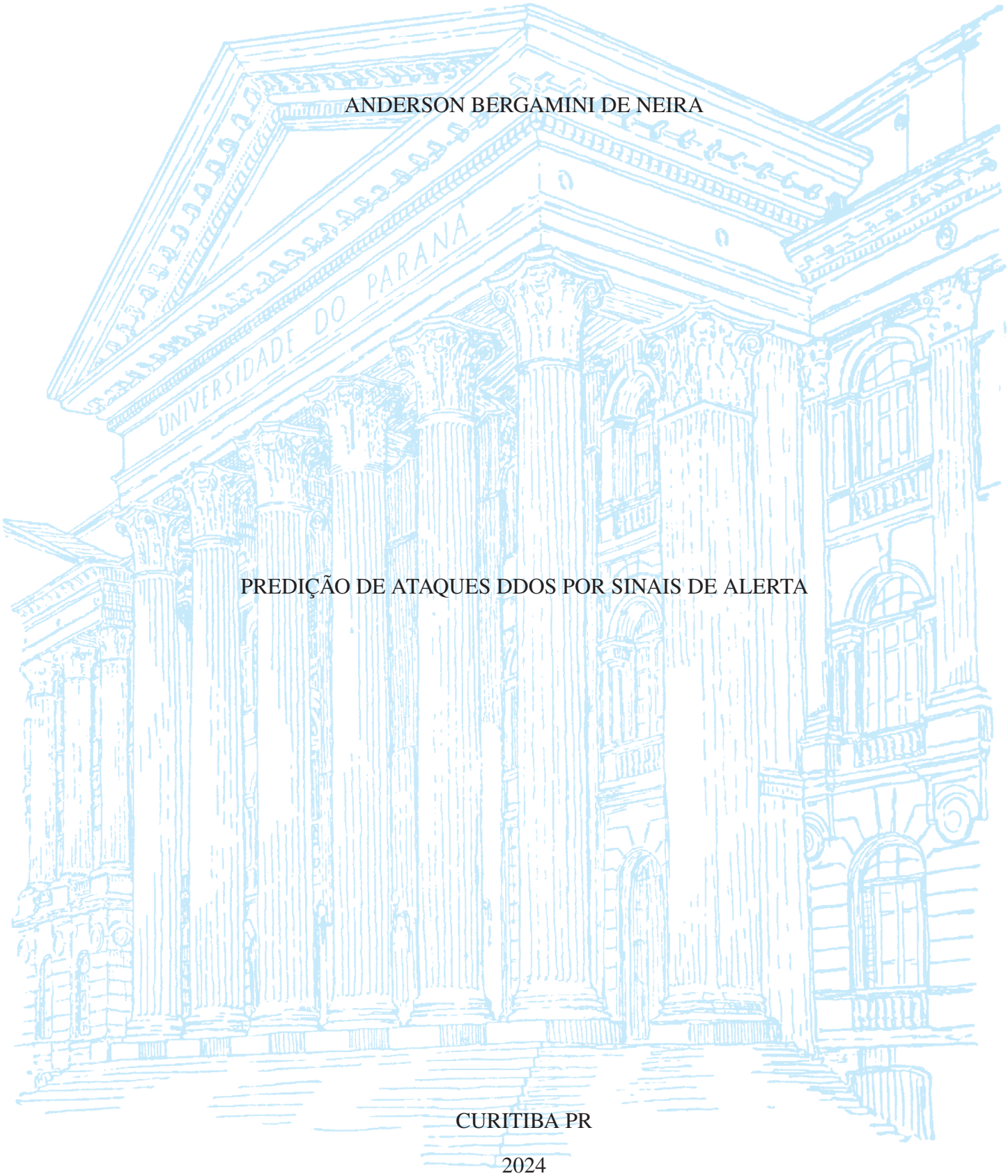
UNIVERSIDADE FEDERAL DO PARANÁ

ANDERSON BERGAMINI DE NEIRA



PREDIÇÃO DE ATAQUES DDOS POR SINAIS DE ALERTA

CURITIBA PR

2024

ANDERSON BERGAMINI DE NEIRA

PREDIÇÃO DE ATAQUES DDOS POR SINAIS DE ALERTA

Tese apresentada como requisito parcial à obtenção do grau de Doutor em Ciência da Computação no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Prof. Dra. Michele Nogueira Lima.

CURITIBA PR

2024

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

Neira, Anderson Bergamini de
Predição de ataques DDOS por sinais de alerta / Anderson Bergamini de Neira. – Curitiba, 2024.
1 recurso on-line : PDF.

Tese (Doutorado) - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática.

Orientador: Michele Nogueira Lima

1. Aprendizado do computador. 2. Redes de computadores - segurança. I. Universidade Federal do Paraná. II. Programa de Pós-Graduação em Informática. III. Lima, Michele Nogueira. IV . Título.

Bibliotecário: Leticia Priscila Azevedo de Sousa CRB-9/2029

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **ANDERSON BERGAMINI DE NEIRA** intitulada: **Predição de Ataques DDoS por Sinais de Alerta**, sob orientação do Prof. Dr. MICHELE NOGUEIRA LIMA, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de doutor está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 18 de Novembro de 2024.

Assinatura Eletrônica
19/11/2024 16:39:49.0
MICHELE NOGUEIRA LIMA
Presidente da Banca Examinadora

Assinatura Eletrônica
20/11/2024 00:27:54.0
LEOBINO NASCIMENTO SAMPAIO
Avaliador Externo (UNIVERSIDADE FEDERAL DA BAHIA)

Assinatura Eletrônica
20/11/2024 10:07:47.0
ALDRI LUIZ DOS SANTOS
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
21/11/2024 00:19:35.0
DANIEL MACEDO BATISTA
Avaliador Externo (UNIVERSIDADE DE SÃO PAULO)

Assinatura Eletrônica
19/11/2024 16:45:25.0
AURORA TRINIDAD RAMIREZ POZO
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Aos meus entes queridos, amigos e colegas, dedico este trabalho como um pequeno passo no contínuo avanço do conhecimento científico.

AGRADECIMENTOS

Agradeço a Deus pelo dom da vida e pela força de acordar todos os dias. Agradeço a toda a minha família, em especial a minha esposa Juliana, a minha mãe Eliza e meu pai Nelson por toda a ajuda, encorajamento, suporte, força e apoio ao longo de toda a jornada. Agradeço imensamente a Prof. Dra. Michele Nogueira Lima pela oportunidade, ensinamentos e paciência ao longo de toda a jornada. A todos do grupo Ciência de Segurança Computacional e de Segurança de Redes (CCSC), em especial os coautores de artigos e a todos do Grupo Núcleo de Redes Sem Fio e Redes Avançadas (NR2) por toda a ajuda e companheirismo ao longo da jornada. Agradeço aos professores Aldri Santos, Aurora Pozo, Daniel Batista e Leobino Sampaio pela participação nas bancas e por todos os comentários e críticas construtivas que contribuíram para a evolução deste trabalho. Por fim, agradeço imensamente a todos que, com sua colaboração, possibilitaram a conclusão desta tese.

RESUMO

Os ataques cibernéticos evoluem continuamente. Dentre eles, o ataque de negação de serviço distribuído (do inglês, *Distributed Denial of Service* — DDoS) é um dos mais comprometedores. Após seu efetivo lançamento, os administradores de rede são surpreendidos pelo rápido crescimento no consumo de recursos, sendo necessário coibi-los antes que eles cheguem a esta fase. A tarefa de identificar indícios da preparação dos ataques é desafiadora, pois a preparação dos ataques evitam impactar os atributos do tráfego de rede. Além disso, devido ao grande desbalanceamento dos dados, a preparação dos ataques é frequentemente ignorada. Entretanto, mesmo sem mudanças claras no comportamento, é possível observar sinais precoces de alerta por meio de novas características e utilizá-las para predizer ocorrências futuras. Apesar do potencial dos sinais precoces de alerta, sua investigação é incipiente na identificação de indícios da preparação dos ataques e, particularmente, dos ataques DDoS. À luz deste problema, este trabalho advoga pela hipótese de identificar a preparação dos ataques por intermédio da criação e uso de sinais de alerta. Com o vasto contexto de aplicação das redes de computadores e das ameaças cibernéticas, são necessárias soluções que se adaptem aos diferentes objetivos e contextos. Assim, este trabalho propõe uma abordagem denominada de **E**ngenharia de **S**inais **P**recoces de **A**lerta (ESPA), a qual processa os dados do tráfego de rede sob o objetivo de geração de sinais precoces de alerta. A proposta deste trabalho cria características capazes de representar mudanças no tráfego da rede e usa aprendizado de máquina para automatizar a predição dos ataques DDoS a partir de novas características. Este trabalho segue quatro casos de usos para demonstrar o poder de customização e a evolução da proposta, além de validar a hipótese. Inicialmente, a abordagem ESPA utiliza o aprendizado de máquina supervisionado para analisar as características criadas, predizendo ataques DDoS com acurácias próximas a 100%. Para predizer diferentes tipos de ataques DDoS, a abordagem ESPA evolui com o uso de técnicas de aprendizado de máquina não supervisionado. Outra característica alvo de evolução foi a configuração autônoma dos modelos de aprendizado de máquina. Assim, a abordagem proposta usa modelos configurados manual ou autonomamente pela própria abordagem. Para aumentar ainda mais a confiabilidade e produzir predições mais assertivas sem depender de treinamentos e rótulos, este trabalho evoluiu abordagem proposta para usar um *ensemble* de técnicas de aprendizado de máquina. Com o *ensemble*, a abordagem ESPA adapta-se à evolução do tráfego de rede produzindo predições mais confiáveis. Por fim, este trabalho evoluiu abordagem proposta visando coletar diferentes atributos do tráfego de rede, em diferentes modos de coleta, utilizá-los de diferentes formas na engenharia de sinais e realizar a seleção de características sem rótulos.

Palavras-chave: DDoS, predição, cooperação, aprendizado de máquina, sinais precoces de alerta, cibersegurança, segurança de redes, sistema distribuído, indicadores estatísticos.

ABSTRACT

Cyberattacks are constantly evolving. Among them, the distributed denial of service (DDoS) attack is one of the most damaging. After its effective launch, network administrators are surprised by the rapid growth in resource consumption, and it is necessary to prevent it before it reaches this stage. Identifying signals of attack preparation is challenging since attacks avoid impacting network traffic attributes. In addition, due to the significant data imbalance, attack preparation is often neglected. However, even without apparent changes in behavior, it is possible to observe early warning signals through new characteristics and use them to predict future occurrences. Despite the potential of early warning signals, their investigation is incipient in identifying signals of attack preparation, particularly DDoS attacks. Regarding this problem, this work advocates the hypothesis of identifying attack preparation by creating and using warning signals. Given the vast application context of computer networks and cyber threats, solutions that adapt to different objectives and contexts are needed. Thus, this work proposes an approach called Early Warning Signals Engineering, which processes network traffic data to generate early warning signals. The proposal of this work creates features representing changes in network traffic and uses machine learning to automate the prediction of DDoS attacks from new features. This work follows four use cases to demonstrate the customization power and evolution of the proposal, in addition to validating the hypothesis. The ESPA approach initially uses supervised machine learning to analyze the created features, predicting DDoS attacks near 100% accuracy. The ESPA approach evolves with unsupervised machine learning techniques to predict different DDoS attacks. Another feature that has been the target of evolution is the autonomous configuration of machine learning models. Thus, the proposed approach uses models configured manually or autonomously by the approach itself. This work has evolved the proposed approach to utilize an ensemble of machine learning techniques to increase trustworthiness and produce more assertive predictions without relying on training and labels. With the ensemble, the ESPA approach adapts to the network traffic evolution, creating more reliable predictions. Finally, this work has evolved the proposed approach to collect different attributes of network traffic in different collection modes, use them in different ways in signal engineering, and perform feature selection.

Keywords: DDoS, prediction, cooperation, machine learning, early warning signals, cybersecurity, network security, distributed system, leading indicators.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ACRÔNIMOS

AC-1	Autocorrelação de *lag-1*
AC-2	Autocorrelação de *lag-2*
AC-3	Autocorrelação de *lag-3*
AC-τ	Autocorrelação de *lag-τ*
Adaboost	*Adaptive Boosting*
AIC	*Akaike's Information Criterion*
AM	Aprendizado de Máquina
ANN	*Artificial Neural Network*
ANOVA	*Analysis of Variance*
AUC	*Area Under the Curve*
AutoML	*Automated Machine Learning*
AR	*Autoregressive Model*
ARIMA	*Autoregressive Integrated Moving Average*
ANS	*Autonomous system number*
BBC	*British Broadcasting Corporation*
BGP	*Border Gateway Protocol*
CAPTCHA	*Completely Automated Public Turing Test to Tell Computers and Humans Apart*
CCPA	*California Consumer Privacy Act*
CCSC	*Computational and network seCurity SCience*
CDbw	C*omposed Density Between and Within Clusters*
CICDDoS2019	*DDoS Evaluation Dataset*
CISA	*Cybersecurity and Infrastructure Security Agency*
CM	Ciclos maliciosos
CN	Ciclos normais
CNN	*Convolutional Neural Network*
CPJ	*Committee to Protect Journalists*
CPU	*Central Process Unit*
CTU-13	*Czech Technical University Datasets*
CV	Coeficiente de Variação
CVE	*Common Vulnerabilities and Exposures*
DBN	*Deep Belief Network*
DNN	*Deep Neural Networks*
DDoS	*Distributed Denial of Service*
DNS	*Domain Name System*

DoS	*Denial of Service*
EM	*Expectation-Maximization*
EUA	Estados Unidos da América
FN	Falso Negativo
FP	Falso Positivo
GAN	*Generative Adversarial Network*
GDPR	*General Data Protection Regulation*
GRU	*Gated Recurrent Unit*
HMM	*Hidden Markov Model*
HPO	*Hyperparameter Optimization*
HTTP	*Hypertext Transfer Protocol*
HTTPS	*Hypertext Transfer Protocol Secure*
IA	Inteligência Artificial
IANA	*Internet Assigned Numbers Authority*
ICA	*Independent component analysis*
ICMP	*Internet Control Message Protocol*
IDS	*Intrusion Detection System*
IoT	*Internet of Things*
IP	*Internet Protocol*
k-NN	*k-Nearest Neighbors*
L-BFGS	*Limited Memory BFGS*
LFA	*Link Flooding Attack*
LGPD	Lei Geral de Proteção de Dados
LSTM	*Long Short-term Memory*
LSTM-AE	*Long Short-term Memory Autoencoder*
MAC	*Media Access Control*
MAE	*Mean Absolute Error*
MNasNet	*Mobile Neural Architecture Search Network*
MSE	*Mean Square Error*
NAS	*Neural Architecture Search*
NASNet	*Neural Architecture Search Network*
ONF	*Open Networking Foundation*
P2P	*Peer to Peer*
PCA	*Principal Component Analysis*
PoD	*Ping of Death*
RBF	*Radial Basis Function*
ReLU	*Rectified Linear Unit*
RF	*Random Forest*
RL	*Reinforcement Learning*

RMSE	*Root-mean-square Deviation*
RNN	*Recurrent Neural Network*
ROC	*Receiver Operating Characteristic*
RU	Reino Unido
SARIMA	S*easonal Autoregressive Integrated Moving Average*
SARIMAX	*Seasonal ARIMA with eXogenous Factors*
SARSA	*State-Action-Reward-State-Action*
SCESN	*Recurrent Neural Echo State Network*
SDN	*Software Defined Networks*
SGD	*Stochastic Gradient Descent*
SSH	*Secure Shell*
SIP	*Session Initiation Protocol*
SO	Sistemas Operacionais
SOM	*Self Organizing Map*
SVM	*Support Vector Machine*
TCP	*Transport Control Protocol*
TSVM	*Transductive Support Vector Machines*
UE	União Europeia
UDP	*User Datagram Protocol*
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo
XAI	*Explainable Artificial Intelligence*

LISTA DE SÍMBOLOS

γ	Valor do Davies-Bouldin e S_Dbw
a	Quantidade de Agentes
$A_k(n)$	Transformada de Fourier
$b(x)$	Distância Média ao Cluster Mais Próximo
c	Centro da Base de Dados
c_i	Centro do I-ésimo Cluster
d^2	Distância Euclidiana ao Quadrado
$e(x)$	Distância Média a Todos os Pontos do Cluster ao Qual x Pertence
ews	Quantidade de Sinais Precoces de Alerta
f	Quantidade de Atributos Coletados
F_1	Ponto de Bifurcação Um
F_2	Ponto de Bifurcação Dois
g	Valor do *Skewness*
i	Quantidade de Indicadores Estatísticos
I_k	Periodograma do Segmento K
k	Quantidade de *Clusters*
k_i	I-ésimo Cluster
K	Quantidade de Segmentos Sobrepostos
L	Tamanho do Segmento
max_A	Maior Valor Observado
max'_A	Maior Valor do Novo Intervalo
min_A	Menor Valor Observado
min'_A	Menor Valor do Novo Intervalo
mm	Valor Após a Transformação Min-Max
M_d	Mediana do Atributo d
n_i	Número de Elementos em Cada Cluster
N	Quantidade Total de Observações
p	Valor Após a Transformação de Padronização
Q_1	Primeiro Quartil
Q_3	Terceiro Quartil
rs	Valor Após a Transformação *RobustScaler*
s	Desvio Padrão
S_{fold}	Indicador de Transição Crítica do Tipo *Fold*
S_{hopf}	Indicador de Transição Crítica do Tipo *Hopf*
S_{max}	Power Spectrum Máximo

S_{null}	Indicador de não Transição Crítica
x	Cada Elemento de Um Cluster
x_1	Primeira Observação
x_2	Segunda Observação
x_3	Terceira Observação
x_i	Valor original
$\bar{x}$	Média Aritmética Simples
$X(j)$	Todas as Observações de uma Série Temporal
$\mathscr{X}$	Conjunto de dados que descrevem um problema
y_i	Valor Reconstruído
$\hat{y}$	Complemento da Fórmula do *Kurtosis*
$\hat{y}^U$	Valor do *Kurtosis*
$\mathscr{Y}$	Conjunto de Rótulos Verdadeiros
$W(j)$	Dados da Série Temporal

SUMÁRIO

1 INTRODUÇÃO

Com a variedade de aplicativos e serviços disponíveis na Internet, os usuários têm acesso a diferentes formas de trabalho e entretenimento. Durante a pandemia de COVID-19, a conectividade possibilitou o aprendizado remoto, conferências online, telemedicina e teletrabalho, entre outros. Impulsionado pelos novos hábitos adquiridos durante a pandemia, em 2020, o comércio eletrônico cresceu 42% em relação a 2019, movimentando US$ 813 bilhões (Adobe, 2021). Seguindo a tendência de crescimento, especialistas estimam que a indústria de jogos digitais movimentou US$ 187 bilhões em 2023, ultrapassando a marca de 3 bilhões de jogadores, um crescimento de 6,3% em relação a 2022 (Wijman, 2023). Em 2023, a cada minuto o serviço da Google recebeu em média 6,3 milhões de pesquisas, usuários de serviços de vídeos ao vivo assistiram a 43 anos de conteúdo e 360 mil *tweets* foram enviados (Domo, 2023). Esses dados reforçam a importância da computação e das redes nas atividades cotidianas.

Apesar dos benefícios proporcionados pela computação, é necessário tomar cuidados para evitar prejuízos. Ataques de negação de serviço (do inglês, *Denial of Service* - DoS), *man-in-the-middle*, *phishing* e ataques baseados em *malware* são alguns tipos de ataques conhecidos (Bendovschi, 2015; Biju et al., 2019). Entre as ameaças cibernéticas, o ataque de negação de serviço distribuído (do inglês, *Distributed Denial of Service* - DDoS) é um dos mais perigosos (He et al., 2017; Luong et al., 2020; Jyoti e Behal, 2021). No segundo semestre de 2022, especialistas da empresa Netscout identificaram 6.797.959 ataques DDoS, um aumento de 13% em relação ao primeiro semestre de 2022 (Hummel et al., 2023). Especialistas da empresa Zayo identificaram que a quantidade de ataques aumentou 387% entre o primeiro e segundo trimestre de 2023 (Zayo, 2023). As plataformas que oferecem ataques DDoS como um serviço a preços baixos contribuem para aumentar o número de ataques. É possível contratar um ataque a um custo de US$ 100,00 por dia (DDoS-Stress, 2023) ou US$ 10,00 por hora (Armor, 2018).

Qualquer organização pode ser alvo de ataques DDoS, de serviços governamentais a usuários domésticos (Mahjabin et al., 2017; CISA, 2019; Kumar et al., 2022). Durante a pandemia de COVID-19, os atacantes diversificaram suas atividades, atacando também serviços de saúde, e-commerce e serviços educacionais (Vijayan, 2020; Hummel et al., 2021). No segundo trimestre de 2023, serviços relacionados às criptomoedas foram os que mais receberam ataques DDoS na Ásia. Na Europa, o setor de jogos e apostas foi o mais visado pelos atacantes. A América Latina surpreendeu ao ter a indústria de artigos esportivos como a mais atacada (Yoachimik e Pacheco, 2023). No Brasil, um ataque DDoS de seis dias interrompeu o site de uma organização que relata crimes de direitos humanos (CPJ, 2021). A bolsa de valores da Nova Zelândia recebeu um ataque DDoS por dois dias consecutivos (BBC, 2020). No segundo semestre de 2022, os países que mais sofreram ataques DDoS foram os Estados Unidos da América (EUA), México e Espanha. Os países que mais originaram ataques foram a China, Índia e EUA (Hummel et al., 2023).

1.1 MOTIVAÇÃO E PROBLEMA DE PESQUISA

A motivação das pesquisas relacionadas a ataques DDoS consiste nos diferentes tipos de prejuízos resultantes dos mesmos. O tipo de prejuízo mais comum causado pelos ataques DDoS está associado com a interrupção dos serviços (sistemas, servidores ou redes). Em 2014, a empresa de consultoria Gartner estimou que o custo médio de interrupção de serviços é de US$ 5.600 por minuto (Lerner, 2014). Em 2020, especialistas reforçaram esta estimativa mostrando que uma hora de interrupção poderia custar cerca de US$ 300.000 (ITIC, 2020). Mesmo com

esses valores de referência, não é trivial quantificar os prejuízos relacionados à interrupção do serviço. Em 2010, uma companhia aérea teve prejuízo superior a US$ 15 milhões devido a onze dias de indisponibilidade no sistema (Winterford, 2010). Em agosto de 2023, sistemas hospitalares dos estados americanos da Califórnia, Texas, *Connecticut*, *Rhode Island* e Pensilvânia sofreram ataques cibernéticos que interromperam cirurgias eletivas, consultas ambulatoriais e outros serviços médicos (SecurityWeek, 2023). Nestes casos, o prejuízo é imensurável devido à quantidade de pessoas que deixaram de receber os devidos cuidados.

O segundo tipo de prejuízo está relacionado com o consumo excessivo de recursos, aumentando o custo dos serviços. Nesses casos, o atacante não deseja interromper a operação do serviço, e sim consumir recursos. Quanto mais recursos os atacantes consomem, mais recursos os administradores precisam alocar, aumentando o custo operacional do serviço (Singh et al., 2014). Por fim, um tipo de prejuízo cada vez mais comum é a extorsão (Ilascu, 2022). Neste caso, os atacantes requerem o pagamento de resgates para interromper ou não lançar ataques DDoS.

Os administradores de rede têm pouco tempo para evitar os prejuízos causados por ataques DDoS. Após o atacante lançar efetivamente o ataque, os administradores de rede são surpreendidos pelo consumo de recursos, prejudicando o acesso dos usuários reais. A quantidade limitada de tempo para responder aos ataques DDoS são influenciados por dois motivos. Os ataques têm evoluído constantemente, atingindo grandes volumes de tráfego rapidamente. É cada vez mais comum que os ataques DDoS atinjam terabits por segundo (AWS-Shield, 2020; Menscher, 2020; Ilascu, 2022; Toh et al., 2022). Em 2021, o recorde de maior ataque reportado foi quebrado duas vezes. Primeiramente, o maior ataque atingiu 17 milhões de solicitações por segundo. Esse ataque foi lançado por mais de 20.000 dispositivos infectados por uma variante do Mirai[1] e tinha como alvo empresas do setor financeiro (Yoachimik, 2021). Este foi superado por um ataque que atingiu 21,8 milhões de solicitações por segundo (Marrow e Stolyarov, 2021). A empresa Yandex, uma gigante tecnológica russa alvo do ataque, não divulgou detalhes sobre esse ataque. Em 2022, a Microsoft Corporation foi alvo de um ataque DDoS que, em apenas um minuto, atingiu 3,47 terabits por segundo. O ataque baseado no Protocolo de Datagrama de Usuário (do inglês, *User Datagram Protocol* — UDP) foi originado por cerca de 10.000 fontes em diferentes países (Toh et al., 2022). Em fevereiro de 2023, a empresa Cloudflare identificou um dos maiores ataques já reportados, alcançando 71 milhões de requisições por segundo. Esse ataque foi originado por 30.000 fontes. Além disso, esse ataque apresentou uma propriedade perigosa ao atingir cerca de 65 milhões de requisições por segundo em menos de 30 segundos após o seu lançamento (Yoachimik et al., 2023). Em outubro de 2023, a Google reportou que foi alvo do maior ataque DDoS conhecido. Esse ataque foi baseado na multiplexação de fluxo e atingiu 398 milhões de requisições por segundo. Esse ataque durou cerca de dois minutos e, em menos de 30 segundos, atingiu o pico de requisições (Kiner e April, 2023).

Aliado à evolução dos ataques, os mecanismos de defesa demandam tempo para mitigá-los. Alocar mais recursos, bloquear possíveis atacantes ou migrar a vítima são ações que demandam tempo para amenizar ou mitigar os ataques. Especialistas da empresa Corero Network Security verificaram que empresas com 0,1% de participação de mercado podem perder até 240 mil dólares em vendas e 2 milhões de dólares em oportunidade de converter vendas se os ataques DDoS demorarem mais de 60 segundos para serem mitigados (Corero, 2021). Outro ponto que atrasa a resposta aos ataques é a necessidade de detecção. Para que a detecção dos ataques DDoS funcione corretamente, é necessário que o ataque apresente sinais de degradação do serviço. E

[1]O Mirai é um *malware* dedicado a infectar dispositivos da Internet das Coisas (IoT), convertendo-os em uma *botnet*. O Mirai utiliza senhas padrão ou vulneráveis para comprometer aparelhos como câmeras e roteadores. *Botnets* baseadas no Mirai foram utilizadas em diversos ataques significativos desde 2016, quando seu código-fonte foi disponibilizado ao público.

mesmo após a detecção do ataque, aliviar os prejuízos não é uma tarefa trivial (Gupta e Dahiya, 2021; Frost e Sullivan, 2022).

Devido à periculosidade do ataque DDoS, ao grau de dificuldade em evitar os prejuízos, e ao tempo limitado para lidar com os ataques, cada segundo é crucial no combate aos ataques DDoS. Gupta e Badve (2017) e Santos et al. (2017) argumentam que a melhor forma de prevenir danos causados pelos ataques DDoS é interromper/mitigar o ataque antes que ele ocorra. Para isso é necessário antecipar e mitigar o ataque antes que o atacante lance o ataque. Assim, o problema de pesquisa tratado nesta tese é:

Se existirem sinais da preparação dos ataques DDoS, como identificá-los?

Os atacantes estão se tornando cada vez mais habilidosos em mascarar a preparação dos ataques (Antonakakis et al., 2017). As características tradicionalmente usadas na detecção dos ataques são pouco representativas para o combate antecipado dos ataques DDoS. Além disso, devido ao grande desbalanceamento dos dados, a preparação dos ataques pode ser confundida com o tráfego normal da rede. Isso ocorre, pois, a fase de preparação produz um volume de tráfego de rede ínfimo comparado ao tráfego do ataque, postergando a identificação do ataque.

Para definir características representativas para a predição dos ataques DDoS, este trabalho explora a teoria dos sinais precoces de alerta. Compreender a dinâmica dos sistemas e aprimorar a capacidade de previsão de eventos futuros é essencial para a humanidade (Drake e Griffen, 2010). A literatura dos sinais precoces de alerta é uma linha de pesquisa capaz de fornecer evidências sobre a dinâmica dos sistemas observáveis, podendo antecipar eventos futuros. Contudo, predizer corretamente eventos futuros ainda é um desafio complexo (Bury et al., 2020; Proverbio et al., 2022). Portanto, para antecipar os ataques e proporcionar mais tempo para os responsáveis lidarem com os ataques DDoS, a hipótese que norteia este trabalho é:

É possível identificar os sinais da preparação dos ataques usando a teoria dos sinais precoces de alerta.

A literatura embasa essa hipótese, pois identificar sinais de eventos ainda não realizados é estudada em áreas como a previsão do tempo (Webster, 2013), predição de terremoto (Geller, 1997), predição do mercado de ações (Iacomin, 2015) e previsão de ações de adversários dos Estados Unidos da América (Tingley, 2021). Para a área de segurança de sistemas de computadores, Abdlhamed et al. (2017) apresentam a predição como uma maneira de fornecer evidências da preparação de um ataque antes que ele efetivamente comece. Embora predizer ataques não seja uma tarefa trivial, Husák et al. (2019) apresentam alguns estudos que demonstram a viabilidade de propor soluções para a predição de ameaças cibernéticas. Apesar da existência de trabalhos relacionados com a predição de ameaças cibernéticas, ainda há oportunidades nesta área de pesquisa criando soluções mais acuradas, customizáveis, autônomas e especializadas em predizer ataques DDoS.

1.2 OBJETIVO DA PESQUISA

Para validar a hipótese, este trabalho tem dois objetivos, o primeiro é fornecer uma pesquisa abrangente da literatura e o segundo é propor uma solução para predizer ataques DDoS. Este trabalho apresenta os resultados de uma extensa e rigorosa revisão bibliográfica sobre predição de ataques DDoS, classificando estudos e apresentando uma visão organizada

das soluções existentes para a predição de ataques DDoS. A classificação considera diferentes aspectos para construir suas ramificações, proporcionando diferentes visões sobre o problema. Assim, os pesquisadores podem usar a classificação como um atalho para identificar os esforços relacionados à predição de ataques DDoS. Este trabalho também destaca questões em aberto. Abordá-las revelará linhas de pesquisa total ou parcialmente inexploradas, evoluindo a área de predição de ataques DDoS. Assim, pesquisadores podem usar essas lacunas para propor estudos com o intuito de resolver o problema de predição de ataque DDoS.

O segundo objetivo deste trabalho é utilizar as questões em aberto para inspirar uma solução capaz de predizer ataques DDoS. Utilizando as lições aprendidas com a revisão bibliográfica, este trabalho apresenta uma abordagem denominada de **E**ngenharia de **S**inais **P**recoces de **A**lerta (ESPA) a fim de sistematizar a predição de ataques DDoS embasada na teoria dos sinais precoces de alerta. A engenharia de sinais compreende o processamento dos dados do tráfego de rede sob a perspectiva da teoria dos sinais precoces de alerta. Os sinais precoces de alerta são gerados ao aplicar indicadores estatísticos como a variância, o *skewness* e o *kurtosis* sobre o tráfego de rede. O resultado dessa ação cria características que podem representar futuras mudanças no comportamento do sistema (Scheffer, 2009). Assim, a abordagem ESPA cria características com base no tráfego de rede que suportem a predição dos ataques DDoS. O aprendizado de máquina é uma alternativa para consumir as novas características do tráfego de rede e automatizar a predição. Deste modo, o objetivo da abordagem ESPA é proporcionar mais tempo para os administradores de rede lidarem com os ataques e evitarem prejuízos. Além disso, este trabalho projetou a abordagem ESPA para ser customizável. Portanto, a abordagem proposta pode adaptar-se a diferentes objetivos e casos de uso.

1.3 CONTRIBUIÇÕES

As contribuições deste trabalho concentram-se em dois grupos, as contribuições relacionadas à revisão bibliográfica e as relacionadas com a abordagem ESPA. A revisão bibliográfica possui três contribuições, sendo elas: 1) Identificar os estudos que predizem ataques DDoS; 2) Propor uma visão organizada (classificação) desses estudos com base nos aspectos principais; e 3) Apresentar uma análise aprofundada de questões em aberto que podem auxiliar na evolução das soluções de predição. Para tanto, este trabalho analisou 2.682 estudos e identificou 27 estudos diretamente relacionados à predição de ataques DDoS. A visão organizada desses estudos apresenta aspectos relevantes que devem ser considerados no planejamento de novas soluções de predição de ataques DDoS. Tempo, arquitetura, metodologia e os tipos de dados usados pelos estudos são aspectos comuns das soluções de predição de ataque DDoS. Além disso, a pesquisa apresenta uma análise expressiva das questões em aberto. Essas questões apresentam oportunidades de pesquisa que podem evoluir o estado da arte da área de predição de ataques DDoS e reduzir os prejuízos relacionados aos ataques. Por fim, a revisão bibliográfica foi publicada no periódico *Computer Networks* em fevereiro de 2023 (Neira et al., 2023d).

O segundo grupo apresenta as contribuições relacionadas com a abordagem ESPA. A primeira contribuição é a abordagem ESPA. Ela permite ao administrador de rede usar configurações variadas para adaptar a proposta a cenários distintos, automatizando a predição utilizando diferentes algoritmos de aprendizado de máquina. Outra contribuição é o planejamento e a execução dos experimentos realizados para avaliar a abordagem proposta. Além de avaliar a abordagem ESPA, o molde dos experimentos pode ser utilizado para avaliar outras soluções. A definição dos trabalhos futuros também é uma contribuição. Eles evoluem a abordagem ESPA e podem auxiliar na evolução da literatura de predição de ataques.

Para validar a hipótese, este trabalho seguiu quatro casos de uso. Os casos de uso avaliam a predição de ataques DDoS realizada pela abordagem ESPA sob diferentes circunstâncias e cenários. Os casos de uso foram projetados para predizer os ataques DDoS adaptando-se aos diferentes cenários. Assim, as versões da abordagem ESPA utilizada nos casos de uso são diferentes e complementares. Isso proporcionou a melhoria e a evolução da proposta. Por fim, cada caso de uso possui um conjunto de experimentos que auxiliam na verificação da hipótese e suportam as contribuições, conclusões e evoluções deste trabalho.

No primeiro caso de uso, a abordagem ESPA foi projetada para ser a primeira solução da literatura a predizer ataques DDoS cooperativamente. Neste caso, a abordagem ESPA coleta distribuidamente o total de pacotes trafegados na rede por segundo. A abordagem ESPA coopera ao aplicar centralizadamente a engenharia de sinais usando indicadores estatísticos sobre o total de pacotes coletados. Ao realizar a engenharia de sinais, a abordagem ESPA cria novas características que formam a base da predição dos ataques DDoS. O *adaptive boosting* (AdaBoost) e a *multilayer perceptron* (MLP), dois algoritmos de aprendizado de máquina supervisionado, foram utilizados para automatizar a predição dos ataques. Os resultados indicam que a abordagem proposta predisse ataques DDoS com até 3 minutos e 55 segundos antes do lançamento do ataque, obtendo uma acurácia de até 99,76%. A verificação da hipótese e as demais contribuições desse caso de uso foram publicadas no periódico IEEE TNSM (Neira et al., 2023c).

O segundo caso de uso diminuiu a dependência de dados rotulados e automatizou a configuração das técnicas de aprendizado de máquina. Além disso, a abordagem ESPA utiliza o total de pacotes e o total de endereços do Protocolo de Internet (do inglês, *Internet Protocol* - IP) usados como origem e como destino dos pacotes. O *skewness*, o *kurtosis* e o coeficiente de variação (CV) são os indicadores estatísticos usados pela engenharia de sinais para gerar as novas características. Para diminuir o uso de dados rotulados, este caso de uso utilizou o *Long Short-Term Memory* (LSTM) *Autoencoder* e a Máquina de Vetores de Suporte de Classe Única (do inglês, *One-Class Support Vector Machine* - One-Class SVM). Esses algoritmos foram escolhidos visto que eles não demandam dados rotulados para realizar a predição dos ataques DDoS. O Autokeras (Jin et al., 2019), um *framework* que implementa o *Automated Machine Learning* (AutoML) (Feurer et al., 2015, 2019), foi usado para identificar a configuração da rede LSTM *Autoencoder* que maximizaria a predição dos ataques DDoS. Os resultados obtidos indicam que a proposta predisse um ataque DDoS com 31 minutos e 2 segundos de antecedência, atingindo uma acurácia máxima de 98,03%. A evolução da literatura foi publicada em anais de conferências internacionais (Silva et al., 2022a; Brito et al., 2023a), de simpósios (Silva et al., 2022b; dos Santos Lima et al., 2023, 2024) e workshop (Brito et al., 2023b).

No terceiro caso de uso, a abordagem ESPA realiza a predição de ataques DDoS de dia zero (do inglês, *zero-day*) ao não demandar uma fase de treinamento e dados rotulados para a predição dos ataques. A abordagem ESPA evoluiu para processar até 51 atributos do tráfego de rede e gerar 306 novas características que evidenciam os sinais de preparação para ataques DDoS. A Abordagem ESPA prediz os ataques DDoS consumindo as novas características com o aprendizado de máquina não supervisionado. O aprendizado de máquina não supervisionado não requer dados rotulados e pode ser utilizado para a predição de ataques de dia zero. O aprendizado de máquina foi configurado de duas maneiras, manual e autonomamente. Para a versão autônoma, a abordagem ESPA foi equipada com um algoritmo criado especialmente para analisar os dados e sugerir o modelo de aprendizado de máquina não supervisionado ideal para o cenário. É importante destacar que esse algoritmo foi projetado para não utilizar dados rotulados no processo de seleção do modelo de aprendizado de máquina não supervisionado. A seleção de características não supervisionada também foi uma evolução deste caso de uso. Os resultados indicam que a abordagem ESPA melhora a explicabilidade em cenários reais e

apresenta um novo conceito para explicar a previsão com base na visualização de dados. Além disso, a abordagem proposta predisse ataques até 30 minutos antes do início efetivo do ataque e 19 minutos e 11 segundos após o início da infecção, com uma acurácia máxima de 100%. A contribuição relacionada com a explicabilidade da predição dos ataques foi publicada no XXVIII WGRS (Neira et al., 2023a) e no *IEEE Globecom* 2023 (Neira et al., 2023b). Os resultados relacionados com a seleção autônoma do modelo de aprendizado de máquina não supervisionado e com a seleção de características para a predição dos ataques foram aceitas para publicação na conferência internacional IEEE LATINCOM 2024.

O Caso de uso 4 foi projetado para evoluir a proposta, produzindo predições ainda mais confiáveis sem a utilização de dados rotulados e treinamentos. A engenharia de sinais realizada neste caso de uso foi a mesma utilizada no Caso de uso 3. Assim, a abordagem ESPA coleta e processa 51 atributos do tráfego de rede sob a perspectiva de seis indicadores estatísticos. As principais contribuições deste caso de uso originam-se na utilização da Análise de Componentes Principais (do inglês, Principal Component Analysis - PCA) e de um *ensemble* de identificadores de *outliers*. O PCA e os identificadores de *outliers* não demandam dados rotulados para realizar a seleção de características e identificar os *outliers*. Assim, a versão da abordagem ESPA deste caso de uso não demanda dados rotulados para nenhuma parte do processo de predição dos ataques DDoS. O *ensemble* de identificadores de *outliers* foi criado com base em 105 modelos derivados dos algoritmos *One-class* SVM, *One-class* SVM com Gradiente Descendente Estocástico (do inglês, *Stochastic Gradient Descent* - SGD) e o *Elliptic Envelope*. Assim, a contribuição de caso de uso é uma versão ainda mais confiável da abordagem ESPA, menos suscetível a erros baseados em ruídos dos dados e que não demanda uma fase de treinamento. Ao não utilizar treinamento e dados rotulados, a abordagem ESPA possui a capacidade de adaptar-se frente às mudanças do tráfego de rede, atingindo assim o aprendizado online Bitit et al. (2024). A acurácia obtida neste caso de uso variou entre 84,56% e 97,01%. O tempo de predição variou entre 9 minutos e 49 minutos antes do lançamento do ataque. Por fim, as contribuições deste caso de uso serão submetidas ao periódico IEEE *Access*.

1.4 ORGANIZAÇÃO DO TEXTO

Esta tese está estruturada da seguinte forma. O Capítulo 2 detalha o funcionamento dos métodos de ataques DDoS e dos mecanismos de defesa contra ataques DDoS. Além disso, esse capítulo apresenta o aprendizado de máquina e a teoria dos sinais precoces de alerta que são conceitos importantes para este manuscrito. O Capítulo 3 apresenta o resumo, a classificação e questões em aberto pertinentes aos estudos relacionados com a predição de ataques DDoS. O Capítulo 4 detalha a proposta deste trabalho. O Capítulo 5 apresenta a avaliação dividida nos quatro casos de uso. Por fim, o Capítulo 6 apresenta a conclusão e os trabalhos futuros.

2 FUNDAMENTOS

Confidencialidade, integridade e disponibilidade são os princípios-chave da segurança de sistemas de computadores (Avizienis et al., 2004). Os mecanismos de ataque e defesa visam danificar ou proteger esses princípios em sistemas digitais. O ataque DoS é um evento que desabilita ou restringe o funcionamento correto de uma rede ou serviço (Gligor, 1984; Wood e Stankovic, 2002; Gruschka e Luttenberger, 2006), tornando-os inacessíveis para usuários legítimos (CISA, 2019), afetando o princípio da disponibilidade. Esses eventos podem estar relacionados a atacantes que exploram as fraquezas da rede para subverter, obstruir ou suprimir a rede (Wood e Stankovic, 2002). Esses ataques geralmente atingem seu objetivo inundando a infraestrutura do serviço com um fluxo intenso de pacotes (Comer, 2015). Além disso, esses eventos podem ser falhas de software ou hardware, recursos insuficientes, condições ambientais ou qualquer combinação desses fatores (Wood e Stankovic, 2002). Nas origens do ataque DoS, era comum atacantes lançarem o ataque a partir de um único ponto. Esta ação simplificou a mitigação do ataque porque era relativamente simples identificar e bloquear a origem do ataque (Comer, 2015).

