de mitigação *Portscan* – já que no caso os alarmes falsos foram em grande maioria para *Portscan* – proporcionará uma mitigação mínima nesses intervalos, considerando que em um tráfego

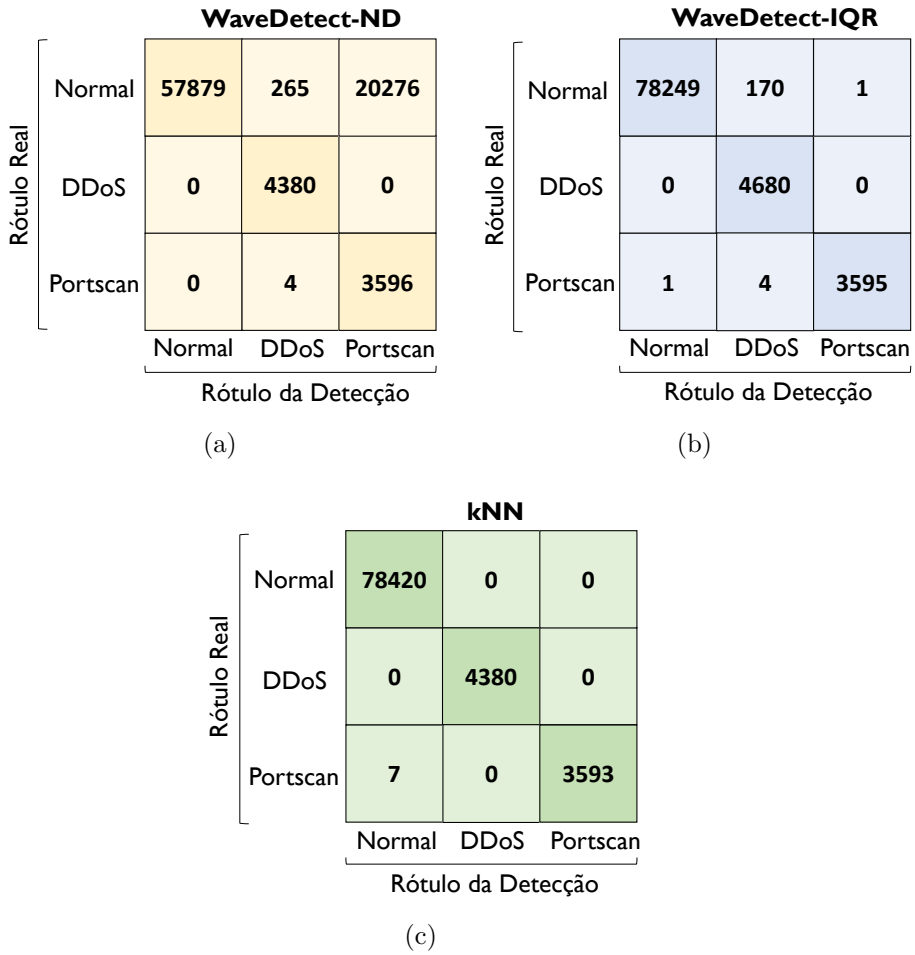


Figura 32 – Matriz de confusão para WaveDetect–ND (a), WaveDetect–IQR (b) e kNN (b).

normal não existem IPs de destino com uma grande variedade de portas acessadas.

Neste cenário, a análise da mitigação por meio do teste de McNemar com $\alpha = 5\%$ e das taxas de pacotes descartados também foi realizado. A Tabela 11 fornece as informações sobre a classificação do tráfego entre anômalo e normal antes e após o processo de mitigação para as três abordagens propostas neste cenário. Aplicando o teste nas informações da tabela para cada um dos modelos, os resultados de p-valor foram abaixo de $2,2 \times e^{-16}$ o qual é menor que o valor de α. Logo, a hipótese nula neste cenário também foi rejeitada, o que indica que houve sim diferença nas frequências e que então, a mitigação foi eficaz em minimizar os efeitos das ameaças. Neste teste, há que se levar em conta que no caso do modelo WaveDetect–ND não existe eficácia na mitigação, devido sua alta quantia de alarmes falsos.

As taxas de remoção de pacotes anômalos e normais são apresentadas na Tabela 12. Como já foi verificado por meio das análises anteriores é possível perceber que tanto WaveDetect–IQR quanto kNN associadas à mitigação são ferramentas eficazes na minimização de anomalias DDoS e *Portscan*. Já WaveDetect–ND foi capaz de mitigar os

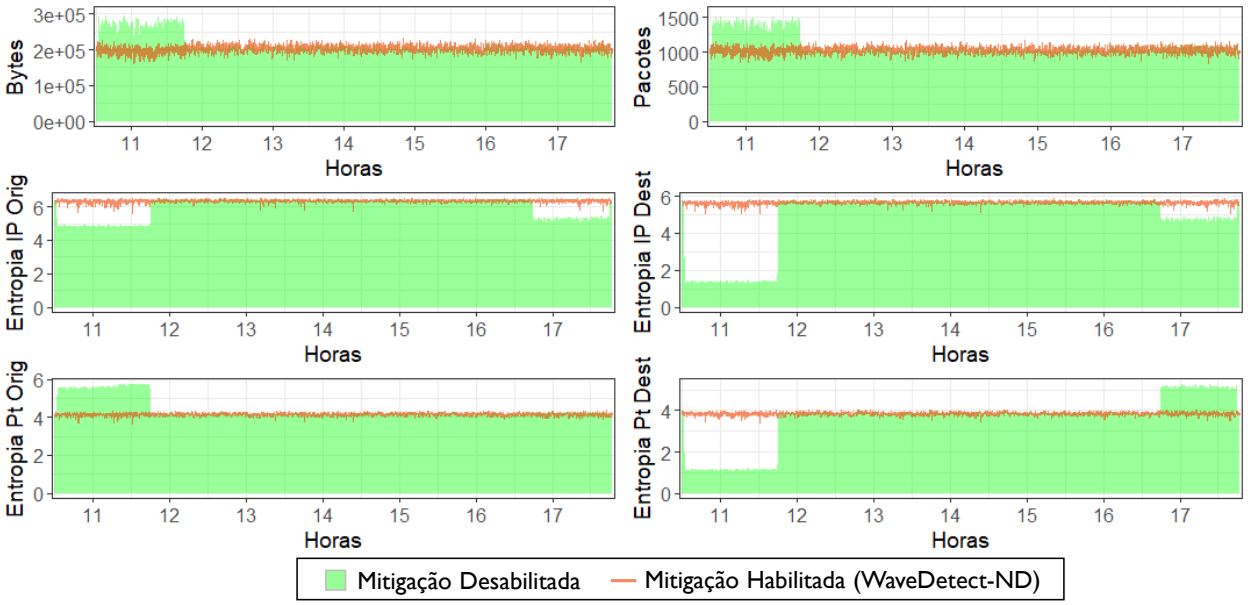


Figura 33 – Gráfico com o tráfego sem mitigação e o tráfego mitigado utilizando WaveDetect–ND.