Por uma questão de eficácia, os atacantes empregam variações de ataque DoS, como o ataque DDoS. Um dos primeiros ataques DDoS registrados foi contra a Panix, um provedor de acesso à Internet de Nova York em 1996. Neste caso, a Panix foi atacada com uma inundação de pacotes do Protocolo de Controle de Transmissão (do inglês, *Transmission Control Protocol* - TCP) com a *flag* SYN durante vários dias. Especialistas sugerem que esse ataque foi motivado pela instalação de filtros de spam no sistema de e-mail da Panix (CALEM, 1196; Brooks et al., 2022). Desde então, os ataques DDoS evoluíram em frequência, volume e sofisticação (Lee et al., 2008; Bhatia et al., 2018; Zhijun et al., 2020). Um ataque DDoS visa limitar ou obstruir o acesso legítimo aos serviços. Então, os atacantes exploram as fraquezas da infraestrutura ou consomem todos os recursos usando vários agentes comprometidos (Beslin Pajila e Golden Julie, 2020). Este capítulo apresenta conceitos utilizados ao longo deste manuscrito. Primeiramente, a Seção 2.1 apresenta o modo operacional de ataques DDoS. A Seção 2.2 apresenta os tipos de ataque DDoS. A Seção 2.3 apresenta os mecanismos de proteção contra os ataques DDoS. A Seção 2.4 apresenta conceitos relacionados ao aprendizado de máquina. A Seção 2.5 apresenta a teoria dos sinais precoces de alerta. Por fim, a Seção 2.6 resume o capítulo.

2.1 ATAQUE DDOS

A popularização da Internet de alta capacidade e o aumento de dispositivos conectados à rede mundial de computadores propiciaram a evolução do ataque DoS. Os atacantes passaram a controlar[1] as máquinas conectadas à Internet, possibilitando contornar a fragilidade do único ponto de lançamento do ataque. Portanto, o ataque DDoS combina vários dispositivos conectados para atacar um alvo (Peng et al., 2007; Keshariya e Foukia, 2010; Comer, 2015; CISA, 2019). No modo clássico, os ataques DDoS esgotam os recursos de computação da infraestrutura vítima, criando várias conexões de fontes diferentes (Douligeris e Mitrokotsa, 2004). Ataque de inundação de link (do inglês, *Link Flooding Attack* - LFA) é outra maneira de degradar ou

[1]Atacantes exploram vulnerabilidades como senhas padrão ou erros no desenvolvimento dos sistemas dos dispositivos conectados com a Internet visando utilizá-los durante os ataques. Deste modo, os atacantes conseguem controlar remotamente esses dispositivos segundo suas necessidades.

interromper o serviço da vítima, congestionando links críticos para isolar a rede da vítima da Internet (ur Rasool et al., 2020; Xing et al., 2021).

Outra forma de causar danos à vítima é realizando a Negação Econômica da Sustentabilidade (do inglês, *Economic Denial of Sustainability* - EDoS). No EDoS, o objetivo é consumir os recursos da vítima, forçando-a a alocar mais recursos. Esta ação aumenta o custo necessário para manter o serviço em execução (Monge et al., 2017; Vlajic e Slopek, 2014; Singh et al., 2014). A negação de serviço interno da nuvem (do inglês, *Cloud-internal Denial of Service* - CIDoS) consome recursos do servidor usando várias máquinas virtuais hospedadas no dispositivo físico da vítima. Atacantes aumentam a carga de trabalho das máquinas virtuais para consumir mais recursos do hospedeiro e, consequentemente, interromper o serviço (Alarifi e Wolthusen, 2013; Ribin e Kumar, 2019). *Ransom DDoS* (RDDoS) é outra forma de causar danos usando variações do ataque DDoS. Neste caso, os atacantes solicitam o pagamento de resgates para suspender ou não lançar ataques DDoS contra a vítima (Sasaki et al., 2020; Cloudflare, 2021h).

As partes padrão de um ataque DDoS são os atacantes, os dispositivos infectados e a vítima. Um zumbi, robô da web ou simplesmente *bot* é um dispositivo infectado por *malware* conectado à Internet que executa tarefas programadas (Ngo et al., 2020). Enviar correspondência eletrônica em massa (spam), capturar informações confidenciais, *phishing*, fraude de cliques, *keylogging*, disseminar software para mineração de criptomoedas e lançar ataques DDoS são ações realizadas por *bots* (Choi e Lee, 2012; Huang et al., 2014; Amini et al., 2015). Uma rede de robôs ou *botnet* é um grupo de vários *bots* controlados remotamente por atacantes ou *botmasters* (Mane, 2017; Santos et al., 2017). Uma vítima é um servidor ou uma rede de computadores que contém os recursos para o correto funcionamento de um serviço (Salim et al., 2020). Para conduzir um ataque DDoS, um *botmaster* envia comandos à *botnet* para iniciar conexões com a vítima (Comer, 2015). A duração dos ataques DDoS varia de minutos a dias (Gutnikov et al., 2021), podendo atingir terabits por segundo (Menscher, 2020; AWS-Shield, 2020; Ilascu, 2022; Toh et al., 2022) ou até mesmo atingir 398 milhões de solicitações por segundo (Kiner e April, 2023).

A Figura 2.1 ilustra a operação de um ataque DDoS. Um *botmaster* gerencia os *bots* por meio do tráfego de controle, fazendo com que os vários *bots* enviem o tráfego de ataque para a vítima. O *Botmaster* explora vários tipos de fraquezas de diferentes dispositivos conectados à Internet para espalhar seu código malicioso (Bhatia et al., 2018). O alvo pode ser dispositivos com mais recursos, como computadores de mesa, servidores, tablets e smartphones (Wlosinski, 2019), dispositivos com recursos limitados, como dispositivos que compõem a Internet das coisas (do inglês, *Internet of Things* - IoT) (Marzano et al., 2018; Wlosinski, 2019), como câmeras de segurança ou *smart* TVs ou dispositivos de infraestrutura, como roteadores residenciais ou pontos de acesso à rede sem fio. Após a infecção, os atacantes enviam comandos para que seus *bots* executem. Um ataque DDoS ocorre quando o atacante instrui os *bots* a criar conexões com a infraestrutura vítima para consumir todos os recursos disponíveis.

A literatura apresenta três arquiteturas para atacantes controlarem as *botnets*, a arquitetura centralizada, ponto a ponto (do inglês, *peer-to-peer* - P2P) e híbrida (il Jang et al., 2009; Karim et al., 2014; Gupta e Dahiya, 2021). Em uma arquitetura centralizada, o atacante se comunica com toda a *botnet* por meio de um ponto central conhecido como servidor de comando e controle (do inglês *command and control* - C&C). Uma limitação dessa abordagem é que o ponto de comunicação central precisa de alta largura de banda para gerenciar a comunicação com a *botnet*. Apesar de facilitar o gerenciamento da *botnet*, o ponto central é um ponto de falha arquitetural. Se o servidor C&C for eliminado, o atacante perderá todo o controle sobre a *botnet* (Zeidanloo e Manaf, 2010; Gupta e Dahiya, 2021).

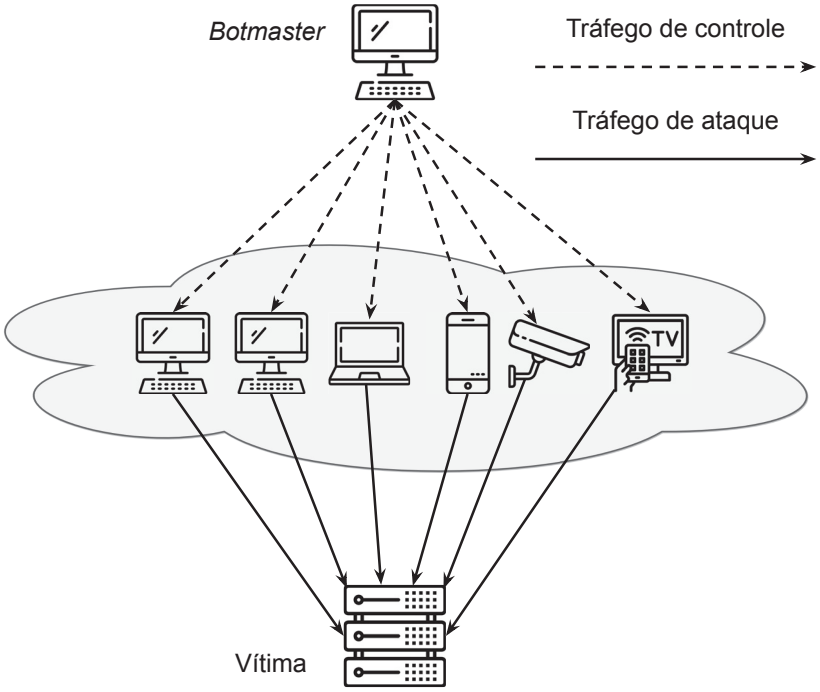


Figura 2.1: Estrutura Básica do Ataque DDoS (adaptado de Bhatia et al. (2018))

Para evitar um único ponto de falha, os atacantes podem usar a arquitetura P2P para operar como o modelo C&C. Nessa arquitetura, os *bots* atuam como clientes, recebendo comandos e como servidores, disponibilizando comandos para outros *bots*. Se um grupo de *bots* for encerrado, a *botnet* pode continuar operando com o restante dos *bots*. Esta arquitetura é mais complexa para o atacante gerenciar e o tempo de disseminação do comando pode ser maior (Zeidanloo e Manaf, 2010; Gupta e Dahiya, 2021). A arquitetura híbrida visa reduzir a complexidade de gerenciamento da arquitetura P2P. Na arquitetura híbrida, apenas um conjunto de *bots* atua como servidor e cliente. Assim, todos os *bots* procuram esses servidores para se atualizarem sobre os novos comandos (Wang et al., 2010; Gupta e Dahiya, 2021).

Segundo Vormayr et al. (2017), para implementar as arquiteturas das *botnets*, é necessário utilizar algum protocolo pré-existente desenvolvido para outra finalidade ou utilizar um protocolo exclusivo, os protocolos neotéricos. O *Internet Relay Chat* (IRC) é um dos protocolos mais importantes para a comunicação com *botnets*. O protocolo IRC opera em sistemas de bate-papo da Internet. Porém, os atacantes passaram a utilizá-lo por sua estrutura de fácil implementação, já que é amplamente utilizado na Internet e permite a conversação entre várias entidades. As desvantagens desse protocolo são que as redes corporativas raramente usam esse protocolo e os firewalls podem bloquear facilmente o tráfego.

Outro protocolo comumente usado na Internet e por atacantes é o Protocolo de Transferência de Hipertexto (do inglês, *Hypertext Transfer Protocol* - HTTP). O HTTP opera no modelo cliente-servidor onde o cliente cria requisições e o servidor responde (Comer, 2015). Tal como acontece com as comunicações IRC, as comunicações HTTP são simples de implementar. Porém, elas possuem latência maior quando comparadas ao IRC e não permitem a comunicação entre grupos, pois o cliente deve iniciar a solicitação, neste caso, o *bot*. Outro protocolo que os atacantes usam para se comunicar com as *botnets* é o *Server Message Block* (SMB). O SMB é um protocolo projetado para compartilhar recursos, como impressoras, arquivos e portas seriais entre computadores (Musik e Jaroensutasinee, 2007). Os atacantes usam esse protocolo para comunicação em redes locais, pois ele pode ser bloqueado diretamente no *gateway* da Internet.

Vários atacantes usam protocolos baseados na arquitetura P2P para manter a comunicação com seus *bots*. A comunicação P2P foi concebida como uma alternativa aos servidores centralizados utilizados para o compartilhamento de arquivos e consequentemente para reduzir o tempo de download desses arquivos. Nesta abordagem, um arquivo está disponível em vários servidores. Quando o cliente precisa baixar um arquivo, ele pode obter pedaços dos servidores mais próximos e não necessariamente o arquivo inteiro do mesmo lugar (Comer, 2015). Quando o cliente acaba de receber as partes do arquivo, ele se torna um servidor de arquivos, aumentando assim o número de servidores (Comer, 2015). Os atacantes aproveitam esse modo de comunicação para distribuir seus comandos entre os *bots* (Vormayr et al., 2017).

Os protocolos neotéricos são protocolos de comunicação criados exclusivamente para a comunicação entre *bots* e *botmasters*. Os atacantes geralmente implementam esses novos protocolos com base no TCP ou no UDP. No entanto, há casos em que os atacantes usam o protocolo de mensagens de controle da internet (do inglês, *Internet Control Message Protocol* - ICMP) para comunicação C&C. O objetivo de propor protocolos exclusivos para *botnets* é dificultar a detecção. No entanto, se eles revelarem padrões raramente usados na rede, os sistemas de segurança podem detectar facilmente a comunicação (Vormayr et al., 2017).

Um ataque DDoS possui pelo menos quatro etapas: reconhecimento, recrutamento, comando e controle e lançamento do ataque (Cisco, 2014; Sahu e Khare, 2020). O reconhecimento consiste em coletar informações sobre a vítima e os candidatos a *bots*. Identificar os dispositivos conectados à Internet ajuda o atacante a definir quais dispositivos podem atuar como servidores C&C ou clientes que realizam o ataque. Além disso, as informações auxiliam os atacantes a definirem a estratégia de recrutamento desses dispositivos. Para recrutar *bots*, os atacantes podem explorar vulnerabilidades para infectar dispositivos ou criar campanhas para espalhar software malicioso em anexo de e-mail ou downloads da web (Yadav e Thakur, 2020).

Ao atingir milhares de dispositivos controlados, o ataque evolui para a etapa de controle. Os atacantes podem realizar manutenção no código da *botnet* para atualizar o código dos *bots* e sincronizá-los conforme as intenções dos atacantes. Além disso, na etapa de controle, os atacantes realizam testes com duração de segundos ou minutos para avaliar a eficácia do ataque e realizar as correções necessárias. A última etapa é lançar o ataque, onde todos os *bots* ativos na *botnet* começam a enviar tráfego malicioso para a vítima. Os atacantes podem evitar o processo de preparação de um ataque DDoS se usarem um serviço de ataque DDoS online (Stresser-App, 2021; DDoS-Stress, 2023). Nesses casos, outros atacantes executaram as etapas iniciais e possuem *botnets* prontas para lançar ataques DDoS por uma taxa (Santanna et al., 2015).

2.2 TIPOS DE ATAQUE DDOS

A literatura apresenta três tipos de ataques DDoS: ataque baseado em volume, ataque baseado em protocolo e ataque baseado em camada de aplicação. Os ataques baseados em volume sobrecarregam a largura de banda da rede da vítima usando grandes volumes de dados maliciosos. Os ataques baseados em protocolo exploram vulnerabilidades principalmente presentes nas camadas de rede e transporte do modelo de referência Protocolo de Controle de Transmissão/Protocolo de Internet (do inglês, T*ransmission Control Protocol/Internet Protocol* - TCP/IP) para sobrecarregar os recursos de computação da vítima. O ataque baseado na camada de aplicação evita que o serviço alcance a quinta camada da pilha do protocolo TCP/IP (Alam et al., 2015; Sahoo et al., 2019; Visalatchi et al., 2020; El-Sofany, 2020; Gupta e Dahiya, 2021; Cloudflare, 2021a). Devido ao número de variações nos ataques DDoS, a literatura tem diferentes taxonomias de acordo com pontos de vista distintos (Douligeris e Mitrokotsa, 2004; Mirkovic e Reiher, 2004; Asosheh e Ramezani, 2008; Gupta e Badve, 2017; Yusof et al., 2019). O grau

de automação do ataque, a vulnerabilidade explorada pelo ataque e a taxa de ataque são os critérios para classificar os ataques DDoS existentes (Gupta e Dahiya, 2021). Esta pesquisa apresenta ataques DDoS usando uma classificação baseada nos impactos dos ataques assim como os estudos em (Specht e Lee, 2004; Mahjabin et al., 2017; Salim et al., 2020). Em Mahjabin et al. (2017) e Salim et al. (2020), os autores dividiram os ataques DDoS em quatro categorias: esgotamento de largura de banda, esgotamento de recursos, ataque de infraestrutura e ataque de dia zero. O objetivo desta taxonomia é garantir que o estado da arte seja coberto completamente. A Figura 2.2 apresenta a taxonomia de ataques DDoS, todas as categorias são descritas a seguir.

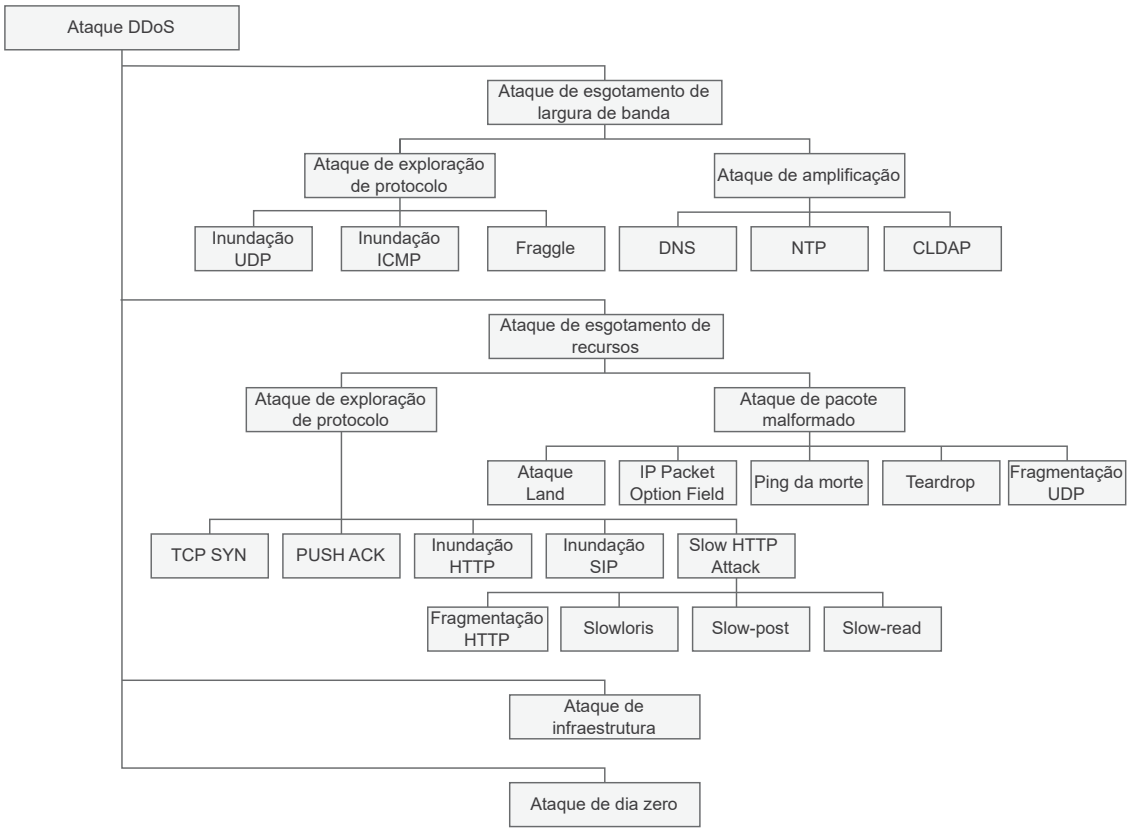


Figura 2.2: Taxonomia de Ataque DDoS (adaptado de Mahjabin et al. (2017); Salim et al. (2020))

2.2.1 Ataque de Esgotamento de Largura de Banda

O objetivo dos ataques de esgotamento da largura de banda é negar acesso a serviços consumindo toda a largura de banda da rede da vítima usando *bots* ou dispositivos amplificadores previamente comprometidos. Este tipo de ataque pode durar longos períodos antes de ser mitigado. Os ataques de esgotamento de largura de banda podem ser classificados como ataques de exploração de protocolo e ataques de amplificação (Mahjabin et al., 2017; Salim et al., 2020; Deshmukh e Devadkar, 2015). A principal propriedade dos **ataques de exploração de protocolo** é usar protocolos em diferentes camadas de rede para levar a vítima à privação de largura de banda. Existem três subtipos de ataques nesta categoria: inundação UDP, inundação ICMP e *fraggle* (Mahjabin et al., 2017; Salim et al., 2020), descritos a seguir:

- **Ataques baseados em inundação UDP** são bastante comuns, constituindo 43% dos ataques no primeiro trimestre de 2021 (Gutnikov et al., 2021). Nesse tipo de ataque, o

atacante manda a *botnet* inundar a largura de banda da rede da vítima com um grande fluxo de pacotes UDP com o endereço de origem falsificado. Este ataque tem duas variantes onde o atacante decide se a porta de destino do pacote UDP será a mesma para toda a *botnet* ou se a porta será aleatória. O ataque funciona porque o servidor recebe um fluxo de pacotes UDP e precisa verificar se há algum programa na porta especificada capaz de responder à solicitação. Se não houver um programa para resolver a solicitação, o servidor responde à solicitação, notificando que o endereço de destino não está acessível. Eventualmente, o servidor consome todos os recursos disponíveis para receber, processar e responder aos pacotes UDP (Mahjabin et al., 2017; Salim et al., 2020; Cloudflare, 2021g).

- **Ataques baseados em inundação de ICMP** visam consumir recursos de rede ao enviar e receber solicitações de ICMP. ICMP é um protocolo para relatar erros aos clientes sendo usado por ferramentas como *ping* e *traceroute* (Comer, 2015). O atacante usa a largura de banda da rede da vítima enviando solicitações ICMP excessivas. O servidor processa essas solicitações e retorna a resposta ICMP para a origem, ocupando a largura de banda (Mahjabin et al., 2017; Salim et al., 2020; Cloudflare, 2021e). Uma maneira simples de evitar esse tipo de ataque é desabilitar ou limitar o recebimento de pacotes ICMP originados em outras redes. O efeito colateral é que o servidor não responderá às solicitações de *ping* ou *traceroute* (Cloudflare, 2021e).

- ***Fraggle attack*** é semelhante à inundação ICMP, mas usa o protocolo UDP ao invés do protocolo ICMP. Para inundar a largura de banda, este ataque usa amplificadores de rede para enviar pacotes UDP_ECHO. Um amplificador pode ser qualquer dispositivo que responda a pelo menos um pacote de uma solicitação anterior. Assim, servidores web, servidores Sistema de Nome de Domínio (do inglês, *Domain Name System* - DNS) ou roteadores podem ser considerados refletores. Os atacantes usam esses refletores com seus endereços reais para enviar pacotes à vítima, simulando a resposta a uma solicitação. Assim, os atacantes permanecem disfarçados, mas o refletor é revelado (Mahjabin et al., 2017; Salim et al., 2020).

Ataques de amplificação objetivam negar o serviço, direcionando grandes respostas para a vítima, originadas de pequenas solicitações. Eles podem ser de três tipos: amplificação DNS, amplificação NTP e amplificação CLDAP (Mahjabin et al., 2017; Salim et al., 2020). Abaixo está a descrição dos ataques de amplificação:

- O **ataque de amplificação de DNS** usa resolvedores de DNS para aumentar a carga de ataque direcionada à vítima e esgotar a largura de banda da rede. Nesse tipo de ataque, o atacante usa a *botnet* para consultar resolvedores de DNS abertos. Nas consultas, o atacante insere o endereço de origem da vítima ao invés do endereço do *bot*. Assim, as respostas não são direcionadas aos dispositivos da *botnet* que criaram a consulta, mas à vítima. O atacante configura a solicitação para exigir o máximo de informações disponíveis. Assim, com uma pequena solicitação, o atacante pode gerar tráfego substancial para a vítima. A resposta originada pelo servidor DNS é válida, então as respostas são um desafio para os sistemas de defesa filtrarem. Para habilitar esse tipo de ataque, o atacante precisa identificar os resolvedores de DNS abertos (Mahjabin et al., 2017; Salim et al., 2020; Cloudflare, 2021b).

- O **ataque de amplificação NTP** usa servidores do protocolo *Network Time Protocol* (NTP) para aumentar a carga de ataque direcionado à vítima e esgotar a largura de banda

da rede. As máquinas conectadas à Internet usam o protocolo NTP para sincronizar seus relógios internos. Para amplificar o ataque, o atacante usa uma função de servidor NTP que retorna dados das últimas 600 consultas realizadas no servidor. O atacante pode, com apenas uma solicitação, criar uma resposta várias vezes maior do que a solicitação original. Assim, o atacante usa uma *botnet* para criar várias consultas com respostas direcionadas ao servidor da vítima. Uma forma de evitar esse tipo de ataque é desabilitar a função que retorna os dados das últimas consultas. Esta é a ação padrão para servidores com versões do software NTP 4.2.7 ou superior (Mahjabin et al., 2017; Salim et al., 2020; Cloudflare, 2021d).

- O **ataque de amplificação CLDAP** usa o protocolo *Connection-less Lightweight Directory Access Protocol* (CLDAP) para aumentar a carga do ataque direcionado à vítima e esgotar a largura de banda da rede. O protocolo CLDAP é uma alternativa ao protocolo *Lightweight Directory Access Protocol* (LDAP) e fornece as informações contidas nos diretórios. Para amplificar o ataque, o atacante realiza várias consultas onde a resposta é maior do que a consulta. As respostas são direcionadas ao servidor da vítima para buscar a negação de serviço (Salim et al., 2020; Arteaga e Mejia, 2017).

2.2.2 Ataque de Esgotamento de Recursos

Outra abordagem para negar serviço a usuários legítimos é consumir outros recursos além da largura de banda da rede. A Unidade de Processamento Central (do inglês, *Central Processing Unit* - CPU), memória ou soquetes são exemplos comuns de recursos que podem ser consumidos por atacantes. Ataques de esgotamento de recursos podem ser ataques de exploração de protocolo e ataques de pacotes malformados (Mahjabin et al., 2017; Salim et al., 2020; Deshmukh e Devadkar, 2015).

Ataques de exploração de protocolo usam vários protocolos para consumir os recursos vitais para a entrega do serviço. Eles podem ser de cinco tipos: ataque TCP SYN, ataque TCP PUSH + ACK, ataque de inundação HTTP, ataque de inundação SIP e ataque *Slow* HTTP (Mahjabin et al., 2017; Salim et al., 2020). Abaixo está a descrição dos ataques de exploração de protocolo:

- **TCP SYN** é um ataque onde o atacante explora uma fraqueza do TCP para consumir o recurso de memória e, consequentemente, negar o serviço. Para o cliente iniciar uma conexão TCP, é necessário realizar o processo de *handshake*. Nesse processo, o cliente, que deseja estabelecer a conexão, envia um pacote do tipo SYN para o servidor. Após o recebimento, o servidor responde com um pacote SYN/ACK para confirmar a comunicação. Nesse ponto, o servidor espera que o cliente envie de volta um pacote ACK para estabelecer a comunicação. Este é o momento em que o atacante atua, pois o pacote ACK esperado pelo servidor nunca chega. Além disso, o atacante cria várias conexões TCP com endereços falsificados até que o servidor fique sobrecarregado e não consiga estabelecer novas conexões. Assim, os usuários que tentarem utilizar o serviço não conseguirão estabelecer conexões (Mahjabin et al., 2017; Salim et al., 2020; Cloudflare, 2021f).

- No **ataque TCP PUSH + ACK**, o atacante força o servidor vítima a descartar pacotes originados por usuários legítimos por meio de uma inundação de pacotes TCP PUSH ACK. Usando a *botnet*, o atacante envia vários pacotes com as *flags* PUSH e ACK com o valor "1" no cabeçalho. Esta ação requer processamento extra do servidor para verificar a necessidade de limpar a memória e encaminhar uma resposta. Consequentemente,

o atacante torna o servidor de destino incapaz de processar todas as solicitações TCP, além de descartar pacotes legítimos (Mahjabin et al., 2017; Salim et al., 2020).

- O **ataque de inundação de HTTP** visa esgotar os recursos de computação da vítima inundando o servidor com muitas solicitações de HTTP. HTTP é um protocolo da camada de aplicação que permite a comunicação entre o cliente e os servidores web (Comer, 2015). Os atacantes podem usar duas abordagens para exaurir os recursos de computação da vítima: HTTP GET e HTTP POST. Em um ataque HTTP GET, os atacantes usam *botnets* para criar muitas solicitações de imagens, vídeos ou qualquer arquivo presente no servidor de destino. Esta ação faz com que o servidor precise encontrar o arquivo solicitado, carregá-lo na memória e dividi-lo para entregar aos solicitantes. No entanto, o servidor pode ficar sem recursos de processamento ou memória para atender a todas as solicitações criadas pelo atacante. Em um ataque HTTP POST, o atacante anexa parâmetros à solicitação POST para causar uma carga de processamento significativa com operações pesadas no servidor da vítima. Eventualmente, o servidor fica sem recursos e o ataque é considerado bem-sucedido (Mahjabin et al., 2017; Salim et al., 2020; Cloudflare, 2021i).

- O **ataque de inundação SIP** esgota os recursos computacionais da vítima inundando o servidor com muitas solicitações de Protocolo de Iniciação de Sessão (do inglês, *Session Initiation Protocol* - SIP). O SIP é usado por aplicativos de chamada de voz sobre IP (do inglês *Voice-over Internet Protocol* - VOIP) (Comer, 2015). Usando uma *botnet*, o atacante envia muitas mensagens de solicitação SIP, como SIP REQUEST ou SIP INVITE. Em seguida, o atacante tenta consumir todos os recursos do servidor de registro SIP (Mahjabin et al., 2017; Salim et al., 2020).

- O **ataque *Slow* HTTP** consome lentamente os recursos de computação da vítima. Em geral, o atacante procura maneiras de manter as conexões cliente-servidor abertas pelo maior tempo possível. Este ataque não requer recursos abundantes para ser lançado e produz tráfego semelhante ao real. Portanto, esse tipo de ataque é difícil de identificar. Existem quatro abordagens para conduzir esse ataque: fragmentação HTTP, *Slowloris*, *Slowpost* e *Slow-read*. Na fragmentação HTTP, o atacante diz a *botnet* para dividir os pacotes HTTP em pequenos fragmentos. Ao enviar os fragmentos o mais lentamente possível para o servidor vítima, o *bot* mantém a conexão aberta o maior tempo possível. O ataque *Slowloris* abre uma conexão com o servidor vítima e envia lentamente cabeçalhos HTTP parciais. Isso faz com que o servidor espere até o final da mensagem, consumindo uma conexão e negando o serviço para usuários legítimos. O ataque *Slowpost*, também conhecido como ataque RUDY, mantém o servidor esperando por várias solicitações HTTP POST que o atacante enviou ao servidor lentamente. O servidor acaba mantendo essas conexões abertas por muito tempo e pode causar a interrupção do serviço. Por fim, o objetivo do ataque *Slow-read* é ler as respostas do servidor o mais lentamente possível, forçando o servidor a manter as conexões abertas o maior tempo possível (Mahjabin et al., 2017; Salim et al., 2020; Cloudflare, 2021c).

Ataques de pacotes malformados visam interromper a operação da vítima usando pacotes inválidos. Eles podem ser de cinco tipos, *Land attack*, *IP Packet Option Field*, Ping da morte, ataque de fragmentação UDP e ataque *Teardrop* (Mahjabin et al., 2017; Salim et al., 2020), conforme descrito a seguir.

- **Land attack** consome os recursos do servidor da vítima causando *loops* entre solicitações e respostas. Para fazer isso, os atacantes modificam o cabeçalho do pacote IP (do inglês,

Internet Protocol) e, em seguida, a origem e o destino tornam-se o endereço IP da vítima. A vítima então responde a si mesma e, eventualmente, consome os recursos disponíveis (Mahjabin et al., 2017; Salim et al., 2020).

- O ***IP Packet Option Field*** consome recursos do servidor aumentando a carga de processamento necessária para analisar cada pacote. Para fazer isso, os atacantes modificam os campos opcionais do pacote inserindo informações adicionais. As *botnets* enviam esses pacotes alterados para a vítima para causar a interrupção do sistema (Mahjabin et al., 2017; Salim et al., 2020).

- O **Ping da morte** (do inglês, *Ping of Death* - PoD) sobrecarrega o servidor vítima. Para isso, os atacantes constroem pacotes ICMP maiores que o tamanho máximo. O pacote se divide em segmentos menores e força a vítima a reconstruí-lo. Este processo de reconstrução pode causar estouro de memória e, consequentemente, impedir que o servidor vítima continue funcionando corretamente (Salim et al., 2020; Samta e Sood, 2020).

- O **ataque *Teardrop*** interrompe o correto funcionamento do servidor explorando a fragmentação e remontagem de pacotes TCP/IP. Para isso, os atacantes alteram o campo "*fragment offset*" dos pacotes TCP/IP para sobrepô-los. O servidor vítima pode sobrecarregar ao receber esses pacotes inválidos e tentar remontá-los (Mahjabin et al., 2017; Salim et al., 2020; Radware, 2021).

- O **ataque de fragmentação UDP** consome os recursos do servidor. Para fazer isso, os atacantes enviam pacotes UDP maiores do que a rede pode aceitar. O pacote se divide em segmentos menores e força a vítima a reconstruí-lo. Este processo de reconstrução é caro para o servidor vítima, o que pode levar ao consumo de todos os recursos disponíveis (Salim et al., 2020; Samta e Sood, 2020).

2.2.3 Ataque de Infraestrutura

O objetivo do ataque à infraestrutura é negar o acesso aos serviços, consumindo toda a largura de banda e os recursos de computação da infraestrutura crítica para o funcionamento da Internet. Um exemplo clássico desse ataque é o ataque a servidores de DNS (Mahjabin et al., 2017). DNS é uma rede hierárquica responsável por traduzir endereços de computador em nomes simbólicos para simplificar o uso de serviços web (Comer, 2015). Caso o DNS não consiga resolver uma solicitação, o usuário pode não conseguir acessar o serviço pretendido. Em outubro de 2016, a empresa DynDNS foi alvo de um dos maiores ataques de infraestrutura conhecidos. Nesse caso, milhares de dispositivos IoT inundaram os servidores da empresa para negar acesso a serviços importantes como GitHub, Twitter e Netflix (Williams, 2016; Mahjabin et al., 2017).

2.2.4 Ataque de Dia Zero

Ataques DDoS de dia zero são ataques com vetores não catalogados, ou seja, ataques sem precedentes. Nesse tipo de ataque, os atacantes exploram vulnerabilidades ou violações de segurança ainda não usadas para conduzir o ataque DDoS. Além dos vetores usados, o impacto do ataque também é desconhecido. O ataque leva este nome porque somente após o ataque no dia 0 é possível reconhecer o vetor de ataque e propor a defesa ou resposta apropriada (Sahoo et al., 2019; Fenil e Mohan Kumar, 2020; Sonar e Upadhyay, 2014; Mahjabin et al., 2017). Manter os sistemas atualizados e corretamente configurados pode ser uma forma de evitar vulnerabilidades

desconhecidas (Cloudflare, 2021h), consequentemente, diminuir as chances de ataques de dia zero. Além disso, algumas empresas oferecem recompensas para os pesquisadores relatarem vulnerabilidades e violações de segurança (Facebook, 2021; Google, 2021; Apple, 2021).

2.3 MECANISMOS DE DEFESA DDOS

Os mecanismos de defesa contra os ataques DDoS definem ações para prevenir ou reduzir os danos causados por ataques. A literatura apresenta algumas abordagens para representar mecanismos de defesa. O nível de atividade, o grau de cooperação, o local de implantação e a estratégia de resposta ao ataque são maneiras de classificar os mecanismos de defesa DDoS existentes (Mirkovic e Reiher, 2004; Chahal et al., 2019). Este trabalho apresenta os mecanismos de defesa DDoS por meio de uma classificação baseada no tempo de atuação do mecanismo como em (Gupta e Badve, 2017; Srinivasan et al., 2020; Radain et al., 2021; Fenil e Mohan Kumar, 2020; Somani et al., 2017; Zargar et al., 2013). O momento de ação dos mecanismos de defesa é a abordagem adequada a ser representada neste trabalho, pois um dos objetivos dos mecanismos de defesa é identificar ataques DDoS o mais rápido possível (Dalmazo et al., 2021; Gupta e Dahiya, 2021). Em Zargar et al. (2013), os autores dividiram os mecanismos de defesa DDoS em três categorias: prevenção, detecção e mitigação (resposta). Atualmente, ambientes como nuvem (Somani et al., 2017), IoT (Salim et al., 2020) e Rede definida por software (do inglês, *Software-defined networking* - SDN) (Sahoo et al., 2019) possuem mecanismos de defesa especializados para lidar com propriedades específicas de cada ambiente. Para ser genérico e apresentar uma visão geral dos mecanismos de defesa contra os ataques DDoS, a Figura 2.3 apresenta a classificação proposta em (Zargar et al., 2013), atualizada com base em (Srinivasan et al., 2020; Gupta e Dahiya, 2021; Radain et al., 2021; Somani et al., 2017), e todas as categorias são descritas a seguir.

2.3.1 Prevenção de Ataques

O principal objetivo do combate a um ataque DDoS é evitá-lo (Gupta e Badve, 2017; Zargar et al., 2013), uma vez que o ataque foi lançado, os danos podem ser inevitáveis (Gupta e Dahiya, 2021). Desativar serviços desnecessários, manter sistemas atualizados, instalar e manter firewalls atualizados e configurados corretamente e replicar serviços em diferentes lugares são mecanismos gerais de prevenção (Zargar et al., 2013; Gupta et al., 2010). A prevenção ocorre antes do ataque ser lançado e visa impedir ou restringir o lançamento do ataque, e pode agir proativamente para aceitar ou rejeitar solicitações suspeitas (Srinivasan et al., 2020; Gupta e Dahiya, 2021; Somani et al., 2017). Em Srinivasan et al. (2020), Gupta e Dahiya (2021) e Somani et al. (2017), Abdlhamed et al. (2017) os autores apresentam desafio-resposta, servidores ou portas ocultas, acesso restritivo, limites de recursos e a predição como abordagens adicionais para prevenir ataques DDoS. Abaixo está a descrição dos mecanismos de prevenção:

- O mecanismo **desafio-resposta** visa distinguir usuários legítimos de *bots*. Para provar a legitimidade, os usuários devem resolver um desafio na forma de um teste de Turing público totalmente automatizado para distinguir computadores de humanos (do inglês, *Completely Automated Public Turing Test to Tell Computers and Humans Apart* - CAPTCHA). Atualmente, existem vários tipos de CAPTCHAs com muitos níveis de complexidade diferentes. Baseado em texto, baseado em imagem, baseado em vídeo e baseado em áudio são os tipos existentes de CAPTCHA (Brodić e Amelio, 2020). Equilibrar segurança e simplicidade é um desafio para esta abordagem, enquanto uma

Figura 2.3: Mecanismos de Defesa Contra os Ataques DDoS

desvantagem é o consumo excessivo de recursos (Srinivasan et al., 2020; Gupta e Dahiya, 2021; Somani et al., 2017).

- O propósito do mecanismo **servidores ou portas ocultas** é ocultar servidores e evitar contato direto entre clientes e servidores. Para isso, dispositivos intermediários, também conhecidos como *proxies*, são implantados entre os clientes e servidores. O *proxy* é responsável por gerenciar a conexão entre os clientes e servidores, seja monitorando ou encaminhando o tráfego entre os servidores. Portas ocultas, alvos móveis, servidores efêmeros e servidores intermediários são exemplos de abordagens usadas para ocultar servidores. Embora este mecanismo oculte possíveis pontos fracos dos servidores, os *proxies* podem ser sobrecarregados ao redirecionar o tráfego (Srinivasan et al., 2020; Somani et al., 2017; Gupta e Dahiya, 2021).

- No mecanismo **acesso restritivo**, o objetivo é gerenciar o acesso priorizando clientes potencialmente confiáveis. Para priorizar os clientes, o servidor atrasa a resposta. Assim, os recursos ficam disponíveis para os clientes com a melhor pontuação de reputação. Existem várias estratégias para definir a reputação dos usuários. Uma estratégia para calcular a reputação é usar o histórico de acesso. Outra estratégia é analisar o tempo para resolver quebra-cabeças criptográficos. Embora esta seja uma abordagem que visa regular os recursos do servidor, é necessário lidar com algumas desvantagens desse mecanismo. O custo de manter as conexões atrasadas, os problemas relacionados à escalabilidade e o aumento do tempo de processamento para analisar as respostas dos quebra-cabeças são exemplos de problemas com esta técnica (Srinivasan et al., 2020; Gupta e Dahiya, 2021; Somani et al., 2017).

- O objetivo do mecanismo **limites de recursos** é evitar grandes custos na conta do servidor causados por ataques DDoS. Nessa abordagem, os administradores de serviço impõem limites de replicação de servidor e restrições aos recursos disponíveis para cada serviço. Assim, um ataque DDoS contra a rede irá ativar o número máximo de servidores, garantindo que a conta não ultrapasse os limites pré-definidos. Este mecanismo só protege o servidor evitando custos elevados na fatura do servidor, porque se o ataque DDoS consumir todos os recursos de rede/serviço, ele conseguirá negar o serviço a usuários legítimos (Gupta e Dahiya, 2021; Somani et al., 2017).

- Em Abdlhamed et al. (2017), os autores apresentam a predição como um mecanismo para auxiliar na detecção e mitigação do ataque, pois a predição ocorre antes do lançamento do ataque. Para realizar as predições, as soluções produzem evidências de ataques futuros. A evidência pode ser uma mensagem de alerta, a probabilidade de ocorrência de um ataque ou qualquer artefato direcionado aos administradores de rede que possa representar um ataque DDoS que ainda não foi lançado. A Figura 2.4 destaca a diferença entre predição e detecção. A predição aparece antes do lançamento do ataque em relação à cronologia do ataque, enquanto os mecanismos de detecção precisam que o ataque esteja em execução para reconhecê-lo. A Seção 2.1 define o início do ataque como seu quarto estágio (lançamento do ataque). A principal vantagem da predição é fornecer mais tempo para os administradores de rede responderem a um ataque. Isso porque, uma vez iniciado o ataque, o dano pode ser irreversível. As soluções de predição podem operar com pouco volume de informações sobre incidentes futuros em comparação com as soluções de detecção e mitigação. Em outras palavras, as soluções de predição devem produzir evidências sobre ataques futuros, processando informações onde os sinais não determinam claramente um ataque, enquanto as soluções de detecção dependem dos sinais de esgotamento causados pelo ataque. Apesar da dificuldade em identificar esses sinais, as predições serão mais precisas à medida que mais sinais forem identificados (Abdlhamed et al., 2017). O Capítulo 3 apresenta a classificação e detalha soluções que contribuem para o mecanismo de predição.

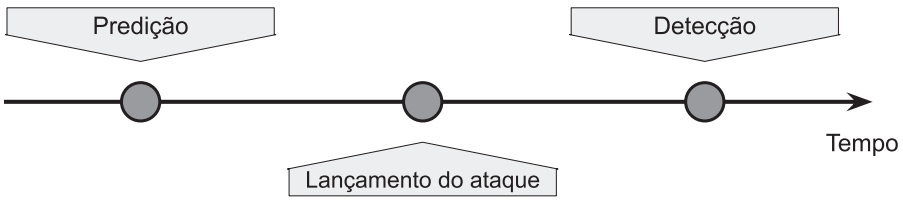


Figura 2.4: Relação entre Predição, Detecção e Lançamento do Ataque DDoS

2.3.2 Detecção de Ataque

Após os procedimentos de prevenção serem determinados, a próxima etapa é definir os mecanismos de detecção de ataques. A detecção de ataques ocorre depois que o atacante lança o ataque e pode ocorrer aos primeiros sinais de sobrecarga ou quando o ataque DDoS está causando danos. Para detectar ataques DDoS, as soluções podem usar métricas como consumo de memória, tempos de resposta e informações relacionadas ao desempenho do servidor. A detecção pode ocorrer na infraestrutura vítima, em intermediários de rede ou na origem do ataque. O local mais viável para a detecção de ataques é na infraestrutura vítima, por ser a vítima que recebe todo o tráfego malicioso. No entanto, a detecção perto da fonte do ataque reduz a quantidade de dano

sofrido pela vítima e pelos intermediários de rede (Zargar et al., 2013; Somani et al., 2017; Gupta e Badve, 2017; Srinivasan et al., 2020). Os mecanismos de detecção existentes são: detecção de anomalias, detecção de assinatura, rastreamento de origem falsificada, filtragem baseada em contagem, *BotCloud Detection* e mecanismos de detecção de uso de recursos (Somani et al., 2017; Srinivasan et al., 2020; Gupta e Dahiya, 2021). Abaixo estão descritos os mecanismos de detecção:

- Os mecanismos de **detecção de anomalias** buscam padrões que diferem dos padrões comuns para detectar ataques DDoS. As soluções rastreiam e analisam pacotes, analisam registros de acesso e conexões de Internet e comparam essas informações aos padrões normais predefinidos. Se o mecanismo identificar uma anomalia, um ataque DDoS pode estar em andamento. Soluções baseadas em aprendizado de máquina vêm ganhando destaque na literatura. No entanto, existem vários tipos de soluções, tais como baseadas em mineração de dados, métodos estatísticos, computação suave e algoritmos de fluxo de dados (Nooribakhsh e Mollamotalebi, 2020). O maior problema com os mecanismos baseados em anomalias é a alta porcentagem de falsos positivos, já que uma anomalia nem sempre representa um ataque em progresso (Somani et al., 2017; Srinivasan et al., 2020; Gupta e Dahiya, 2021; Radain et al., 2021).

- Os mecanismos de **detecção de assinatura** detectam ataques com padrões conhecidos. As soluções que usam esse mecanismo comparam os dados atuais com as assinaturas de ataques anteriores para tentar detectar ataques. Apesar de ser eficaz contra os ataques conhecidos, manter a base de ataques atualizada não é trivial. Outra limitação das soluções que implementam detecção de assinatura é que as soluções dificilmente detectarão ataques com padrões ainda não registrados (Radain et al., 2021).

- O objetivo do mecanismo **rastreamento de origem falsificada** é identificar o endereço da requisição. O resultado esperado não é o endereço do atacante, mas o endereço dos dispositivos usados pelos atacantes para conduzir o ataque (Belenky e Ansari, 2003). Os mecanismos de mitigação usam a origem real da solicitação para conter o ataque. Rastreamento de ICMP, marcação de pacote probabilístico e rastreamento de IP baseado em *hash* são exemplos de mecanismos de rastreamento de origem falsificada (Belenky e Ansari, 2003). Para que este mecanismo funcione corretamente, é necessária a cooperação entre diferentes entidades da rede, limitando a adoção deste mecanismo (Somani et al., 2017; Gupta e Dahiya, 2021).

- O mecanismo de **filtragem baseado em contagem** visa filtrar o tráfego malicioso usando recursos de rede. As soluções que implementam este mecanismo enumeram o número de conexões, o número de requisições feitas por cada origem ou o número de saltos. Sempre que o resultado excede um limite predeterminado, a solução marca o pacote como malicioso. Esse mecanismo é simples de implementar e fornece uma resposta rápida ao ataque. No entanto, é necessário monitorar a taxa de sucesso porque as soluções podem classificar incorretamente muitos pacotes (Gupta e Dahiya, 2021; Somani et al., 2017).

- O mecanismo ***BotCloud Detection*** visa identificar ou detectar máquinas virtuais comprometidas usadas como *bots* para conduzir ataques internos na nuvem. Uma maneira de identificar máquinas virtuais comprometidas é comparar as ações realizadas pelas máquinas virtuais com uma lista de ações típicas de máquinas virtuais comprometidas. Ou usando mecanismos de detecção baseados em anomalias. Estabelecer os

limites para reconhecer o comportamento malicioso é o principal problema deste mecanismo (Srinivasan et al., 2020; Somani et al., 2017; Gupta e Dahiya, 2021).

- No mecanismo **uso de recursos**, as soluções monitoram os recursos do servidor com frequência e, se o servidor começar a usar mais recursos do que o normal, pode estar ocorrendo um ataque DDoS. Como vários ataques visam exaurir os recursos do servidor, métricas como processamento e consumo de memória podem ajudar a descobrir ataques DDoS. O principal desafio das soluções baseadas neste mecanismo é identificar corretamente quando os recursos são consumidos por usuários legítimos ou por *bots* como resultado de um ataque DDoS (Somani et al., 2017; Srinivasan et al., 2020; Radain et al., 2021). Outra dificuldade com esses mecanismos é que, quando o consumo de recursos é discriminatório para detectar um ataque DDoS, pode não haver tempo suficiente para tomar precauções contra o ataque.

2.3.3 Mitigação de Ataque

Após identificar o ataque, é necessário tomar ações adequadas para responder e mitigar o ataque. A mitigação consiste em ações para evitar ou minimizar os danos sofridos pelo serviço. Além disso, esses mecanismos definem ações para a recuperação do serviço no final do ataque. Este trabalho apresenta mecanismos de mitigação atribuídos à vítima, e aponta a necessidade do esforço coletivo de outras entidades da rede para proteger os seus ativos e aumentar a segurança global contra os ataques DDoS. Após a ocorrência de um ataque, os métodos de prevenção e detecção podem ser atualizados usando as lições aprendidas durante a mitigação de ataque (Srinivasan et al., 2020; Gupta e Badve, 2017; Somani et al., 2017; Zargar et al., 2013). Em Srinivasan et al. (2020), Gupta e Dahiya (2021), Somani et al. (2017), Zargar et al. (2013), os autores apresentam cinco mecanismos para mitigar ataques DDoS, sendo os seguintes: escalonamento de recursos, migração de vítima, gerenciamento de recursos de SO, rede definida por software e mitigação de DDoS como serviço. A seguir está a descrição dos mecanismos de mitigação:

- O **escalonamento de recursos** define o aumento de recursos seguindo a progressão do ataque. Nesse caso, as soluções de escalonamento automático alocam mais recursos para as máquinas virtuais existentes ou instalam novas máquinas virtuais no(s) mesmo(s) servidor(es) para processar todas as solicitações, minimizando os danos. O dimensionamento de recursos pode salvar o serviço de um ataque DDoS, mas pode aumentar a sobrecarga devido à utilização de recursos extras. Essa sobrecarga é inevitável para manter o serviço disponível, e esse é o objetivo dos ataques EDoS (Srinivasan et al., 2020; Gupta e Dahiya, 2021; Somani et al., 2017; Radain et al., 2021).

- Na **migração da vítima**, os administradores movem o serviço atacado para um servidor físico diferente, isolado do ataque. Assim, o serviço está disponível para usuários legítimos e o ataque continua consumindo o servidor antigo. Para ser eficaz, o mecanismo depende da possibilidade de haver um servidor disponível para os clientes, mas não sob ataque. Existem casos em que os atacantes preferem essa mudança porque isso aumenta os custos relacionados à alocação de um novo servidor (Srinivasan et al., 2020; Gupta e Dahiya, 2021; Somani et al., 2017; Radain et al., 2021).

- O **gerenciamento de recursos de sistemas operacionais (SO)** restringe o consumo de recursos pelo serviço atacado. Assim, este mecanismo visa impedir o consumo excessivo de recursos para não prejudicar o desempenho de outros métodos de mitigação.

A principal limitação desse mecanismo é que o serviço pode ficar sobrecarregado rapidamente, negando o serviço a usuários em potencial (Somani et al., 2017).

- A **Rede definida por software** fornece algumas oportunidades de reconfiguração da rede para mitigar os ataques DDoS. Essas oportunidades de reconfiguração se devem ao desacoplamento físico das funções de roteamento e controle da rede, auxiliando no gerenciamento dos serviços (Comer, 2015; ONF, 2021). Além do poder de reconfiguração da SDN, ela fornece uma inspeção profunda de pacotes. Apesar de seus benefícios, a SDN não é imune a ataques DDoS e exige mecanismos de defesa especializados (Somani et al., 2017; Gupta e Dahiya, 2021; Srinivasan et al., 2020).

- A **mitigação DDoS como um serviço** requer que os gerentes de rede recrutem soluções especializadas para mitigar ataques. Uma das maneiras de realizar a mitigação é adicionar um dispositivo intermediário que filtra e direciona os pacotes para o servidor. Outra maneira é usar soluções baseadas em nuvem para mitigar ataques. O uso de abordagens remotas pode causar problemas de privacidade se os dados do usuário se tornarem públicos. Outra limitação das abordagens de mitigação remota é que as ações de mitigação podem atrasar a contenção de ataques (Radain et al., 2021; Somani et al., 2017; Gupta e Dahiya, 2021).

2.4 APRENDIZADO DE MÁQUINA

O Dicionário de Cambridge define a palavra aprender como o ato ou ação de adquirir um novo conhecimento, ou uma nova habilidade[2]. Muito antes da invenção dos computadores, o ser humano já tentava emular o aprendizado em máquinas. No fim da Idade Média, Roger Bacon teria construído bonecos que simulavam a fala, Leonardo da Vinci teria construído um leão que possuía a capacidade de andar (Crevier, 1993). Com o poder de automatização proporcionado pela computação, cientistas identificaram a possibilidade de evoluir a pesquisa da automatização do aprendizado. Assim, desde o início da computação, cientistas desenvolvem formas para prover às máquinas a capacidade de aprender e realizar tarefas. Propor e evoluir formas de aprendizado para máquinas é um desafio contínuo da Inteligência Artificial (IA), uma importante área da computação (Carbonell et al., 1983). O objetivo da IA é construir máquinas inteligentes capazes de imitar o comportamento humano para as máquinas lidarem com várias situações de forma segura e eficaz (Kour e Gondhi, 2020; Russell e Norvig, 2021). O campo de atuação da IA engloba áreas extremamente relevantes como o aprendizado de máquina e o *deep learning*. A Figura 2.5 representa graficamente subáreas de IA. A IA é a grande área de concentração, que engloba todos os esforços relacionados com o aprendizado de máquina, que por sua vez, engloba todos os esforços do *deep learning* (Kaluarachchi et al., 2021; Kavlakoglu, 2020).

A Figura 2.6 apresenta dois paradigmas de programação de sistemas computacionais, a programação tradicional e o aprendizado de máquina. Na programação tradicional (Figura 2.6(a)) um conjunto de regras pré-determinadas agem sobre os dados para gerar as respostas. Por exemplo, um modo simplista de quantificar ganhos sobre a venda de um produto é calcular as receitas desse produto e subtrair os custos. Deste modo, regras predefinidas atuam sobre os dados para gerar a saída desejada pelo operador do sistema (O'Reilly, 2021).

A disponibilidade abundante de certos tipos de dados permite usá-los de modo diferente. Deste modo, a literatura do aprendizado de máquina propõe algoritmos capazes de analisar e aprender relações entre os dados (Figura 2.6(b)). Assim, a saída esperada do processo de

[2] https://dictionary.cambridge.org/dictionary/english-portuguese/learn?q=Learn

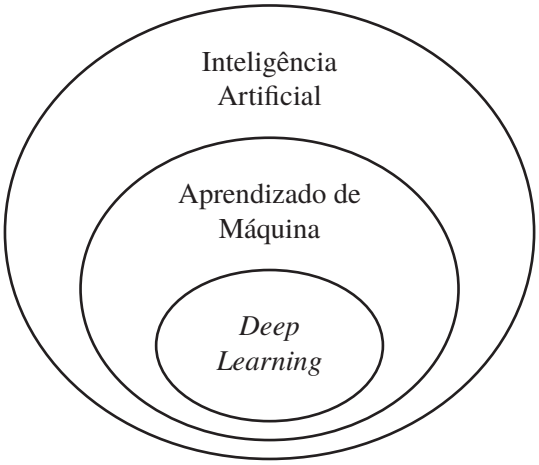
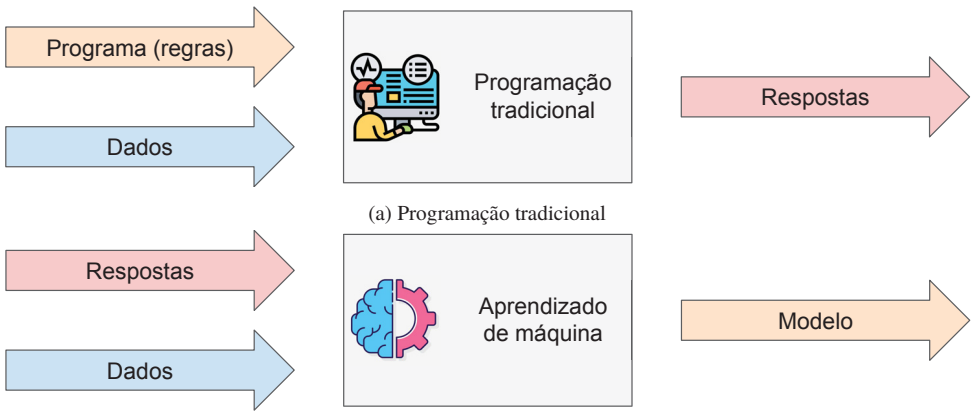


Figura 2.5: Inteligência Artificial e Suas Subáreas (adaptado de Kaluarachchi et al. (2021))

análise de dados é um modelo capaz de identificar os padrões aprendidos com a análise prévia quando confrontado com novas observações (Mitchell, 1997; Kour e Gondhi, 2020; O'Reilly, 2021). Soluções baseadas no aprendizado de máquina estão presentes em várias áreas como reconhecimento de fala (Padmanabhan e Premkumar, 2015), classificação de imagem (Li et al., 2019a) e detecção de anomalias (Liu e Lang, 2019; Pang et al., 2021). Uma questão fundamental para o sucesso da aplicação do aprendizado de máquina é a disponibilidade de dados. Vieses causados pela manipulação humana, por exemplo, na seleção de atributos ou na rotulação dos dados e a baixa representatividade dos dados podem induzir os algoritmos de aprendizado de máquina a aprender erradamente. Assim, o modelo gerado fica especialista para os dados de treinamento e pode não generalizar os resultados para outros cenários (Monroe, 2021).



(a) Programação tradicional

(b) Aprendizado de máquina

Figura 2.6: Paradigmas de Programação (adaptado de Raschka (2020); O'Reilly (2021))

2.4.1 Classificação dos Algoritmos de Aprendizado de Máquina

Ao longo da evolução do aprendizado de máquina, pesquisadores propuseram algoritmos com propriedades e focos diferentes. Utilizando essas diferentes propriedades, a literatura classifica os algoritmos de aprendizado de máquina quanto à tarefa realizada pelo algoritmo, à estratégia de aprendizado utilizada pelo algoritmo e à profundidade do algoritmo (Ibitoye et al., 2020). A Figura 2.7 apresenta a classificação dos algoritmos de aprendizado de máquina e cada ramo é apresentado a seguir.

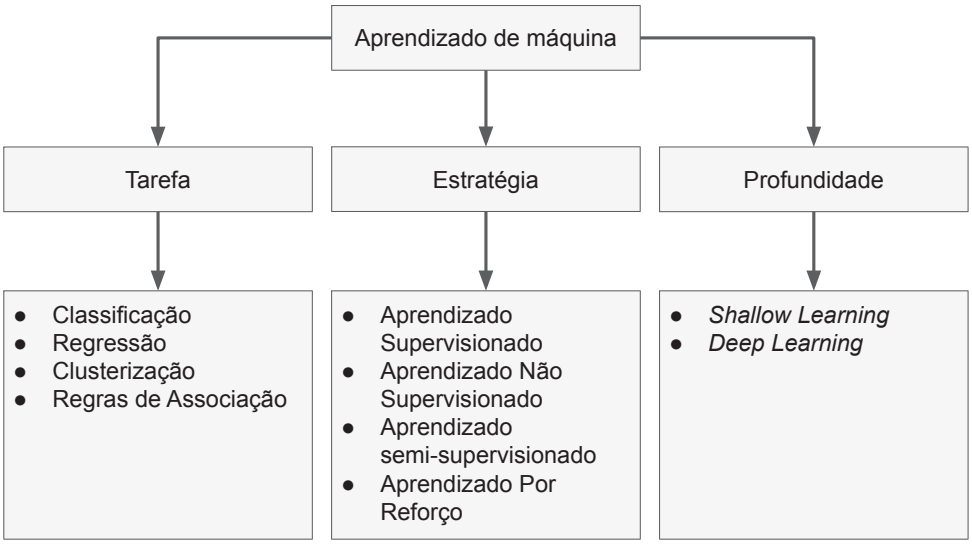


Figura 2.7: Classificação do Aprendizado de Máquina (adaptado de Ibitoye et al. (2020))

2.4.1.1 Tarefas do Aprendizado de Máquina

O ramo Tarefa, o primeiro ramo da Figura 2.7, apresenta quatro tarefas que podem ser realizadas com os algoritmos de aprendizado de máquina, sendo eles: classificação, regressão, clusterização e as regras de associação. A **classificação** é o processo de identificar a classe real de dados ainda não rotulados utilizando algoritmos de aprendizado de máquina (Muhammad e Yan, 2015). O rótulo identifica a natureza da ação, por exemplo, o rótulo pode distinguir transações de cartão de crédito fraudulentas ou verdadeiras. Ou ainda, o rótulo pode distinguir o tráfego em uma rede de computadores, onde as opções podem ser o tráfego normal ou tráfego de ataque. A principal diferença entre a classificação e os outros tipos de tarefas é que o rótulo é uma variável categórica ou discreta (Singh, 2019). Ou seja, o resultado da classificação será interpretado como uma classe do problema.

A Figura 2.8 ilustra o conceito da tarefa de classificar dados. Os círculos e os quadrados representam observações de diferentes classes. Deste modo, algoritmos de classificação constroem modelos onde é possível distinguir os dados. A linha separando os círculos dos quadrados representa esse modelo. Caso uma nova observação esteja no topo da figura (acima da linha), o modelo irá classificá-la como círculo. Caso a nova observação esteja na parte de baixo da figura (depois da linha) o modelo irá classificá-la como quadrado. Algoritmos de aprendizado de máquina que podem realizar a classificação incluem o perceptron, algoritmo passivo agressivo, *Support Vector Machines* (SVM), classificadores *ensemble* entre outros. A biblioteca Scikit-Learn, desenvolvida para a linguagem Python, implementa esses e outros algoritmos que realizam classificação[3].

A segunda tarefa que pode ser realizada com aprendizado de máquina é a **regressão**. A principal diferença entre a regressão e a classificação está no tipo da variável alvo (rótulo). Na regressão, o rótulo é composto por uma variável contínua ou numérica. Isso significa que a regressão visa identificar um número, como o preço de um imóvel de 50 metros quadrados ou o valor do salário baseado na quantidade de anos de experiência. A Figura 2.9 ilustra o conceito de regressão. Os círculos apresentam o salário de um indivíduo em relação ao tempo de trabalho. Por exemplo, os algoritmos de regressão podem identificar ou ainda relacionar o total de bytes transferidos por dispositivos com total de requisições realizadas. Assim, quando

[3] `https://scikit-learn.org/stable/supervised_learning.html`

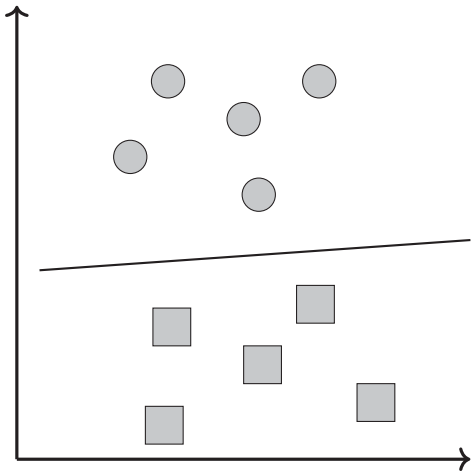


Figura 2.8: Exemplo de Classificação (adaptado de Singh (2019))

uma nova observação estiver disponível, o algoritmo irá buscar na reta onde a nova observação está disposta. Assim, o algoritmo de regressão identifica o valor do salário ou o total de bytes transferidos como resposta para a experiência do novo indivíduo (dispositivo). Algoritmos de aprendizado de máquina que podem realizar a regressão incluem o SVM, *k-Nearest Neighbors* (k-NN), processo gaussiano, árvores de decisão, regressores *ensemble* entre outros. A biblioteca Scikit-Learn implementa esses e outros algoritmos que realizam regressão[2].

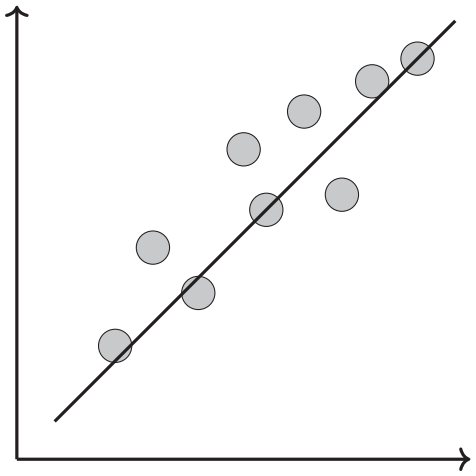
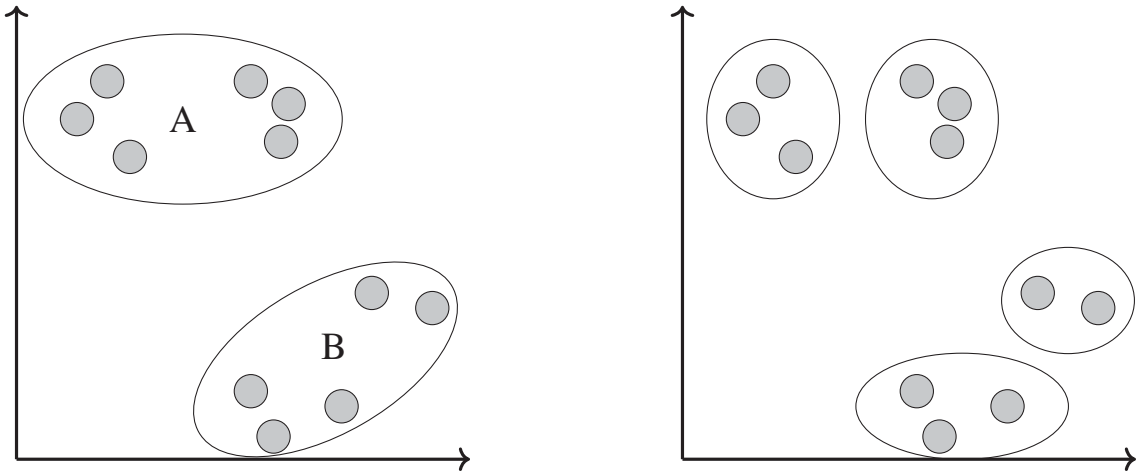


Figura 2.9: Exemplo de Regressão (adaptado de Singh (2019))

A terceira tarefa que pode ser realizada com aprendizado de máquina é a **clusterização**. Diferentemente da regressão e da classificação, os algoritmos de clusterização têm o objetivo de segmentar ou dividir os dados seguindo as propriedades dos dados. Assim, os algoritmos de aprendizado de máquina para a clusterização analisam os dados e possuem como saída agrupamentos de observações com padrões similares. A Figura 2.10 ilustra o conceito da clusterização. Na Figura 2.10(a) os dados observados foram agrupados nos conjuntos A e B. O conjunto A possui seis observações e o conjunto B possui cinco. Caso uma nova observação esteja disposta próxima ao agrupamento A, ela fará parte desse agrupamento. O agrupamento B pode crescer caso uma nova observação seja similar às observações contidas no agrupamento B. Dado limites como o tamanho dos agrupamentos ou quantidade de agrupamentos, é possível obter resultados diferentes. A Figura 2.10(b) apresenta a mesma disposição dos círculos cinza, porém agrupados em quatro grupos diferentes. Esse resultado pode ser obtido adicionando uma

regra onde é necessário encontrar quatro grupos. Algoritmos de aprendizado de máquina que podem realizar a clusterização incluem o *K-means*, DBSCAN, *Spectral clustering* entre outros. A biblioteca Scikit-Learn implementa esses e outros que realizam clusterização[4].



(a) Clusterização com dois agrupamentos

(b) Clusterização com quatro agrupamentos

Figura 2.10: Exemplo de Clusterização (adaptado de Singh (2019))

A clusterização em diferentes agrupamentos (Figura 2.10) pode ser utilizada para resolver problemas relacionados com a segurança de sistemas de computadores. Por exemplo, se o objetivo for identificar os dispositivos maliciosos e os normais, aqueles que não oferecem danos à segurança, é possível utilizar apenas dois agrupamentos (Figura 2.10(a)). Assim, um grupo representará os dispositivos normais e outro representará os maliciosos. Outra abordagem é refinar os agrupamentos para identificar grupos específicos. Por exemplo, é possível existir diferentes perfis de dispositivos normais. Assim, cada perfil compreenderá um agrupamento diferente. Outro exemplo é a possibilidade de existirem múltiplos ataques simultâneos. Isso faria com que mais agrupamentos fossem necessários para separar os dispositivos maliciosos segundo os tipos de ataques (Figura 2.10(b)).

A quarta tarefa que pode ser realizada com aprendizado de máquina é nomeada de **regras de associação**. Nesse contexto, a associação tem o sentido de coocorrência. Isso ocorre, pois o objetivo é minerar dados comerciais em busca de conjuntos das variáveis que aparecem frequentemente. O exemplo clássico da adoção de regras de associação é na análise de vendas em uma loja. Neste caso, as variáveis representam todos os produtos vendidos pela loja. O objetivo é encontrar conjuntos de itens que são adquiridos juntos. Várias decisões estratégicas podem ser embasadas pelos resultados das regras de associação identificadas. Por exemplo, facilitar a compra definindo quais itens devem ficar nas prateleiras próximas. Melhorar a experiência dos usuários ao manusear catálogos de produtos. Segmentar os diferentes tipos de clientes. Além de auxiliar na definição de estratégias de marketing em promoções (Hastie et al., 2009; Singh, 2019). Uma regra relacionada aos ataques DDoS pode ser a utilização de protocolos. Por exemplo, usuários normais usam geralmente o protocolo HTTPS para acessar o Serviço. Durante os ataques DDoS, os *bots* usam os protocolos HTTP. Essa e outras regras identificadas podem auxiliar os administradores de rede a identificar os ataques DDoS. Os algoritmos AIS, SETM e APRIORI são exemplos de algoritmos que podem ser utilizados para realizar o aprendizado por regras de associação (Kumbhare e Chobe, 2014).

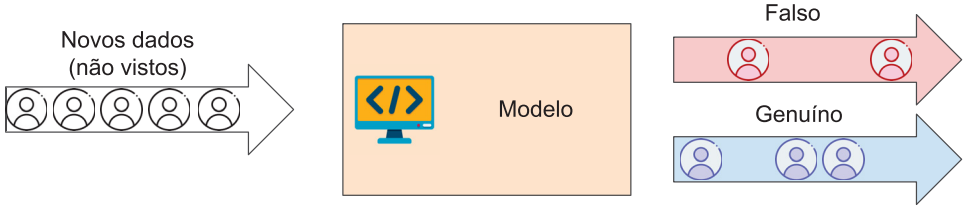
[4] `https://scikit-learn.org/stable/unsupervised_learning.html`

2.4.1.2 *Estratégias do Aprendizado de Máquina*

O ramo Estratégia, o segundo ramo da Figura 2.7, apresenta três tipos de estratégias para aplicar o aprendizado de máquina: o aprendizado supervisionado, o aprendizado não supervisionado, e o aprendizado por reforço. Hastie et al. (2009) e Crisci et al. (2012) definem o **aprendizado supervisionado** como a ação de analisar um problema utilizando variáveis explicativas definidas por $X \in \mathscr{X}$, onde o objetivo é aprender a prever uma variável aleatória $Y \in \mathscr{Y}$. Ou seja, o processo de aprendizado é guiado pela busca de padrões nos dados que descrevem o problema (conjunto $\mathscr{X}$). Esses padrões são relacionados aos rótulos verdadeiros (conjunto $\mathscr{Y}$) dos elementos já conhecidos e vão ser utilizados para identificar os rótulos dos elementos ainda não conhecidos pelo aprendizado de máquina (Singh et al., 2016; Singh, 2019). O aprendizado supervisionado foi utilizado na robótica (Amarjyoti, 2017), em análise de imagens médicas (Erickson et al., 2017), na detecção de transação fraudulenta de cartão de crédito (Dhankhad et al., 2018), na detecção de anomalias em redes de computadores (Omar et al., 2013), entre outros.

O aprendizado supervisionado pode ser utilizado para tarefas de regressão ou de classificação. Independentemente do tipo tarefa, o aprendizado supervisionado acontece em duas fases, a fase de treinamento e a fase de teste. Durante o treinamento, o algoritmo de aprendizado de máquina supervisionado analisa a relação entre os dados e os rótulos para construir modelos capazes de produzir saídas corretas quando for confrontado com novos dados. A fase de teste é utilizada para atestar a qualidade do modelo. Nesta fase o modelo gerado no treinamento é utilizado para catalogar as classes de novos dados. Os rótulos reais desses novos dados são conhecidos pelo desenvolvedor, mas não pelo modelo. Deste modo, os rótulos reais são comparados com os resultados obtidos pelo modelo treinado. Após o treinamento e o teste o modelo pode ser utilizado em produção, onde ele irá receber dados que nem o desenvolvedor conhece o verdadeiro rótulo. A Figura 2.11 apresenta o funcionamento do aprendizado supervisionado. Primeiramente, o algoritmo de aprendizado de máquina selecionado aprende com os dados do conjunto de treinamento (Figura 2.11(a)). Um novo conjunto de dados contendo informações sobre observações não utilizadas no treinamento é apresentado para o modelo treinado (Figura 2.11(b)). O intuito é verificar se o modelo pode identificar corretamente os rótulos das novas observações (Singh, 2019).

(a) Ilustração da fase treinamento no aprendizado supervisionado



(b) Ilustração da fase teste no aprendizado supervisionado

Figura 2.11: Aprendizado Supervisionado (adaptado de Singh (2019))

A segunda estratégia de aprendizado é o **aprendizado não supervisionado**. O aprendizado não supervisionado diversifica as possibilidades e aumenta a relevância do aprendizado de máquina. Isso ocorre, pois o aprendizado não supervisionado visa resolver um dos maiores problemas relacionados ao aprendizado supervisionado, a generalização dos resultados devido ao uso de rótulos. Durante o treinamento, algoritmos de aprendizado supervisionado relacionam os dados observados com os rótulos disponíveis, aprendendo assim sobre a natureza do problema. Porém, existem casos em que os dados observados representam apenas parte do problema (Sutton e Barto, 2018). Caso uma nova observação não siga os padrões pré-estabelecidos na base de dados, o modelo pode produzir saídas incorretas. Uma nova fraude relacionada a cartões de créditos pode não ser detectada antes de causar prejuízos. Bem como um tipo diferente de ataque DDoS pode não ser detectado caso este tipo de ataque difira dos ataques que o algoritmo de aprendizado de máquina foi treinado. Deste modo, o objetivo do aprendizado não supervisionado é inferir relacionamentos entre os dados sem o auxílio dos rótulos reais ou recompensas do ambiente (Dayan et al., 1999; Ghahramani, 2004; Hastie et al., 2009; James et al., 2023).

O aprendizado não supervisionado pode ser realizado por intermédio da clusterização ou das regras de associação (Hastie et al., 2009; Singh, 2019). Apesar das vantagens do aprendizado não supervisionado, escolher o algoritmo e os parâmetros corretos não são triviais (Tomašev e Radovanović, 2016). Como citado anteriormente (Figura 2.10), diferentes parâmetros podem ocasionar em diferentes resultados. A literatura apresenta o uso do aprendizado não supervisionado na medicina (Nattkemper e Wismüller, 2005), no mercado financeiro (de Prado e Lewis, 2019) e em sistemas de detecção de intrusão (Hanselmann et al., 2020), entre outros.

A terceira estratégia de aprendizado é o **aprendizado semi-supervisionado**. O aprendizado semi-supervisionado compreende algoritmos que aprendem com parte dos dados rotulados e parte sem rótulos. Existem cenários onde rotular toda a base de dados não é uma tarefa trivial. Nesses casos, o custo de tempo e dinheiro para rotular toda a base pode ser alto, dificultando o uso do aprendizado de máquina supervisionado. Por outro lado, existem cenários em que é possível melhorar significativamente os resultados apenas com parte da base rotulada. Para diminuir as desvantagens relacionadas à obtenção dos rótulos, e para melhorar os resultados dos algoritmos não supervisionados, a literatura apresenta o aprendizado semi-supervisionado (Zhu e Goldberg, 2009; Zhou e Belkin, 2014). Áreas como a medicina (Yang et al., 2019b), robótica (Xu et al., 2021a) e detecção de anomalias em redes de computadores (Gao et al., 2018) utilizam o aprendizado semi-supervisionado. *Transductive support vector machines* (TSVMs), *co-training*, *Expectation-Maximization* (EM) são exemplos de algoritmos de aprendizado de máquina semi-supervisionado (Pise e Kulkarni, 2008).

A quarta estratégia de aprendizado é o **aprendizado por reforço**. Como no aprendizado não supervisionado, o aprendizado por reforço é uma alternativa nos cenários onde coletar dados de todas as classes é impraticável. Porém, o aprendizado por reforço também difere do aprendizado não supervisionado, pois o objetivo do aprendizado por reforço não é inferir relacionamentos entre as observações sem conhecimento prévio (Sutton e Barto, 2018). O objetivo do aprendizado por reforço é interagir com o ambiente a fim de aprender a lidar com o ambiente. O aprendizado acontece baseado em recompensas e penalidades. A cada geração, os agentes interagem com o ambiente testando-o em busca de aprender a aumentar as recompensas e minimizar as penalidades recebidas. A cada geração, o conhecimento acumulado é repassado para as gerações com o intuito de melhorá-las. Assim, ao utilizar o aprendizado por reforço é possível obter automaticamente habilidades comportamentais que maximizam as recompensas e minimizam as penalidades (Ghahramani, 2004; Sutton e Barto, 2018; Levine et al., 2020). *Q-Learning*, *Temporal Difference Learning* e SARSA, (do inglês, *State-Action-Reward-State-Action*) são exemplos de algoritmos de aprendizado por reforço (Sewak, 2019).

2.4.1.3 Profundidade do Aprendizado de Máquina

O terceiro ramo da Figura 2.7 divide os algoritmos de aprendizado de máquina em relação à profundidade dos algoritmos. *Shallow learning* e *deep learning* são as duas classificações disponíveis na literatura. Aprendizado raso ou aprendizado superficial são traduções para o termo *shallow learning*. Os termos raso ou superficial se referem ao modo que os algoritmos utilizam para gerar os modelos de aprendizado de máquina. Portanto, esse nome não deve ser interpretado como um demérito desses algoritmos, pois não tem relação com a qualidade dos resultados obtidos. Assim, para evitar interpretações dúbias acerca do ramo profundidade, este trabalho utiliza apenas os termos em inglês (*shallow learning* e *deep learning*).

Shallow learning compreende algoritmos de aprendizado de máquina tradicionais que não utilizam várias camadas ocultas (Liu e Lang, 2019; Kaluarachchi et al., 2021). O número de camadas ocultas não é unanimidade na literatura, porém algoritmos que utilizam menos de duas camadas ocultas são geralmente associados ao *shallow learning* (Kawaguchi, 2016). SVM, k-NN, *Naïve Bayes*, redes neurais artificiais (do inglês, *Artificial Neural Network* - ANN) e árvores de decisão são exemplos de algoritmos classificados como *shallow learning* (Liu e Lang, 2019). Como cada algoritmo tem sua peculiaridade, é possível utilizar o *shallow learning* em várias áreas. Recentemente o *shallow learning* vem sendo empregado na predição do consumo de energia elétrica (Shao et al., 2020), na descoberta de novos remédios (Houssein et al., 2020), na medicina (Nilashi et al., 2020), entre outros.

Nos últimos anos, algoritmos de ***deep learning*** têm evoluído muito (Pak e Kim, 2017; Mahmoud et al., 2019; Liu e Lang, 2019). Zhang et al. (2018) definem *deep learning* como o processo de aprender a relação entre várias variáveis, a relação que governa as variáveis e o conhecimento que dá significado para a relação entre as variáveis. O *deep learning* diferencia-se do *Shallow Learning* pois compreende algoritmos que utilizam várias camadas ocultas para realizar o aprendizado (Liu e Lang, 2019; Kaluarachchi et al., 2021). Deste modo, a principal diferença entre os tipos de aprendizado reside no número de transformações que os dados de entrada sofrem até alcançar a saída (Wang et al., 2020). Um grande benefício que os algoritmos de *deep learning* possuem é o fato de não precisar de seleção de atributos. Em muitos casos, a seleção de atributos pode melhorar o resultado obtido pelos algoritmos de aprendizado de máquina. Porém, se a seleção de atributos não for realizada corretamente, esta pode enviesar os resultados ou até os prejudicar. Além disso, é necessário despender tempo para realizar essa ação. Como o *deep learning* pode aprender a representação dos atributos apenas observando os dados originais, este consegue escolher quais os atributos mais relevantes para atingir os melhores resultados (Liu e Lang, 2019).

Para casos em que grandes volumes de dados estejam disponíveis, é possível que algoritmos de *deep learning* apresentem resultados melhores em comparação com algoritmos de *shallow learning* (Liu e Lang, 2019). Isso ocorre, pois os algoritmos de *deep learning* comparam os dados reais com a saída gerada pelo algoritmo. É esperado que no início do treinamento o resultado esteja longe do real. Porém, ao longo das iterações, o *deep learning* vai ajustando as configurações e os resultados começam a se aproximar dos reais. Como ocorrem muitas iterações, é necessário poder computacional suficiente para lidar com todas as iterações (Janos, 2020).

Os algoritmos de *deep learning* são utilizados em várias áreas como na geração de imagens (Dosovitskiy e Brox, 2016), robótica (Amarjyoti, 2017) e no reconhecimento de imagens. Especialmente no reconhecimento de imagens, Wu e Chen (2015) obtiveram altas acurácias com o uso de *deep learning* para o reconhecimento de caracteres escritos à mão. Outro exemplo de uso de algoritmos de *deep learning* foi no combate à COVID-19. Song et al. (2021) desenvolveram um sistema que diferencia tomografias computadorizadas de pacientes com a doença de pacientes saudáveis. Redes neurais convolucionais (do inglês, *convolutional neural network* - CNN), *deep*

belief network (DBN), *deep neural network* (DNN), e a *generative adversarial network* (GAN) são algoritmos classificados como *deep learning* (Liu e Lang, 2019).