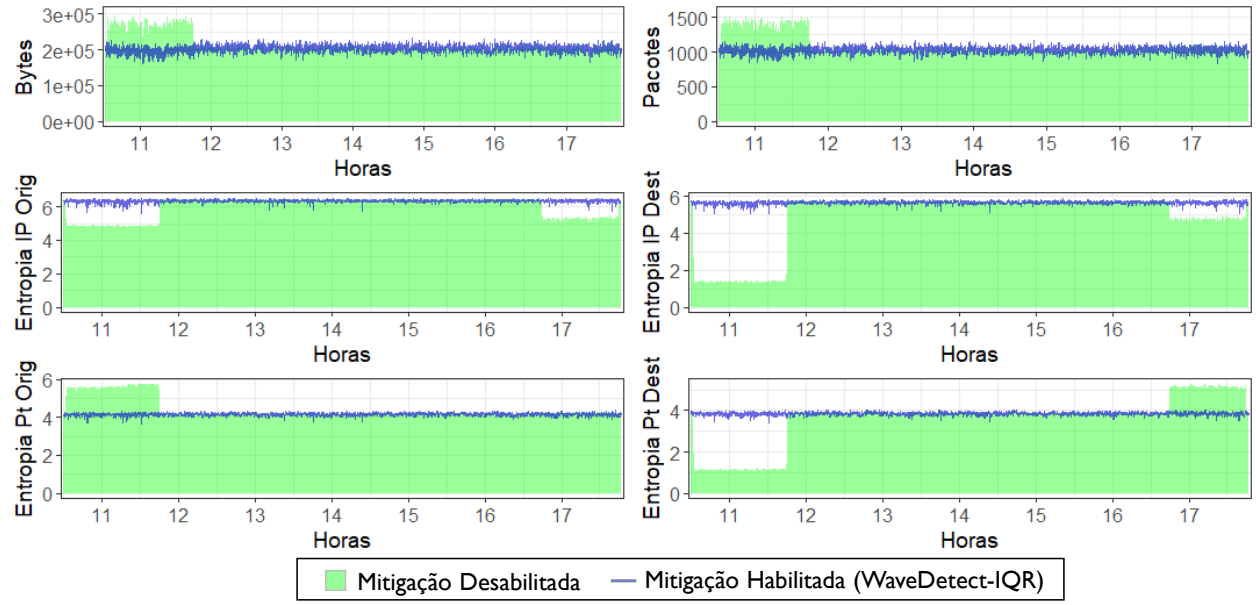


Figura 34 – Gráfico com o tráfego sem mitigação e o tráfego mitigado utilizando WaveDetect–IQR.

intervalos com anomalias, mas devido à sua alta taxa de falso-positivo não se mostrou uma abordagem adequada na mitigação de ataques DDoS e *Portscan*. Isso pode ser comprovado pelas taxas de descarte de pacotes nos intervalos sem anomalias, em que kNN não descartou nenhum pacote, WaveDetect–IQR descartou apenas $0,0025\%$ dos pacotes, e, por fim, WaveDetect–ND descartou $0,32\%$, o que representa uma taxa de descarte cem

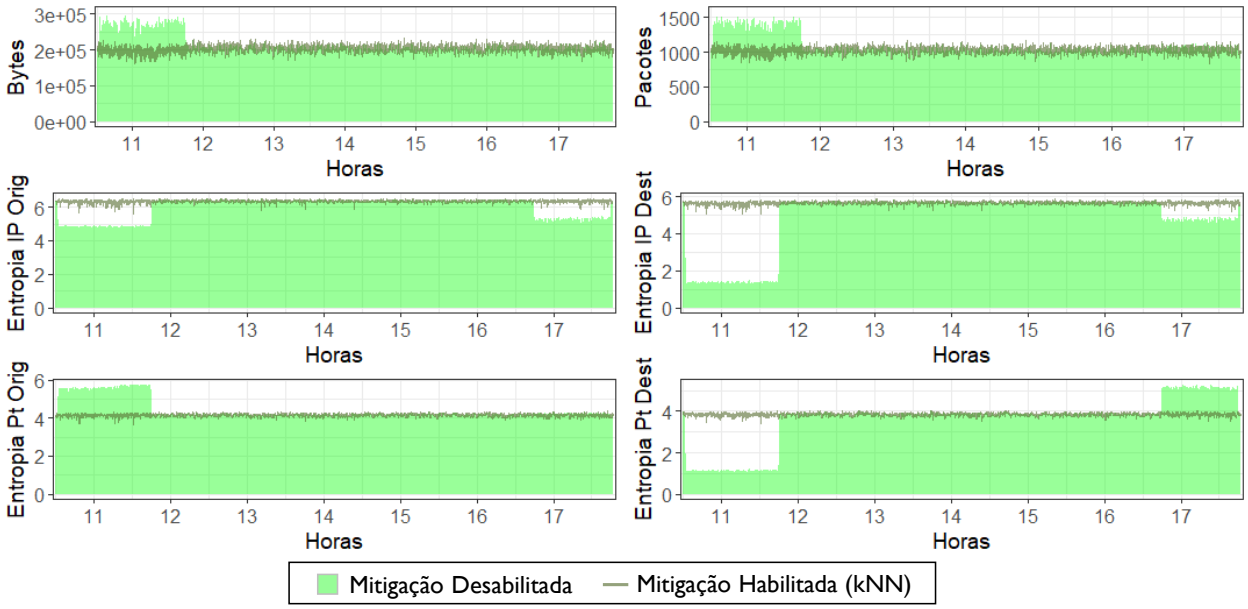


Figura 35 – Gráfico com o tráfego sem mitigação e o tráfego mitigado utilizando kNN.

Tabela 11 – Tabela de contingência para WaveDetect–ND, WaveDetect–IQR e kNN (Teste de McNemar - se $p - valor < 0,0001$ então rejeita H_0.