2.4.2 Medidas de Avaliação do Aprendizado de Máquina

Ao fim do processo de treinamento dos algoritmos de aprendizado de máquina é imprescindível avaliar o desempenho. Em geral, os algoritmos de aprendizado que realizam tarefas de classificação são submetidos a testes para identificar o rótulo real de observações não utilizadas no treinamento. A partir desse teste é possível extrair a matriz de confusão. Para problemas com duas classes, a matriz de confusão possui duas linhas e duas colunas (Tabela 2.1). As classes são chamadas genericamente de positivo e negativo. A partir dos resultados apresentados na matriz de confusão é possível identificar algumas medidas para mensurar a qualidade do modelo. A primeira medida é a quantidade de verdadeiros positivos (VP) que o algoritmo gerou. Para gerar um VP é necessário que o sistema rotule uma observação como pertencente a classe positiva e o rótulo real também seja positivo. A segunda medida é o verdadeiro negativo (VN). Similar ao VP, o VN acontece quando o modelo classifica corretamente uma observação da classe negativa. VP e VN são os dois casos de acertos, porém a matriz de confusão também apresenta os erros. O falso positivo (FP) ocorre quando o modelo de aprendizado de máquina rotula uma observação com o rótulo positivo, mas o rótulo real é negativo. O falso negativo (FN) ocorre quando o rótulo real é positivo, mas o algoritmo de aprendizado de máquina o rotula como negativo.

Tabela 2.1: Composição da Matriz de Confusão

Matriz de confusão		Classe real	
		Positivo	Negativo
Classe hipotética	Positivo	VP	FP
	Negativo	FN	VN

Uma das medidas de avaliação mais utilizadas é a acurácia. A acurácia divide o total de acertos positivos (VP) e negativos (VN) pelo total de observações da base. A Fórmula 2.1 apresenta o modo para calcular a acurácia. A acurácia é uma medida adequada para quantificar a qualidade dos modelos de aprendizado de máquina, mas em alguns casos pode ser interpretada equivocadamente. Alguns problemas são naturalmente desbalanceados. Por exemplo, a quantidade de fraudes de cartão de crédito representa pequena parte do conjunto total de transações. Antes do lançamento de um ataque DDoS, a quantidade de *bots* é menor que a quantidade de dispositivos normais. Depois do início do ataque, a quantidade de tráfego gerado pelos *bots* supera a quantidade de tráfego gerado pelos usuários normais. Deste modo, apresentar acurácias próximas a 100% não garante que o modelo de aprendizado de máquina é adequado. Pois ele pode ter errado a rotulação de todas as observações da classe minoritária. A Tabela 2.2 apresenta um exemplo de um modelo de aprendizado de máquina que atingiu acurácia de 99% em um problema desbalanceado. No exemplo da figura existem 100 observações, 99 da classe positiva e uma da classe negativa. O modelo de aprendizado de máquina rotulou todas as observações na classe positiva. Seguindo a Fórmula 2.1 esse modelo tem 99% de acurácia $(99+0/(99+1+0+0) = 99\%)$. Porém, a única observação da classe minoritária, classe negativa, foi incorretamente rotulada.

$$Acurácia = \frac{VP + VN}{VP + FP + FN + VN} \tag{2.1}$$

Tabela 2.2: Matriz de Confusão para um Exemplo com Classes Desbalanceadas

Matriz de confusão		Classe real	
		Positivo	Negativo
Classe hipotética	Positivo	99	1
	Negativo	0	0

Deste modo, é oportuno complementar a análise dos algoritmos de AM com outras medidas. A Fórmula 2.2 apresenta o modo para calcular a precisão de um modelo de aprendizado de máquina. A precisão é obtida a partir da divisão dos VP com a soma dos VP com os FP. O valor de 100% para a precisão indica que todas as observações rotuladas pelo modelo de aprendizado de máquina como sendo da classe positiva realmente eram da classe positiva. Assim, a precisão avalia o quanto o modelo de aprendizado de máquina é preciso quanto a classificação das observações da classe positiva. É possível obter a precisão para a classe negativa, para isso basta usar substituir os termos VP e FP por VN e FN respectivamente.

$$Precisão = \frac{VP}{VP + FP} \tag{2.2}$$

Outra medida comumente utilizada é o *recall*. Algumas traduções apresentam o *recall* como revocação ou sensibilidade. Para evitar interpretações incorretas, este trabalho utiliza o termo em inglês. O *recall* complementa a precisão analisando a relação entre todas as observações do tipo positivo e quantas observações do tipo positivo o modelo de aprendizado de máquina rotulou corretamente. Com o *recall* é possível verificar o quão sensível às observações da classe positiva o modelo é. O valor de 100% no *recall* indica que o modelo acertou todas as rotulações para a classe positiva. A Fórmula 2.3 apresenta o modo para calcular o *recall*. Para obter o *recall* para a classe negativa basta substituir os termos VP e o FN por VN e FP respectivamente. Em geral, obter altas taxas de precisão e *recall* é o objetivo dos desenvolvedores, porém pode não ser uma tarefa trivial.

$$Recall = \frac{VP}{VP + FN} \tag{2.3}$$

A medida *F1-score* foi desenvolvida para facilitar a visualização da relação entre precisão e *recall*. O *F1-score* é a média harmônica entre precisão e *recall* (Fórmula 2.4). Assim, o *F1-score* apresenta em uma única medida um bom indicativo sobre qualidade do modelo. Por fim, a área sob a curva (do inglês *area under the curve* - AUC) *receiver operating characteristic* (ROC) também podem complementar a análise dos resultados. O valor de AUC igual a 1 significa que o modelo de aprendizado classificou todas as amostras corretamente. É necessário criar a curva ROC para calcular a medida AUC. A curva ROC é baseada em diferentes valores de limite, taxas de verdadeiros positivos e falsos positivos. Portanto, a AUC condensa a relação entre limiares, taxa de verdadeiros positivos e taxa de falsos positivos em apenas uma medida.

$$F1 - score = 2. \frac{Precisão\ .\ recall}{Precisão + recall} \tag{2.4}$$

As medidas de avaliação citadas anteriormente demandam dados rotulados durante os testes para ser possível quantificá-las (calculá-las). Contudo, ao usar o aprendizado de máquina não supervisionado, é plausível que os rótulos originais não estejam disponíveis para que essas medidas possam ser calculadas. Assim, a literatura propõe medidas para avaliar a qualidade da

clusterização realizada pelo aprendizado de máquina não supervisionado. O índice de silhueta (do inglês, *silhouette index*), o índice de Calinski–Harabasz e o índice de Davies–Bouldin são medidas clássicas para avaliar o resultado do aprendizado de máquina não supervisionado. A literatura evoluiu esses índices clássicos propondo novos índices como o S_Dbw e o CDbw.

O índice de silhueta avalia a qualidade da clusterização analisando propriedades entre e intra-clusters. Quanto mais distantes os clusters estão uns dos outros (propriedade entre clusters) e quanto mais compactos estão os clusters (propriedade intra-clusters) maior será o valor do índice de silhueta. A distância euclidiana pode ser utilizada para verificar se os clusters estão compactos e/ou distantes. O resultado do índice de silhueta varia entre [-1 e +1]. Quanto mais próximo de +1, melhor é o resultado da clusterização, pois as distâncias intra-clusters são pequenas, representando clusters "bem agrupados", e as distâncias entre-clusters são grandes, indicando que os clusters estão "distantes" uns dos outros. Resultados próximos de zero indicam que clusters podem estar sobrepostos, pois as amostras estão próximas. Assim, as amostras poderiam estar alocadas em diferentes clusters. Por fim, o -1 indica que, em geral, as amostras foram clusterizadas erroneamente. Pois as amostras estão mais próximas de um cluster diferente da qual foi clusterizada (Rousseeuw, 1987). A Fórmula 2.5 define o cálculo do índice de silhueta. Onde, k representa todos os clusters identificados, k_i representa o i-ésimo cluster, n_i é o número de elementos em k_i. O termo $b(x)$ é a distância média ao cluster mais próximo a cada amostra x e o termo $e(x)$ indica a distância média a todos os pontos do cluster ao qual x faz parte. A Figura 2.12 apresenta graficamente o conceito do índice de silhueta incluindo os termos $e(x)$ e $b(x)$ para o caso em que o x pertence ao cluster A (quadrado vermelho no cluster A). Graficamente, $b(x)$ é a média das linhas azuis tracejadas entres os clusters A e B, pois o cluster B é o mais próximo da amostra i que está sendo analisada na figura. O termo $e(x)$ é dado pela média da distância euclidiana de todas as linhas no cluster A. Esse processo é repetido para todas as amostras de todos os clusters e uma média aritmética simples dos resultados indica o índice de silhueta da clusterização (Rousseeuw, 1987; Liu et al., 2010; Poulakis, 2020; Şenol, 2022). Por fim, a biblioteca Scikit-learn possui uma implementação do índice de silhueta para a linguagem de programação Python[5].

$$\textit{Índice de silhueta} = \frac{1}{k}\sum_{i}\{\frac{1}{n_i}\sum_{x\in k_i}\frac{b(x)-e(x)}{max\{e(x),b(x)\}}\} \tag{2.5}$$

O índice Calinski–Harabasz é outra medida para avaliar a qualidade da clusterização que também utiliza propriedades entre e intra-clusters. A diferença entre o índice Calinski–Harabasz para com o índice de silhueta é que o índice Calinski–Harabasz baseia-se na soma média dos quadrados das distâncias entre e dentro dos clusters. O índice Calinski–Harabasz não tem um limite superior. Maiores valores para o índice Calinski–Harabasz indicam que os clusters estão densos (amostras nos clusters estão "próximas") e os clusters estão longes uns dos outros ("bem separados"). Portanto, quanto maior for o valor desse índice, melhor é a clusterização. A Fórmula 2.6 define o modo para calcular o índice Calinski–Harabasz. Onde, k representa todos os clusters identificados, k_i representa o i-ésimo cluster, n é o total de elementos em toda a base de dados, n_i é o número de elementos em k_i. O termo d^2 representa a distância euclidiana ao quadrado. A distância é medida de duas formas, a primeira compara o c_i, o qual é o centro do i-ésimo cluster (k_i) e o centro da base de dados (c). A segunda distância é entre cada elemento do cluster (x) com o centro do cluster (c_i) (Caliński e Harabasz, 1974; Liu et al., 2010; Poulakis, 2020; Şenol, 2022). A Figura 2.13 auxilia no entendimento do índice. O numerador da

[5] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

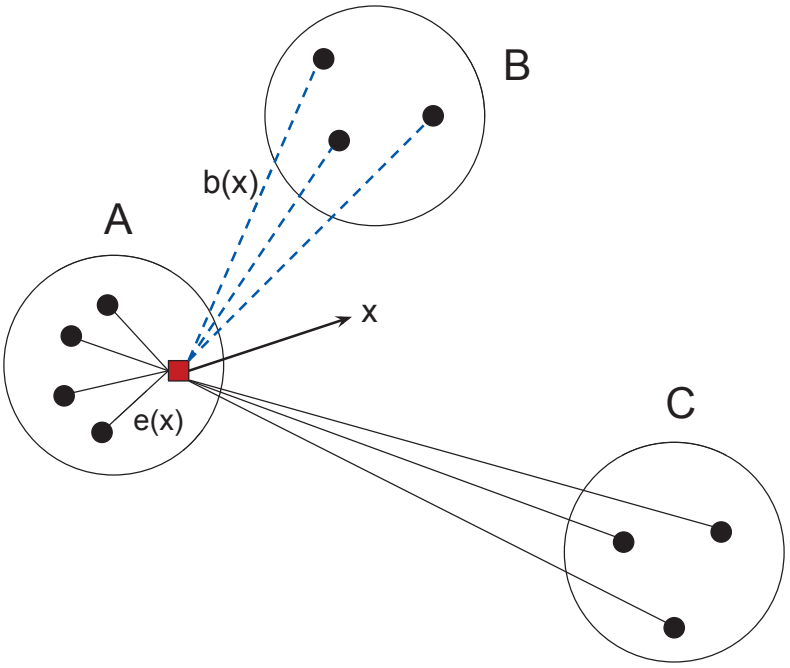


Figura 2.12: Ilustração do Índice de Silhueta (adaptado de Rousseeuw (1987))

Fórmula 2.6 foi representado graficamente pela manipulação das linhas pontilhadas que ligam os centros dos clusters, representados pelos triângulos, até o centro da base de dados, representado pela estrela. Assim, tem-se a soma do quadrado da média das distâncias dos centros dos clusters até o centro da base de dados. O denominador é obtido pela manipulação das linhas nos clusters seguindo a Fórmula 2.6. Assim, tem-se a soma do quadrado da média das distâncias de todas as amostras para os centros dos respectivos clusters. Por fim, a biblioteca Scikit-learn possui uma implementação do índice Calinski–Harabasz para Python[6].

$$Índice\ Calinski - Harabasz = \frac{\sum_{i=1}^{k} n_i d^2(c_i, c)/(k-1)}{\sum_{i=1}^{k} \sum_{x \in k_i} d^2(x, c_i)/(n-k)} \tag{2.6}$$

O índice Davies-Bouldin também utiliza propriedades entre e intra-clusters para avaliar a qualidade da clusterização. Este índice baseia-se na similaridade entre os clusters. A Figura 2.14 representa graficamente o conceito base do índice Davies-Bouldin. No exemplo da figura, o cluster analisado é o cluster A. Primeiramente, é necessário calcular a média da distância de todos os elementos do cluster A para com o centro do cluster A (linhas no cluster A). O resultado é somado com a média da distância dos elementos do cluster B para com o centro do cluster B (linhas no cluster B). Esse resultado é dividido pela distância entre os centros dos clusters A e B (linha pontilhada entre A e B). O mesmo processo é repetido entre os clusters A e C. O maior valor obtido entre os pares A-B e A-C é o marcado como o índice Davies-Bouldin para o cluster A por representar o cluster mais similar. Todo o processo é repetido para todos os pares de clusters, ao final do processo, o índice Davies-Bouldin geral é definido pela média dos índices Davies-Bouldin de cada cluster. Diferentemente dos casos anteriores, onde valores maiores indicam melhores clusterizações, no índice Davies-Bouldin menores valores indicam as melhores clusterizações. Pois o ideal é que os clusters tenham o mínimo de similaridade entre si (Halkidi

[6] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.calinski_harabasz_score.html

[7] https://velog.io/@hyangki0119/Clustering-analysis-Key-concepts-and-Implementation-in-Python

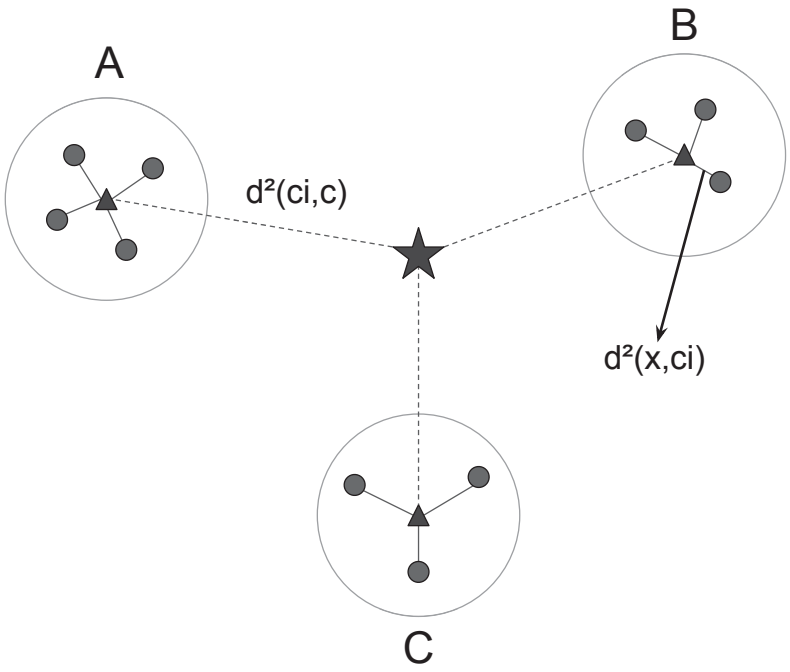


Figura 2.13: Ilustração do Índice Calinski–Harabasz (adaptado de[7])

et al., 2001). A Fórmula 2.7 define o modo para calcular o índice Davies-Bouldin para k clusters (média dos índices Davies-Bouldin de cada cluster). Onde o k_i representa o i-ésimo cluster, n_i é o número de elementos em k_i. A expressão $\frac{1}{n_i}\sum_{x\in k_i} d(x,c_i)$ calcula, para cada cluster, a média das distâncias de cada elemento do cluster (x) para com o centro cluster c_i (linhas nos clusters na Figura 2.14). Esse resultado é somado com a média da distância ($\frac{1}{n_j}\sum_{x\in k_j} d(x,c_j)$) para o cluster j. O termo $d(c_i,c_j)$ indica a distância do centro do cluster que está sendo analisado (c_i) para o centro do cluster j (c_j) (linhas pontilhadas na Figura 2.14) (Davies e Bouldin, 1979; Liu et al., 2010; Poulakis, 2020; Şenol, 2022). A média dos maiores valores de similaridade definem o valor do índice Davies-Bouldin. Por fim, a biblioteca Scikit-learn possui uma implementação do índice Davies-Bouldin para Python[8].

$$\textit{Índice Davies} - \textit{Bouldin} = \frac{1}{k}\sum_{i}\max_{j,j\neq i}\{[\frac{1}{n_i}\sum_{x\in k_i} d(x,c_i) + \frac{1}{n_j}\sum_{x\in k_j} d(x,c_j)]/d(c_i,c_j)\} \quad (2.7)$$

O índice S_Dbw baseia-se na variação intra-cluster e na densidade entre clusters, sendo definido como a soma da dispersão média nos clusters e da densidade entre os clusters (Halkidi e Vazirgiannis, 2001; Liu et al., 2010; Poulakis, 2020). A Fórmula 2.8 define que o índice S_Dbw combina a compacidade e separação para avaliar os resultados do agrupamento. O primeiro termo $Scat(k)$, indica o espalhamento médio das amostras dos k clusters. Assim, valores pequenos para $Scat(k)$ indicam clusters compactos. Já o $Dens\_bw(k)$ representa o número médio de pontos entre os clusters, oferecendo uma medida da densidade entre clusters em comparação com a densidade dos clusters. Um valor reduzido de $Dens\_bw(k)$ sugere clusters bem separados (Halkidi e Vazirgiannis, 2001). Assim, quanto menor for o valor final do índice

[8]https://scikit-learn.org/stable/modules/generated/sklearn.metrics.davies_bouldin_score.html

[9]https://pyshark.com/davies-bouldin-index-for-k-means-clustering-evaluation-in-python/

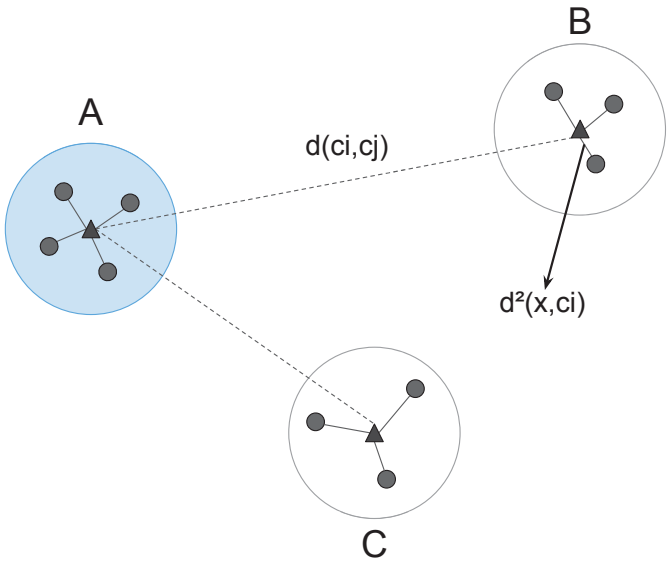


Figura 2.14: Ilustração do Índice Davies-Bouldin (adaptado de[9])

S_Dbw, melhor é a clusterização obtida (Halkidi e Vazirgiannis, 2001; Liu et al., 2010; Poulakis, 2020). A implementação do índice S_Dbw está disponível no PYPI[10], um repositório de software para a linguagem Python.

$$Índice\ S\_Dbw(k) = Scat(k) + Dens\_bw(k) \tag{2.8}$$

O índice densidade composta entre e dentro dos clusters (do inglês, *composed density between and within clusters* - CDbw) é outro índice de avaliação de clusters desenvolvidos pelos mesmos autores do S_Dbw (Halkidi e Vazirgiannis, 2008). O índice CDbw utiliza a coesão, compacidade e separação dos clusters, critérios essenciais para avaliar agrupamentos. Isso propicia robustez na análise dos resultados referentes à qualidade da clusterização. A Fórmula 2.9 define o modo de obtenção deste índice. Valores elevados para ambos os termos da Fórmula 2.9, ou seja, coesão e separação em relação à compacidade, são alcançados quando o agrupamento exibe clusters compactos, bem separados e uma baixa variação na distribuição de densidade nos clusters. Assim, quanto maior for o valor do índice CDbw, melhor é a clusterização obtida (Halkidi e Vazirgiannis, 2008; Poulakis, 2020). Na Fórmula 2.9, Halkidi e Vazirgiannis (2008) definem o termo $Coesão(k)$ como a densidade dos clusters em relação às variações de densidade observadas internamente a eles. Já o termo $CS(k)$ (Compactação em relação à Separação) avalia a compactação dos clusters relacionando-a com a separação dos clusters (Halkidi e Vazirgiannis, 2008). A implementação do índice CDbw está disponível no PYPI[11] para a linguagem Python.

$$Índice\ CDbw(k) = Coesão(k) \cdot SC(k) \tag{2.9}$$

2.4.3 Automated Machine Learning

Com base nos esforços de especialistas em aprendizado de máquina, a literatura definiu o *Automated Machine Learning* (AutoML), uma área de pesquisa cujo objetivo é democratizar, simplificar e reduzir o custo do aprendizado de máquina. *Frameworks* AutoML visam encontrar

[10] https://pypi.org/project/s-dbw/
[11] https://pypi.org/project/cdbw/

e configurar algoritmos de aprendizado de máquina que minimizem erros de classificação para o conjunto de dados utilizados pelo usuário (Feurer et al., 2015). Os *frameworks* AutoML atuam como especialistas em aprendizado de máquina para sugerir o algoritmo de aprendizado de máquina adequado para cada conjunto de dados. Assim, o AutoML acelera e automatiza o processo de obtenção de modelos de aprendizado de máquina (Isingizwe et al., 2021).

Atualmente, existem *frameworks* AutoML de dois tipos, o primeiro é do tipo otimização de hiperparâmetros (do inglês, *Hyperparameter Optimization* - HPO) e o segundo é do tipo busca de arquiteturas de redes neurais (do inglês, *Neural Architecture Search* - NAS). A literatura apresenta diferentes *frameworks* AutoML do tipo HPO. Exemplos de *frameworks* de AutoML de código aberto são: AutoGluon AutoML (Holland et al., 2021), AutoML mljar-supervised (Isingizwe et al., 2021), Auto-sklearn (Feurer et al., 2019), H2O AutoML (Thatipalli et al., 2022) e o *Tree-based Pipeline Optimization Tool* (TPOT) (Ruijun et al., 2021; Liuliakov e Hammer, 2021). Singh et al. (2022) e Horsanali et al. (2021) criaram seus próprios *frameworks* AutoML do tipo HPO. Em Horsanali et al. (2021), os autores definiram que o *framework* iria avaliar seis algoritmos de aprendizado de máquina: Decision Tree, K-nearest neighbors, Logistic Regression, Naive Bayes, Random Forest, e o SVM. Assim, o *framework* AutoML proposto treina e testa todos os algoritmos com os mesmos dados. O algoritmo de aprendizado de máquina que maximiza a acurácia é selecionado.

Cada *framework* AutoML do tipo HPO define sua estratégia para selecionar e configurar os algoritmos de aprendizado de máquina adequados para o contexto dos dados explorados pelo usuário. Isto permite que os *frameworks* AutoML apresentem propriedades únicas, como tempo de execução, algoritmos de aprendizado de máquina avaliados e linguagem de programação (Feurer et al., 2015). A Figura 2.15 mostra o funcionamento geral dos *frameworks* AutoML do tipo HPO. Na Etapa 1, os *frameworks* AutoML do tipo HPO definem os algoritmos de aprendizado de máquina candidatos. Os algoritmos candidatos podem variar entre os diferentes *frameworks*. Por exemplo, os *frameworks* AutoML podem usar todos os algoritmos implementados por uma biblioteca (por exemplo, Scikit-learn e Weka) ou restringir o espaço de pesquisa a um conjunto de algoritmos que funcionam bem na maioria dos casos.

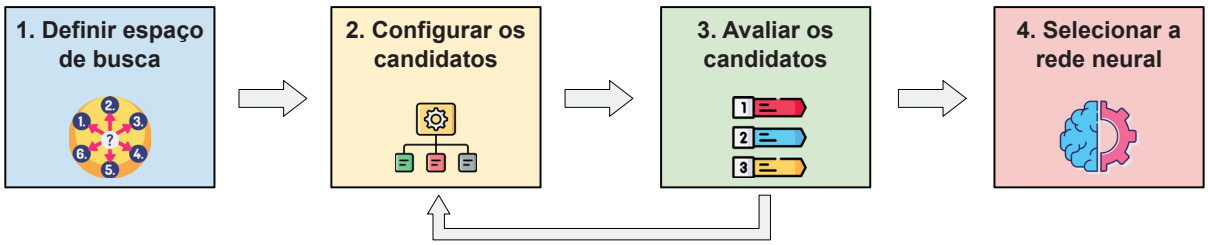


Figura 2.15: Operação Geral dos *Frameworks* AutoML (adaptado de Ren et al. (2021))

A Etapa 2 visa configurar um subconjunto ou todos os algoritmos de aprendizado de máquina candidatos. Normalmente, esta etapa usa algum processo de otimização como a otimização bayesiana (Feurer et al., 2019). Assim, não é necessário que os *frameworks* AutoML avaliem todas as combinações de configuração de algoritmos de aprendizado de máquina. Na Etapa 3, os *frameworks* AutoML treinam e testam algoritmos de aprendizado de máquina candidatos configurados usando o conjunto de dados selecionado pelo usuário do *framework*. Os resultados da acurácia podem ser um critério para avaliar algoritmos (Imran et al., 2021; Xu et al., 2021b). Porém, é comum que o usuário escolha diferentes critérios de avaliação, como precisão, *recall* ou F1-score.

Ao final da etapa de avaliação, o *framework* retorna à Etapa 2 para que os algoritmos de aprendizado de máquina recebam novas configurações para maximizar os critérios de avaliação.

O ciclo entre as Etapas 2 e 3 se repete até que o *framework* encontre um critério de parada. O critério de parada pode ser o tempo de execução ou o número de iterações. Por exemplo, no Auto-sklearn, um *framework* AutoML do tipo HPO, o critério de parada é 60 minutos. Portanto, se o usuário não alterar este parâmetro, o Auto-sklearn será executado por 60 minutos. Na Etapa 4, o *framework* AutoML escolhe o algoritmo de aprendizado de máquina que maximiza os critérios de avaliação. Alguns *frameworks* AutoML podem combinar algoritmos de aprendizado de máquina que maximizam os critérios de avaliação para construir um conjunto de algoritmos de aprendizado de máquina ideais para os dados selecionados pelos usuários.

O AutoML do tipo NAS difere do HPO em termos dos algoritmos de aprendizado de máquina usados. Os *frameworks* AutoML do tipo HPO concentram-se em algoritmos de aprendizado de máquina do tipo *shallow learning*, enquanto os *frameworks* NAS usam algoritmos de aprendizado de máquina do tipo *deep learning*. Tanto os *frameworks* AutoML do tipo HPO quanto o NAS têm o mesmo propósito, identificar e configurar o algoritmo de aprendizado de máquina adequado conforme a necessidade dos usuários. Alguns *frameworks* que implementam o AutoML do tipo NAS são o Autokeras (Imran et al., 2021), o MetaQNN (Xu et al., 2021b), o *Neural Architecture Search Network* (NASNet) e o *Mobile Neural Architecture Search Network* (MNasNet) (Lam e Abbas, 2020). A Google possui o AutoML Vision, um produto comercial que implementa um AutoML do tipo NAS. Além de ser um *framework* pago, com período de teste gratuito, o *framework* do Google limita a customização da execução do AutoML. Por exemplo, o usuário não pode limitar o espaço de pesquisa do AutoML selecionando apenas um conjunto de algoritmos (Ioulianou et al., 2022).

O AutoML do tipo NAS é especializado em selecionar e configurar algoritmos de aprendizado de máquina do tipo *deep learning* para maximizar a acurácia. Para isso, os *frameworks* AutoML do tipo NAS definem a arquitetura dos modelos *deep learning*. O *deep learning* opera com redes neurais, cuja arquitetura inclui o número de camadas ocultas, pesos, número de neurônios e funções de ativação (Lam e Abbas, 2020; Imran et al., 2021). Portanto, para definir a arquitetura das redes neurais, os *frameworks* AutoML do tipo NAS escolhem a combinação de componentes que maximizam a acurácia. Apesar da automação que os *frameworks* AutoML do tipo NAS apresentam, os *frameworks* requerem tempo para identificar a arquitetura apropriada. Em Lam e Abbas (2020), os autores executaram o NASNet por 24 horas e o MNasNET por três horas para encontrar a arquitetura adequada.

Cada *framework* AutoML do tipo NAS define sua estratégia para selecionar a arquitetura de rede neural que minimiza erros. A Figura 2.15, apresentada anteriormente, expõe o funcionamento geral de *frameworks* AutoML do tipo NAS. Na Etapa 1, cada *framework* define o espaço de busca com um conjunto potencialmente grande de arquiteturas de redes neurais (Lam e Abbas, 2020). Na Etapa 2, cada *framework* usa uma estratégia diferente para selecionar a função de ativação, o número de camadas ocultas, os pesos e o número de neurônios para as arquiteturas candidatas (Lam e Abbas, 2020; Imran et al., 2021). Assim, ao final da Etapa 2, cada *framework* possui arquiteturas candidatas para minimizar erros no conjunto de dados inserido pelo usuário.

O *framework* avalia cada arquitetura candidata usando o conjunto de dados inserido pelo usuário na Etapa 3. O *framework* usa as arquiteturas selecionadas para minimizar erros para criar novas arquiteturas candidatas. Portanto, o processo de seleção da arquitetura da rede neural retorna à Etapa 2 e avalia as novas arquiteturas na Etapa 3. O *framework* repete a iteração entre as Etapas 2 e 3 até atingir o critério de parada. O critério de parada pode ser o tempo de execução do *framework* ou a quantidade de iterações. Por exemplo, Autokeras, um *framework* AutoML do tipo NAS, define o critério de parada como um máximo de 100 tentativas. Portanto, caso o usuário não edite este parâmetro, o Autokeras repetirá os passos 2 e 3 até 100 vezes. Na

Etapa 4, o *framework* seleciona a arquitetura que minimiza erros durante o processo de avaliação e a sugere ao usuário (Lam e Abbas, 2020).

2.5 SINAIS PRECOCES DE ALERTA

O estudo de sinais precoces de alerta baseia-se em conceitos como sistemas com equilíbrio estáveis, sistemas que mudam de comportamento e na análise de séries temporais. Um exemplo clássico de sistemas com equilíbrio estáveis é a temperatura no planeta Terra. O equilíbrio estável da temperatura ocorre quando o planeta ganha calor da radiação solar e perde parte desse calor enviando-o para a atmosfera. Esse equilíbrio indica que mesmo que as condições (parâmetros) variem, o sistema tende a compensar essa variação e tende a retomar o equilíbrio. Por exemplo, se a Terra apresentasse um comportamento mais quente que o ponto de equilíbrio, a quantidade de calor da radiação solar que a Terra recebe iria manter-se a mesma. Contudo, o planeta tenderia ao equilíbrio da temperatura apenas enviando mais calor para a atmosfera. Caso a Terra apresente temperaturas mais baixas, a tendência é que menos calor esteja disponível para ser irradiado para a atmosfera, mantendo o sistema equilibrado (Scheffer, 2009).

Outro exemplo é a população hipotética que atingiu o equilíbrio populacional frente à capacidade do ambiente de proporcionar os recursos para a população viver. Esse equilíbrio é verificado pela taxa de mortalidade e a taxa de natalidade. Caso a densidade populacional exceda a capacidade do ambiente em produzir recursos, a diminuição na taxa de natalidade iria equilibrar o sistema. Caso um evento adverso assole parte da comunidade, a quantidade de recursos disponíveis seguiria inalterada. A taxa de nascimento tenderia a crescer devido à fartura de recursos, levando o sistema ao equilíbrio. A Figura 2.16 ilustra graficamente o conceito de equilíbrio. Na figura, o círculo está equilibrado no ponto mais baixo da curva. Mesmo que eventos (mudanças nas condições/parâmetros observados) movam o círculo para a direita ou para a esquerda, a tendência é de retorno ao ponto do equilíbrio (Scheffer, 2009).

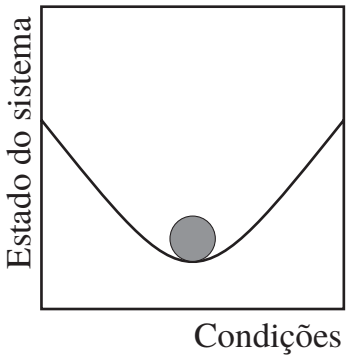


Figura 2.16: Ilustração do Conceito de Equilíbrio (adaptado de Scheffer (2009))

Scheffer (2009) cita que conforme as condições mudam, alguns sistemas podem transitar de um ponto de equilíbrio para outro ponto de equilíbrio. O novo ponto de equilíbrio pode manter o comportamento do sistema, por ser normal que os sistemas tenham múltiplos pontos de equilíbrio. A Figura 2.17 apresenta um sistema com dois pontos de equilíbrio. No exemplo da figura, o sistema está equilibrado no estado A. Porém, a mudança nas condições pode obrigar o sistema a transitar do estado A para o estado B. O novo ponto de equilíbrio pode apresentar boas propriedades ou apresentar uma mudança sem volta. Por exemplo, uma família pode construir um caminho para sair da pobreza utilizando pequenos empréstimos. Além disso, é possível salvar um lago contendo a superpopulação de peixes por meio de grandes esforços de pesca. Os recifes de coral do Caribe apresentam um exemplo de mudança negativa. Mesmo com várias mudanças

externas como um furacão dizimando grande parte do recife, em alguns meses o recife de corais começou a se regenerar. Porém, com a mortalidade em massa de ouriços-do-mar presentes na localidade, uma espécie de algas marinhas ficou sem predadores naturais. Consequentemente, o crescimento populacional dessas algas marinhas dizimou o recife de corais, desordenando o equilíbrio natural. Mesmo após muito tempo da tomada do recife pelas algas, os recifes de corais ainda não conseguiram retomar o ponto de equilíbrio (Scheffer, 2009).

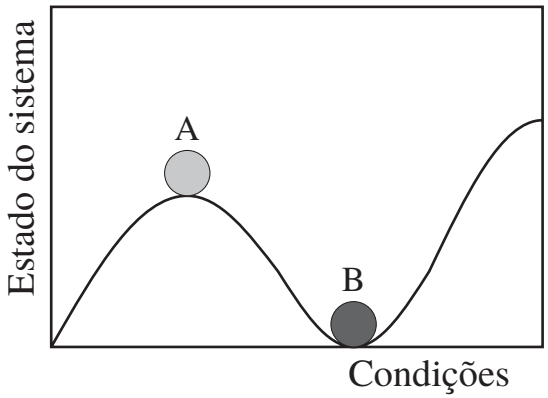
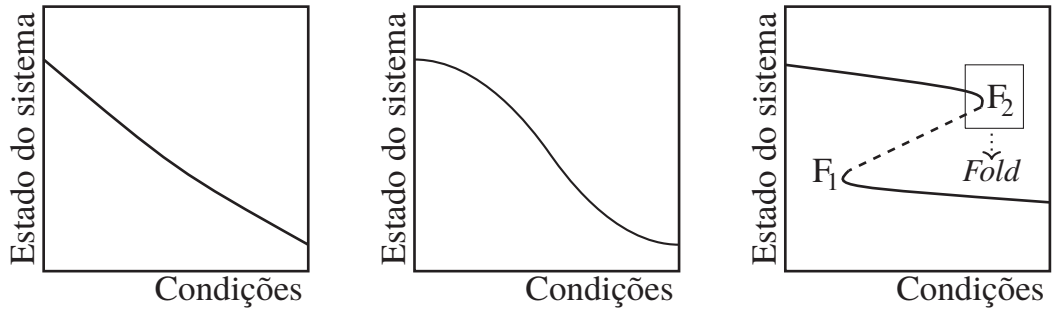


Figura 2.17: Ilustração do Conceito de Sistema com Múltiplos Pontos de Equilíbrio (adaptado de Scheffer (2009))

Um sistema pode transitar entre os estados de equilíbrio de diferentes formas frente às mudanças ocorridas nas condições observadas. A Figura 2.18 apresenta três exemplos de transições. Um sistema em equilíbrio pode transitar suavemente para outro estado de equilíbrio. A Figura 2.18(a) apresenta a suave transição do sistema observado frente às mudanças das condições. Por outro lado, a mudança entre estados de equilíbrio pode ser abrupta, assim como na Figura 2.18(b). Isso significa que, algumas mudanças nas condições impactam mais o sistema do que outras. Esse fato faz com que o novo estado de equilíbrio seja atingido subitamente. Por exemplo, o equilíbrio populacional de uma espécie hipotética pode seguir o comportamento normal até ser afetada pelo aparecimento de um agente tóxico.



(a) Transição suave (b) Transição abrupta (c) Transição crítica

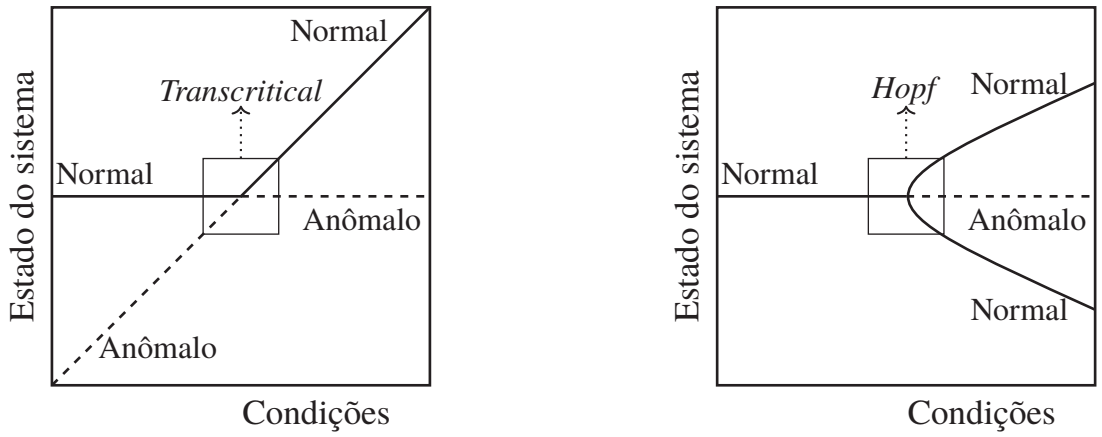
Figura 2.18: Exemplos de Transição dos Sistemas (adaptado de Scheffer (2009))

O terceiro exemplo de transição entre estados representa os casos em que os sistemas não conseguem transitar suave ou abruptamente entre os estados de equilíbrio. Assim, quando as condições mudam, o sistema é forçado a transitar criticamente entre estados de equilíbrio alternativos. A Figura 2.18(c) apresenta um exemplo de transição crítica. A parte superior da curva na Figura 2.18(c) apresenta o comportamento inicial do sistema. Com a evolução das condições observadas, o sistema irá variar a ponto de criar uma dobra na representação gráfica do sistema. Quando as condições observadas variam o suficiente, o sistema supera o ponto F_2 e

encaminha-se para o novo comportamento do sistema iniciado em F_1. Porém, devido à dobra, o sistema não pode transitar suavemente para a parte inferior da curva quando as condições observadas mudarem.

A transição dos pontos F_2 para F_1 é marcada por um equilíbrio instável, sendo denominada de transição crítica. Essa transição crítica pode representar mudanças nos sistemas. Por exemplo, durante a transição crítica, um sistema pode apresentar instabilidade ou uma espécie pode ser extinta. Outra propriedade relevante das transições críticas é que não bastam as condições retornarem ao patamar anterior à transição crítica para o sistema voltar ao ponto de equilíbrio anterior. Deste modo, recuperar sistemas que passam por transições críticas não é uma tarefa trivial (Scheffer, 2009). A literatura descreve F_1 e F_2 como pontos de bifurcação, bifurcação catastrófica ou pontos de inflexão (do inglês, *tipping point*). A transição crítica apresentada na Figura 2.18(c) é comumente chamada de bifurcação de dobra (do inglês, *fold bifurcation*) (Scheffer, 2009; Bury et al., 2021).

A literatura apresenta outros tipos de transições críticas além da transição de dobra (Figura 2.18(c)). A Figura 2.19 apresenta uma bifurcação do tipo *transcritical* e uma bifurcação do tipo *hopf*. Na bifurcação do tipo *transcritical* (Figura 2.19(a)), mesmo as condições evoluindo, o sistema observado possui um padrão constante até o ponto de bifurcação. A partir do ponto de bifurcação o comportamento normal do sistema muda e o padrão constante começa a representar um comportamento anômalo. Enquanto um padrão novo, representado pela reta crescente, é o novo comportamento normal. Outro exemplo de transição crítica é a bifurcação *hopf*, também conhecida como bifurcação *Pitchfork* (Figura 2.19(b)). Inicialmente, a bifurcação *hopf* apresenta um comportamento constante, similar a bifurcação *transcritical*. Porém, mudanças nas condições do sistema permitem que o sistema siga por dois caminhos diferentes. Um exemplo de transição deste tipo é quando uma viga está em posição vertical sustentando um peso. Caso o peso exceda os limites, a viga pode entortar tanto para a direita como para a esquerda. Deste modo, o sistema, antes linear, pode ser enviesado e seguir outras direções (Strogatz, 2018; Bury et al., 2021).



(a) Bifurcação do tipo *transcritical*

(b) Bifurcação do tipo *hopf*

Figura 2.19: Exemplos de Transições Críticas (adaptado de Strogatz (2018); Bury et al. (2021))

Prever e entender como ocorrem as mudanças na natureza ou na sociedade auxiliam no gerenciamento das mudanças. Controlar mudanças pode evitar catástrofes ou até mesmo a extinção de espécies (Scheffer, 2009). As séries temporais representam uma importante ferramenta para a análise das mudanças. Box et al. (2015) definem que uma série temporal é constituída de observações realizadas sequencialmente no tempo. Dados epidemiológicos dos últimos dias, a quantidade de vendas nos últimos meses, o preço médio da gasolina nos últimos

anos, e a quantidade de pacotes trafegando em uma rede de computadores contabilizada nos últimos segundos são exemplos de sistemas expressos na forma de séries temporais.

Uma das principais funções das séries temporais é auxiliar a criação de modelos matemáticos que descrevem satisfatoriamente os dados observados nas séries temporais (Shumway e Stoffer, 2017). Para ilustrar o conceito de séries temporais, Shumway e Stoffer (2017) apresentam o seguinte exemplo. Dadas às observações sequenciais x_1 e x_2, x_3, ..., a variável randômica x_1 representa o valor da primeira observação, o x_2 representa o valor no segundo ponto de observação, o x_3 representa o valor da terceira observação e assim por diante. A Figura 2.20 apresenta a série temporal da variação da temperatura registrada de 1880 até 2015. O eixo horizontal (eixo x) representa o tempo, nesse caso os anos em que a observação foi realizada. O eixo vertical (eixo y) apresenta o valor da variação da temperatura.

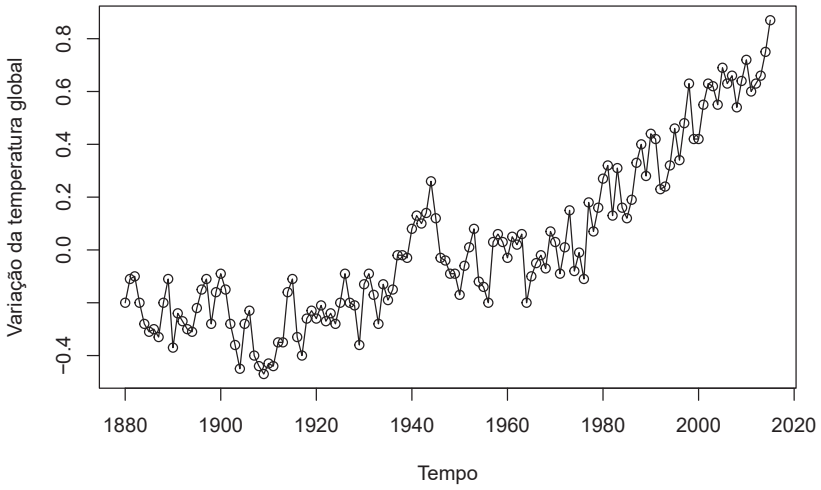


Figura 2.20: Exemplo de Série Temporal (adaptado de Shumway e Stoffer (2017))

2.5.1 Indicadores Estatísticos

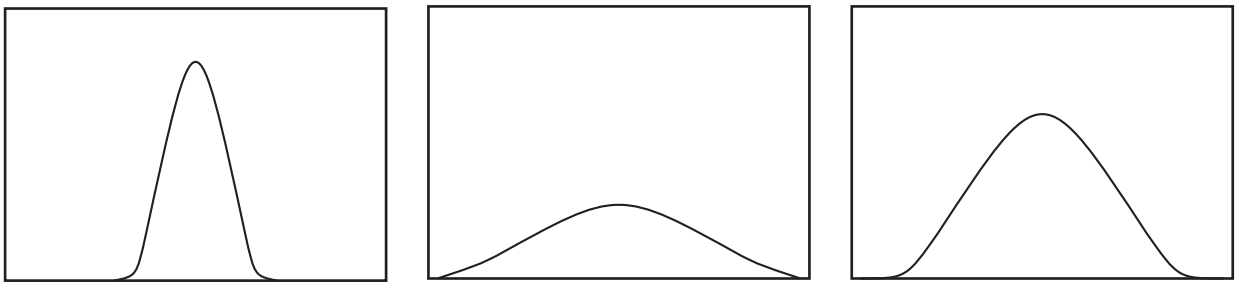
A literatura evoluiu a ponto de identificar evidências capazes de representar a ocorrência de transições críticas durante as observações dos sistemas (séries temporais). As evidências representam mudanças observáveis nos valores calculados a partir de indicadores estatísticos antes de um ponto de inflexão. Ou seja, é possível utilizar indicadores estatísticos sobre séries temporais para identificar sinais precoces de alerta e antecipar transições críticas (Bury et al., 2020). A autocorrelação de *lag*-τ (AC-τ), variância, *kurtosis*, *skewness* e variações do *power spectrum* representam alguns indicadores estatísticos capazes de produzir sinais precoces de alerta (Dakos et al., 2012). Em geral, os indicadores estatísticos são genéricos. Isso significa que eles se apoiam em propriedades comuns das transições críticas para calcular os sinais precoces de alerta. Isso é benéfico por poderem ser utilizados em diferentes aplicações.

O primeiro indicador estatístico utilizado neste trabalho é o ***Kurtosis***. A Fórmula 2.10 apresenta o método para a obtenção do *Kurtosis* de uma série temporal. O termo N representa a quantidade total de itens observados. Para calcular o *Kurtosis* ($\hat{y}^U$) é necessário o resultado de $\hat{y}$, definido pela Fórmula 2.11. O termo x_t refere-se a cada item observado na série temporal juntamente com seu índice. E o termo $\bar{x}$ refere-se a média aritmética simples do conjunto ($\bar{x} = \frac{\sum x_i}{N}$) (An e Ahmed, 2008; Joanes e Gill, 1998; Cramér, 1946). Segundo Oja (1981), existem

divergências acerca da interpretação do *Kurtosis*. Apesar das diferentes interpretações para o *Kurtosis*, uma das mais difundidas e controversas relaciona o valor do *Kurtosis* com grau de achatamento da curva da série temporal em comparação com a distribuição normal. A Figura 2.21 apresenta três exemplos de achatamentos de curvas. Quando o valor do *Kurtosis* for positivo ($\hat{y}^U > 0$) a tendência da distribuição é possuir um pico, cujo nome é *Leptokurtic* (Figura 2.21(a)). Para *Kurtosis* negativo, ($\hat{y}^U < 0$) a tendência da série temporal é possuir uma distribuição plana, nomeada de *Platykurtuc* (Figura 2.21(b)). Por fim, quando o valor do *Kurtosis* for igual a zero ($\hat{y}^U = 0$), a distribuição assemelha-se a distribuição normal, nomeada de *Mesiokurtic* (Figura 2.21(c)) (DeCarlo, 1997; Čisar e Čisar, 2010; Zhong et al., 2017). Atualmente, alguns autores defendem que a interpretação do valor do *Kurtosis* deve deixar de lado a relação com o pico da distribuição e se concentrar na cauda da distribuição (Westfall, 2014). Mesmo com as diferentes interpretações relacionadas ao *Kurtosis* e o fato do *Kurtosis* não ser um indicador estatístico projetado para produzir sinais precoces de alerta, o *Kurtosis* pode ser utilizado como sinal precoce de alerta. Isso ocorre, pois Biggs et al. (2009); Dakos et al. (2012) verificaram que o valor do *Kurtosis* pode crescer ou apresentar picos próximo de transições críticas. Deste modo, uma distribuição pode mudar de classificação variando para a distribuição *Leptokurtic* próximo às transições críticas (Dakos et al., 2012).

$$\hat{y}^U = \frac{(N-1)}{(N-2)(N-3)}(N-1)\hat{y} + 6 \tag{2.10}$$

$$\hat{y} = \frac{N\sum(x_t - \bar{x})^4}{[\sum(x_t - \bar{x})^2]^2} \tag{2.11}$$

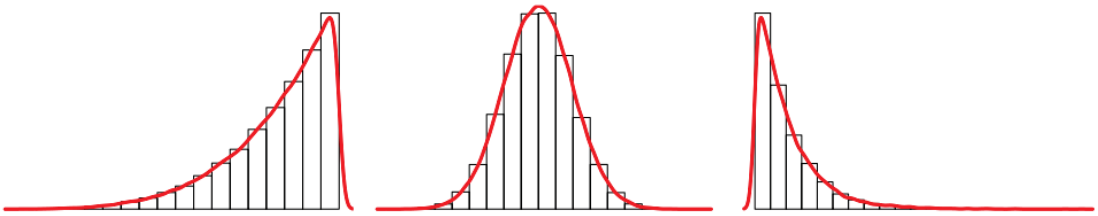


(a) Leptokurtic ($\hat{y}^U > 0$) (b) Platykurtuc ($\hat{y}^U < 0$) (c) Mesiokurtic ($\hat{y}^U = 0$)

Figura 2.21: Interpretação do *Kurtosis* (adaptado de DeCarlo (1997); Čisar e Čisar (2010); Zhong et al. (2017))

O segundo indicador estatístico utilizado neste trabalho é o ***Skewness***. O *Skewness* mensura o grau de assimetria das observações de uma série temporal. O grau de simetria pode indicar se a série temporal é simetricamente centralizada na média ou tende para alguma direção. A Figura 2.22 apresenta graficamente a interpretação dos resultados do *Skewness*. Para o *Skewness* negativo (menor que 0) a tendência dos valores é para a esquerda (Figura 2.22(a)). Caso o valor do *Skewness* seja nulo (próximo de zero), a série temporal é simétrica, ou seja, é centralizada na média (Figura 2.22(b)). Caso o valor do *Skewness* seja positivo (maior que 0) a distribuição tende a direita (Figura 2.22(c)) (Salas et al., 1980; Kardian et al., 2018). Apesar de não ser um indicador projetado para produzir sinais precoces de alerta, Guttal e Jayaprakash (2008) verificaram que aumentos na assimetria da distribuição podem indicar um alerta precoce confiável. A Fórmula 2.12 apresenta o modo para a obtenção do *Skewness*. O termo x_t refere-se a cada item observado na série temporal juntamente com seu índice. O N representa a quantidade

total de itens observados. O $\bar{x}$ refere-se a média aritmética simples do conjunto ($\bar{x} = \frac{\sum x_i}{N}$). Por fim, o s representa o desvio padrão dos dados ($s = \sqrt{\frac{\sum x - \bar{x}^2}{(N-1)}}$).



(a) Enviesado para a esquerda ($g < 0$) (b) Simétrico Normal ($g = 0$) (c) Enviesado para a direita ($g > 0$)

Figura 2.22: Interpretação do *Skewness* (adaptado de Doane e Seward (2011); Pipis (2020))

$$g = \frac{N \sum_{t=1}^{N} (x_t - \bar{x})^3}{(N-1)(N-2)s^3} \tag{2.12}$$

O **CV** é o terceiro indicador estatístico utilizado por este trabalho. Para obter o CV (Fórmula 2.13) basta dividir o desvio padrão dos dados analisados (s) pela média dos dados ($\bar{x}$), onde $\bar{x} \neq 0$ (Bedeian e Mossholder, 2000). Para obter o CV em porcentagem basta multiplicar a Fórmula 2.13 por 100 (Fórmula 2.14). Este trabalho utiliza o CV como na Fórmula 2.13. O CV é utilizado como indicador de diversidade em relação à média dos conjuntos de dados analisados. Assim, o limite inferior de CV ($CV = 0$) indica uniformidade completa do conjunto de dados (Bedeian e Mossholder, 2000). O CV permite comparar séries de valores que apresentam unidades de medida distintas. A literatura identificou que variações no CV indicam a ocorrência de uma transição crítica. Portanto, o CV pode ser utilizado como um sinal precoce de alerta (Carpenter e Brock, 2006; Dakos et al., 2012).

$$CV = \frac{s}{\bar{x}} \tag{2.13}$$

$$CV(\%) = \frac{s}{\bar{x}}.100 \tag{2.14}$$

Ao processar sinais utilizando o ***power spectrum***, pesquisadores podem identificar a ausência ou a existência de padrões repetitivos e estruturas de correlação. Aplicações como radares, codificação de sinais, previsão de dados, reconhecimento de padrões, detecção de sinais e sistemas de tomada de decisão utilizam o *power spectrum* (Vaseghi, 1996). Welch (1967) estima *power spectrum* utilizando a transformada de Fourier. Para isso, Welch propõe o cálculo da média aritmética simples de periodogramas sobrepostos. A Figura 2.23 apresenta o modo para calcular os periodogramas sobrepostos, apresentado por Welch (1967). Primeiramente, todas as observações ($X(j)$) de uma série temporal são divididas em K segmentos sobrepostos. Para cada segmento K, o periodograma é calculado. Ao fim do cálculo do periodograma para todos os segmentos, é necessário realizar a média de todos os periodogramas. O resultado da média representa o valor do *power spectrum*. Quando uma nova observação estiver disponível, todo o processo é executado novamente para a definição do novo valor do *power spectrum*. Welch define o periodograma como $I_k(f_n) = \frac{L}{U}|A_k(n)|^2$, onde k varia como $k = 1, 2, 3...K$. Welch define f_n como $f_n = \frac{n}{L}$ onde $n = 0, ..., L/2$. L representa o tamanho do segmento analisado pelo periodograma e $U = \frac{1}{L}\sum_{j=0}^{L-1} W^2(j)$. $W(j)$ representa os dados de série temporal analisada pelo periodograma, ou seja, $j = 0, ..., L-1$. A expressão $A_k(n)$ é a transformada de Fourier

definida pela Fórmula 2.15. Por fim, a biblioteca Scipy[12] para a linguagem de programação Python implementa o método de Welch para aproximação do *power spectrum*.

O quarto indicador estatístico utilizado neste trabalho é obtido por meio do cálculo do *power spectrum* sendo nomeado de *power spectrum* máximo (S_{max}). Para calcular S_{max}, é necessário identificar o valor máximo entre as últimas observações. Assim, o S_{max} não é definido a cada janela temporal, sendo então definido após um número arbitrário e contínuo de janelas. Por exemplo, a cada 20 observações, o maior valor do *power spectrum* é definido como o valor do S_{max}. Após outras 20 observações, um novo S_{max} é definido com base nas últimas observações. O S_{max} é importante, pois picos nos seus valores podem ser indicadores da aproximação de transições críticas. Assim, o S_{max} é utilizado como sinal precoce de alerta (Bury et al., 2020).

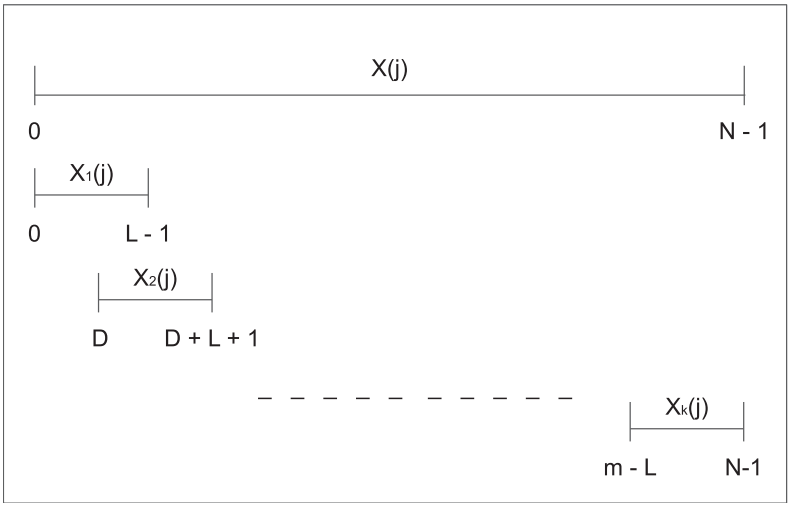


Figura 2.23: Sobreposição das Janelas em Uma Série Temporal (adaptado de Welch (1967))

$$A_k(n) = \frac{1}{L}\sum_{j=0}^{L-1} X_k(j)W(j)e^{-2kijn/L} \tag{2.15}$$

Sistemas como a maximização dos lucros, prescrição de remédios e a minimização de atrasos são sistemas reais que podem ser modelados matematicamente (Meerschaert, 2013). Existem casos em que os problemas possuem comportamentos tão complexos e dependem de tantas variáveis que identificar o modelo que rege perfeitamente o problema não é trivial. Um problema amplamente difundido e ainda não solucionado pela matemática é o P versus NP (Fortnow, 2009; Ruiz-Vanoye et al., 2020). Conforme a literatura propõe diferentes modelos com o intuito de apresentar soluções aproximadas para alguns problemas, surge a demanda de identificar qual é o melhor. Com o intuito de sanar esta demanda, Akaike propôs o uso da informação de Kullback-Leibler (distância) como um modo para a seleção do modelo. Akaike propôs o AIC (do inglês, *Akaike's information criterion*) o qual é uma forma de estimar as distâncias entre os modelos candidatos e o mecanismo verdadeiro desconhecido. Deste modo, calcula-se o AIC para todos os modelos candidatos. O modelo com menor valor de AIC é o "mais próximo" da função real (desconhecida) que gerou os dados observados (Burnham e Anderson, 2004).

[12] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.welch.html`

Utilizando a teoria do *power spectrum* e a teoria do AIC, Bury et al. (2020) propuseram três variações do AIC com o intuito de tipificar as transições críticas, e este trabalho utiliza essas variações também como indicador estatístico para gerar sinais precoces de alerta. As variações dos AICs são: S_{null}, S_{fold} e S_{hopf}. Caso o S_{null} (Fórmula 2.16) apresente um espectro relativamente plano, provavelmente o sistema observado está longe de transição crítica. O S_{fold} (Fórmula 2.17) e o S_{hopf} (Fórmula 2.18) tendem a crescer conforme transições críticas do tipo *fold* ou *hopf* se aproximam. A parte real dos autovalores da matriz Jacobiana do sistema, em geral, negativa, corresponde aos parâmetros λ e π. σ corresponde ao desvio padrão da distribuição e ω corresponde ao *power spectrum*. Por fim, ω_0 é uma constante que pode assumir valores diferentes entre os modelos. A biblioteca Ewstools[13] escrita em Python implementa os cálculos de S_{null}, S_{fold} e S_{hopf}.

$$S_{null}(\omega;,\sigma) = \frac{\sigma^2}{2\pi} \tag{2.16}$$

$$S_{fold}(\omega;\sigma,\lambda) = \frac{\sigma^2}{2\pi}\frac{1}{\omega^2+\lambda^2} \tag{2.17}$$

$$S_{hopf}(\omega;\sigma,\mu,\omega_0) = \frac{\sigma^2}{4\pi}\left(\frac{1}{(\omega-\omega_0)^2+\mu^2}+\frac{1}{(\omega+\omega_0)^2+\mu^2}\right) \tag{2.18}$$

O último indicador estatístico capaz de gerar sinais precoces de alerta utilizado neste trabalho é a **autocorrelação de *lag*-τ (AC-τ)**. A AC-τ computa a correlação entre pontos de dados espaçados em τ unidades de tempo (Bury et al., 2020). Ou seja, a AC-τ compara uma versão de uma série temporal com uma versão atrasada dela mesma. O resultado indica o nível de concordância entre os valores previamente registrados em uma série temporal e os valores em um momento subsequente (Riley e Greenhall, 2004; Torrecilla et al., 2011; Dakos et al., 2012; Box et al., 2015). O τ em AC-τ controla quanto tempo a versão anterior da série temporal atrasada será resgatada para comparação. Por exemplo, autocorrelação de *lag-1* compara a versão atual da série temporal com a versão imediatamente anterior a ela. O τ ideal para cada problema pode variar. Contudo, é comum na literatura que sejam calculados a AC-1 ($\tau = 1$), AC-2 ($\tau = 2$) e a AC-3 ($\tau = 3$). A Fórmula 2.19 apresenta o modo para a obtenção da autocorrelação de *lag*-τ. Onde, z_t é a série temporal atual, $z_{t+\tau}$ é a série temporal atrasada em τ unidades de tempo. O termo μ indica o valor médio, o termo *sigma* representa a variância de z_t e o E simboliza o valor esperado (Riley e Greenhall, 2004; Torrecilla et al., 2011; Dakos et al., 2012; Box et al., 2015). A Fórmula 2.20 apresenta uma alternativa para a AC-τ. Onde, N é o total de amostras da série temporal e $\bar{z}$ é o valor médio da série temporal (Riley e Greenhall, 2004; Box et al., 2015). A literatura verificou que variações na autocorrelação podem indicar a aproximação de transições críticas (Bury et al., 2020). Portanto, a AC-τ pode ser utilizada para gerar sinais precoces de alerta. Por fim, a biblioteca Pandas implementa o cálculo da autocorrelação de lag-τ para a linguagem de programação Python[14].

$$AC-\tau = \frac{E[(z_t-\mu)(z_{t+\tau}-\mu)]}{\sigma_z^2} \tag{2.19}$$

$$AC-\tau\ (alternativa) = \frac{\frac{1}{N}\sum_{t=1}^{N-\tau}(Z_t-\bar{z})(z_{t+\tau}-\bar{z})}{\frac{1}{N}\sum_{t=1}^{N}(Z_t-\bar{z})} \tag{2.20}$$

[13] https://github.com/ThomasMBury/ewstools

[14] https://pandas.pydata.org/docs/reference/api/pandas.Series.autocorr.html

2.6 RESUMO

Este capítulo apresentou conceitos essenciais para o decorrer do trabalho. Primeiramente este capítulo apresentou conceitos relacionados com o ataque DDoS. Devido ao potencial danoso, o ataque DDoS define o escopo deste trabalho. Deste modo, é importante conhecer o funcionamento dos ataques bem como os modos de defesa que a literatura apresenta. Como a solução proposta (Capítulo 4) é pautada nos conceitos do aprendizado de máquina, é oportuno descrevê-los. Assim, este capítulo apresentou como a literatura classifica os diferentes esforços que englobam o aprendizado de máquina. É oportuno destacar o aprendizado supervisionado e não supervisionado para a classificação de observações, pois algoritmos dessas classes são comumente utilizados na literatura, e fazem parte da avaliação deste trabalho. Também é oportuno destacar os algoritmos de *deep learning* devido à grande evolução que esses algoritmos obtiveram nos últimos anos. Por fim, os sinais precoces de alerta constituem a espinha dorsal deste trabalho. Os sinais precoces de alerta são produzidos por indicadores estatísticos sobre séries temporais. A principal propriedade dos sinais é a capacidade de apresentar variações em seus resultados conforme transições críticas se aproximam. Essas variações podem acontecer mesmo quando o sistema observado não apresente sinais claros de mudança. Desse modo, os sinais precoces de alerta são extremamente importantes para prevenir ações que possam causar prejuízos para a humanidade.

3 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta estudos que propõem soluções para a predição de ataques DDoS. Para organizar a apresentação da revisão bibliográfica, este trabalho propõe uma classificação desses estudos. Esta classificação auxilia os pesquisadores a identificarem como o estado da arte aborda a predição de ataques DDoS (Karim et al., 2014) e revela oportunidades de pesquisa (Salam et al., 2016). Este capítulo está dividido da seguinte forma. A Seção 3.1 apresenta os aspectos da predição considerados neste trabalho para a criação da classificação. As Seções 3.2 e 3.3 apresentam os estudos identificados que realizam predições de curto e longo prazo respectivamente. A Seção 3.4 apresenta uma discussão sobre os estudos presentes na literatura. A Seção 3.5 apresenta as questões em aberto e oportunidades de pesquisa. A Seção 3.6 resume o capítulo. Dois apêndices complementam este capítulo. O Apêndice A detalha o rigoroso planejamento e execução da revisão bibliográfica apresentada neste capítulo. O Apêndice B complementa a revisão bibliográfica apresentando trabalhos relacionados com a teoria dos sinais precoces de alerta usada neste trabalho.

3.1 ASPECTOS DOS ESTUDOS PARA A PREDIÇÃO DE ATAQUES DDOS

A execução da revisão da literatura foi baseada na revisão de 2.682 artigos relacionados com ataques DDoS. Desses artigos, 27 apresentaram soluções para predição de ataques DDoS. Este trabalho constatou que o tempo antes do lançamento do ataque, a arquitetura da solução, a metodologia da solução e o tipo de dados usados pela solução para predizer os ataques são os aspectos comuns aos estudos identificados. Isso leva a uma classificação dos estudos com base no aspecto temporal, arquitetônico, metodológico e de dados. Esta classificação em quatro aspectos fornece diferentes pontos de vista sobre os estudos identificados, refletindo os principais aspectos para a evolução das pesquisas em predição de ataques DDoS.

A Figura 3.1 apresenta a classificação dos estudos usando os quatro principais aspectos dos estudos divididos em quatro camadas. Na primeira camada, os estudos são classificados quanto ao aspecto temporal. Na segunda camada, arquiteturas centralizadas versus distribuídas são usadas como critérios de classificação. Os estudos classificados na categoria curto prazo seguem ambas as abordagens (centralizada e distribuída), enquanto estudos presentes na categoria longo prazo, seguem apenas a abordagem centralizada. Na terceira camada, os estudos classificados na categoria curto prazo e centralizado seguem quatro possíveis aspectos metodológicos: Aprendizado de Máquina, Baseado em Markov, Modelo Estatístico e Híbrido. O único estudo classificado como curto prazo distribuído utiliza os Modelos Estatísticos como aspecto metodológico. Os estudos classificados na categoria de longo prazo e na categoria arquitetura centralizada usam Aprendizado de Máquina, Modelos Estatísticos ou uma abordagem híbrida baseada em Aprendizado de Máquina e Modelos Estatísticos. Por fim, na camada de aspecto de dados, os estudos usam tráfego de rede, alertas de Sistema de Detecção de Intrusão (do inglês, *Intrusion Detection System* - IDS), *logs* de aplicações e fontes de dados externas.

O **aspecto temporal** está associado ao tempo entre a predição do ataque e o lançamento do ataque. Em geral, os estudos empregam os termos curto e longo prazo referentes ao período mencionado. No entanto, não foi possível localizar uma definição precisa de curto e longo prazo em relação à predição em andamento. Por exemplo, no trabalho de Laing et al. (2021), o termo curto prazo refere-se a ações a serem realizadas dentro de seis meses, o termo médio prazo incorpora ações que ocorrerão nos próximos seis ou 18 meses, e longo prazo inclui ações

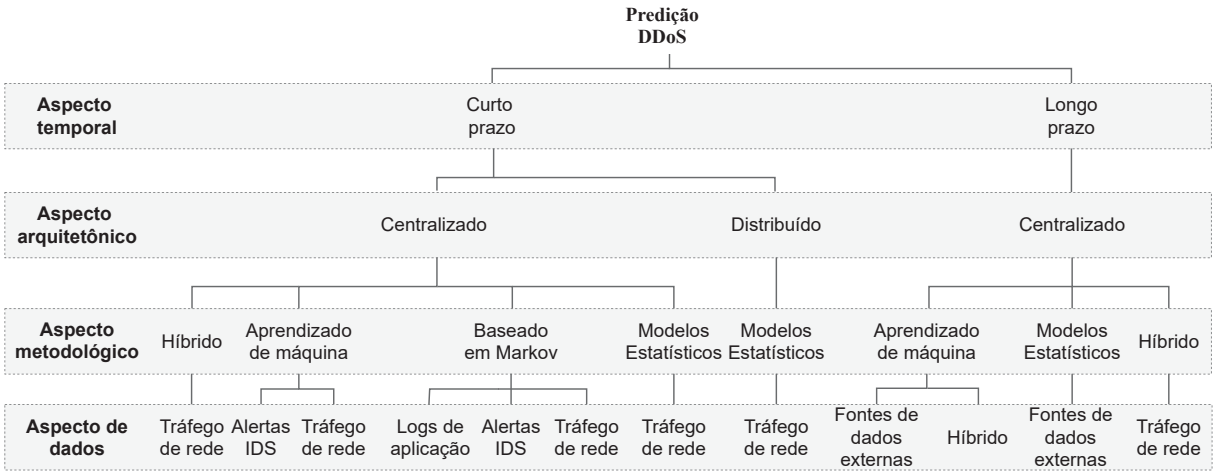


Figura 3.1: Classificação das Soluções para a Predição de Ataque DDoS

que ocorram além de 18 meses. Como os atacantes DDoS usam vários métodos para evitar predição e detecção (Abu Rajab et al., 2006; Dei Rossi et al., 2016), é importante definir um padrão adequado para esses termos.

Este trabalho observou dois caminhos em relação à predição de ataques nos estudos. Existe um grupo de estudos que realiza a predição próximo ao momento em que o ataque é lançado, no máximo algumas horas antes do atacante lançar o ataque. Nesses casos, a predição ocorre com menos de um dia de antecedência. Em relação ao segundo grupo, este trabalho identificou que a predição ocorre com um ou mais dias de antecedência. Como conclusão desta análise, este trabalho definiu a fronteira de 24 horas para dividir os grupos.

A Figura 3.2 ilustra os dois comportamentos identificados em relação ao aspecto temporal. Na figura, a seta representa uma linha do tempo em que as predições e lançamentos de ataque são representados. Os círculos preenchidos antes da fronteira de 24 horas representam os seis estudos que predisseram ataques com mais de um dia de antecedência. Esses estudos variam de um dia a três meses de antecedência. Os diamantes preenchidos representam os 20 estudos que predisseram um ataque com menos de 24 horas de antecedência. Esses estudos variam a criação de evidências de 14 horas a segundos perto do lançamento do ataque. Assim, este trabalho classificou como estudos de curto prazo, os estudos que realizam predição de ataque até 23 horas, 59 minutos e 59 segundos antes do lançamento do ataque. Caso o estudo realize a predição pelo menos um dia antes do atacante iniciar o ataque, este trabalho o classificou como estudo de longo prazo. Por fim, o estudo de Muhammad et al. (2020) não apresentou resultados que indicassem que a predição era de curto ou longo prazo. Portanto, este trabalho não possui classificação quanto ao aspecto temporal.

O **aspecto arquitetônico** refere-se a como os dados são processados. A abordagem clássica é coletar os dados usados pelas soluções e centralizar seu processamento. Há casos em que essa ação é vital porque a união dos dados pode representar indícios de um ataque. A dificuldade dessa abordagem é que, dependendo do tipo, da quantidade de dados usados e do método empregado para realizar a predição de ataques DDoS, a centralização do processamento de dados requer grande poder computacional (Chen et al., 2021). Além da arquitetura centralizada, este trabalho observou a arquitetura distribuída. Em arquiteturas distribuídas, os processos em diferentes pontos da infraestrutura dividem o problema processando pequenas porções dos dados. Com os dados analisados, os processos identificam e extraem evidências sobre ataques DDoS.

A arquitetura distribuída é desejável para soluções de predição de ataque DDoS devido à complexidade da infraestrutura de rede e ao grande volume de dados gerados pelos usuários

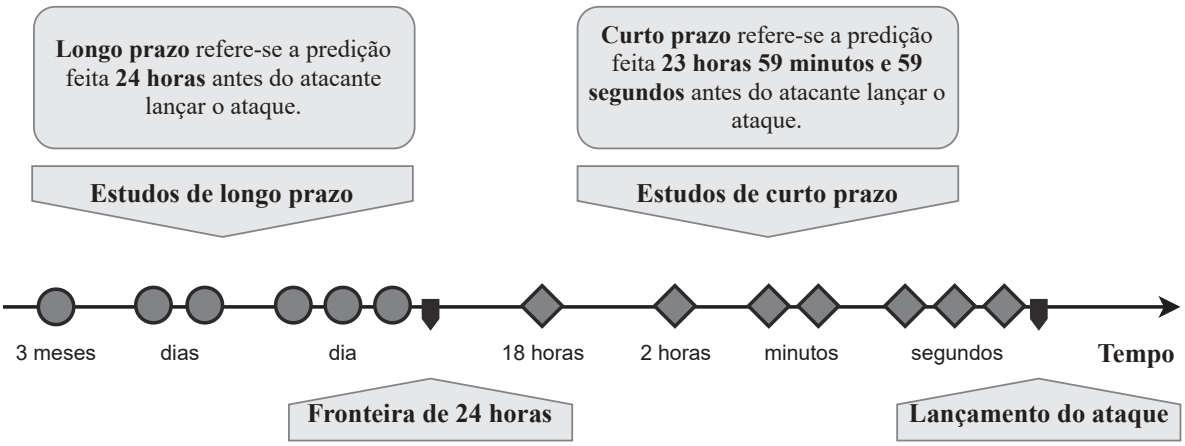


Figura 3.2: Relação Entre Predições de Curto e Longo Prazo

normais. No entanto, propor e testar soluções distribuídas para resolver o problema de predição de ataques DDoS não é uma tarefa trivial. Este trabalho classificou apenas um dos 27 estudos como distribuído. Os outros 26 estudos processam informações de forma centralizada. Apesar de serem mais comuns na literatura, esses estudos devem considerar a quantidade de dados a ser analisada, uma vez que os recursos podem não ser suficientes para a realização das tarefas.

Os **aspectos metodológicos** envolvem as técnicas que os estudos empregaram nas soluções. Esse aspecto é essencial porque as metodologias garantem os aspectos temporais e arquitetônicos ao manipular os dados e fazer predições. Este trabalho identificou quatro conjuntos de técnicas: Aprendizado de Máquina/*Deep Learning*, baseado em Markov, Modelo Estatístico e Híbrido. Modelos estatísticos foram as primeiras abordagens utilizadas para predizer ataques DDoS. Em geral, o objetivo era usar técnicas como a média móvel integrada autorregressiva (do inglês, *Autoregressive integrated moving average* - ARIMA) ou a *Gray Theory* para predizer o comportamento e identificar com antecedência se um ataque poderia ocorrer. Esse tipo de técnica evoluiu e outras abordagens surgiram na literatura para possibilitar a identificação de indícios antes que o atacante lance o ataque.

A evolução do poder computacional e a criação de milhares de gigabytes de dados todos os meses (Barnett et al., 2018; Domo, 2023) apoiou o uso de algoritmos de aprendizado de máquina em vários contextos (Shinde e Shah, 2018), incluindo a predição de ataques DDoS. Os algoritmos de aprendizado de máquina usam os dados disponíveis para construir modelos capazes de predizer a ocorrência de um ataque DDoS. Além de lidar bem com a abundância de dados, os algoritmos de aprendizado de máquina têm uma variedade de modelos que podem inspirar soluções para predizer ataques DDoS. Como em Liu e Lang (2019); Kavlakoglu (2020); Kaluarachchi et al. (2021), este trabalho considerou o *deep learning* como um ramo do aprendizado de máquina (Seção 2.4). Portanto, este trabalho considera como aprendizado de máquina todas as soluções que utilizam algoritmos de *deep learning*.

Soluções baseadas em cadeias de Markov (Kemeny e Snell, 1976) também fazem parte da classificação apresentada neste trabalho. Estudos usam os possíveis estados de um serviço ou ataque para construir cadeias de Markov e predizer ataques DDoS. Além disso, as cadeias de Markov mostram a probabilidade de mudança de estado. Se a solução puder identificar indícios de que um ataque está nas fases de coordenação, como comunicação com a central de C&C (Silva et al., 2013), a solução apresenta a probabilidade desse ataque evoluir antes mesmo que o atacante lance o ataque, alcançando assim a predição de ataques DDoS. Por fim, as soluções híbridas combinam duas ou mais técnicas mencionadas anteriormente para predizer ataques DDoS.

A camada **aspectos dos dados** destaca o tipo de dados empregados pelas soluções para realizar as predições. Este trabalho identificou que os estudos usam alertas provenientes de IDS, tráfego de rede, *logs* de aplicações, dados coletados de fontes de dados externas ou uma abordagem híbrida. Alertas de IDS são avisos criados por diferentes IDS que algumas soluções usam para identificar possíveis ações, como tentativas de propagação de *malware* e comunicações incomuns. Assim, as soluções que usam alertas IDS tentam identificar conjuntos de violações para predizer o ataque antes que o atacante lance o ataque.

O tráfego de rede é o aspecto de dados mais utilizado entre os estudos analisados. Estudos podem inferir sinais capazes de representar ataques antes mesmo dos atacantes lançarem o ataque. Dois estudos usam *logs* de aplicações para construir o modelo de predição de ataque. Esses *logs* podem ser operações do dispositivo, tentativas de login ou estados de aplicações. Alguns estudos usam dados coletados de fontes externas de dados para predizer alvos e campanhas de ataques DDoS. Os dados podem ser provenientes de redes sociais, sites de notícias, blogs, fóruns e listas de reputação. Por fim, um artigo enriquece o tráfego da rede com dados coletados externamente. Nesse caso, essa combinação é denominada de abordagem híbrida.

3.2 PREDIÇÕES DE CURTO PRAZO

Esta seção apresenta todos os estudos classificados como predições de curto prazo. Como citado anteriormente, os estudos de curto prazo podem possuir aspectos centralizados ou distribuídos para a classe de aspectos arquitetônicos. Além disso, este trabalho identificou que estudos com aspectos centralizados utilizam aprendizado de máquina, modelos estatísticos baseados em Markov ou híbridos como aspectos metodológicos. Este trabalho identificou que o estudo com aspecto distribuído utiliza modelos estatísticos como aspecto metodológico. Para simplificar a apresentação dos estudos, este trabalho os divide conforme o aspecto metodológico utilizado em cada estudo. No entanto, cada estudo tem sua classificação completa ao longo do texto e na Tabela 3.1 (apresentada na Seção 3.3).

3.2.1 Modelos Híbridos (Aspecto metodológico)

Em Salemi et al. (2021), os autores propõem uma solução para predizer ataques DDoS em ambientes de saúde, apesar de não conseguir testar a solução com um conjunto de dados que representa ambientes de saúde. A solução processa centralizadamente o tráfego da rede e projeta o comportamento do tráfego da rede nos próximos segundos. Com a projeção do tráfego futuro da rede, a solução prediz o erro dessa projeção utilizando a *Recurrent Neural Echo State Network* (SCESN). A SCESN é pré-treinada com base em dados históricos da rede. A solução analisa esse erro de predição, usando o expoente de Lyapunov para predizer o ataque. Como resultado, a solução identifica sinais de ataques DDoS até 50 segundos antes do ataque. O resultado mais preciso foi predizer ataques que ocorrerão nos próximos 20 segundos.

3.2.2 Aprendizado de Máquina

Em Olabelurin et al. (2015), os autores propõem a análise de alertas gerados por IDS para a predição de ataques DDoS. A solução centraliza o processamento dos alertas e realiza análises medindo o grau de aleatoriedade das informações contidas nos alertas (entropia dos alertas). Com a entropia, o sistema agrupa os alertas usando K-means. K-means é um algoritmo de aprendizado de máquina não supervisionado que agrupa dados em *k clusters* não sobrepostos (Wu, 2012). O objetivo é identificar grupos de alertas que podem simbolizar a orquestração de um ataque DDoS

antes que o atacante inicie o ataque. A solução fornece informações sobre grupos com atividades que podem representar um ataque segundos antes de acontecer.

O estudo de Fadlullah et al. (2011) apresenta o uso da *Gaussian Process Regression* para a predição de ataques DDoS usando tráfego de rede coletado por sensores. O Processo Gaussiano é um algoritmo de aprendizado de máquina supervisionado usado em tarefas de regressão de classificação (Scikit-learn, 2021). Devido ao baixo poder de processamento do sensor, os autores citam ser mais adequado centralizar o processamento dos dados em vez de distribuir o processamento dos dados. Durante as simulações usadas para testar a proposta, a solução pode predizer quanto tempo levaria para um ataque sobrecarregar a vítima, geralmente alguns segundos antes da sobrecarga. Embora não esteja claro quando a solução fez a predição, este estudo foi considerado nesta revisão, pois a solução pode criar notificações de sobrecarga antes que o atacante lance o ataque. Isso é possível desde que as ações se manifestem por tempo suficiente antes do lançamento do ataque, caracterizando o estudo como uma solução para a predição de ataques DDoS.

Em Jaber et al. (2017), os autores usam a estratégia *Learning from Examples based on Rough Sets* (LERS) para construir um conjunto de regras que podem predizer ataques e intrusões. Embora não seja uma solução específica para DDoS, é relatado que a solução pode operar com ataques DDoS. Primeiro, a solução processa as informações de forma centralizada. O processamento de dados consiste em dividir os dados históricos usando a Análise de Componentes Principais (do inglês, *Principal Component Analysis* - PCA) (Wold et al., 1987) e análise de covariância. Em seguida, a solução aplica o conjunto de regras criadas com o LERS. LERS é um modo de aquisição de inteligência baseado na abordagem de aprendizado de máquina (Grzymala-Busse, 1992). Assim, usando o tráfego de rede antes do lançamento do ataque, a solução pode predizer um ataque que ocorrerá nos próximos segundos.

Em Kwon et al. (2017), os autores usam a análise de regressão linear multivariada, a qual é considerada um modelo de aprendizado de máquina (Ray, 2019), para predizer o volume de um ataque DDoS que pode ocorrer nos próximos segundos. O modelo de predição consiste em centralizar a análise do tráfego de rede coletado anteriormente. Assim, foi possível estimar o número de possíveis agentes maliciosos (*bots*) que podem lançar ataques DDoS com base no número de usuários na rede. Com o número estimado de *bots* e o número total de serviços protegidos, os autores estimam o volume do ataque DDoS que pode ocorrer nas próximas horas.

Machaka et al. (2022) compararam três algoritmos de regressão para predizer ataques DDoS. Os algoritmos comparados são a *Logistic Regression* (LGR), *Support Vector Regression* e *Kernel Ridge Regression*. Para realizar a comparação, os autores utilizaram o tráfego de rede do conjunto de dados DARPA 1999. Os autores treinaram os algoritmos de regressão usando o número de pacotes por 10 segundos em 80% do conjunto de dados e avaliaram a solução com os 20% restantes. O LGR obteve os melhores resultados, alcançando uma acurácia de 98,60% para predizer ataques DDoS nos próximos 15 minutos.