Abordagem	normal (antes)	anômalo (antes)	normal (depois)	anômalo (depois)	p-valor
WaveDetect-ND	57883	28517	64726	21674	$< 0,0001$
WaveDetect-IQR	78250	8150	86150	250	$< 0,0001$
kNN	78427	7973	86399	1	$< 0,0001$

vezes maior que a abordagem com IQR.

Tabela 12 – Taxa de pacotes anômalos e normais descartados para WaveDetect–ND, WaveDetect–IQR e kNN.

Ataque	Pacotes removidos (WaveDetect–ND)	Pacotes removidos (WaveDetect–IQR)	Pacotes removidos (kNN)
Ataque 1	$97,66\%$	$97,66\%$	$97,66\%$
Ataque 2	$99,94\%$	$99,94\%$	$99,97\%$
Intervalo Normal	$0,32\%$	$0,0025\%$	0%

6.5 Análise de Complexidade Computacional

Uma DWT de uma dimensão consiste em sucessivas multiplicações entre matrizes, entre uma matriz esparsa contendo os filtros com tamanho $N \times N$ e uma matriz coluna com os dados de entrada, de tamanho $N \times 1$. O custo computacional de uma DWT está na soma das convoluções realizadas em cada um dos filtros [94]. O tamanho da entrada

de dados é reduzido pela metade após cada nível de decomposição da DWT, portanto, a complexidade total da DWT é descrita em (6.7).

$$\mathcal{O}\left(N + \frac{N}{2} + \frac{N}{4} + ... + 1\right) = \mathcal{O}\left(N\right) \tag{6.7}$$

Dependendo da abordagem, uma quantidade de níveis diferente foi utilizada. Esta diferença influencia na complexidade θ de cada abordagem, no entanto não modifica a complexidade $\mathcal{O}$ que para todos os casos se mantém $\mathcal{O}\left(N\right)$.

Levando em conta o filtro utilizado, a complexidade $\mathcal{O}$ também se modifica, pois o filtro Daubechies possui o dobro de filtros que Haar. Isso influencia na quantidade de operações realizadas dentro de uma convolução. Portanto, sua complexidade é mais elevada se comparada ao filtro de Haar, que possui quatro filtros, ao passo que DB4 possui oito.

6.6 Considerações sobre o capítulo

Este capítulo teve o objetivo de realizar a análise quanto à eficácia do sistema proposto, utilizando três cenários, todos utilizando Mininet para criação da estrutura da rede. O primeiro cenário utilizando controlador POX e os cenários dois e três utilizando controlador Floodlight. O primeiro cenário foi desenvolvido com amostragem de trinta segundos, o segundo com amostras a cada cinco segundos, e o terceiro e último com amostragem de um segundo.

O primeiro cenário visa comparar o modelo WaveDetect–ND com dois diferentes filtros DWT, Haar e Daubechies. Os resultados para este cenário indicaram que ambos filtros foram eficazes na detecção, no entanto utilizando o filtro de Haar, melhores resultados foram obtidos, o que pode ser verificado pelos valores de AUC que para o filtro de Haar foi de $99,7\%$ e para o filtro de Daubechies foi de 99%. Isso pode ser explicado pelo fato do filtro DB4 possuir sobreposições entre suas iterações na forma matricial e também pela quantidade de filtros de passa-alta e passa-baixa, o que produz coeficientes de aproximação mais distantes do dado orignal, se comparada ao filtro de Haar.

Para o segundo cenário a ideia é confrontar o modelo WaveDetect–ND ao modelo WaveDetect–IQR utilizando dois filtros, Haar e Daubechies a fim de analisar se os parâmetros pré-definidos foram adequados, e ainda observar se houve ou não detecção e as influências deste no processo de mitigação. Por meio dos testes, foi possível concluir que os modelos WaveDetect–IQR com Haar e DB4 foram mais adequados para este cenário, detectando os intervalos anômalos com eficácia provendo um valor de AUC de $99,72\%$ e $99,8\%$ contra um valor de AUC de $89,2\%$ de WaveDetect–ND. Esta diferença tem relação direta com a taxa de alarmes falsos, visto que os modelos utilizando IQR proporcionaram taxas abaixo de $0,5\%$, enquanto WaveDetect–ND forneceu taxa de falso-positivo superior

à 21%. Quanto ao filtro utilizado neste cenário não houve uma diferença significativa entre os filtros, o que pode ser explicado pela quantidade de níveis utilizados pela abordagem WaveDetect–IQR, que proporciona uma aproximação menos distante do dado original, visto que utiliza apenas um nível.

Neste cenário a mitigação também foi experimentada, de forma que ambos modelos que utilizam IQR aliados à mitigação proporcionaram um mecanismo efetivo na mitigação de intervalos anômalos, preservando os intervalos sem anomalias, o que não ocorre com WaveDetect–ND, que apesar de mitigar todos os intervalos com anomalias também ocasiona uma perda nos intervalos que são normais, o que pode ser verificado pelas taxas de descarte de pacotes dos intervalos normais, que em comparação com a abordagem utilizando IQR foi cerca de vinte vezes maior, com $0,56\%$ contra $0,0225\%$.

O terceiro e último cenário propõe uma análise das abordagens WaveDetect–ND e WaveDetect–IQR, ambas com filtro de Haar, em comparação a um modelo presente na literatura, o kNN. Os testes foram realizados para a detecção e a mitigação, o que indicou que as abordagens WaveDetect–IQR e kNN foram eficazes na detecção de intrusos, com taxas de $99,99\%$ e $99,91\%$ respectivamente. Já WaveDetect–ND apesar de detectar e mitigar os intervalos anômalos, assim como no segundo cenário, forneceu altas taxas de falso-positivo, superando os 26%. Vale destacar também que WaveDetect–IQR proporcionou uma detecção ligeiramente mais precisa dos intervalos anômalos, ao passo que kNN produziu uma taxa mais baixa de alarmes falsos. A eficácia de ambos modelos se reflete na mitigação, com taxa média de eliminação de pacotes anômalos de $98,8\%$ para WaveDetect–IQR e de $98,82\%$ para kNN. Neste cenário, WaveDetect–ND também se mostrou pouco eficaz que as outras duas abordagens, com taxas de remoção de pacotes normais cem vezes maior que as abordagens WaveDetect–IQR e kNN.

Após todas estas análises é possível verificar que ambas abordagens propostas e também o sistema de mitigação aqui apresentados se mostraram eficazes na detecção e minimização dos efeitos das anomalias. WaveDetect–ND se comportou muito bem no primeiro cenário, ao contrário do segundo e do terceiro, que com os mesmos parâmetros foi eficaz na detecção mas produziu uma grande quantidade de alarmes falsos. Já o segundo modelo se comportou de maneira eficaz, detectando com altas taxas de acurácia e *recall* as anomalias presentes nos cenários dois e três, providenciando baixa taxa de falso-positivo.

Uma análise entre os filtros permite verificar que o filtro de Haar de maneira geral se comportou de maneira mais eficaz na detecção, se comparado ao filtro DB4. Já com relação às duas abordagens propostas, WaveDetect–IQR tão eficaz quanto WaveDetect–ND na detecção dos intervalos anômalos. No entanto, a abordagem que utiliza IQR necessita de menos ajustes de parâmetros, o que torna sua implementação na prática mais eficiente, visto que com o mesmo intervalo de confiança foi capaz de detectar diferentes tipos de ataques. Isso demonstra que o parâmetro IQR e o intervalo de confiança gerado por meio

dele se mostram descritores mais precisos do tráfego considerado normal, auxiliando nas altas taxas de detecção de diferentes ataques e também na quantidade de falsos alarmes abaixo de 1%.

7 CONCLUSÃO

A detecção de anomalias em redes é uma área em expansão, o que se deve à popularização da Internet e também ao crescimento expressivo na quantidade de dispositivos na rede bem como a introdução de dispositivos IoT, fatos que também ocasionaram no aumento de ações maliciosas dentro do ambiente de redes. Além dos sistemas de detecção e anomalias, outra necessidade também emergiu, a otimização dos processos de configuração e funcionamento da rede. Para tal propósito, as redes SDN surgem, com o objetivo de auxiliar e facilitar processos de configuração e programabilidade dentro do ambiente de rede. Este trabalho apresenta duas abordagens para a detecção de ataques do tipo DDoS e *Portscan* e um sistema para mitigação destes ataques.

A primeira abordagem, nomeada WaveDetect–ND, utiliza uma DWT e um intervalo de confiança que se baseia em propriedades da distribuição normal. A segunda, WaveDetect–IQR, também utiliza DWT, no entanto, o intervalo utilizado é obtido por meio de uma distância interquartil. Ambas abordagens utilizam-se duas janelas, uma contendo o tráfego classificado como normal e outra contendo o tráfego legado e a amostra de interesse (amostra analisada). Ambos métodos possuem baixa complexidade computacional, que juntamente com um baixos intervalos de amostragens, produzem uma solução *fast* para detecção de intrusos.

Além do mecanismo de detecção, também foi proposto um mecanismo de mitigação, em que são geradas políticas para mitigar ataques do tipo DDoS e *Portscan*.

Com relação a análise do sistema proposto, por meio dos testes realizados e dos resultados obtidos foi possível verificar que o sistema com a abordagem WaveDetect–ND não proporcionou uma solução eficaz e funcional para detecção e mitigação de anomalias, visto que produziu altas taxas de alarmes falsos, tornando o sistema pouco confiável, ocasionando o descarte e desta forma a perda de diversos fluxos legítimos. Já o sistema utilizando a abordagem WaveDetect–IQR proporcionou uma solução eficaz para detecção e mitigação de intervalos anômalos, com taxas de falso-positivo abaixo de $0,5\%$ e uma detecção dos intervalos anômalos acima de 99%, que são resultados tão eficazes quanto a abordagem kNN.

Uma última análise realizada foi com relação ao filtro utilizado. Por meio dos testes, foi possível verificar que o filtro de Haar se mostrou mais efetivo que o filtro DB4, principalmente em abordagens que se utilizavam de níveis mais profundos da transformada DWT. Diversos trabalhos que se utilizam das Wavelets para a detecção de anomalias em redes obtiveram resultados também efetivos na detecção de intervalos anômalos, como o trabalho de Tian *et al.* [24] com resultados de detecção próximo de cem por cento e taxa

de falso-positivo de cerca de 6%. Trabalhos como os de Kanarachos [26] *et al.* ou ainda de Dainotti *et al.* [25] também obtiveram taxas de detecção aceitáveis, com valores próximo dos 90%.

Uma característica importante das Wavelets que são trabalhadas em alguns trabalhos da literatura como o trabalho de Dalmazo *et al.*[95] e de Perlin *et al.* [51] que afirmam que as Wavelets são eficazes na redução das taxas de falso-positivo. De fato, ao realizar os testes para escolha de níveis foi possível verificar que de maneira geral ao utilizar uma quantidade profunda o bastante de níveis as taxas tendem a diminuir, ao passo que as taxas de detecção também. Os testes de escolha que foram realizados trataram de encontrar o melhor nível que oferecesse um equilíbrio entre as duas taxas, proporcionando ao final, um sistema capaz de detectar os intervalos anômalos com taxas próximas a 100% e taxas de falso-positivo abaixo de $0,5\%$.

Como proposta futura, considerando que ambas abordagens WaveDetect possuem baixa complexidade computacional as duas podem vir a se complementar, com o intuito de enriquecer o processo de detecção, explorando o que há de mais vantajoso em cada uma delas. Testes com outros filtros DWT também podem ser realizados, a fim de buscar uma função wavelet que se encaixe ao problema. Com relação à mitigação, pôde ser verificado que mesmo com a detecção da totalidade de alguns intervalos anômalos a mitigação não atingiu 100% de remoção de pacotes anômalos. Isso pode ser enriquecido por meio de políticas que realizem uma análise do tráfego pregresso, com uma *blacklist* dos IPs suspeitos ou ainda com a exploração de mais particularidades de cada uma das anomalias.

REFERÊNCIAS

[1] RAWAT, D. B.; REDDY, S. R. Software defined networking architecture, security and energy efficiency: A survey. *IEEE Communications Surveys Tutorials*, v. 19, n. 1, p. 325–346, Firstquarter 2017. ISSN 1553-877X. DOI: 10.1109/COMST.2016.2618874.

[2] DALLAS, G. *Wavelets for Dummies: Signal Processing, Fourier Transforms and Heisenberg*. 2014. Disponível em: "<https://georgemdallas.wordpress.com/2014/05/14/wavelets-4-dummies-signal-processing-fourier-transforms-and-heisenberg/>".