O estudo de SU et al. (2018) prediz ataques DDoS usando amostras de tráfego de rede e redes neurais de função de base radial (do inglês, *Radial Basis Function* - RBF) (Liu, 2013). O objetivo deste estudo é propor uma solução capaz de processar dados de forma centralizada e predizer a variação do tráfego de forma mais fiel à realidade do que outras técnicas clássicas como o ARIMA. A predição do tráfego nos próximos segundos serve como entrada para um algoritmo de detecção de ataque DDoS. Em outras palavras, a solução completa não apresenta evidências de quando o ataque foi detectado, se foi antes ou depois do ataque ter sido lançado. No entanto, a predição do tráfego futuro pode gerar evidências de um possível ataque DDoS antes mesmo que o atacante lance o ataque. Além disso, a solução proposta pode se tornar uma solução que atenda aos requisitos de predição de ataque DDoS se for possível alterar o algoritmo

de detecção para usar a predição de tráfego futuro e gerar evidências antes do lançamento do ataque DDoS.

O estudo de Muhammad et al. (2020) propõe uma solução focada na detecção de *botnets* durante o estágio inicial de comunicação C&C. Para isso, os autores analisam centralizadamente o tráfego da rede que contém rastros de ações realizadas por *botnets*. A partir dessa análise, os autores selecionam atributos com base no tráfego de rede e comparam a taxa de acerto entre quatro algoritmos de aprendizado de máquina: *Random Forest* (Breiman, 2001), SVM (Kecman, 2005), *Logistic Regression* (Kleinbaum e Klein, 2010), e *Multilayer Perceptron* (MLP) (Taud e Mas, 2018). Usando 37 dos 40 atributos, os autores obtiveram 97,8% de acurácia. Embora a solução não preveja especificamente um ataque DDoS, detectar a comunicação C&C é a evidência de um possível ataque que ocorrerá no futuro. Não está claro quanto tempo antes do ataque a solução pode identificar a comunicação C&C e, consequentemente, identificar sinais de um ataque futuro. Assim, este trabalho não identificou uma classificação para o aspecto temporal para o estudo de Muhammad et al. (2020). No entanto, este trabalho identifica e classifica todos os outros aspectos sobre (Muhammad et al., 2020).

3.2.3 Baseado em Markov

No estudo de Ali e Al-Shaer (2013), os autores usam cadeias de Markov para predizer eventos de segurança cibernética de infraestruturas de medição avançada. A solução apresentada não foca em DDoS, mas os autores citam a possibilidade de que a solução também funcione no contexto de ataques DDoS. A solução realiza a modelagem da cadeia de Markov com base nos *logs* de aplicação gerados nos centralizadores das medições. Os autores coletaram os *logs* de eventos durante o período de duas semanas e os processaram centralizadamente. O diferencial deste estudo é que os autores identificaram que, por meio da modelagem da cadeia de Markov, eles predizem a mudança entre os estados com um erro menor que 2%. Se a predição de variação de estado representa uma mudança para um estado de ataque, a solução pode predizer ataques, incluindo ataques DDoS segundos antes de eles ocorrerem.

Em Moudoud et al. (2021), os autores propõem uma arquitetura capaz de predizer ataques DDoS em redes IoT. Centralizadamente, a solução analisa o comportamento de cada dispositivo conectado à rede com base nos *logs* de aplicação. Esses registros podem ser estados da aplicação, operações do dispositivo ou tentativas de login. Com o comportamento de cada dispositivo, a solução extrai o estado do dispositivo e gera uma cadeia de Markov. Apenas com o estado atual e a cadeia de Markov, os autores podem medir a probabilidade de o dispositivo evoluir para um estado de ataque antes do lançamento do ataque. Assim, é possível predizer se um dispositivo pode lançar ataques nos próximos segundos.

O estudo de Abaid et al. (2016) visa identificar e modelar o comportamento típico de *botnets* em uma cadeia de Markov. Assim, os autores propuseram uma metodologia para predizer ataques com base na probabilidade de evolução do estado atual para um estado de ataque no futuro próximo. Para treinar a solução da cadeia de Markov, os autores obtêm e processam centralizadamente vários alertas IDS. Durante os experimentos, a solução prediz a comunicação de C&C com acurácia de 99%. Os resultados ainda mostram que a predição do ataque pode variar de alguns segundos a 14 horas antes do lançamento do ataque.

O estudo de Holgado et al. (2020) propõe o uso de alertas produzidos por IDS para predizer ataques utilizando o Modelo Oculto de Markov (HMM). A primeira ação dos autores foi criar um conjunto de dados de Vulnerabilidades e Exposições Comuns (do inglês, *Common Vulnerabilities and Exposures* - CVE) para associá-las aos alertas para evitar o sobreajuste (do inglês, *overfitting*). Os atributos utilizados para predizer os ataques são a descrição do alerta, a gravidade e os CVEs associados aos alertas. A ação subsequente é treinar a solução no modo

off-line para identificar a evolução dos ataques para predizê-los. Durante o treinamento, a solução analisa os diferentes alertas IDS, identifica os estados de ataque e calcula a matriz de probabilidade. Essa matriz descreve a probabilidade de transição entre os estados da rede que os alertas IDS representam. Após o treinamento, a solução pode identificar o estado atual e medir a probabilidade da rede evoluir para um estado de ataque. Os autores verificaram que a solução proposta predisse um ataque DDoS cerca de 11 minutos antes do lançamento do ataque.

O estudo de Shin et al. (2013) propõe o uso da cadeia de Markov para predizer e detectar intrusões na rede. Para predizer intrusões, a solução coleta e centraliza o processamento do tráfego da rede. Após a coleta, a solução define os estados da rede, constrói a matriz de probabilidade e analisa os novos dados em tempo real. Assim, a solução pode fornecer um grau de risco com base na análise da cadeia de Markov. Embora a solução não se concentre em ataques DDoS, os autores realizaram um teste experimental usando um cenário de ataque DDoS (DARPA 2000 (Laboratory, 2000)). Durante os experimentos, a solução produziu evidências de um ataque segundos antes do lançamento do ataque. Especificamente, neste caso, as evidências produzidas mostraram probabilidade da rede evoluir para um estado de ataque.

3.2.4 Modelos Estatísticos

O estudo de Adegboyega (2015) usa uma melhoria incremental sobre o ARIMA com os Modelos de Pontuação Condicional Adaptativa (do inglês, *Adaptive Conditional Score Models* - ACS) para predizer o *throughput* (taxa de pacotes por minuto da rede). A solução proposta centraliza o processamento do tráfego de rede para predizer variações na taxa de pacotes por minuto causadas por possíveis ataques. Embora não seja específico para predizer ataques DDoS, a solução proposta pode identificar sinais da preparação do ataque próximo do momento do lançamento de um ataque DDoS. Caso a solução antecipe o aumento da taxa de transferência antes que aconteça, ela pode fornecer aos administradores mais tempo para se prepararem para os ataques DDoS.

O estudo de Fan et al. (2018) propõe a predição de índices de risco dividindo os índices de risco em três camadas, onde os índices de risco das camadas inferiores são a entrada para a definição dos índices de maior risco. Cada índice de risco tem sua metodologia estatística a ser construída, e os autores usaram *Fuzzy Cognitive Maps* além de usar dados de inteligência de ameaças para unir e ajustar os resultados das predições. A solução processa o tráfego da rede de forma centralizada. Embora a solução não se concentre na predição de ataques DDoS, um índice representa o risco de um ataque DDoS ocorrer nos próximos segundos.

Leros e Andreatos (2019) propõem uma solução para predizer o tráfego de rede próximo ao real para detectar ataques DDoS. Os autores dividem a solução em duas partes: determinística e estocástica. Na parte determinística, a solução possui um módulo que utiliza observações anteriores para treinar o modelo autorregressivo (AR) incrementado com o filtro de Kalman. A solução utiliza o modelo AR para predizer o volume de tráfego de rede (bits por segundo) nos próximos cinco minutos. Os autores projetaram a parte estocástica para ter três componentes: um módulo para calcular o coeficiente de difusão adaptativo, um módulo para calcular a raiz quadrada do próprio processo estocástico e um módulo para capturar a dinâmica da evolução do resultado do modelo AR. A solução utiliza as duas partes (determinística e estocástica) e o *Mean-Reversion Stochastic Process* para predizer o tráfego a cada segundo nos próximos cinco minutos. A saída representa atividades normais ou anormais nos próximos cinco minutos.

O estudo de Pelloso et al. (2018) usa a teoria de metaestabilidade (modelos estatísticos) para identificar sinais antes do lançamento do ataque. De forma centralizada, a solução proposta captura o tráfego da rede, prepara os dados capturados e calcula os indicadores. A solução analisa esses indicadores para verificar a existência de evidências de ataques futuros e para produzir

alertas de ataque DDoS, se necessário. A solução proposta foi avaliada experimentalmente em dois conjuntos de dados. O primeiro é o *Defense Advanced Research Projects Agency* (DARPA 2000) (Laboratory, 2000) e um dos 13 cenários fornecidos pelo conjunto de dados da *Czech Technical University* (CTU-13) (Garcia et al., 2014). Como resultado, a proposta conseguiu identificar sinais da preparação do ataque duas horas antes dos atacantes lançarem os ataques.

Em Savchenko et al. (2020), os autores analisam a viabilidade de detectar ataques DDoS, predizendo o comportamento dos usuários do serviço. Centralizando o processamento do tráfego da rede, os autores propõem um modelo estatístico para predizer o comportamento dos usuários nos próximos segundos. Durante a simulação, a solução proposta prediz o atraso médio entre os pacotes transmitidos. Como a solução usou dados coletados perto do lançamento do ataque, o modelo se tornou mais preciso. Este trabalho considera o estudo de Savchenko et al. (2020) um estudo relacionado à predição de ataques DDoS, pois Savchenko et al. (2020) usaram a predição do comportamento realizado pela solução antes do lançamento do ataque como evidência para predição de ataques DDoS.

O estudo de Yin e Nianqing (2009) propõe a predição do coeficiente de risco cibernético para predizer ataques, incluindo ataques DDoS. Centralizadamente, os autores calculam a entropia das informações presentes no cabeçalho do pacote, como IP de origem e destino e porta de origem e destino. Os autores relacionam a entropia de momentos anteriores à entropia atual para obter o risco atual. Com essas informações, os autores calculam o risco nos próximos segundos usando a *Gray Theory*. Este risco futuro pode indicar a ocorrência de um ataque antes de seu lançamento. Portanto, este estudo está apto a ser considerado neste trabalho.

Kivalov e Strelkovskaya (2022) criaram uma solução para predizer ataques DDoS com base na extrapolação *spline* do tráfego de rede. A extrapolação de *spline* prediz o pico de tráfego com base no tráfego autossemelhante. Os autores utilizaram extrapolações lineares e cúbicas para predizer ataques DDoS. A solução utiliza o tráfego de rede antes, durante e depois de um ataque DDoS para predizer ataques semelhantes. O atributo utilizado pela solução durante a avaliação é a intensidade do tráfego. No entanto, os autores não mencionam como quantificaram esse atributo. As *splines* cúbicas apresentaram os melhores resultados para predizer o ataque nos próximos segundos.

O estudo de Jog et al. (2015) é o único estudo identificado neste trabalho que propõe uma solução distribuída para predição de ataques DDoS. Ou seja, a solução proposta coleta e processa o tráfego da rede de maneira distribuída. A solução é instalada em pontos estratégicos da infraestrutura da vítima. A solução analisa o tráfego e prediz quando pode ocorrer uma sobrecarga. O ARIMA é usado para analisar o tráfego da rede e inferir quando uma possível sobrecarga pode ocorrer. Se algum nó identificar uma possível sobrecarga, esse nó emite um alerta com os detalhes da possível sobrecarga. Caso essa solução possa criar alertas segundos antes do lançamento do ataque, este trabalho pode classificar o estudo de Jog et al. (2015) como uma solução para predição de ataque DDoS.

3.3 PREDIÇÕES DE LONGO PRAZO

Esta subseção apresenta todos os estudos classificados como predições de longo prazo. O aspecto arquitetônico desses estudos é predominantemente centralizado. Além disso, os estudos com aspectos centralizados utilizam aprendizado de máquina, modelos estatísticos ou híbridos como aspectos metodológicos. Para simplificar a apresentação dos estudos, o aspecto metodológico é utilizado para classificar os trabalhos de predição de longo prazo. Porém, cada estudo tem sua classificação completa ao longo do texto. A Tabela 3.1 apresenta um resumo da classificação de todos os estudos identificados nesta revisão bibliográfica. Nesta tabela, cada

linha descreve um dos estudos, bem como a classificação para todos os aspectos. Além dos critérios de classificação, este trabalho incluiu uma coluna para relatar os métodos de avaliação.

3.3.1 Aprendizado de Máquina

O estudo de Liu et al. (2015) propõe o uso de fontes de dados externas para treinar o SVM para predizer eventos de segurança. As fontes de dados externas são compostas por listas de reputação associadas a eventos de segurança. As listas representam dispositivos que, possivelmente, estão envolvidos em ataques. Os eventos de segurança são ocorrências de ataques relatados na Internet. Os autores processam esses dados centralizadamente e preveem a ocorrência de eventos relacionados à segurança cibernética com uma média de VP de 69% para os próximos três meses do estudo. Embora não focado em ataques DDoS, os autores observam a possibilidade de que a solução funcione para a predição de ataques DDoS. Por esse motivo, este trabalho considera o estudo de Liu et al. (2015) nesta revisão.

Em Wang e Zhang (2017), os autores investigam a possibilidade de propor soluções que monitorem textos relevantes encontrados em fontes de dados externas. Redes sociais como o Twitter são fontes de dados relevantes para a análise proposta. O objetivo de monitorar textos de mídia social é centralizar o processamento de tweets e usá-los para construir modelos de *deep learning* para predizer a probabilidade de um ataque DDoS. Com base nas palavras contidas nos tweets, os modelos podem predizer a ocorrência de alguns ataques no dia seguinte.

Em Anuar et al. (2018), os autores usam uma Rede Neural Artificial (do inglês, *Artificial Neural Network* - ANN) (Shanmuganathan, 2016) para predizer quais *botnets* podem realizar ataques no dia seguinte. A informação utilizada como entrada da ANN é uma combinação dos ataques que ocorreram anteriormente na forma de tráfego de rede e outras informações fornecidas por fontes externas. Centralizadamente, a ANN processa dados como IP de origem e destino do pacote, a carga útil (do inglês, *payload*), o tipo de *botnet* e outras informações e resulta em um erro quadrático médio (do inglês, *Mean Square Error* - MSE) de 0,0053.

3.3.2 Modelos Estatísticos

O estudo de Sapienza et al. (2018) propõe uma solução para a predição de eventos relacionados à segurança cibernética. Embora não se concentre na predição de ataques DDoS, pode ser utilizado como predição de ataques DDoS. Centralizadamente, a solução coleta dados de redes sociais, blogs de segurança cibernética e fóruns da *dark web*. Com os dados coletados de fontes externas, a solução analisa o texto em busca de novos termos que possam representar ataques. Quando o modelo estatístico identifica um novo termo na base, ele gera um alerta indicando a possibilidade do ataque. Assim, é possível identificar um ataque com antecedência de dias se houver menção do ataque nas bases analisadas.

O estudo de Tse e Carley (2017) avalia como predizer a tendência de ataques DDoS usando eventos políticos, diplomáticos, informativos, militares e econômicos coletados de fontes de dados externas. Para isso, os autores fazem simulações centralizadas com eventos alinhados a dados para representar a relação entre os países para verificar se a tendência de ataque DDoS aumentará. A proposta é modelar cada evento conforme a hostilidade, severidade, origem e tipo. Os autores usam métodos estatísticos para predizer se haveria aumento nos ataques DDoS na China e na Rússia com mais de um dia de antecedência. Além disso, os autores também apresentam situações em que as simulações indicam uma diminuição dos ataques DDoS, embora a realidade mostre um aumento dos ataques DDoS.

3.3.3 Modelos Híbridos

Os autores do estudo de Wang et al. (2017) projetam dois modelos complementares para predizer os ataques DDoS. O primeiro modelo prediz a magnitude dos ataques (modelo temporal) usando ARIMA, e o segundo faz a modelagem comportamental dos atacantes (modelo espacial) usando redes neurais. Os autores propuseram um terceiro modelo, a união dos dois primeiros, para predizer ataques DDoS de modo mais acurado. Durante os testes, os autores utilizaram uma base de dados privada com 50.704 ataques DDoS. Os autores utilizaram 40.563 ataques para treinamento e 10.141 ataques para o teste. Combinando os dois primeiros modelos, a solução prediz ataques DDoS com um desvio quadrático médio (do inglês, *Root-mean-square deviation* - RMSE) de 2,72 dias, mais de um dia antes do lançamento do ataque.

Tabela 3.1: Resumo dos Estudos de Predição de DDoS com Relação aos Critérios de Classificação

Referência	Tempo	Arquitetura	Método	Dados	Avaliação
Salemi et al. (2021)	CP	CE	Híbrido	TR	Experimentação
Olabelurin et al. (2015)	CP	CE	AM	AL	Simulação
Fadlullah et al. (2011)	CP	CE	AM	TR	Simulação
Jaber et al. (2017)	CP	CE	AM	TR	Experimentação
Kwon et al. (2017)	CP	CE	AM	TR	Experimentação
Machaka et al. (2022)	CP	CE	AM	TR	Experimentação
SU et al. (2018)	CP	CE	AM	TR	Simulação
Muhammad et al. (2020)	-	CE	AM	TR	Experimentação
Ali e Al-Shaer (2013)	CP	CE	MK	LA	Experimentação
Moudoud et al. (2021)	CP	CE	MK	LA	Simulação
Abaid et al. (2016)	CP	CE	MK	AL	Experimentação
Holgado et al. (2020)	CP	CE	MK	AL	Experimentação
Shin et al. (2013)	CP	CE	MK	TR	Experimentação
Adegboyega (2015)	CP	CE	ME	TR	Experimentação
Fan et al. (2018)	CP	CE	ME	TR	Experimentação
Leros e Andreatos (2019)	CP	CE	ME	TR	Simulação
Pelloso et al. (2018)	CP	CE	ME	TR	Experimentação
Savchenko et al. (2020)	CP	CE	ME	TR	Simulação
Yin e Nianqing (2009)	CP	CE	ME	TR	Simulação
Kivalov e Strelkovskaya (2022)	CP	CE	ME	TR	Simulação
Jog et al. (2015)	CP	DI	ME	TR	Simulação
Liu et al. (2015)	LP	CE	AM	DE	Experimentação
Wang e Zhang (2017)	LP	CE	AM	DE	Experimentação
Anuar et al. (2018)	LP	CE	AM	Híbrido	Experimentação
Sapienza et al. (2018)	LP	CE	ME	DE	Experimentação
Tse e Carley (2017)	LP	CE	ME	DE	Simulação
Wang et al. (2017)	LP	CE	Híbrido	TR	Experimentação

Aprendizado de máquina = (AM), Modelos estatísticos = (ME), Baseado em Markov = (MK), Centralizado = (CE), Distribuído = (DI), Curto prazo = (CP), Longo prazo = (LP), Tráfego de rede = (TR), *Logs* de aplicações = (LA), Alertas IDS = (AL) Fontes de dados externas = (DE).

3.4 DISCUSSÃO SOBRE OS TRABALHOS RELACIONADOS

Embora a literatura apresente diversos estudos para detecção de ataques DDoS, o mesmo não ocorre com a predição de ataques DDoS. Dentre os 2.682 estudos analisados, este trabalho identificou apenas 27 estudos que realizam predições. Propor soluções para predizer ataques DDoS é uma tarefa desafiadora, mas necessária. O interesse e a necessidade de predição de ataques DDoS tendem a crescer nos próximos anos, contribuindo para o campo da segurança cibernética. O primeiro motivo para esse crescimento potencial é a vantagem que a predição de ataques oferece aos administradores de rede sobre os atacantes, uma vez que predizer ataques é uma das poucas maneiras de permitir que os administradores de rede não sejam surpreendidos pelos atacantes. Assim, os administradores de rede podem ter mais tempo para lidar com o ataque, agindo para reduzir os danos causados pelos atacantes. Embora predizer ataques DDoS não seja uma tarefa trivial, a literatura apresenta a viabilidade de predizer ataques. Apesar da dificuldade em desenvolver soluções para realmente predizer ataques DDoS, pesquisas nesta área são de fundamental importância. O segundo motivo pelo qual a predição de ataques DDoS irá evoluir é que existem muitas oportunidades de pesquisa. O estudo mais antigo identificado data de 2009. Isso demonstra que a preocupação com a predição de ataques DDoS é uma ação relativamente recente. Este trabalho identificou que 66% dos estudos citados foram publicados nos últimos sete anos, entre 2017 e 2023. O pico de publicações ocorreu entre 2017 e 2018, com cinco estudos em cada ano. Assim, a área da pesquisa tem potencial para evoluir cada vez mais.

Também é oportuno discutir sobre a definição de termos utilizados na área. Como a predição de ataque DDoS está crescendo, é importante evitar erros de interpretação quanto ao uso de termos comuns. Por exemplo, o uso do termo predição em outras áreas. O aprendizado de máquina emprega comumente o termo predição em duas situações: (*i*) a predição pode estar relacionada à classificação de um evento que ainda não ocorreu; (*ii*) a classificação de um evento que já aconteceu. Deste modo, durante a revisão foi comum identificar estudos que utilizam o termo predição para se referir a ação de detectar ataques após o atacante lançar os ataques. Apesar do uso da palavra predição, esse tipo de trabalho não se enquadra no conceito de identificar sinais/evidências do ataque antes que o atacante lance o ataque.

O campo de predição de ataque DDoS visa evitar que as equipes de segurança estejam despreparadas frente aos ataques (Nogueira et al., 2016). Por proporcionarem mais tempo para as equipes de segurança, as soluções com aspectos de longo prazo são desejáveis, mesmo que realizar tais predições não seja trivial. Wang et al. (2017); Liu et al. (2015) e Wang e Zhang (2017) apresentam estudos que ilustram isso. Liu et al. (2015) alcançaram uma média de verdadeiro positivo de 69% para predizer eventos relacionados à segurança cibernética que ocorreriam nos próximos três meses usando dados coletados da Internet. Wang e Zhang (2017) previram a ocorrência de ataques para os próximos dias com uma AUC baixa (inferior a 50%). Mesmo usando dados de mais de 40 mil ataques para fazer engenharia reversa dos ataques, o estudo de Wang et al. (2017) apresentou um RMSE de 2,72 dias. Salemi et al. (2021) obtiveram um *f1-score* de 82,81% prevendo ataques que ocorrerão nos próximos 20 segundos, e em Abaid et al. (2016), 81% das previsões ocorreram menos de um minuto antes do lançamento do ataque, sendo que o valor máximo obtido foi de 865 minutos (14 horas e 25 minutos). Esses resultados reforçam a hipótese de um compromisso (do inglês, *trade-off*) entre o tempo de predição e a precisão das soluções. Aumentar o tempo de predição pode causar um aumento na taxa de erro. Diminuir o tempo de predição pode tornar as soluções mais precisas. No entanto, o tempo para lidar com os ataques diminui.

Uma propriedade inerente à predição de ataques DDoS é o desbalanceamento das classes. As soluções para predição de ataques DDoS devem predizer ataques usando poucas informações

relacionadas aos ataques (Abdlhamed et al., 2017). Portanto, é razoável argumentar que as soluções operam em ambientes onde a quantidade mais significativa de dados é originada por usuários normais, não de atacantes. Essa diferença na quantidade de dados normais e maliciosos dificulta a criação de soluções para predizer ataques porque os autores devem demonstrar que as soluções podem lidar com possíveis desequilíbrios de dados. O aprendizado de máquina pode ilustrar esse problema. Algoritmos clássicos de aprendizado de máquina podem apresentar altas taxas de erro quando os dados tratados são desbalanceados. Uma maneira de resolver isso é pré-processar os dados. O *under-sampling* é uma estratégia para equalizar os dados da classe diminuindo o número de instâncias da classe majoritária, enquanto o *over-sampling* aumenta a quantidade de dados da classe minoritária (Kaur et al., 2019). Embora essas estratégias auxiliem no processo de aprendizado, é imprescindível cautela, pois essas técnicas de amostragem podem influenciar negativamente no resultado das soluções. Por exemplo, a solução pode ficar super adaptada para dados de treinamento não funcionando corretamente em ambientes reais. Por fim, empregar métricas de avaliação adequadas é essencial. A acurácia é uma métrica amplamente utilizada na literatura para avaliar os resultados das soluções. Porém, quando há dados desbalanceados, é fundamental complementá-los com outras métricas. Precisão, *recall* e F1-score são métricas que complementam a avaliação de soluções (Wang et al., 2021).

Os mecanismos de ataques DDoS que as soluções podem predizer é uma discussão importante para a evolução das soluções de predição de ataques. A Tabela 3.2 destaca o aspecto dos dados, os conjuntos de dados empregados durante a avaliação, os ataques avaliados e os atributos utilizados em cada estudo, apresentando-os na mesma ordem da Tabela 3.1. A Tabela 3.2 mostra que as soluções predizem diferentes mecanismos de ataque empregando diversas combinações de atributos. Os mecanismos de ataque com mais estudos identificados foram TCP SYN com cinco estudos e ICMP *flood* com dois. Não foi identificado o tipo de ataque avaliado em alguns estudos, como em Olabelurin et al. (2015) e SU et al. (2018). Estudos como Muhammad et al. (2020) e Abaid et al. (2016) utilizam ações de *botnet* para predizer ataques independentemente do mecanismo utilizado pelo atacante. A literatura complementa a discussão sobre quais mecanismos de ataque é possível predizer, apresentando a possibilidade dos atacantes realizarem testes contra a vítima antes de lançar efetivamente o ataque com toda a *botnet* (Jaber et al., 2017). Além disso, as fases anteriores ao ataque podem causar variação no tráfego de rede, fornecendo os sinais para predizer os ataques (Zi et al., 2010). Portanto, definir quais mecanismos de ataque a literatura pode predizer não é uma tarefa trivial porque existem vários mecanismos de ataque, os atacantes evoluem constantemente os mecanismos de ataque e diferentes topologias de rede precisam de proteção. Apesar da quantidade limitada de estudos identificados e da não trivialidade em identificar os tipos de ataques que podem ser preditos, no futuro, a literatura pode definir os limites para a predição de ataques DDoS.

Também é essencial discutir como avaliar as soluções propostas. Os autores avaliam as soluções de forma diferente para demonstrar que as soluções são viáveis, a Tabela 3.2 resume os conjuntos de dados empregados durante as avaliações. Para realizar a experimentação, alguns estudos avaliam soluções com diferentes conjuntos de dados disponíveis na literatura, como DARPA 1998, KDD CUP 1999 e CTU-13. Quatro estudos avaliam suas soluções utilizando o conjunto de dados DARPA 2000 porque sua documentação descreve claramente os estágios do ataque. Outros estudos coletam seus dados e podem não os disponibilizar. Além disso, alguns autores utilizam dados reais coletados em colaboração com empresas privadas. Esses dados geralmente não estão disponíveis para outros pesquisadores para evitar que informações confidenciais se tornem públicas. Além da experimentação, alguns autores utilizam software como MATLAB e NS-3 para simular cenários de ataque DDoS e avaliar soluções. A utilização de dados na literatura incentiva a replicação, comparação e inspeção de estudos. A simulação

permite a avaliação de cenários onde a literatura não possui conjuntos de dados. Avaliar as soluções é essencial porque demonstra que a solução funciona nas condições apresentadas, esclarece as limitações da solução e ajuda a compreender a proposta.

Tabela 3.2: Atributos e conjuntos de dados utilizados pelos estudos para predizer os ataques DDoS

Referência	**Conjunto de dados**	**Ataque**	**Atributos**
Salemi et al. (2021)	DARPA 1998	ICMP *flood*	Total de pacotes por segundo
Olabelurin et al. (2015)	Simulado	-	Descrição dos alertas, prioridade, protocolo, informações do sensor, endereços IPs e portas de origem/destino, hora e tipo do alerta
Fadlullah et al. (2011)	Simulado	Pacote malformado	Número de *hosts* defeituosos e o número de tentativas de autenticação por unidade de tempo
Jaber et al. (2017)	KDD CUP 1999	TCP SYN	Serviço, protocolo, origem, destino e número de logins com falha, entre outros
Kwon et al. (2017)	Coletado	Esgotamento de largura de banda	O número de endereços IP não duplicados que tentam se conectar via SSH nos servidores e o nível de importância dos servidores que podem ser alvos
Machaka et al. (2022)	DARPA 1999	Esgotamento de largura de banda	Total de pacotes a cada 10 segundos
SU et al. (2018)	Simulado	-	-
Muhammad et al. (2020)	CCC *dataset*	Comunicação C&C	A proporção de pacotes recebidos cujo tamanho está entre 1200-1299 bytes, a proporção de pacotes recebidos cujo tamanho é superior a 1400 bytes, a proporção de pacotes com a *flag* RST, a proporção de pacotes com a *flag* FIN e o menor tempo de intervalo entre pacotes recebidos, entre outros
Ali e Al-Shaer (2013)	Coletado	-	Tempo de captura, a origem e o destino do *log*, o tamanho do *log* e o tipo de evento
Moudoud et al. (2021)	Simulado	-	Logins de usuários, estado do dispositivo e operações do sistema
Abaid et al. (2016)	SysNet e ISCX *Botnet*	Tráfego de *bots*	-
Holgado et al. (2020)	DARPA 2000	TCP SYN	A descrição do alerta, a gravidade e os CVEs associados aos alertas

Shin et al. (2013)	DARPA 2000	TCP SYN	A entropia da porta de origem/destino, a entropia do endereço IP de origem/destino, a entropia do tipo de pacote, a quantidade de pacotes ICMP, UDP, TCP SYN e a quantidade total de pacotes por hora
Adegboyega (2015)	Wand Group	-	Taxa de pacotes por minuto
Fan et al. (2018)	DARPA 2000	TCP SYN	-
Leros e Andreatos (2019)	Simulado	-	Bits por segundo
Pelloso et al. (2018)	CTU-13 e DARPA 2000	TCP SYN e ICMP e UDP *flood*	Tamanho médio do pacote por segundo
Savchenko et al. (2020)	Simulado	*Slow-post*	Endereços IP de origem/destino, tempo de chegada do pacote e tamanho da janela TCP
Yin e Nianqing (2009)	Simulado	-	IP de origem/destino e porta
Kivalov e Strelkovskaya (2022)	Simulado	-	Intensidade do tráfego
Jog et al. (2015)	Simulado	-	-
Liu et al. (2015)	Coletado	-	Endereços IP envolvidos em ataques, eventos de segurança e prefixos BGP
Wang e Zhang (2017)	Coletado	-	Texto dos tweets
Anuar et al. (2018)	Cybersecurity Malaysia	Ataques de *botnets*	Endereço IP de origem e destino do ataque, porta de origem e destino do ataque, a carga útil da mensagem, tipo de *botnet* que realizou o ataque, dados geográficos dos endereços IP presentes nos ataques e dados do protocolo HTTP
Sapienza et al. (2018)	Coletado	-	Texto de blogs e tweets
Tse e Carley (2017)	Simulado	-	Destino, hostilidade, gravidade, origem e tipo
Wang et al. (2017)	Team Cymru	Tráfego de *bots*	Nível de atividade, magnitude do ataque, número do sistema autônomo de origem, duração dos ataques e início do ataque

Também é importante discutir sobre os atributos usados pelas soluções para predizer os ataques. Cada aspecto de dados define os limites para obtenção dos atributos. Por exemplo, é possível obter mais de 80 atributos analisando o tráfego de rede (Sharafaldin et al., 2018; Mahfouz et al., 2020). A Tabela 3.2 destaca os atributos utilizados nos estudos. É importante

ressaltar que este trabalho não identificou os atributos utilizados pelos estudos Jog et al. (2015); SU et al. (2018); Abaid et al. (2016); Fan et al. (2018). Esses estudos geralmente mencionam que utilizam tráfego de rede ou alertas IDS, mas não especificam os atributos extraídos do tráfego de rede ou dos alertas IDS. A Tabela 3.2 mostra a diversidade de atributos utilizados para a predição. O estudo Jaber et al. (2017) utiliza todas os 41 atributos disponíveis no conjunto de dados KDD CUP 1999[1]. Estudos como Salemi et al. (2021); Adegboyega (2015); Leros e Andreatos (2019) e Pelloso et al. (2018) usam apenas um atributo para predizer os ataques. O endereço IP de origem/destino e a porta são os atributos mais utilizados, presentes em 14,81% dos estudos (Olabelurin et al., 2015; Shin et al., 2013; Yin e Nianqing, 2009; Anuar et al., 2018) identificados neste trabalho. Por fim, destaca-se que os estudos de Holgado et al. (2020); Shin et al. (2013) e Pelloso et al. (2018) usam o mesmo conjunto de dados (DARPA 2000), mas contam com atributos diferentes.

3.5 OPORTUNIDADES DE PESQUISA RELACIONADAS COM A PREDIÇÃO

Esta seção apresenta as questões em aberto sobre a predição de ataques DDoS identificadas neste trabalho. As questões em aberto apresentam oportunidades de pesquisa que não foram abordadas na literatura ou que podem ser evoluídas. Para apresentar as oportunidades de pesquisa, esta seção está estruturada conforme a Figura 3.1.

3.5.1 Predição de Curto Prazo

Dentre os dez estudos que usam o aprendizado de máquina para predição de ataques DDoS, este trabalho observou o uso de algoritmos como SVM e K-means, bem como redes neurais como ANN e RBF. No entanto, é possível evoluir a literatura seguindo pelo menos sete hipóteses: 1) Diminuindo a dependência de informações rotuladas. As soluções podem usar algoritmos de aprendizado de máquina semi-supervisionado (Zhou e Belkin, 2014) ou não supervisionado, como DBScan (Ester et al., 1996) e *Self-Organizing Map* (SOM) (Kohonen, 1997). 2) Uso de técnicas de ensemble para melhorar a acurácia dos modelos. As técnicas de ensemble visam combinar classificadores para melhorar a acurácia, precisão e o *recall* (Rokach, 2005). O *Adaptive Boosting* (Adaboost) (Freund e Schapire, 1997) e o *Gradient Boosting* (Friedman, 2002) são exemplos de algoritmos que podem aumentar a precisão de soluções baseadas em aprendizado de máquina. 3) Usando algoritmos de *deep learning* para predizer ataques DDoS. Recentemente, soluções baseadas em *deep learning* ganharam impulso (Pak e Kim, 2017; Mahmoud et al., 2019) na solução de problemas em vários domínios, como reconhecimento visual (Pan et al., 2020), geração de imagens (Dosovitskiy e Brox, 2016) e robótica (Amarjyoti, 2017). Apesar dos esforços de Wang e Zhang (2017) usando a *Long short-term memory* (LSTM), algoritmos como *Gated recurrent unit* (GRU) (Dey e Salem, 2017), o *Autoencoder* (Nguyen et al., 2021) e o *Deep belief network* (DBN) (Chen et al., 2015) podem avançar o estado da arte. 4) Modelos de inteligência artificial explicável (do inglês, *Explainable artificial intelligence* - XAI). A explicabilidade é um problema crítico em soluções baseadas em IA, ao mesmo tempo que evita perdas causadas por erros de predição em potencial. Isso ocorre porque esses modelos se concentram na transparência e interpretabilidade (Došilović et al., 2018). Assim, se os administradores reconhecem predições supostamente incorretas, eles podem escolher não utilizar as predições. 5) Sistemas autoconfiguráveis. Os sistemas autoconfiguráveis podem identificar as condições ideias para operar (He et al., 2021; Kedziora et al., 2020). Assim, a solução extrai o melhor do algoritmo de aprendizado de máquina com o mínimo de interação humana

[1] https://kdd.ics.uci.edu/databases/kddcup99/kddcup.names

possível. 6) Aprendizado federado. O aprendizado federado é uma opção para incentivar a criação de soluções distribuídas utilizando aprendizado de máquina. As soluções de aprendizado federado criam modelos de aprendizado de máquina usando dados coletados em diferentes locais. Consequentemente, as soluções utilizam vários dados para realizar tarefas de classificação (Yang et al., 2019a). 7) Aprendizado por reforço (do inglês, *Reinforcement Learning* - RL). O RL é uma alternativa para diminuir a dependência de dados rotulados. A literatura apresenta a possibilidade de detectar ataques usando RL (Feng et al., 2020; Dake et al., 2021). Portanto, este trabalho levanta a hipótese da possibilidade de soluções usarem RL para predizer ataques DDoS.

Outro desafio a ser considerado no futuro das soluções é o aprendizado de máquina adversário (do inglês, *Adversarial Machine Learning*) (Huang et al., 2011; Liu et al., 2022). O aprendizado de máquina é benéfico para atuar em prol da segurança da informação (Stamp, 2017; Liu e Lang, 2019) devido a vantagens como a facilidade de lidar com a abundância de dados e a diversidade de algoritmos existentes. Contudo, os atacantes podem influenciar o treinamento dos modelos de aprendizado de máquina ou aprender como as soluções que usam aprendizado de máquina operam (Barreno et al., 2010). Assim, os atacantes podem explorar vulnerabilidades para enganá-las, tornando o resultado da predição do ataque DDoS incorreto ou mesmo criando maneiras de conduzir ataques para evitar a predição (Kianpour e Wen, 2020). Os atacantes podem criar alarmes falsos para ativar os mecanismos de mitigação, aumentando os custos relacionados ao serviço. Portanto, é oportuno que a literatura discuta maneiras de evitar as limitações do aprendizado de máquina adversário em soluções para a predição de ataques.

Este trabalho identificou cinco estudos que usam técnicas baseadas na teoria de Markov, três usam uma cadeia de Markov, um estudo utiliza um processo estocástico de Markov e um estudo usa o HMM. Uma maneira de avançar os estudos anteriores é avaliar variações da cadeia de Markov e do HMM, como *Markov random field*, *Hierarchical Markov models*, *Tolerant Markov model* e *Markov-chain forecasting models*. Visto que, essas varições podem fornecer vantagens como melhor modelagem do problema, mais detalhes entre os estados ou ser especializada para prever eventos futuros. Outra forma de avançar nos estudos anteriores é resolver as limitações apontadas nas soluções atuais. Algumas limitações ocorrem devido ao número limitado de estados que não representam completamente a realidade (Abaid et al., 2016), a dificuldade em generalizar a solução para predizer diferentes mecanismos de ataque (Holgado et al., 2020) e a necessidade de melhorar a precisão e a acurácia das predições (Shin et al., 2013). Este trabalho também argumenta que o uso do processo de decisão de Markov (Puterman, 1990) e do processo de decisão de Markov parcialmente observável (Spaan, 2012) podem avançar o estado da arte.

Este trabalho identificou dez estudos que utilizam modelos estatísticos para a predição de ataques DDoS, um dos aspectos mais estudados. Uma oportunidade de melhoria e evolução é criar soluções utilizando variações do ARIMA (Adegboyega, 2015). A Média Móvel Integrada Autorregressiva Sazonal (do inglês, *Seasonal Autoregressive Integrated Moving Average* - SARIMA) e a ARIMA Sazonal com Fatores Exógenos (do inglês, *Seasonal ARIMA with eXogenous Factors* - SARIMAX) são variações de ARIMA que lidam com dados sazonais. Portanto, existe a hipótese de que as soluções podem usar o SARIMA e SARIMAX para predizer ataques DDoS considerando a sazonalidade, por exemplo, do tráfego de rede (Hanbanchong e Piromsopa, 2012). A literatura reforça essa oportunidade ao mostrar que SARIMA e SARIMAX podem ser empregados para predizer o consumo de eletricidade (Erdogdu, 2007; Elamin e Fukushige, 2018) e o consumo de gás natural (Manigandan et al., 2021).

Devido ao sucesso alcançado com soluções baseadas em redes neurais, este trabalho apresenta a possibilidade de novas soluções predizerem ataques DDoS utilizando a computação bioinspirada. A computação bioinspirada tem recebido atenção de pesquisadores, sendo utilizada em áreas como Astronomia, Ciência da Computação e Matemática (Kar, 2016). Estudos sobre o

combate a ataques DDoS também utilizam computação bioinspirada (Rauf, 2018; Prathyusha et al., 2019). Em Tian et al. (2019), os autores propõem uma solução de detecção de ataques baseada em uma ANN com seus parâmetros iniciais definidos por um algoritmo de colônia de abelhas. Assim, a computação bioinspirada pode compor soluções para predizer ataques e evoluir o estado da arte.

3.5.2 Predição de Longo Prazo

Este trabalho identificou três estudos capazes de produzir predições de longo prazo usando aprendizado de máquina (Liu et al., 2015; Wang e Zhang, 2017; Anuar et al., 2018). Dentre as questões relacionadas à predição de ataques DDoS, este trabalho destaca a baixa acurácia (Wang e Zhang, 2017), a necessidade de apresentar probabilidades de ocorrência de ataques DDoS (Liu et al., 2015), e a preocupação com o desempenho computacional (Anuar et al., 2018). Produzir predições corretas de longo prazo é uma das questões em aberto mais desafiadoras. As predições realizadas incorretamente podem causar custos desnecessários e as predições não realizadas podem deixar um serviço vulnerável. Portanto, minimizar a taxa de erro necessita de evolução constante. Uma maneira de transferir a responsabilidade e aliviar esse problema é produzir probabilidades sobre um ataque futuro. Assim, as equipes de segurança podem decidir que tipo de ações tomar em diferentes casos. Por fim, o desempenho para fazer a predição do ataque também é crucial. Se a solução atrasar a produção da predição, parte da vantagem da predição acaba perdida.

Este trabalho hipotetiza a possibilidade de propor soluções de longo prazo para predizer ataques DDoS inspirados em modelos estatísticos de previsão do tempo. Nos últimos anos, a área de previsão do tempo recebeu grandes investimentos (RU, 2020). Os países desenvolvidos preveem o tempo com alguns dias de antecedência, e algumas soluções preveem enchentes com até 10 dias de antecedência, correlacionando a previsão do tempo com modelos hidrológicos (Webster, 2013). Embora os modelos estatísticos de previsão do tempo sejam complexos e específicos, a área de previsão do tempo está evoluída. Assim, a hipótese deste trabalho é que, usando as lições aprendidas com a previsão do tempo, é possível aumentar o tempo para predizer ataques DDoS.

Este trabalho identificou que estudos baseados em Markov que utilizam tráfego de rede ou alertas IDS no aspecto de dados são de curto prazo no aspecto temporal. Uma hipótese para mudar o aspecto temporal dessas soluções para longo prazo é o uso de dados de fontes externas. A hipótese poderia ser confirmada se fosse possível mapear os estágios precursores do ataque e produzir a probabilidade da evolução para um estado de ataque antes que o atacante lance o ataque. Este trabalho acredita que a consolidação dessa hipótese não seja uma ação trivial. Porém, como existem soluções que utilizam alertas de IDS, este trabalho hipotetiza a possibilidade de utilizar esses estudos como base para a proposição de novos estudos que utilizem fontes de dados externas.

A união de diferentes estudos é uma questão em aberto que pode resultar em novas soluções capazes de amenizar as desvantagens de cada solução. Um exemplo disso é a união de soluções que utilizam e processam diferentes tipos de dados, gerando uma solução híbrida sobre o aspecto dos dados e o aspecto metodológico. Uma hipótese plausível é a união de soluções que usam tráfego de rede com soluções que usam dados disponíveis de fontes de dados externas. Desta forma, é possível aumentar o tempo de predição das soluções e melhorar a precisão. Este trabalho acredita nessa possibilidade porque classificou como predição de longo prazo os quatro estudos que utilizam fontes de dados externas, o que indica que eles podem predizer ataques DDoS com pelo menos um dia de antecedência. A desvantagem desses estudos é a possibilidade da solução não identificar um ataque ou a chance elevada de gerar falsos positivos. Além disso, este trabalho classificou 13 dos 15 estudos que usam tráfego de rede como estudos que realizam predições

de curto prazo. Assim, se for possível combinar soluções que utilizam dados disponíveis em fontes externas de dados com soluções que utilizam tráfego de rede, possivelmente a precisão poderá ser aumentada. Porque quando uma solução falha em predizer o ataque ou gerar um falso positivo, a outra complementa a predição. Além de proporcionar ao usuário um aumento no tempo de predição de soluções que utilizam exclusivamente o tráfego de rede.

3.5.3 Outras oportunidades de pesquisa

Uma das questões em aberto mais importantes é a quantidade limitada de artigos que tratam do aspecto da arquitetura. Dentre todos os 27 estudos identificados, este trabalho classificou apenas um que processa os dados de modo distribuído. Nos casos em que a quantidade de dados a serem processados pela solução são extremamente volumosos, é plausível imaginar que soluções que centralizam o processamento em um único ponto podem não analisar todas as informações a tempo de realizar a predição do ataque DDoS. Porém, é um desafio propor soluções capazes de processar informações em diferentes pontos e ainda conseguir predizer ataques DDoS. Mesmo com os desafios relacionados à criação de soluções que operem distribuidamente, elas apresentam vantagens como a facilidade no escalonamento. Eles exigem menos capacidade de computação e é possível instalá-los em dispositivos com menos recursos computacionais disponíveis, como roteadores. Além disso, a solução não para de funcionar completamente se uma parte ficar indisponível.

Zargar et al. (2013) e Gupta e Badve (2017) destacam a necessidade de cooperação entre diferentes entidades de rede para criar estratégias para melhorar a defesa contra os ataques DDoS. Estudos como os de Oikonomou et al. (2006); Rodrigues et al. (2017); Tayfour e Marsono (2020) propõem soluções cooperativas para detecção e/ou mitigação. As soluções de colaboração para a defesa contra os ataques DDoS podem ser entre dispositivos alocados na vítima e na fonte ou entre dispositivos alocados nas redes centrais e a vítima (Sachdeva et al., 2009). Em Zhou et al. (2010), os autores citam a possibilidade de detectar ataques em seus estágios iniciais combinando informações de várias redes. Arquiteturas distribuídas, correlação de alerta, privacidade do usuário e alta precisão são desafios que as soluções colaborativas devem abordar (Zhou et al., 2010). Mesmo com muitos desafios, este trabalho hipotetiza a possibilidade de propor soluções colaborativas para predizer ataques DDoS.

A seleção de características utilizadas para predizer ataques DDoS pode gerar muitas contribuições. O Capítulo 2 apresenta a diversidade de mecanismos de ataque e as fases típicas de uma *botnet*. A Tabela 3.2 mostra que ainda não há consenso sobre os atributos ideais para predizer ataques DDoS. Portanto, identificar os atributos mais afetados pelas fases que antecedem o início de cada tipo de ataque é essencial para evoluir na predição de ataques DDoS e diminuir a taxa de erros. Esta questão em aberto é relevante para todos os aspectos dos dados (Figura 3.1) porque as soluções podem investir recursos computacionais apenas para processar os dados que ajudam a predizer ataques DDoS. Além disso, os atributos podem variar conforme os mecanismos de ataque, e podem variar segundo a metodologia empregada pela solução (Figura 3.1). Por exemplo, o conjunto de atributos ideal para construir modelos de aprendizado de máquina pode não ser ideal para construir soluções baseadas na cadeia de Markov. Portanto, cada solução pode ter um processo de seleção de características diferente. A seleção de características para detecção de ataques DDoS é um tema abordado na literatura (Chandrashekar e Sahin, 2014; Miao e Niu, 2016; Venkatesh e Anuradha, 2019; Batchu e Seetha, 2021; Kshirsagar e Kumar, 2021). Portanto, é possível iniciar a análise para seleção de características coletadas no tráfego de rede para a predição tendo como modelo os estudos que realizaram a seleção de características para detecção de ataques DDoS. Por fim, cada fase de ataque pode ser uma seleção

de características especializadas. Por exemplo, o estudo de (Feng et al., 2018) concentra-se na seleção de características para a comunicação C&C.

É oportuno que as novas soluções de predição de ataques DDoS forneçam detalhes sobre os ataques. Os resultados da análise dos estudos apresentados neste capítulo indicam que, geralmente, as soluções apenas notificam os administradores de rede sobre um ataque futuro. Em alguns casos, como em soluções baseadas em cadeias de Markov, apresentam a probabilidade de ocorrência do ataque. Este trabalho levanta a hipótese de que a não trivialidade do tema e a quantidade limitada de estudos que realizam a predição de ataques DDoS causam a falta de detalhes sobre ataques DDoS. Portanto, novas oportunidades de pesquisa podem apontar: (*i*) a probabilidade de o ataque acontecer; (*ii*) o momento da ocorrência do ataque; (*iii*) a magnitude do ataque; (*iv*) o mecanismo de ataque, entre outros detalhes sobre o ataque. Esses detalhes tornarão as soluções mais fáceis de usar porque os administradores de rede podem realizar diferentes ações conforme os detalhes variam.

A IoT proporcionou uma revolução cibernética nos últimos anos (Mrabet et al., 2020), aplicada a soluções em diferentes áreas (Riazul Islam et al., 2016) como indústria (Xu et al., 2018), saúde (Yassein et al., 2019) e agricultura (Zou e Quan, 2017). Mas a IoT também revolucionou o nível de ameaças cibernéticas (Yu e Guo, 2019; Li et al., 2020), introduzindo novos desafios para as soluções de segurança cibernética. Um exemplo relevante de novos desafios de segurança cibernética é o ataque realizado em 2016, em que os atacantes usaram dispositivos IoT para conduzir um ataque interrompendo o acesso a serviços importantes, como GitHub, Twitter e Netflix (Williams, 2016; Mahjabin et al., 2017). Dentre os estudos identificados neste trabalho, apenas dois estudos consideram redes IoT (Sapienza et al., 2018; Moudoud et al., 2021). Esse fato não indica que os outros estudos mencionados acima não operem para redes IoT. Mas pode haver uma degradação do desempenho devido às peculiaridades das redes IoT. Assim, é apropriado que novos estudos considerem este novo paradigma e produzam soluções específicas para lidar com as propriedades e desafios únicos das redes IoT (Alhanahnah et al., 2018), ou pelo menos, apresentem resultados que considerem as redes IoT.

Assim como acontece com as redes IoT, a arquitetura SDN está recebendo mais atenção na indústria e na academia. A arquitetura SDN oferece algumas vantagens, como economia de custos relacionada à aquisição e instalação de equipamentos (Benzekki et al., 2016), oportunidades de reconfiguração de rede (Somani et al., 2017) e inspeção profunda de pacotes (Somani et al., 2017). Como nas redes IoT, as soluções especializadas para o paradigma SDN podem superar a precisão das soluções generalistas. Assim, este trabalho identificou pelo menos dois estudos relacionados à predição de ataques DDoS em SDN (Adegboyega, 2015; Wang et al., 2017). No entanto, novos esforços ainda são necessários para lidar com as especificidades da arquitetura SDN para fornecer aos administradores mais tempo para lidar com os próximos ataques DDoS.

A teoria dos grafos resolve problemas em áreas como física, química (Foulds, 2012) e geometria (Bondy et al., 1976). Soluções baseadas em grafos preveem a demanda de tráfego (Xiong et al., 2019) e velocidades de tráfego futuras (Xie et al., 2020). As soluções de detecção de ataques DDoS também podem usar grafos (Li et al., 2019b). Este tipo de solução mapeia padrões de comunicação entre *bots* e vítimas (Jing e Wang, 2020) para identificar relacionamentos que representam ataques. Devido ao amadurecimento das técnicas baseadas na teoria dos grafos e o surgimento de ferramentas como as *Graph Neural Networks* (Scarselli et al., 2009), este trabalho hipotetiza a possibilidade de propor soluções para a predição de ataques DDoS com base em grafos. Contudo, atingir esse objetivo não é uma tarefa trivial.

A proteção da privacidade dos usuários deve ser uma preocupação inerente a todas as soluções de predição de ataques DDoS. A *General Data Protection Regulation* (GDPR) (UE, 2016) válido para os países da União Europeia, a *California Consumer Privacy Act* (CCPA) (Califórnia,

2018) usado no estado da Califórnia nos EUA, e a Lei Geral de Proteção de Dados (LGPD) (Brasil, 2018) válida no Brasil são leis que visam garantir a privacidade dos usuários. Por exemplo, o tráfego de rede é o aspecto dos dados mais utilizado pelas soluções identificadas neste trabalho. Essas soluções devem se preocupar com o tratamento do tráfego de rede. Visto que, caso os dados relativos ao tráfego da rede se tornem públicos, a solução pode infringir a privacidade dos usuários. É mais prudente propor soluções que evitem a manipulação e o armazenamento das cargas úteis dos pacotes de rede. Portanto, soluções menos invasivas são desejáveis (Klement et al., 2020).

3.6 RESUMO

Devido ao grau de periculosidade dos ataques DDoS, a literatura deve considerar todas as maneiras de reduzir possíveis danos. A predição de ataques é um mecanismo de defesa que vem ganhando atenção na literatura. As técnicas de predição criam evidências de um ataque DDoS que ainda não aconteceu, para fornecer mais tempo para os administradores de rede lidarem com o ataque. Este capítulo apresentou 27 estudos que propõem soluções para a predição de ataques DDoS. A partir desses estudos, este trabalho propôs uma classificação com base nos principais aspectos das soluções. O primeiro aspecto está relacionado ao tempo antes do lançamento do ataque que as soluções podem prevê-lo. Este trabalho identificou duas possibilidades, estudos que realizam predições de curto prazo com menos de 24 horas de antecedência e de longo prazo, que podem predizer ataques com mais de 24 horas de antecedência. Os estudos de curto prazo são a maioria dos estudos encontrados, com 20 dos 27 estudos, e apenas seis estudos possuem aspecto de longo prazo. O aspecto da arquitetura está relacionado ao modo como as soluções processam os dados. Este trabalho identificou que as soluções atualmente executam processamento centralizado ou distribuído. Este trabalho identificou apenas um artigo com aspecto distribuído, enquanto o restante apresentou aspectos centralizados. O aspecto metodológico apresenta as técnicas utilizadas pelas soluções para predição dos ataques. Este trabalho identificou estudos que usam modelos estatísticos, aprendizado de máquina, soluções baseadas na teoria de Markov e híbridos. Modelos estatísticos e aprendizado de máquina são os aspectos metodológicos mais comuns, com dez estudos cada. O aspecto dos dados mostra quais informações as soluções usam. Este trabalho identificou que os estudos usam alertas IDS, tráfego de rede, *logs* de aplicações, dados coletados de fontes de dados externas ou uma abordagem híbrida. O tráfego de rede é o aspecto de dados mais comum, por ser utilizado em 15 estudos. Quatro estudos utilizaram fontes externas de dados, sendo todos com aspectos de longo prazo. Dentre os estudos encontrados, este trabalho identificou estudos que utilizam métodos clássicos como a cadeia de Markov, ARIMA, SVM e ANN. A criação de soluções distribuídas e cooperativas, a especialização de soluções para diferentes ambientes como IoT e SDN, a redução da dependência de dados rotulados para treinamento de modelos, o uso de algoritmos de *deep learning* e a criação de novos estudos capazes de realizar predições de longo prazo com alta taxa de sucesso ainda são questões em aberto. Assim, muitos novos estudos podem desenvolver técnicas de predição e ajudar a combater ataques DDoS.

4 A ABORDAGEM ESPA

Este capítulo apresenta a abordagem ESPA para a predição de ataques DDoS. A abordagem foi projetada para auxiliar os administradores de rede e as equipes de segurança no combate aos ataques DDoS, fornecendo-lhes mais tempo para lidar com os ataques. A hipótese que embasa a abordagem ESPA é a possibilidade da abordagem identificar sinais da preparação de ataques DDoS usando a teoria dos sinais precoces de alerta. Para seguir essa hipótese, a abordagem foi projetada observando as questões em aberto identificadas no capítulo anterior. Deste modo, este trabalho contribui com o estado da arte relacionado a predição de ataque DDoS com o intuito de prover uma vantagem para os administradores de rede frente aos ataques. Este capítulo está dividido da seguinte forma. A Seção 4.1 define a evolução da proposta. A Seção 4.2 detalha o funcionamento da abordagem ESPA para atingir a evolução definida. A Seção 4.3 apresenta uma estratégia para a seleção autônoma de algoritmos de aprendizado de máquina não supervisionado que a abordagem ESPA pode utilizar. A Seção 4.4 apresenta a análise da complexidade da engenharia de sinais proposta. Por fim, a Seção 4.5 resume o capítulo.

4.1 PROJETO DA ABORDAGEM ESPA

A Figura 4.1 ilustra o poder de adaptação e a evolução para qual a abordagem ESPA foi projetada para suportar. Para verificar, ou não, a validade da hipótese defendida neste trabalho, a abordagem ESPA foi inicialmente projetada para utilizar dados rotulados durante a etapa de treinamento dos modelos de aprendizado de máquina supervisionado (AM supervisionado na Figura 4.1). Em geral, utilizar o aprendizado de máquina supervisionado apresenta resultados com altas taxas de acertos. Ao treinar os modelos de aprendizado de máquina supervisionado, é possível que a solução limite-se a funcionar apenas para cenários onde o ataque opera similarmente ao comportamento treinado. É plausível existirem casos onde esse é o objetivo dos administradores de rede e das equipes de segurança. Contudo, essa possibilidade gerou uma evolução da proposta onde ela foi equipada para não demandar dados rotulados. Para isso, a abordagem ESPA utilizou o aprendizado de máquina não supervisionado com identificadores de *outliers* e redes neurais do tipo LSTM-Autoencoder. Mesmo verificando a validade da hipótese, a interpretabilidade dos resultados poderia ser aperfeiçoada. Para isso, este trabalho valeu-se de algoritmos clusterizadores, aprendizado de máquina não supervisionado, para automatizar a predição dos ataques DDoS, focando na interpretabilidade dos resultados.

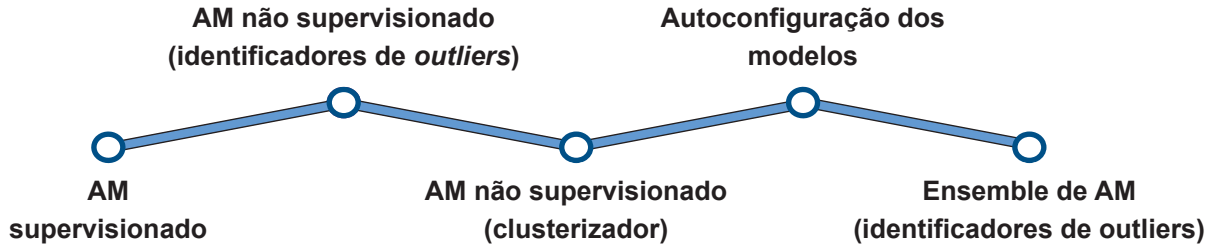


Figura 4.1: Visão Geral da Evolução Projetada para a Abordagem ESPA

Mesmo sendo uma forma de realizar a predição dos ataques DDoS, a configuração manual dos modelos poderia não estar extraindo todo o potencial do aprendizado de máquina. Assim, a abordagem ESPA usou soluções disponíveis na literatura e criou soluções para automatizar a seleção dos modelos de aprendizado de máquina que maximizam a predição dos ataques DDoS.

Portanto, a abordagem ESPA deve ser capaz de utilizar modelos configurados manualmente por especialistas ou autonomamente pela própria abordagem sem a utilização de dados rotulados. Outra forma de maximizar as predições dos ataques DDoS e reforçar a verificação da hipótese é utilizar o conhecimento de múltiplos identificadores de *outliers* por meio de um *ensemble*. Assim, a abordagem evita a geração de predições incorretas que tenham sido originadas por ruídos dos dados. Pois, mesmo que ruídos nos dados indiquem a possível preparação de um ataque DDoS para alguns modelos, majoritariamente o *ensemble* de identificadores de *outliers* deve concordar com a predição do ataque DDoS.

4.2 DEFINIÇÃO DA ABORDAGEM ESPA

Esta seção descreve a proposta deste trabalho para verificar a validade da hipótese e auxiliar no combate aos ataques DDoS. A proposta consiste na definição de uma abordagem para predição dos ataques DDoS. O dicionário Collins[1] define o termo abordagem como uma maneira de lidar com um problema. Ao propor a abordagem ESPA, este trabalho cria um meio para identificar os sinais da preparação dos ataques DDoS e predizê-los. Além disso, ao usar o termo abordagem, a proposta deste trabalho não fica limitada e pode ser adaptável a diferentes necessidades. Portanto, a abordagem ESPA foi projetada para criar características com base no tráfego de rede que suportem a predição dos ataques DDoS adaptando-se a várias necessidades.

A Figura 4.2 apresenta a visão geral da abordagem ESPA. A Etapa 1 da proposta é a preparação da abordagem. É necessário preparar a abordagem para que ela possa se adaptar às necessidades dos administradores de redes. Por exemplo, definindo os atributos do tráfego de rede a serem coletados ou o tipo do aprendizado de máquina a ser utilizado. Após a preparação, a abordagem ESPA começa a coletar o tráfego de rede seguindo os parâmetros definidos (Etapa 2). Na Etapa 3, a abordagem realiza a engenharia de sinais ao processar o tráfego de rede com a teoria dos sinais precoces de alerta (Seção 2.5) sobre o tráfego de rede coletado para criar as novas características. Como resultado, a abordagem ESPA cria características para identificar alterações no tráfego de rede e antecipar possíveis ataques DDoS. A engenharia de sinais está detalhada na Subseção 4.2.3. Na Etapa 4, um algoritmo de aprendizado de máquina utiliza as novas características para identificar os sinais da preparação dos ataques. Na Etapa 5, os administradores de rede são notificados sobre a ocorrência de um futuro ataque DDoS caso a abordagem proposta identifique sinais da preparação dos ataques DDoS. Cada etapa é detalhada a seguir.

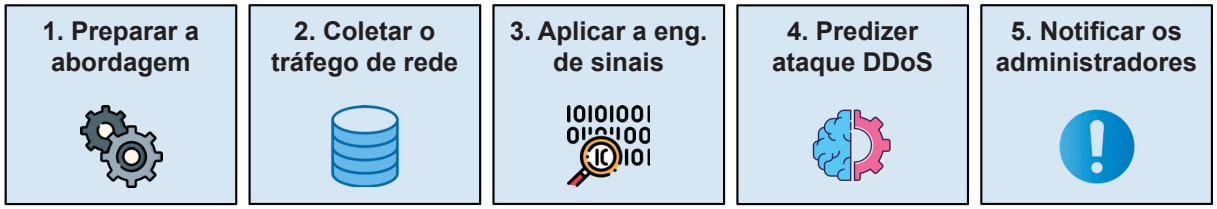


Figura 4.2: Visão Geral da Abordagem ESPA

4.2.1 Preparar a Abordagem ESPA

Para a abordagem funcionar corretamente é necessário poder adaptar-se aos diferentes objetivos (Figura 4.1) é necessário definir alguns parâmetros (Etapa 1 da Figura 4.2). Os atributos

[1] https://www.collinsdictionary.com/dictionary/english/approach

que coletados no tráfego de rede, os indicadores estatísticos que gerarão as novas características, o tamanho dos ciclos de captura e o algoritmo de aprendizado de máquina que automatiza a predição dos ataques são exemplos de parâmetros que a abordagem ESPA flexibiliza a definição. Esses parâmetros variam conforme as especificidades das redes nas quais a abordagem ESPA estiver inserida. Esse trabalho não predefiniu esses parâmetros com o intuito de apresentar uma abordagem capaz de ser adaptável para diferentes casos de uso.

A abordagem proposta foi projetada para coletar tráfego de rede centralizadamente ou distribuidamente. Em ambos os casos os responsáveis por realizar a coleta são os agentes. Agentes são instâncias de software que coletam o tráfego de rede. Na coleta centralizada, o roteador ou o firewall encaminha uma cópia do cabeçalho do pacote para um dispositivo que executa o agente. O processamento centralizado pode requerer muitos recursos dependendo do volume de dados transferidos. Assim, o administrador da rede pode utilizar a abordagem ESPA coletando o tráfego da rede de forma distribuída. A principal vantagem em distribuir os agentes é descentralizar e dividir a coleta do tráfego de rede. Portanto, quanto mais agentes, menos tráfego de rede cada agente terá de coletar. A necessidade do hardware estar distribuído para alocar os agentes é uma dificuldade a ser considerada no planejamento da adoção da abordagem ESPA. Apesar de distribuir a coleta do tráfego de rede ser vantajoso, é possível que os pouco sinais da preparação dos ataques também sejam divididos. Isso pode dificultar a identificação dos sinais e inviabilizar a predição. Como a abordagem é independente do modo de coleta do tráfego de rede, o administrador da rede define a coleta com base nos recursos computacionais disponíveis. Assim, em cada caso de uso da abordagem ESPA, é necessário encontrar o equilíbrio ideal para definir a quantidade de agentes e o modo de coleta.

A localização dos agentes dependerá do modelo de rede onde a abordagem ESPA atuará. A literatura sobre a defesa de ataques DDoS apresenta que os agentes podem estar posicionados em diferentes locais. Segundo Zargar et al. (2013), os agentes podem estar posicionados próximos à origem do ataque, nas redes intermediárias ou próximo da rede vítima. O posicionamento dos agentes interfere nas propriedades das soluções de defesa de DDoS como acurácia e o tempo de detecção do ataque. Quanto mais próximo da origem os agentes forem posicionados, mais cedo é possível detectar o ataque. Isso ocorre, pois os primeiros sinais de ataque estão presentes em redes próximas da origem do ataque. Por outro lado, quanto mais próximo da vítima, melhor são os resultados das soluções. Isso ocorre, pois a rede vítima concentra todos os dados do ataque, e não apenas parte do ataque como ocorre nas redes próximas aos atacantes (Zargar et al., 2013).

A Figura 4.3 ilustra as possíveis localizações dos agentes citados anteriormente. Na parte superior da figura, está representada a origem de ataque. Na parte inferior da imagem está representado o destino do ataque, ou seja, a rede vítima do ataque. No centro da figura, estão representadas as redes intermediárias que conectam a origem com o destino do ataque. Quanto mais próximo ao topo da figura, fonte do ataque, prematuramente o ataque será detectado. Quanto mais próximo da parte inferior da figura, maior tende a ser a acurácia das soluções de detecção de ataque DDoS (Zargar et al., 2013). Apesar da literatura não prover evidências que comprovem a validade dessas propriedades para soluções de predição de ataques DDoS, possivelmente elas serão verificadas para o contexto de predição de ataques DDoS.

A próxima definição é a escolha dos atributos do tráfego de rede que serão coletados pelos agentes. Para que a abordagem ESPA possa ser adaptável a diferentes casos de uso, este trabalho não predefiniu os atributos coletados no tráfego de rede. Assim, os usuários, os administradores de rede ou as equipes de segurança podem defini-los conforme suas necessidades e recursos. Apesar de não definir os atributos coletados no tráfego de rede, este trabalho discute importantes aspectos sobre a definição dos atributos e avalia algumas combinações no Capítulo 5.

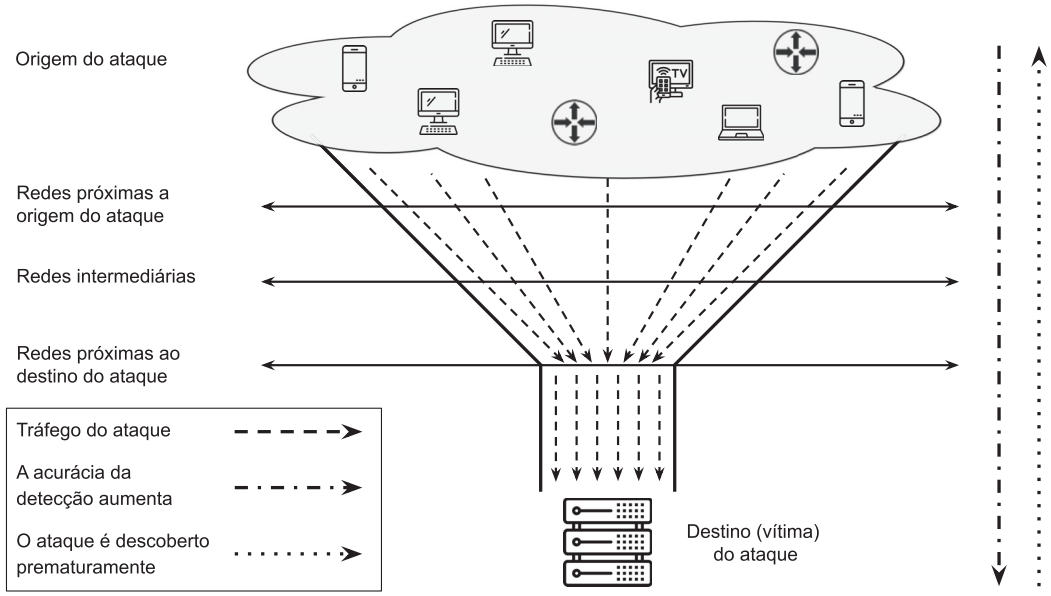


Figura 4.3: Localização dos Agentes (adaptado de Zargar et al. (2013))

O imperativo para a abordagem funcionar é que os atributos do tráfego de rede utilizados possam ter impacto diante de uma preparação do ataque. Por exemplo, Jaber et al. (2017) citam que atacantes podem realizar testes antes do ataque. Assim, atributos como a quantidade de dispositivos trocando pacotes ou quantidade de pacotes enviados/recebidos podem ser influenciados pela preparação dos ataques (Jaber et al., 2017). A busca por dispositivos ativos na rede, a identificação por portas abertas em servidores e em dispositivos da rede, a tentativa de invasão dos dispositivos e a execução de *malware* (Garcia et al., 2014; Sharafaldin et al., 2019; Holgado et al., 2020) são outros exemplos de ações de preparação dos ataques DDoS que podem impactar os atributos do tráfego de rede. As ações de preparação do ataque ocorrem nas fases de reconhecimento, recrutamento e comando e controle, que ocorrem antes do lançamento do ataque DDoS (Cisco, 2014; Sahu e Khare, 2020) (Seção 2.1).

Um importante compromisso (do inglês, *trade-off*) a ser considerado é a quantidade de atributos a serem coletados no tráfego de rede em relação ao poder computacional do dispositivo que vai alocar o agente. Pois quanto mais atributos do tráfego de rede forem escolhidos, maior será o poder computacional necessário para coletá-los. Quanto mais atributos do tráfego de rede forem processados, melhor tende a ser a acurácia obtida pela abordagem ESPA. Isso ocorre, pois cada vez mais os atacantes buscam esconder suas ações. Assim, o prelúdio do ataque pode provocar variação em poucos atributos do tráfego de rede. Processando mais atributos, maior é a chance da abordagem identificar sinais da preparação dos ataques. Contudo, processar mais atributos necessita de hardware compatível com a demanda, fato esse que pode não ser a realidade do usuário da abordagem proposta.

A abordagem ESPA foi projetada para consumir coletas do tráfego de rede na forma de valores numéricos e não violar a privacidade do usuário. A abordagem aceita atributos do tráfego de rede desde que o atributo corresponda a valores mensurados a partir da observação do tráfego na rede (Campbell, 2015). Por exemplo, a quantidade de pacotes por tempo, o tamanho médio dos pacotes por tempo, quantidade de pacotes enviados de um protocolo específico, entre outros. Caso o administrador de rede queria utilizar variáveis categóricas, essas devem ser transformadas em números. Por exemplo, protocolos como ICMP, TCP e o UDP podem ser representados usando o mapeamento de protocolos definido pela Autoridade para Atribuição de Números da

Internet (do inglês, *Internet Assigned Numbers Authority* -IANA)[2]. Assim, o protocolo ICMP seria expresso pelo código 1, o TCP seria o 6 e o UDP seria o 17. Quando o administrador de rede necessita usar códigos como alto e baixo, esses valores podem ser substituídos por 1 e 2, por exemplo. Essa peculiaridade é importante, pois, o tráfego de rede será transformado em sinais (Subseção 4.2.3), e, atualmente, essa transformação está limitada a variáveis numéricas. Além disso, a abordagem visa preservar a privacidade do usuário e estar em conformidade com as leis que regulamentam a proteção dos dados. Portanto, este trabalho projetou a proposta para utilizar somente dados disponíveis nos cabeçalhos dos pacotes, e nunca dados da carga útil do pacote. Mesmo que, somente o cabeçalho possa fornecer pistas para quebrar a privacidade dos usuários, utilizar somente o cabeçalho diminui o risco a privacidade dos usuários comparado com técnicas que utilizam todo o pacote.

A literatura apresenta alguns trabalhos que definem os melhores atributos do tráfego de rede para lidar com ataques DDoS. A abordagem ESPA toma como base o trabalho de Feng et al. (2018), onde os autores realizaram uma seleção de atributos para detectar comunicações de C&C. Os autores iniciaram a análise com 55 atributos, e após as investigações, os autores identificaram que utilizando 40 atributos o desempenho de detecção não variava. Além do conjunto de 40 atributos, os autores apresentaram os 10 atributos mais importantes para a detecção de C&C (Tabela 4.1). Dentre os atributos apresentados, este trabalho destaca a quantidade de pacotes enviados (Spc), a variância do tamanho dos pacotes recebidos (Rvar), o tempo de intervalo máximo no pacote recebido (RITmax) e a variância do intervalo de tempo dos pacotes recebidos (RITvar). Esses atributos foram destacados devido à utilização prévia deles em trabalhos anteriores conduzidos por membros do grupo de pesquisa CCSC Pelloso et al. (2018); de Neira et al. (2020); Araújo et al. (2022). A análise dos melhores atributos do tráfego de rede para a predição de ataques DDoS utilizando a proposta apresentada neste trabalho ainda necessita de esforços complementares que serão realizados futuramente (Seção 6.2). Por fim, a Tabela 5.17, apresentada no Capítulo 5 (Avaliação de Desempenho), apresenta outros 51 atributos do tráfego de rede que a abordagem ESPA pode utilizar. Isso mostra que, além dos 40 atributos citados nesta seção, a abordagem ESPA pode utilizar outros conforme a necessidade dos administradores de rede.

Tabela 4.1: Atributos Relevantes Para Detecção do Tráfego C&C (adaptado de Feng et al. (2018))

Atributo	Descrição
Spc	**Quantidade de pacotes enviados**
FlagR	Proporção de pacotes com *flag Reset* em uma sessão
Sip1	Proporção dos pacotes enviados com o tamanho de 100-199 bytes
Sip0	Proporção dos pacotes enviados com o tamanho de 0-99 bytes
FlagS	Proporção de pacotes com *flag* SYN em uma sessão
Svar	Variância de pacotes enviados
RITmax	**Tempo de intervalo máximo no pacote recebido**
RITvar	**Variância do intervalo de tempo dos pacotes recebidos**
Sip5	Proporção dos pacotes enviados com o tamanho de 500-599 bytes
Rvar	**Variância do tamanho dos pacotes recebidos**

A próxima definição é a escolha dos indicadores estatísticos. Os indicadores estatísticos definem como a abordagem estimará os sinais precoces de alerta utilizando como entrada os atributos coletados do tráfego de rede (engenharia de sinais). A definição dos indicadores

[2]https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml

estatísticos é extremamente importante para a abordagem. Pois a abordagem proposta baseia-se na possibilidade desses indicadores estatísticos apresentarem variações anteriores a transições críticas. Assim, é possível identificar essas variações e alertar os administradores de rede sobre a ocorrência de possíveis ataques DDoS. Além disso, as definições descritas anteriormente vão impactar diretamente o desempenho da abordagem. Isso ocorre, pois todo o tráfego de rede coletado pelos agentes serão convertidos em sinais precoces de alerta, utilizando a teoria definida nos indicadores estatísticos. A coleta demasiada de atributos do tráfego de rede, juntamente com a definição de muitos indicadores estatísticos, pode impactar no tempo de processamento da abordagem ESPA. A abordagem foi projetada para aceitar quaisquer indicadores estatísticos que possam gerar sinais precoces de alerta capazes de representar uma transição crítica, consequentemente representar um ataque futuro. A Seção 2.5 apresenta alguns indicadores estatísticos já definidos na literatura. Este trabalho não definiu previamente os indicadores estatísticos a serem utilizados por existir a possibilidade de que, no futuro, a literatura proponha indicadores estatísticos que melhor se adaptem a predição de ataque DDoS. Além disso, é possível que um conjunto de indicadores estatísticos possam ser adequados para um grupo específico de ataques. Assim, esse grupo de indicadores estatísticos podem se adequar melhor às necessidades do usuário. Por fim, trabalhos futuros podem explorar a seleção dos melhores indicadores estatísticos conforme o estado atual da rede.

Escolher os ciclos de captura é a próxima definição da abordagem ESPA. Os ciclos de captura correspondem ao tempo em que todos os agentes vão considerar para iniciar e finalizar cada coleta dos atributos da rede. A abordagem ESPA pode utilizar diferentes unidades de tempo como horas, minutos, segundos e milissegundos. Novamente, a abordagem ESPA demanda um compromisso entre as definições e o ambiente. A escolha de pequenos ciclos de captura podem não ser significativos para realizar a predição do ataque, devido à pouca informação disponível para ser analisada. Por outro lado, ciclos demasiadamente grandes podem ser processados somente após o lançamento do ataque, perdendo assim a vantagem da realização da predição de ataques DDoS.

A central de inteligência é a outra instância de software a ser preparada. A central de inteligência é responsável por processar a coleta do tráfego de rede para evidenciar os sinais da preparação dos ataques. Além disso, a central de inteligência usa algoritmos de aprendizado de máquina para automatizar a identificação dos sinais de preparação do ataque DDoS. Este trabalho projetou a central de inteligência para ser compatível com vários algoritmos de aprendizado de máquina. A central de inteligência pode utilizar algoritmos *shallow learning* supervisionados como o SVM e o Adaboost, bem como algoritmos de *deep learning* supervisionadas como o CNN e a MLP. Além do aprendizado supervisionado, a central de inteligência pode usar de aprendizado de máquina não supervisionado. Para isso, algoritmos como o K-means, o *One-class* SVM e os *Autoencoders* (Liu e Lang, 2019) podem compor a central de inteligência. Por fim, é necessário que a central de inteligência seja acessível para os agentes, pois ao longo do ciclo de vida da abordagem os agentes precisam se comunicar com a central de inteligência enviando os dados coletados do tráfego de rede.

4.2.2 Coletar o Tráfego de Rede

A Figura 4.2 introduz a coleta do tráfego de rede como a Etapa 2 da proposta. A primeira ação realizada pelos agentes na etapa de coleta do tráfego de rede é a sincronização. As definições de ciclo de captura ficam disponíveis para todos os agentes na central de inteligência. Assim, para iniciar a coleta, todos os agentes solicitam as definições para a central de inteligência e sincronizam o início da coleta de tráfego de rede. Após a sincronização, os agentes estão prontos para operar. Na abordagem ESPA, cada agente usa um monitor de rede para coletar o tráfego

da rede. Em ambientes reais, os agentes podem ser implantados em dispositivos físicos como computadores ou roteadores, ou em máquinas virtuais. O monitor de rede da abordagem funciona em modo de análise de pacote para a coleta de dados continuamente, e em modo off-line, usando dados históricos como entrada. O agente repete o processo de coleta enquanto a abordagem estiver ativa. Assim como representado pela Figura 4.4, ao final de cada ciclo de captura, o agente possui os dados coletados na rede na forma dos atributos previamente definidos. Os agentes enviam para a central de inteligência os atributos coletados, um identificador único local, a data de início e do fim do ciclo de captura. O processo de coleta de tráfego de rede é o mesmo para a coleta distribuída e centralizada. Pois, ao final da coleta, os dados são disponibilizados para a central de inteligência processá-los e predizer os ataques.

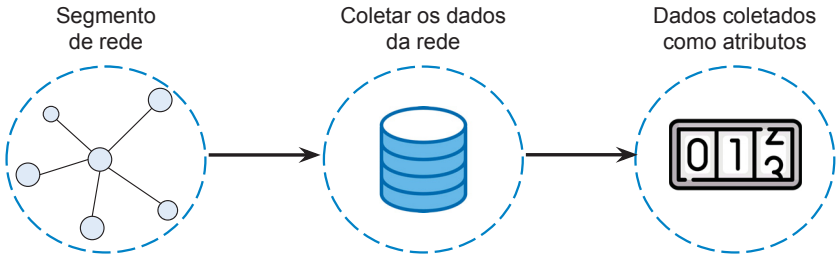


Figura 4.4: Representação do Processo de Coleta do Tráfego de Rede

4.2.3 Aplicar a Engenharia de Sinais

O processamento do tráfego de rede sob a perspectiva dos indicadores estatísticos (engenharia de sinais) é uma das etapas mais importantes da abordagem ESPA (Etapa 3 da Figura 4.2) e é um dos diferenciais deste trabalho frente a literatura. A central de inteligência processa os dados coletados por todos os agentes para predizer os ataques DDoS. A central de inteligência é responsável por sincronizar os dados coletados, processar os dados faltantes, realizar a engenharia de sinais, regularizar as novas características e classificá-las. A Figura 4.5 apresenta as ações realizadas pela central de inteligência que fazem parte da Etapa 3 (Figura 4.2). Todas essas ações estão detalhadas nesta subseção.

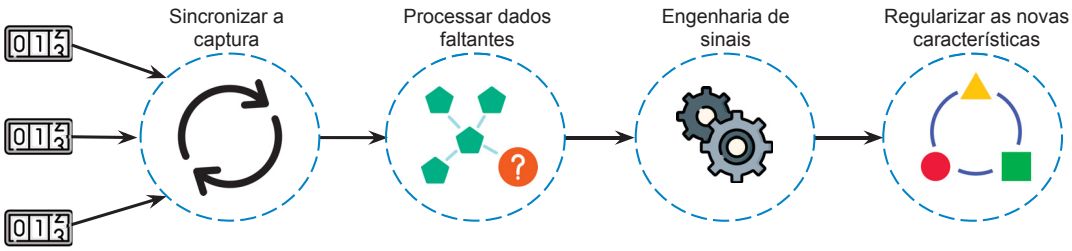


Figura 4.5: Ações Realizadas na Etapa 3 da Abordagem ESPA

O Algoritmo 1 apresenta o pseudocódigo da central de inteligência. A primeira ação da central de inteligência é recuperar as configurações definidas pelos administradores de rede (linha 2 do Algoritmo 1). Este trabalho projetou a central de inteligência para recuperar as configurações a cada novo ciclo de captura. Assim, a central de inteligência sempre realizará a predição dos ataques DDoS usando as últimas configurações definidas pelos administradores de rede. Isso é possível, pois a função de recuperar as configurações apenas realiza uma consulta no banco de dados para buscar as configurações e coleta o modelo de aprendizado de máquina

já definido anteriormente. Essas ações não impactam negativamente o tempo de execução da abordagem ESPA. Visto que, a literatura possui estratégias para aperfeiçoar as buscas.