[3] ZHANG, D.; NING, H.; XU, K. S.; LIN, F.; YANG, L. T. Internet of things. *jucs*, v. 18, n. 9, p. 1069–1071, may 2012. <|>http://www.jucs.org/jucs$_1$8$_9$/$internet_of_things$|.

[4] BIZANIS, N.; KUIPERS, F. A. SDN and virtualization solutions for the internet of things: A survey. In: . [S.l.: s.n.], 2016. v. 4, p. 5591–5606. ISSN 2169-3536. DOI: 10.1109/ACCESS.2016.2607786.

[5] SANTOS, R. R. dos; MOREIRAS, A. M.; REIS, E. A.; ROCHA, A. S. da. *Curso IPv6 Básico*. 2010. Disponível em: "<http://ipv6.br/media/arquivo/ipv6/file/48/IPv6-apostila.pdf>".

[6] ALMEIDA, J. M. F. *Breve História da Internet*. 2005. Disponível em: "<http://repositorium.sdum.uminho.pt/bitstream/1822/3396/1/INTERNET.pdf>".

[7] LEE, S.; BAE, M.; KIM, H. Future of iot networks: A survey. *Applied Sciences (Switzerland)*, v. 7, n. 10, 10 2017. ISSN 2076-3417. DOI: 10.3390/app7101072.

[8] CARVALHO, L. F. *Metaheurística Ant Colony Optimization e Análise de Fluxos IP Aplicados a Detecção de Anomalias e a Gerência de Redes*. Dissertação (Mestrado) — Universidade Estadual de Londrina, 118f, fevereiro 2014. Dissertação de Mestrado, Londrina, 2014.

[9] LOPES, R. V.; SAUVÉ, J. P.; NICOLLETTI, P. S. *Gerência de Redes*. 2001. 16-41 p.

[10] PROENÇA, M. L. J. *Baseline Aplicado a Gerência de Redes*. Tese (Doutorado) — Universidade Estadual de Campinas, julho 2005.

[11] FERNANDES, G. J. *Caracterização de tráfego e detecção de anomalias utilizando a análise de componentes principais e fluxos IP*. Dissertação (Mestrado) — Universidade Estadual de Londrina, 96f, 2013. Dissertação de Mestrado, Londrina, 2013.

[12] ASSIS, M. V. O. de; RODRIGUES, J. J. P. C.; PROENÇA, M. L. A novel anomaly detection system based on seven-dimensional flow analysis. In: *2013 IEEE Global Communications Conference (GLOBECOM)*. [S.l.: s.n.], 2013. p. 735–740. ISSN 1930-529X. DOI: 10.1109/GLOCOM.2013.6831160.

[13] JUNIOR, G.; PENA, E.; CARVALHO, L.; RODRIGUES, J.; PROENÇA, M. L. J. Statistical, forecasting and metaheuristic techniques for network anomaly detection. In: . [S.l.: s.n.], 2015. DOI: 10.1145/2695664.2695852.

[14] PENA, E. H. M.; CARVALHO, L. F.; BARBON, S.; RODRIGUES, J. J. P. C.; PROENÇA, M. L. Correlational paraconsistent machine for anomaly detection. In: *2014 IEEE Global Communications Conference*. [S.l.: s.n.], 2014. p. 551–556. ISSN 1930-529X. DOI: 10.1109/GLOCOM.2014.7036865.

[15] CARVALHO, L. F.; FERNANDES, G.; ASSIS, M. de; RODRIGUES, J.; PROENÇA, M. L. Digital signature of network segment for healthcare environments support. *IRBM*, v. 35, n. 6, p. 299 – 309, 2014. ISSN 1959-0318. DOI: <https://doi.org/10.1016/j.irbm.2014.09.001>.

[16] ASSIS, M. V. O. de; CARVALHO, L. F.; RODRIGUES, J. J. P. C.; PROENÇA, M. L. Holt-winters statistical forecasting and aco metaheuristic for traffic characterization. In: *2013 IEEE International Conference on Communications (ICC)*. [S.l.: s.n.], 2013. p. 2524–2528. ISSN 1550-3607. DOI: 10.1109/ICC.2013.6654913.

[17] COLE, E. *Network Security Bible*. 2nd. ed. Wiley, 2011. (Bible). ISBN 9780470570005. Disponível em: <https://books.google.com.br/books?id=Iq8lPbhGRuYC>.

[18] MEHDI, S. A.; KHALID, J.; KHAYAM, S. A. Revisiting traffic anomaly detection using software defined networking. In: _____. *Recent Advances in Intrusion Detection: 14th International Symposium, RAID 2011, Menlo Park, CA, USA, September 20-21, 2011. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 161–180. ISBN 978-3-642-23644-0. DOI: 10.1007/978-3-642-23644-0$_9$.

[19] MASOUDI, R.; GHAFFARI, A. Software defined networks: A survey. *Journal of Network and Computer Applications*, v. 67, n. Supplement C, p. 1 – 25, 2016. ISSN 1084-8045. DOI: <https://doi.org/10.1016/j.jnca.2016.03.016>.

[20] FARHADY, H.; LEE, H.; NAKAO, A. Software-defined networking: A survey. *Computer Networks*, v. 81, p. 79 – 95, 2015. ISSN 1389-1286. DOI: <https://doi.org/10.1016/j.comnet.2015.02.014>.

[21] BERA, S.; MISRA, S.; VASILAKOS, A. V. Software-defined networking for internet of things: A survey. *IEEE Internet of Things Journal*, v. 4, n. 6, p. 1994–2008, Dec 2017. DOI: 10.1109/JIOT.2017.2746186.

[22] UMER, M. F.; SHER, M.; BI, Y. Flow-based intrusion detection: Techniques and challenges. *Journal of Computers & Security*, v. 70, 06 2017. DOI: <https://doi.org/10.1016/j.cose.2017.05.009>.

[23] JAFARPOUR, S.; POLATKAN, G.; BREVDO, E.; HUGHES, S.; BRASOVEANU, A.; DAUBECHIES, I. Stylistic analysis of paintings usingwavelets and machine learning. In: *2009 17th European Signal Processing Conference*. [S.l.: s.n.], 2009. p. 1220–1224. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7077807&isnumber=7077262>.

[24] TIAN, H.; DING, M. Diffusion wavelet-based anomaly detection in networks. In: *2016 17th International Conference on Parallel and Distributed Computing,*

Applications and Technologies (PDCAT). [S.l.: s.n.], 2016. p. 382–386. DOI: 10.1109/PDCAT.2016.087.

[25] DAINOTTI, A.; PESCAPE, A.; VENTRE, G. Nis04-1: Wavelet-based detection of dos attacks. In: *IEEE Globecom 2006*. [S.l.: s.n.], 2006. p. 1–6. ISSN 1930-529X. DOI: 10.1109/GLOCOM.2006.279.

[26] KANARACHOS, S.; MATHEW, J.; CHRONEOS, A.; FITZPATRICK, M. Anomaly detection in time series data using a combination of wavelets, neural networks and hilbert transform. In: *Information, Intelligence, Systems and Applications (IISA), 2015 6th International Conference on*. [S.l.: s.n.], 2015. p. 1–6. DOI: 10.1109/IISA.2015.7388055.

[27] GAO, J.; HU, G.; YAO, X.; CHANG, R. K. C. Anomaly detection of network traffic based on wavelet packet. In: *2006 Asia-Pacific Conference on Communications*. [S.l.: s.n.], 2006. p. 1–5. ISSN 2163-0771. DOI: 10.1109/APCC.2006.255840.

[28] FERNANDES, G.; CARVALHO, L. F.; RODRIGUES, J. J.; PROENÇA, M. L. Network anomaly detection using IP flows with principal component analysis and ant colony optimization. *Journal of Network and Computer Applications*, v. 64, p. 1 – 11, 2016. ISSN 1084-8045. DOI: <https://doi.org/10.1016/j.jnca.2015.11.024>.

[29] ADANIYA, M.; AO, T. A.; PROENÇA, M. L. J. Anomaly detection using metaheuristic firefly harmonic clustering. v. 8, p. 82–91, 01 2013. DOI: 10.4304/jnw.8.1.82-91.

[30] PROENÇA, M. J. L.; COPPELMANS, C.; BOTTOLI, M.; ALBERTI, A.; MENDES, L. The hurst parameter for digital signature of network segment. v. 3124, p. 772–781, 08 2004. DOI: 10.1007/978-3-540-27824-5.

[31] PENA, E. H. M.; CARVALHO, L. F.; BARBON., S.; RODRIGUES, J. J.; PROENÇA, M. L. Anomaly detection using the correlational paraconsistent machine with digital signatures of network segment. *Information Sciences*, v. 420, n. Supplement C, p. 313 – 328, 2017. ISSN 0020-0255. DOI: <https://doi.org/10.1016/j.ins.2017.08.074>.

[32] HAMAMOTO, A. H.; CARVALHO, L. F.; SAMPAIO, L. D. H.; ABRÃO, T.; PROENÇA, M. L. Network anomaly detection system using genetic algorithm and fuzzy logic. *Expert Systems with Applications*, v. 92, p. 390 – 402, 2018. ISSN 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2017.09.013>.

[33] FERNANDES, G.; RODRIGUES, J. J.; ProenÇa, M. L. J. Autonomous profile-based anomaly detection system using principal component analysis and flow analysis. *Applied Soft Computing*, v. 34, p. 513 – 525, 2015. ISSN 1568-4946. DOI: <https://doi.org/10.1016/j.asoc.2015.05.019>.

[34] AMARAL, A. A.; MENDES, L. de S.; aO, B. B. Z.; JUNIOR, M. L. P. Deep ip flow inspection to detect beyond network anomalies. *Computer Communications*, v. 98, p. 80 – 96, 2017. ISSN 0140-3664. DOI: <https://doi.org/10.1016/j.comcom.2016.12.007>.

[35] HAJISALEM, V.; BABAIE, S. A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection. *Computer Networks*, v. 136, p. 37 – 50, 2018. ISSN 1389-1286. DOI: <https://doi.org/10.1016/j.comnet.2018.02.028>.

[36] ABUROMMAN, A. A.; REAZ, M. B. I. A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing*, v. 38, p. 360 – 372, 2016. ISSN 1568-4946. DOI: <https://doi.org/10.1016/j.asoc.2015.10.011>.

[37] SILVA, A. S. da; WICKBOLDT, J. A.; GRANVILLE, L. Z.; SCHAEFFER-FILHO, A. ATLANTIC: A framework for anomaly traffic detection, classification, and mitigation in SDN. In: *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. [S.l.: s.n.], 2016. p. 27–35. DOI: 10.1109/NOMS.2016.7502793.

[38] SAHAY, R.; BLANC, G.; ZHANG, Z.; DEBAR, H. ArOMA: An SDN based autonomic DDoS mitigation framework. *Computers & Security*, v. 70, p. 482 – 499, 2017. ISSN 0167-4048. DOI: <https://doi.org/10.1016/j.cose.2017.07.008>.

[39] PENG, H.; SUN, Z.; ZHAO, X.; TAN, S.; SUN, Z. A detection method for anomaly flow in software defined network. *IEEE Access*, v. 6, p. 27809–27817, 2018. ISSN 2169-3536. DOI: 10.1109/ACCESS.2018.2839684.

[40] PRASATH, M. K.; PERUMAL, B. A meta-heuristic bayesian network classification for intrusion detection. *International Journal of Network Management*, v. 0, n. 0, p. e2047, 2018. DOI: 10.1002/nem.2047.

[41] ASSIS, M. V. O. D.; HAMAMOTO, A. H.; ABRÃO, T.; PROENÇA, M. L. A game theoretical based system using holt-winters and genetic algorithm with fuzzy logic for DoS/DDoS mitigation on SDN networks. *IEEE Access*, v. 5, p. 9485–9496, 2017. ISSN 2169-3536. DOI: 10.1109/ACCESS.2017.2702341.

[42] CARVALHO, L. F.; Abrão, T.; MENDES, L. de S.; ProenÇa Jr., M. L. An ecosystem for anomaly detection and mitigation in software-defined networking. *Expert Systems with Applications*, 2018. ISSN 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2018.03.027>.

[43] CARVALHO, L. F.; BARBON, S.; MENDES, L. de S.; PROENÇA, M. L. Unsupervised learning clustering and self-organized agents applied to help network management. *Expert Systems with Applications*, v. 54, p. 29 – 47, 2016. ISSN 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2016.01.032>.

[44] PROENÇA, M. J. L.; ZARPELÃO, B. B.; MENDES, L. Anomaly detection using digital signature of network segment aiming to help network management. *Journal of Communication and Information Systems (JCIS)*, v. 23, 08 2008. DOI: 10.14209/jcis.2008.1.

[45] ABREU, F. R.; PIRES, H. D. *Gerência de Redes*. Disponível em: "<http://www.midiacom.uff.br/~debora/redes1/pdf/trab042/SNMP.pdf>".

[46] MAURO, D. R.; SCHMIDT, K. J. *Essentials SNMP*. 2nd. ed. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 2001. ISBN 0-596-00020-0.

[47] PROENÇA, M. L.; FERNANDES, G.; CARVALHO, L. F.; ASSIS, M. V. O.; RODRIGUES, J. J. P. C. Digital signature to help network management using flow analysis. *Netw.*, Wiley-Interscience, New York, NY, USA, v. 26, n. 2, p. 76–94, mar. 2016. ISSN 0028-3045. DOI: 10.1002/nem.1892.

[48] ASSIS, M. V. de; RODRIGUES, J. J.; PROENÇA, M. L. A seven-dimensional flow analysis to help autonomous network management. *Information Sciences*, v. 278, p. 900 – 913, 2014. ISSN 0020-0255. DOI: <https://doi.org/10.1016/j.ins.2014.03.102>.

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

[50] PATCHA, A.; PARK, J.-M. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, v. 51, n. 12, p. 3448 – 3470, 2007. ISSN 1389-1286. DOI: <https://doi.org/10.1016/j.comnet.2007.02.001>.

[51] PERLIN, T. J.; NUNES, R. C.; KOZAKEVICIUS, A. de J. *Detecção de Anomalias em Redes de Computadores através de Transformadas Wavelets*. 2011. 2-15 p. DOI: 10.5335/RBCA.2011.002.

[52] AHMED, M.; MAHMOOD, A. N.; HU, J. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, v. 60, p. 19 – 31, 2016. ISSN 1084-8045.

[53] aO, B. B. Z. *Detecção de Anomalias em Redes de Computadores*. Tese (Doutorado) — Universidade Estadual de Campinas, setembro 2010.

[54] BRASIL, C. G. da Internet no. *Cartilha de Segurança para Internet*. [S.l.]: CERT.br 4th ed., 2012.

[55] ROTHENBERG, C. E.; NASCIMENTO, M. R.; SALVADOR, M. R.; MAGALHÃES, M. F. OpenFlow e redes definidas por software: um novo paradigma de controle e inovação em redes de pacotes. In: *Cadernos CPqD Tecnologia, Campinas*. [S.l.: s.n.], 2010. v. 7, n. 1, p. 65–76.

[56] SDN Resources. [Online]. Disponível em: "<https://www.sdxcentral.com/sdn/definitions/>".

[57] ASSIS, M. V.; NOVAES, M. P.; ZERBINI, C. B.; CARVALHO, L. F.; ABRAO, T.; PROENÇA, M. L. Fast defense system against attacks in software defined networks. *IEEE Access*, PP, p. 1–1, 10 2018. DOI: 10.1109/ACCESS.2018.2878576.

[58] CARVALHO, L. F.; FERNANDES, G.; RODRIGUES, J. J. P. C.; MENDES, L. S.; PROENÇA, M. L. A novel anomaly detection system to assist network management in sdn environment. In: *2017 IEEE International Conference on Communications (ICC)*. [S.l.: s.n.], 2017. p. 1–6. ISSN 1938-1883. DOI: 10.1109/ICC.2017.7997214.

[59] MELTZLER, J.; MELTZLER, A. *The 2015 Guide to SDN and NFV. [Online]*. 2015. Disponível em: "<http://www.nuagenetworks.net/wp-content/uploads/2015/02/2015Ebook-Nuage.pdf>".

[60] Software-Defined Networking (SDN) Definition. [Online]. Disponível em: "<https://www.opennetworking.org/sdn-definition/>".

[61] SINGH, S.; JHA, R. K. A survey on software defined networking: Architecture for next generation network. *J. Netw. Syst. Manage.*, Plenum Press, New York, NY, USA, v. 25, n. 2, p. 321–374, abr. 2017. ISSN 1064-7570. DOI: 10.1007/s10922-016-9393-9.

[62] PROJECT Floodlight. [Online]. Disponível em: "<http://www.projectfloodlight.org/floodlight>".

[63] SOUZA, J. A. M. F. de. *Análise de Sinais*. 2010. Disponível em: "<http://webx.ubi.pt/~felippe/main_pgs/mat_didp.htm>".

[64] LOURENÇO, E. J. *Transformada de Fourier: Teoria como base para Aplicações em Mecânica Celeste*. Dissertação (Mestrado) — Universidade Estadual Paulista, 2014.

[65] LIMA, P. C. de. *Wavelets: Uma Introdução*. 2002. 13-44 p. ICEX - UFMG.

[66] CHUI, C. K. *An Introduction to Wavelets*. [S.l.]: Academic Press, 1992.

[67] DAUBECHIES, I. *Ten Lectures on Wavelets*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1992. ISBN 0-89871-274-2.

[68] SANTOS, D. T. dos. *Compressão de Imagens Usando a Função de Peano e a Transformada Wavelet 1D*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2012.

[69] VILANI, M. T.; SANCHES, L. *Análise de Fourier e Wavelets Aplicada à Temperatura do Ar em Diferentes Tipologias de Ocupação*. 2013. 1340–1346 p. Disponível em: "<http://www.scielo.br/pdf/rbeaa/v17n12/v17n12a13.pdf>".

[70] LOTZE, T.; SHMUELI, G.; MURPHY, S. *A Wavelet-based Anomaly Detector for Early Detection of Disease Outbreaks*. [S.l.]: Pittsburgh, 2006. Disponível em: "<http://web.engr.oregonstate.edu/~wongwe/workshops/icml2006/papers/lotze.pdf>".

[71] GONZALEZ, R.; WOODS, R. *Processamento Digital De Imagens*. 3rd. ed. [S.l.]: ADDISON WESLEY BRA, 2011. v. 3. ISBN 9788576054016.

[72] LU, W.; TAVALLAEE, M.; GHORBANI, A. A. Detecting network anomalies using different wavelet basis functions. In: *Communication Networks and Services Research Conference, 2008. CNSR 2008. 6th Annual*. [S.l.: s.n.], 2008. p. 149–156. DOI: 10.1109/CNSR.2008.75.

[73] LIMTHONG, K.; WATANAPONGSE, P.; KENSUKE, F. A wavelet-based anomaly detection for outbound network traffic. In: *8th Asia-Pacific Symposium on Information and Telecommunication Technologies*. [S.l.: s.n.], 2010. p. 1–6.

[74] GAUCINISKI, J. *Estudo das Transformadas Wavelets para Utilização em Reconhecimento e Classificação de Comandos de Voz*. Dissertação (Mestrado) — Universidade de Santa Cruz do Sul, 2009.

[75] KAPLAN, I. *The Daubechies D4 Wavelet Transform*. 2001. Disponível em: "<http://bearcave.com/misl/misl_tech/wavelets/daubechies/index.html>".

[76] MELLO, M. P.; PETERNELLI, L. A. *Conhecendo o R Uma Visão mais que Estatística*. 1st. ed. [S.l.]: Editora UFV, 2013.

[77] TEAM, R. C. *R: A language and environment for statistical computing. R Foundation for Statistical Computing*. 2018. Disponível em: "<https://www.R-project.org/>".

[78] ALDRICH, E. *Package wavelets*. 2015. Disponível em: "<https://cran.r-project.org/web/packages/wavelets/wavelets.pdf>".

[79] Open Networking Foundation. *The openflow specification version 1.0.0*. 2009. Available: "<https://3vf60mmveq1g8vzn48q2o71a-wpengine.netdna-ssl.com/wp-content/uploads/2013/04/openflow-spec-v1.0.0.pdf>".

[80] RAZALI, N. M.; WAH, Y. B. Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests. *Journal of Statistical Modeling and Analytics*, v. 2, n. 1, p. 21–33, 2011.

[81] JORDAN, S. *The 68-95-99.7 Rule for Normal Distributions*. 2014. Disponível em: <http://faculty.atu.edu/sjordan/College\%20Math\%20notes/5.9\%20The\%2068-95-99.7\%20Rule.pdf>.

[82] PENA, E. H. M.; BARBON, S.; RODRIGUES, J. J. P. C.; PROENÇA, M. L. Anomaly detection using digital signature of network segment with adaptive arima model and paraconsistent logic. In: *2014 IEEE Symposium on Computers and Communications (ISCC)*. [S.l.: s.n.], 2014. p. 1–6. ISSN 1530-1346. DOI: 10.1109/ISCC.2014.6912503.

[83] ADANIYA, M. H. A. C.; LIMA, M. F.; RODRIGUES, J. J. P. C.; ABRÃO, T.; PROENÇA, M. L. Anomaly detection using dsns and firefly harmonic clustering algorithm. In: *2012 IEEE International Conference on Communications (ICC)*. [S.l.: s.n.], 2012. p. 1183–1187. ISSN 1938-1883. DOI: 10.1109/ICC.2012.6364088.

[84] PROENÇA, M. L. J.; SAKURAY, F.; COPPELMANS, C.; BOTTOLI, M.; ALBERTI, A. M.; MENDES, L. de S. A pratical approach for automatic generation of network segment traffic baselines. *Journal of Communication and Information Systems*, v. 20, n. 1, Jun. 2015. DOI: 10.14209/jcis.2005.2. Disponível em: <https://jcis.sbrt.org.br/jcis/article/view/307>.

[85] HOAGLIN, D. C. John w. tukey and data analysis. p. 311–318, 2003.

[86] CARVALHO, L. F. *Um ecossistema para detecção e mitigação de anomalias em redes definidas por software*. Tese (Doutorado) — Universidade Estadual de Campinas, agosto 2018.

[87] OLIVEIRA, R. L. S. de; SCHWEITZER, C. M.; SHINODA, A. A.; PRETE, L. R. Using mininet for emulation and prototyping software-defined networks. In: *2014 IEEE Colombian Conference on Communications and Computing (COLCOM)*. [S.l.: s.n.], 2014. p. 1–6. DOI: 10.1109/ColComCon.2014.6860404.

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

[89] PROJECT Floodlight. [Online]. Disponível em: "<http://www.projectfloodlight.org/floodlight>".

[90] DOVER, J. M. *A Denial of Service Attack Against the Open Floodlight SDN Controller. [Online]*. Disponível em: "<http://dovernetworks.com/wp-content/uploads/2013/12/OpenFloodlight-12302013.pdf>".

[91] HAMAMOTO, A. H. *Detecção de anomalias em tráfego de rede usando algoritmos genéticos e lógica fuzzy com análise de fluxos IP*. Dissertação (Mestrado) — Universidade Estadual de Londrina, 79p, MARÇO 2017. Dissertação de Mestrado, Londrina, 2017.

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

[93] XIAO, P.; QU, W.; QI, H.; LI, Z. Detecting ddos attacks against data center with correlation analysis. *Computer Communications*, v. 67, p. 66 – 74, 2015. ISSN 0140-3664. DOI: <https://doi.org/10.1016/j.comcom.2015.06.012>.

[94] KOC, C. K.; CHEN, G.; CHUI, C. K. Complexity analysis of wavelet signal decomposition and reconstruction. *IEEE Transactions on Aerospace and Electronic Systems*, v. 30, n. 3, p. 910–918, Jul 1994. ISSN 0018-9251. DOI: 10.1109/7.303759.

[95] DALMAZO, B. L.; PERLIN, T.; NUNES, R. C.; KOZAKEVICIUS, A. d. J. *Filtro de Alarmes de Anomalias através de Wavelets*. 2009. 85-100 p.

TRABALHOS PUBLICADOS PELO AUTOR

Trabalhos publicados:

1. ASSIS, M. V.; NOVAES, M. P.; ZERBINI, C. B.; CARVALHO, L. F.; ABRAO, T.; PROENÇA, M. L. Fast defense system against attacks in software defined networks. IEEE Access, ISSN: 2169-3536. V. 6, p. 69620-69639, 10 2018. DOI: 10.1109/ACCESS.2018.2878576. (**Qualis B3**).

2. ZERBINI, C. B.; CARVALHO, L. F.; ABRAO, T.;PROENÇA, M. L. Wavelet against Random Forest for Anomaly Mitigation in Software-defined Networking. Applied Soft Computing, ISSN: 1568-4946. V. 80, p. 138 - 153, 2019. DOI: 10.1016/j.asoc.2019.02.046 (**Qualis A1**).

Trabalhos em avaliação:

1. (Submetido) ZERBINI, C. B.; NOVAES, M. P.; CARVALHO, L. F.; BARBON, S.; PROENÇA, M. L. Wavelets and Naive Bayes for Intrusion Detection and Mitigation in Software-Defined Network Environment, 2019. Transactions on Emerging Telecommunications Technologies, (**Qualis B1**).