Algoritmo 1 Pseudocódigo da Central de Inteligência

```
1:  while True do
2:      tempo_ciclo, tempo_espera, modelo_ml ← recuperar_configurações()
3:      esperar(tempo_ciclo)
4:      coleta_tráfego_rede, ciclo_completo ← recuperar_coleta_tráfego_rede()
5:      if not ciclo_completo then
6:          esperar(tempo_ciclo * tempo_espera)
7:          coleta_tráfego_rede, ciclo_completo ← recuperar_coleta_tráfego_rede()
8:          if not ciclo_completo then
9:              marcar_agentes_como_offline(coleta_tráfego_rede)
10:         end if
11:     end if
12:     coleta_tráfego_rede ← processar_dados_faltantes(coleta_tráfego_rede)
13:     sinais ← engenharia_de_sinais(coleta_tráfego_rede)
14:     sinais ← reescalar_os_sinais(coleta_tráfego_rede)
15:     tem_aviso_ataque ← modelo_ml.predizer(sinais)
16:     if tem_aviso_ataque then
17:         notificar_administradores()
18:     end if
19:     marcar_coleta_como_processada(coleta_tráfego_rede)
20: end while
```

Um grande desafio para a central de inteligência é sincronizar os sinais recebidos se a coleta for distribuída. Isso ocorre, pois as latências entre a central de inteligência e os diversos agentes podem ser diferentes. Assim, a central de inteligência pode receber as informações fora de sincronia, fato que pode implicar na incorreta classificação dos sinais. Para evitar erros por falta de sinais, a central de inteligência aguarda receber os sinais de todos os agentes pelo tempo de um ciclo completo. Caso o agente não envie os sinais, a central de inteligência aguarda por mais três ciclos. Caso a central não receba os dados, ela marca o agente como off-line. Portanto, a central de inteligência não ficará atrasando a predição (linhas 3 a 11 do Algoritmo 1). O tempo de três ciclos é o padrão da abordagem, porém essa definição é alterável. Por fim, se a coleta for centralizada, o tempo de espera de três ciclos não é necessário. Assim, sempre que existir uma nova captura, a central de inteligência processa.

A recuperação da coleta do tráfego de rede (linha 4 do Algoritmo 1) utiliza o conceito de janela deslizante de tamanho fixo para eliminar tendências errôneas na geração das novas características e propiciar a avaliação da solução proposta ao longo do tempo (Zivot e Wang, 2003; Bury et al., 2020). A Figura 4.6 ilustra o conceito de janelas deslizantes de tamanho fixo. Ela apresenta uma série temporal com a medição da quantidade de pacotes recebidos em uma rede por segundo. Para uma janela deslizante de tamanho dois, a central de inteligência processa os dados presentes nas duas primeiras posições. Quando uma nova observação estiver disponível, nesse caso representado pela terceira posição da série temporal, a abordagem desliza e utiliza os dados das posições dois e três. A abordagem ESPA segue até o fim da série temporal ou enquanto recebe novos dados. A literatura ainda não é unânime sobre o tamanho ideal da janela deslizante de tamanho fixo. Portanto, para otimizar o tempo de predição dos ataques, este trabalho avaliou diferentes valores como 5% e 10% da base analisada na avaliação de desempenho (Capítulo 5). O

resultado da função de recuperação do tráfego de rede (linha 4 do Algoritmo 1) é uma matriz com a quantidade de linhas do tamanho da janela deslizante e uma coluna para cada atributo coletado pelos agentes. Ao projetar essa função deste modo, este trabalho separa as responsabilidades de cada função. Isso faz com que as próximas funções invocadas pela central de inteligência concentrem-se apenas em processar a coleta do tráfego de rede. Além disso, ao utilizar janelas deslizantes, a abordagem ESPA pode descartar os dados obsoletos e economizar recursos.

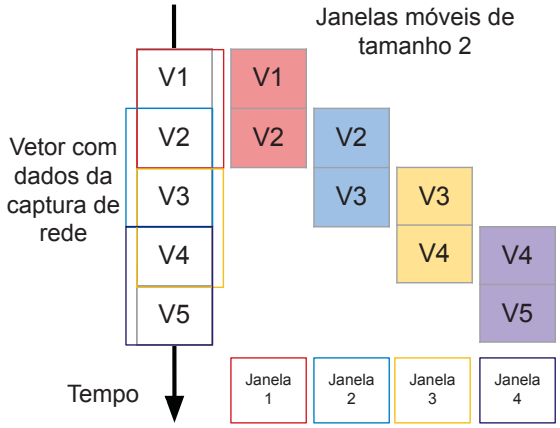


Figura 4.6: Exemplo do Conceito de Janela Deslizante de Tamanho Fixo

Após recuperar a coleta do tráfego de rede, a central de inteligência lida com dados faltantes. A abordagem ESPA pode regularizar (Figura 4.5) os sinais faltantes de três modos (linha 12 do Algoritmo 1). O usuário pode definir que a abordagem não deve utilizar os sinais com variáveis faltantes, assim a abordagem ESPA remove os sinais cujos dados não estão disponíveis. O usuário pode definir que a abordagem adicionará um valor baseado na moda, mediana ou média das últimas observações. Porém, a ação padrão definida para a abordagem é a atribuição da constante "-1" sempre que um dado não estiver disponível. É importante que a abordagem contemple a ocorrência de dados faltantes por ser possível que inconsistências na coleta ou na transmissão dos dados impliquem na inexistência de algum dado.

Após processar os dados faltantes, a central de inteligência transforma os atributos do tráfego de rede coletados em novas características utilizando os indicadores estatísticos definidos na teoria dos sinais precoces de alerta. Este trabalho definiu essa função como a **engenharia de sinais** (linha 13 do Algoritmo 1). Em inglês, a expressão *signal engineering*, foi utilizada na literatura com outros objetivos. Ao citar *signal engineering*, Yates (1920) refere-se a ramos especializados em sinalização automática, segura e confiável na engenharia ferroviária. Nafea e Ismail (2022) citam que lidar com sinais de eletroencefalografia não é uma tarefa trivial. Portanto, é possível aplicar uma engenharia de sinais sobre os dados para melhorar a viabilidade de análise para modelos de aprendizado de máquina. Um exemplo da engenharia de sinais é a aplicação da entropia sobre os dados. Assim, é possível medir a irregularidade e a imprevisibilidade das flutuações de sinal (Nafea e Ismail, 2022). O termo *signal engineering* também pode estar associado ao *deep learning* (Sunindyo et al., 2010) e à eletrônica (Schneider, 1999). Contudo, neste trabalho, o termo engenharia de sinais refere-se à aplicação da teoria dos sinais precoces de alerta sobre o tráfego de rede e está detalhado a seguir.

Como a engenharia de sinais é de suma importância, este trabalho apresenta o Algoritmo 2 com o pseudocódigo desta função. Primeiramente, a engenharia de sinais inicia a variável *sinais* que irá receber todas as novas características (linha 2 do Algoritmo 2). Esta variável é uma matriz que armazenará o resultado da engenharia de sinais. A estratégia padrão da abordagem ESPA para a geração das novas características é aplicar todos os indicadores estatísticos para todos os

atributos (linha 3 do Algoritmo 2). Assim, a quantidade de novas características por ciclo de captura gerado pela abordagem ESPA varia segundo a expressão $ews = a \times f \times i$. Onde o termo a é a quantidade de agentes que coletaram os dados, o termo f é a quantidade de atributos coletados por cada agente e o termo i corresponde à quantidade de indicadores estatísticos pré-definidos. O resultado dessa operação é o termo ews que indica quantidade de novas características geradas pela abordagem proposta (linhas 4 a 11 do Algoritmo 2).

Algoritmo 2 Pseudocódigo da Engenharia de Sinais

```
1:  function ENGENHARIA_DE_SINAIS(coleta_trfego_rede)
2:      sinais ← inicia_estrutura_dados()
3:      for atributo em coleta_tráfego_rede do    // iterar todos os atributos (dos agentes)
4:          dados ← atributo.dados()
5:          skew ← skewness(dados)
6:          kurt ← kurtosis(dados)
7:          cv ← cv(dados)
8:          s_max ← s_max(dados)
9:          s_fold ← s_fold(dados)
10:         s_hopf ← s_hopf(dados)
11:         sinais.acrescentar(skew, kurt, cv, s_max, s_fold, s_hopf)
12:     end for
13:     return sinais
14: end function
```

A engenharia de sinais é repetida para os todos os atributos do tráfego de rede coletados por todos os agentes a cada ciclo de coleta. Considerando os recursos disponíveis, os administradores podem optar limitar a produção de novas características. Isso significa que cada atributo do tráfego de rede será processado por apenas um indicador estatístico (par a par). Esse processo gerará apenas uma nova característica para atributo coletado. A combinação entre atributo do tráfego de rede e indicador estatístico pode ser definida pelo administrador de rede. Após gerar todas as novas características, a função as retorna para a central de inteligência que invocou a função (linha 13 do Algoritmo 2).

A Figura 4.7 exemplifica a engenharia de sinais em um ciclo de captura. No exemplo, a coleta do tráfego de rede é realizada por três agentes distintos em três segmentos de rede distintos. Cada um dos três agentes ($a = 3$) coleta dois atributos da rede, denominados genericamente de "Atributo 1" e "Atributo 2" ($f = 2$). Os agentes enviam os dados coletados para a central de inteligência realizar a engenharia de sinais, transformando cada atributo em novas características. O exemplo utilizou três indicadores estatísticos ($i = 3$). Assim, a central de inteligência transforma cada atributo coletado do tráfego de rede em três novas características diferentes baseadas nos indicadores estatísticos. Portanto, em um ciclo de captura com três agentes distribuídos, cada um coletando dois atributos do tráfego de rede e utilizando três indicadores estatísticos, a solução geraria 18 novas características ($ews = 3 \times 2 \times 3 = 18$). Caso a coleta fosse centralizada, o exemplo apresentaria o uso de apenas um agente ($a = 1$). Mantendo as outras definições, dois agentes ($f = 2$) e três indicadores estatísticos ($i = 3$) a solução geraria 6 ($ews = 1 \times 2 \times 3 = 6$) novas características. Assim, a matriz gerada pela engenharia de sinais teria 6 colunas, uma para cada nova característica.

Para lidar com dados em escalas diferentes, a abordagem ESPA pode utilizar a padronização, a estratégia *RobustScaler* e o dimensionamento Min-Max (linha 14 do Algoritmo 1). A padronização dimensiona os atributos removendo a média das observações dividido pelo

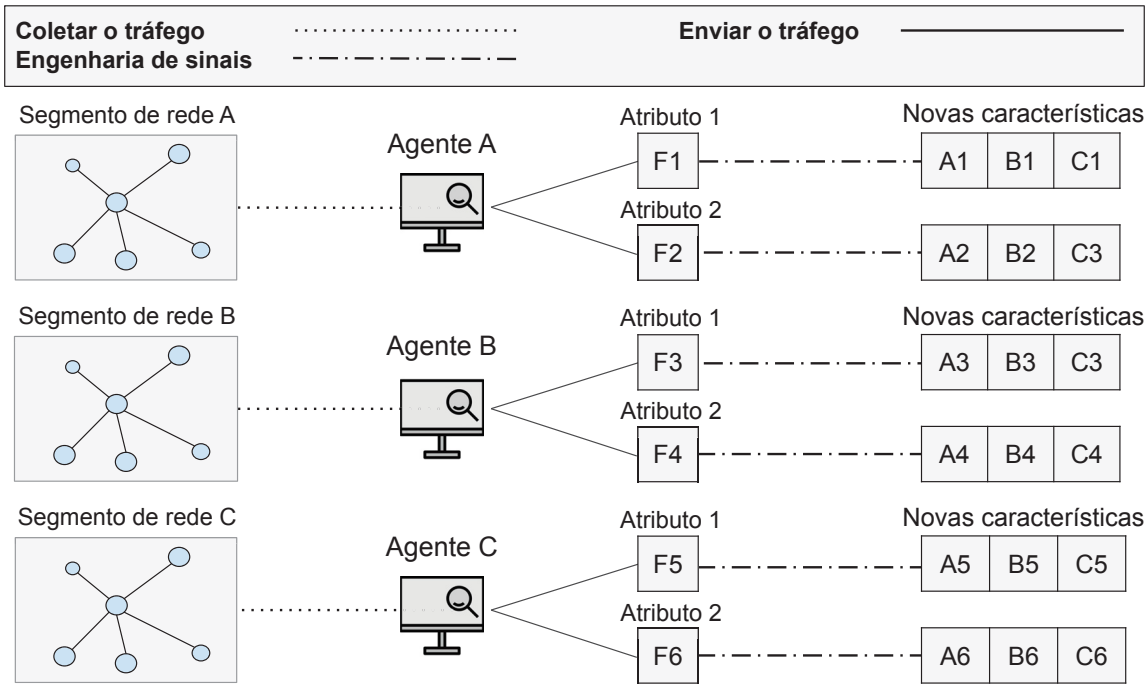


Figura 4.7: Exemplo da Engenharia de Sinais

desvio padrão das observações. A Fórmula 4.1 define a padronização, onde o termo $X(j)$ representa o valor de cada amostra do atributo analisado, o termo $\bar{x}$ representa a média e o termo s desvio padrão. A estratégia *RobustScaler* utiliza a mediana do conjunto de atributos para diminuir a observação analisada. O resultado dessa operação é dividido pelo valor do interquartil, representado pela subtração do terceiro com o primeiro quartil. A Fórmula 4.2 define o *RobustScaler*. O termo $X(j)$ representa o valor de cada amostra do atributo analisado, o termo M_d representa a mediana, o $Q_1(x)$ e o $Q_3(x)$ representam o primeiro e o terceiro quartis respectivamente (Sefara, 2019). A Min-Max é a estratégia padrão da abordagem ESPA para dimensionamento dos dados. A estratégia Min-Max dimensiona os valores dos atributos entre um intervalo pré-definido como zero e um. A Fórmula 4.3 define o cálculo do Min-Max. O $X(j)$ representa o valor de cada amostra do atributo analisado, o min_A e o max_A são respectivamente menor e o maior valor observado para o atributo. O min'_A e o max'_A representam o novo intervalo (Cao e Obradovic, 2015).

$$Padronização = \frac{(X(j) - \bar{x})}{s} \tag{4.1}$$

$$RobustScaler = \frac{X(j) - M_d}{Q_3(x) - Q_1(x)} \tag{4.2}$$

$$Min - Max = \frac{X(j) - min_A}{max_A - min_A}(max'_A - min'_A) + min'_A \tag{4.3}$$

A abordagem ESPA tem que lidar com a possibilidade de existirem sinais irrelevantes. Isso pode ocorrer quando algum sinal não ajuda no processo de predição de ataques DDoS. Além disso, sinais irrelevantes consomem recursos desnecessariamente. A abordagem ESPA pode remover os sinais onde valores são majoritariamente iguais, apresentando assim baixa variação. Por exemplo, caso algum sinal apresente 90% das observações passadas contendo o mesmo valor, a abordagem pode não utilizar esse sinal para predizer ataques. A abordagem ESPA pode

selecionar os melhores sinais dentre todo o conjunto. A escolha é realizada baseada no cálculo estatístico denominado qui-quadrado. Assim, o qui-quadrado é calculado para todos os atributos da base. O conjunto dos sinais selecionado corresponde a metade (50%) dos sinais com maior qui-quadrado. O limite de 50% pode ser previamente alterado. A ação padrão da abordagem ESPA é não realizar a seleção de sinais. Contudo, os Caso de uso 3 e 4 (Capítulo 5) apresentam a abordagem ESPA equipada com uma técnica de seleção de características (sinais).

4.2.4 Predizer Ataques DDoS

Na Etapa 4 (Figura 4.2), a central de inteligência realizará a ação mais importante de toda a proposta, a predição do ataque DDoS. A Figura 4.8 exemplifica a predição dos ataques DDoS. O quadro A apresenta uma transição crítica do tipo *fold* (Seção 2.5). O quadro B representa os indicadores estatísticos em um momento longe do ataque DDoS (transição crítica). Os resultados obtidos no quadro B determinam o perfil dos dados no dado momento. Por exemplo, os resultados poderiam indicar que os dados observados apresentam características de uma distribuição plana. No momento retratado no quadro C, os indicadores estatísticos determinam novamente o perfil dos dados. A diferença entre os quadros B e C é o momento em que a medição foi realizada, pois o quadro C está em um momento mais próximo ao ataque DDoS (transição crítica). Os novos valores dos indicadores estatísticos indicam que a distribuição dos dados mudou. O novo resultado indica uma distribuição dos dados mais pontiaguda do que plana. Essa variação pode ser um indicativo de que, no futuro, uma transição crítica acontecerá. Este trabalho relaciona a ocorrência de transições críticas com a ocorrência de ataques DDoS. Assim, ao antecipar uma transição crítica, a abordagem ESPA consegue predizer ataques DDoS.

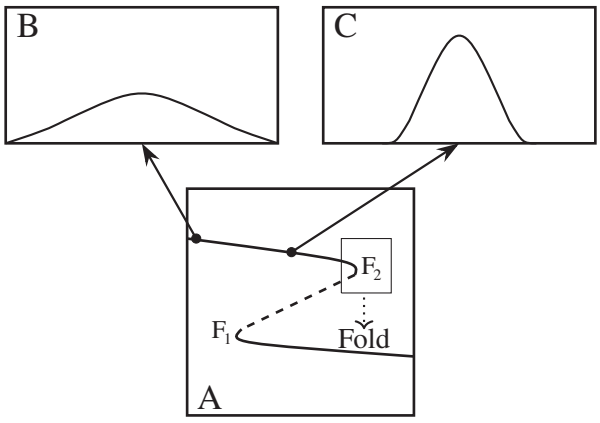


Figura 4.8: Representação da Aproximação de Uma Transição Crítica (adaptado de Guttal e Jayaprakash (2008))

Para automatizar a predição dos ataques DDoS (linha 15 do Algoritmo 1) a abordagem ESPA utiliza algoritmos de aprendizado de máquina (Seção 2.4). A abordagem proposta foi projetada para operar com diferentes tipos de algoritmos de aprendizado de máquina. Os algoritmos de aprendizado supervisionado tendem a apresentar resultados mais acurados. Contudo, a dependência de dados rotulados pode inviabilizar o uso desse tipo de aprendizado. Além disso, a solução pode ficar especializada e identificar somente a preparação dos ataques similares aos treinados. Para diminuir a dependência de dados rotulados, a abordagem ESPA pode utilizar o aprendizado de máquina não supervisionado. Ao não usar rótulos é possível que a abordagem ESPA apresente resultados menos acurados. Contudo, não depender de dados rotulados facilita o uso em ambientes reais. A central de inteligência usa os algoritmos de aprendizado de máquina para consumir as novas características geradas a partir da engenharia de

sinais. A cada ciclo de captura, a central de inteligência visa identificar variações antecessoras aos ataques DDoS (Figura 4.8). Caso encontre um sinal da preparação dos ataques, a central de inteligência gera um aviso de ataque. Caso não encontre, a central de inteligência não gera avisos.

4.2.5 Notificar os Administradores

Na Etapa 5 (Figura 4.2), a abordagem ESPA utiliza a saída da central de inteligência para emitir notificações sobre a ocorrência de possíveis ataques DDoS. A abordagem ESPA envia mensagens para os administradores quando a saída da central de inteligência identificar sinais da preparação dos ataques DDoS (linhas 17 a 19 do Algoritmo 1). Assim, a abordagem notifica os administradores de rede para conduzirem as medidas cabíveis. Existem algumas opções para essa notificação, como o envio de mensagens de texto via celular ou notificações do tipo *push*. Uma opção para automatizar a resposta a incidentes é enviar uma mensagem JSON (do inglês, *JavaScript Object Notation*) contendo os dados da ocorrência. Ela serve como uma entrada para automatizar um firewall ou sistema de alerta. A ação padrão da abordagem ESPA é notificar os administradores de rede por e-mail contendo os dados da notificação.

4.3 SELEÇÃO AUTÔNOMA DO APRENDIZADO DE MÁQUINA NÃO SUPERVISIONADO

A abordagem ESPA foi projetada para adaptar-se a diferentes casos de uso. Uma dificuldade comum em soluções baseadas em aprendizado de máquina é a seleção e configuração dos algoritmos de aprendizado de máquina ideal para cada contexto. Para tornar a abordagem ESPA mais adaptável e simplificar a adoção da abordagem proposta, este trabalho projetou-a para utilizar o AutoML. Assim, é possível utilizar soluções que implementam o AutoML em conjunto com a abordagem ESPA.

Como citado na Subseção 2.4.3, existem diferentes *frameworks* de AutoML. O AutoML tem sido explorado como uma ferramenta para a automatização da seleção e configuração de algoritmos de aprendizado de máquina supervisionados. Contudo, selecionar autonomamente algoritmos de aprendizado de máquina não supervisionados é um grande desafio. Isso ocorre, pois, em geral, usa-se o aprendizado de máquina não supervisionado quando os rótulos reais não estão disponíveis. Assim, identificar qual é o algoritmo adequado não é uma tarefa trivial.

Tendo em vista a dificuldade em selecionar autonomamente algoritmos do aprendizado de máquina não supervisionado sem o uso de dados rotulados, este trabalho apresenta o Algoritmo 3. Este algoritmo apresenta o pseudocódigo de uma estratégia para seleção autônoma sem a dependência de dados rotulados. É importante ressaltar que a utilização do Algoritmo 3 ou de *frameworks* AutoML é opcional. Portanto, o Capítulo 5 apresenta a avaliação da abordagem ESPA com e sem a utilização do AutoML. O Algoritmo 3 utiliza os índices de silhueta, Calinski–Harabasz, Davies–Bouldin, S_Dbw e o CDbw para identificar o algoritmo de aprendizado de máquina ideal para o conjunto de dados que está sendo analisado. A seguir, este trabalho detalha o algoritmo linha a linha.

A primeira decisão que o Algoritmo 3 toma é definir como usar os índices de avaliação do aprendizado de máquina. Na linha 1, o Algoritmo 3 define como padrão que a métrica utilizada para indicar qual é o algoritmo ideal é a média ($metrica\_avaliacao\_cluster \leftarrow$ "avg") dos índices de silhueta, Calinski–Harabasz, Davies–Bouldin, S_Dbw e o CDbw (Subseção 2.4.2). Assim, o algoritmo que, em média, maximizar os cinco índices será identificado como ideal e será sugerido para o usuário. Contudo, o usuário pode optar por usar qualquer índice independentemente. Para isso, basta que o usuário indique qual dos cinco índices é o desejado

no momento da invocação do algoritmo. Por exemplo, indicando que a métrica de avaliação é o índice CDbw (*metrica_avaliacao_cluster* ← "*cdbw*").

Algoritmo 3 Pseudocódigo da Seleção Autônoma de Algoritmo de Aprendizado de Máquina não Supervisionado

```
1: function AUTO_UNSUPERVISED_LEARNING(dados, metrica_avaliacao_cluster ← "avg", numero_clusters ← 2)
2:     lista_algoritmos ← []
3:     algoritmos ← algoritmos(numero_clusters)
4:     for algoritmo in algoritmos do
5:         pontuacao, sdbw, cdbw, ch, db, ss ← 0
6:         rotulos ← algoritmo.fit_predict(dados)
7:         if metrica_avaliacao_cluster == "avg" then
8:             ch ← calcular_calinski_harabasz(dados, rotulos)
9:             ss ← calcular_silhouette(dados, rotulos)
10:            db ← calcular_davies_bouldin(dados, rotulos)
11:            sdbw ← calcular_sdbw(dados, rotulos)
12:            cdbw ← calcular_cdbw(dados, rotulos)
13:            lista_algoritmos.acrescentar(algoritmo, sdbw, cdbw, ch, db, ss)
14:        else
15:            pontuacao ← computar_metrica_especifica(dados, rotulos, metrica_avaliacao_cluster)
16:            lista_algoritmos.acrescentar(algoritmo, pontuacao)
17:        end if
18:    end for
19:    if metrica_avaliacao_cluster == "avg" then
20:        lista_algoritmos ← escalar_dados_entre_zero_um(lista_algoritmos)
21:        lista_algoritmos ← transformar_resultados(lista_algoritmos)
22:        media_algoritmos ← calular_media(lista_algoritmos)
23:        lista_algoritmos = media_algoritmos.ordernar_decrescente()
24:    else
25:        lista_algoritmos.ordernar_decrescente()
26:    end if
27:    retornar lista_algoritmos
28: end function
```

Outra configuração padrão apresentada na linha 1 do Algoritmo 3 é a quantidade de clusters que cada algoritmo irá apresentar. A versão padrão foi criada usando dois clusters. Pois, o objetivo da proposta deste trabalho é predizer ataques DDoS. Para isso, a abordagem ESPA separa o tráfego normal (primeira classe) do tráfego de preparação do ataque (segunda classe). Contudo, o Algoritmo 3 foi projetado para ser adaptado pelo usuário. Caso o usuário não queira utilizar um número predefinido de clusters, basta que ele invoque o algoritmo informando o parâmetro "numero_clusters"como nenhum. Assim, o algoritmo avaliará algoritmos com até oito clusters e indicará a quantidade ideal de clusters.

A linha 2 do Algoritmo 3 inicia a lista que armazenará e retornará os resultados para o invocador do algoritmo. Na linha 3, a função "algoritmos(numero_clusters)" define o espaço de busca do Algoritmo 3. O espaço de busca compreende os algoritmos de aprendizado de máquina não supervisionado candidatos a serem os escolhidos para serem os ideias para o conjunto de

dados que a abordagem ESPA estiver analisando. Este trabalho usou como base nove algoritmos implementados pela biblioteca Scikit-Learn, sendo eles KMeans, Birch, Bisecting K-Means, Gaussian Mixture, Agglomerative Clustering, DBSCAN, OPTICS, MeanShift e HDBSCAN. Além desses, o algoritmo SOM está disponível na biblioteca PYPI[3] e também compõe a lista de algoritmos candidatos.

O Algoritmo 3 configura os nove algoritmos da biblioteca Scikit-Learn com a versão padrão definida pelos desenvolvedores da biblioteca, exceto pela quantidade de clusters definida pelo usuário da abordagem ESPA. Além dessas configurações, o Algoritmo 3 configura outras 28 variações de configurações alternativas. Por exemplo, na biblioteca Scikit-Learn, o algoritmo K-Means possui um parâmetro chamado *algorithm* que pode assumir os valores *lloyd* ou *elkan*. O valor padrão definido pela biblioteca é *lloyd*. Alterando o valor desse parâmetro é possível que o resultado da clusterização seja diferente. Assim, o algoritmo proposto avalia diferentes configurações alternativas para o parâmetro *algorithm*. A Tabela 4.2 apresenta todas as configurações diferentes utilizadas pela abordagem ESPA para escolher a ideal para cada caso. Por fim, versões futuras da abordagem ESPA podem evoluir o espaço de busca do Algoritmo 3 para incorporar outros algoritmos e outras configurações.

Tabela 4.2: Definição dos Algoritmos Usados Pela Abordagem ESPA

Algoritmo	**Configuração**
1. KMeans	n_clusters=2, random_state=0, algorithm = “lloyd”, tol=0.0001
2. KMeans	n_clusters=2, random_state=0, algorithm = “lloyd”, tol=0.001
3. KMeans	n_clusters=2, random_state=0, algorithm = “lloyd”, tol=0.01
4. KMeans	n_clusters=2, random_state=0, algorithm = “lloyd”, tol=0.1
5. KMeans	n_clusters=2, random_state=0, algorithm = “lloyd”, tol=1
6. KMeans	n_clusters=2, random_state=0, algorithm = “elkan”, tol=0.0001
7. KMeans	n_clusters=2, random_state=0, algorithm = “elkan”, tol=0.001
8. KMeans	n_clusters=2, random_state=0, algorithm = “elkan”, tol=0.01
9. KMeans	n_clusters=2, random_state=0, algorithm = “elkan”, tol=0.1
10. KMeans	n_clusters=2, random_state=0, algorithm = “elkan”, tol=1
11. Birch	n_clusters=2, threshold=0.3
12. Birch	n_clusters=2, threshold=0.4

[3]https://pypi.org/project/sklearn-som/

13. Birch	n_clusters=2, threshold=0.5
14. Birch	n_clusters=2, threshold=0.6
15. Birch	n_clusters=2, threshold=0.7
16. Bisecting K-Means	n_clusters=2, init="random", algorithm = "lloyd", bisecting_strategy = "biggest_inertia", random_state=0
17. Bisecting K-Means	n_clusters=2, init="k-means++", algorithm = "lloyd", bisecting_strategy = "biggest_inertia", random_state=0
18. Bisecting K-Means	n_clusters=2, init="k-means++", algorithm = "elkan", bisecting_strategy = "biggest_inertia', random_state=0
19. Bisecting K-Means	n_clusters=2, init="k-means++", algorithm = "lloyd", bisecting_strategy = "largest_cluster", random_state=0
20. Bisecting K-Means	n_clusters=2, init="random", algorithm = "lloyd", bisecting_strategy = "largest_cluster", random_state=0
21. Gaussian Mixture	n_components=2, covariance_type="full", init_params="kmeans", random_state=0
22. Gaussian Mixture	n_components=2, covariance_type="tied", init_params="kmeans", random_state=0
23. Gaussian Mixture	n_components=2, covariance_type="diag", init_params="kmeans", random_state=0
24. Gaussian Mixture	n_components=2, covariance_type="spherical", init_params="kmeans", random_state=0
25. Gaussian Mixture	n_components=2, covariance_type="full", init_params="k-means++", random_state=0
26. Gaussian Mixture	n_components=2, covariance_type="full", init_params="random", random_state=0
27. Gaussian Mixture	n_components=2, covariance_type="full", init_params="random_from_data", random_state=0
28. Agglomerative Clustering	n_clusters=2, linkage="ward"
29. Agglomerative Clustering	n_clusters=2, linkage="average",
30. Agglomerative Clustering	n_clusters=2, linkage="complete",
31. Agglomerative Clustering	n_clusters=2, linkage="single",
32. SOM	m=2, n=1, dim=dimensão padrão dos dados, random_state = 0
33. DBSCAN	Configuração padrão
34. OPTICS	Configuração padrão
35. HDBSCAN	min_cluster_size=2, cluster_selection_method="eom"
36. HDBSCAN	min_cluster_size=2, cluster_selection_method="leaf"
37. MeanShift	Configuração padrão

Entre as linhas 4 e 18, o algoritmo proposto avalia todas as configurações de algoritmos de aprendizado de máquina não supervisionado em busca do candidato ideal para o conjunto de dados. Primeiramente o Algoritmo 3 inicia algumas variáveis auxiliares (linha 5). Na linha 6 o algoritmo realiza a clusterização dos dados adicionados pelo usuário e a variável rótulo receberá o resultado da clusterização. Se o algoritmo proposto executar a versão padrão onde o índice de avaliação de cluster é a média dos resultados de todos os índices, o algoritmo executará as instruções das linhas 8 a 13. Assim, o algoritmo calcula os valores para todos os cinco índices (linhas 8 a 12 do Algoritmo 3). Na linha 13, o Algoritmo 3 armazena temporariamente os resultados na variável "lista_algoritmos". Caso o usuário não utilize a versão padrão e informe um índice específico para avaliar a clusterização, o algoritmo vai direto para a linha 15. Nessa linha, o algoritmo calcula o resultado da clusterização do algoritmo de aprendizado de máquina não supervisionado avaliado para o índice específico e armazena os resultados na linha 16. É importante ressaltar que todo esse processo é realizado para todas as configurações dos algoritmos de aprendizado de máquina não supervisionado descritas na Tabela 4.2.

Após calcular os índices para todos os algoritmos, o Algoritmo 3 verifica novamente se o índice de avaliação de cluster é baseado na média de todos os índices (linha 19). Isso é necessário, pois antes de obter a média dos índices é necessário lidar com as diferentes escalas dos resultados. Como nem todos os índices possuem limite superior (Subseção 2.4.2) o algoritmo redimensiona os dados entre 0 e 1 (linha 20 do Algoritmo 3). Assim, nenhum índice impacta mais do que outro. Isso é importante para que todos os índices influenciem no resultado e apresentando um resultado justo e equilibrado.

Além da escala, a interpretação dos índices Davies–Bouldin e o S_Dbw indicam que os melhores resultados das clusterizações são os que minimizam os valores obtidos (Subseção 2.4.2). Portanto, fazer a média desses valores beneficiaria os algoritmos de aprendizado de máquina não supervisionado que não obtiveram bons clusters segundo esses índices. Assim, a finalidade da linha 21 é transformar proporcionalmente os valores desses índices aplicando a Fórmula 4.4. Seguindo a Fórmula 4.4 o algoritmo decrementa de 1 o valor dos índices Davies–Bouldin e o S_Dbw representados por γ. Por exemplo, supondo que duas instâncias de resultados de qualquer um dos dois índices sejam 0,3 e 0,6. Seguindo a interpretação desses índices (Davies–Bouldin e o S_Dbw), a clusterização que obteve 0,3 é melhor do que a que possuiu 0,6 com 0,4 de diferença entre elas. Ao aplicar a Fórmula 4.4, o valor de 0,3 torna-se 0,7 $(1 - 0,3)$ e o valor de 0,6 torna-se 0,4 $(1 - 0,6)$. Agora, o Algoritmo 3 pode utilizar os maiores valores como sinônimos de melhores. Além disso, a diferença entre o melhor e o pior foi mantida em 0,4. Portanto, a transformação não prejudica nem beneficia os índices. Por fim, é importante ressaltar que a aplicação da Fórmula 4.4 só é possível, pois o Algoritmo 3 lidou com as diferentes escalas redimensionando os dados entre 0 e 1.

$$Transformar\ Davies - Bouldin\ e\ S\_Dbw = 1 - \gamma \tag{4.4}$$

Na linha 22, o Algoritmo 3 calcula média para todos os índices já preparados (índice de silhueta, índice de Calinski–Harabasz, índice de Davies–Bouldin, S_Dbw e CDBW). Na sequência, o algoritmo ordena de forma descrescente as médias dos índices de cada algoritmo. O mesmo processo é realizado para quando apenas um índice foi definido (linha 25). Assim, os primeiros resultados retornados para o usuário (linha 27) são os algoritmos que maximizam o critério de seleção (média dos índices ou índice específico). Caso dois ou mais algoritmos de aprendizado de máquina não supervisionado apresentem a mesma clusterização, o resultado para os índices será o mesmo. A versão atual do Algoritmo 3 apresenta todas as opções e deixa

que o usuário defina qual é usar. Por fim, trabalhos futuros irão avaliar o tempo de execução, a complexidade dos algoritmos ou outras características que possam ser utilizadas para desempatar a escolha pelo algoritmo de aprendizado de máquina não supervisionado ideal para o contexto.

4.4 COMPLEXIDADE DA ABORDAGEM ESPA

Este trabalho propôs a abordagem ESPA apresentada na Seção 4.2 e avaliada no Capítulo 5. A abordagem ESPA está pautada na possibilidade de identificação dos sinais da preparação dos ataques usando a teoria dos sinais precoces de alerta. Assim, a abordagem ESPA pode lidar com o problema de pesquisa e predizer os ataques DDoS. Com o intuito de complementar a apresentação da proposta, esta seção apresenta a complexidade da engenharia de sinais realizada pela abordagem ESPA. O objetivo da análise da complexidade dos algoritmos é identificar o total de recursos que os algoritmos necessitam. Largura de banda, consumo de memória e tempo são recursos que geralmente são alvos da avaliação da complexidade dos algoritmos. Assim, ao comparar diferentes algoritmos que realizam a mesma tarefa é possível identificar o algoritmo que consome menos recurso (Cormen et al., 2012).

Assim como em Cormen et al. (2012), a análise do tempo de execução do algoritmo é o foco desta seção. Cormen et al. (2012) definem que o tempo de execução de um algoritmo está relacionado com a quantidade de operações/passos que o algoritmo executa referente a entrada do algoritmo. Como cada instrução (linha de código) necessita de um tempo para executar, é necessário somar os tempos de execução de todas as instruções executadas. Apesar de ser possível definir o tempo exato de execução de diferentes algoritmos, nem sempre essa é uma tarefa trivial. Segundo Cormen et al. (2012), existem casos em que o esforço para definir o tempo exato de execução de um algoritmo não apresenta benefícios concretos para justificar o esforço. Para os casos em que o tamanho da entrada é consideravelmente grande, a ordem de crescimento apresentada pelo tempo de execução é o suficiente para realizar a análise do algoritmo. Esse processo é chamado de análise da complexidade assintótica dos algoritmos. Por fim, a análise assintótica descreve o tempo de execução dos algoritmos em termos de seu comportamento, focando no tempo de execução no pior caso. O pior caso, ou limite superior, é dado pela notação O (Cormen et al., 2012).

A análise da complexidade concentra-se no Algoritmo 2 que define a engenharia de sinais. Esse algoritmo foi o escolhido, pois a engenharia de sinais precoces de alerta é a espinha dorsal da abordagem ESPA. Análise assintótica do pseudocódigo da engenharia de sinais foi realizada considerando as implementações das bibliotecas Pandas[4] e Ewstools[5]. Este trabalho considerou a implementação dessas bibliotecas, pois o Algoritmo 2 define que é necessário realizar os cálculos relacionados ao *skewness*, ao *kurtosis* entre outros. Contudo, o Algoritmo 2 não apresenta como esses cálculos são feitos. Portanto, as bibliotecas citadas como referência auxiliaram no processo de mensurar a complexidade do Algoritmo 2.

A Tabela 4.3 apresenta a análise da complexidade do Algoritmo 2. A primeira coluna da tabela apresenta as instruções do Algoritmo 2 e a segunda coluna (coluna custo) apresenta o valor da quantidade de operações/passos para cada instrução no pior caso. A Linha 1 possui custo constante O (1) por ser a definição da função. A Linha 2 instancia uma estrutura de dados também em tempo constante (O (1)). A Linha 3 indica que o algoritmo irá executar algum processo repetidamente para os todos os atributos do tráfego de rede coletados por todos os agentes a cada ciclo de coleta. A variável $coleta\_trfego\_rede$ armazena os dados coletados na forma de uma matriz, onde as colunas representam atributos do tráfego de rede e as linhas os

[4] https://pandas.pydata.org/

[5] https://github.com/ThomasMBury/ewstools

dados coletados. Como a abordagem ESPA pode ter vários atributos do tráfego de rede para cada agente, a quantidade de colunas na matriz $coleta\_trfego\_rede$ é igual $a.f$. Portanto, a instrução da Linha 3 será executada O($a.f$) vezes, onde a é a quantidade de agentes e f é a quantidade de atributos coletados por cada agente. Para simplificar a Tabela 4.3, o termo O($a.f$) foi substituído por O(m^2).

A linha quatro simplesmente inicia uma nova variável com os dados do atributo coletado pelo agente. Assim, o custo dessa instrução é O(1). Nas bibliotecas Pandas e Ewstools, as funções *skewness*, *kurtosis*, *cv*, s_{max}, s_{fold} e s_{hopf} podem iterar a lista completa dos dados. Ou seja, as Linhas 5 a 10 analisam cada uma das colunas da matriz $coleta\_trfego\_rede$. Assim, no pior caso, o custo para as Linhas entre 5 e 10 é O(d). Onde d é o tamanho da janela deslizante (linha 4 do Algoritmo 1). As linhas 11 a 14 executam apenas instruções cujo custo é linear O(1). No pior caso, o laço de repetição executa m^2 vezes instruções que demandam até O(d), pois as Linhas 5 a 10 estão em um laço de repetição. Portanto, a complexidade assintótica do Algoritmo 2 é O($m^2.d$) ou O(($a.f$).d). Visando generalizar a complexidade assintótica do Algoritmo 2, a complexidade pode ser escrita como O(n^3), onde n representa a quantidade de linhas e colunas da variável $coleta\_trfego\_rede$.

Tabela 4.3: Análise da Complexidade Assintótica do Pseudocódigo da Engenharia de Sinais (Algoritmo 2)

Instruções	**Custo**
1. function engenharia_de_sinais(coleta_tráfego_rede)	O(1)
2. sinais ← inicia_estrutura_dados()	O(1)
3. for atributo em coleta_tráfego_rede **do**	O(m^2)
4. dados ← atributo.dados()	O(1)
5. skew ← skewness(dados)	O(d)
6. kurt ← kurtosis(dados)	O(d)
7. cv ← cv(dados)	O(d)
8. s_{max} ← s_{max}(dados)	O(d)
9. s_{fold} ← s_{fold}(dados)	O(d)
10. s_{hopf} ← s_{hopf}(dados)	O(d)
11. sinais.acrescentar(skew, kurt, cv, s_{max}, s_{fold}, s_{hopf})	O(1)
12. end for	O(1)
13. return sinais	O(1)
14. end function	O(1)

4.5 RESUMO

Este capítulo apresentou a proposta de uma abordagem para a predição de ataques DDoS. A abordagem ESPA coleta o tráfego de rede de modo distribuído ou centralizado por meio dos agentes, os quais são instâncias de software. Os agentes cooperam para a predição de ataques DDoS enviando o tráfego de rede coletados para outra instância de software, a central de inteligência. Utilizando indicadores estatísticos (Seção 2.5), a central de inteligência transforma o tráfego da rede em sinais precoces de alerta capazes de antecipar as transições críticas. Essa ação de transformação foi definida neste trabalho como a engenharia de sinais. A abordagem ESPA compreende as transições críticas como possíveis ataques DDoS. Assim, ao identificar sinais capazes de antecipar as transições críticas, este trabalho identifica os sinais da preparação dos ataques DDoS e consegue predizê-los. A central de inteligência utiliza o aprendizado de

máquina para automatizar a identificação dos sinais da preparação de ataques DDoS antes que o ataque seja lançado. A abordagem ESPA utiliza a saída da central de inteligência para notificar os administradores sobre a possibilidade de um ataque vindouro. Por fim, a abordagem ESPA foi projetada para ser adaptável pelo administrador de rede. Assim, a abordagem ESPA proporciona a personalização de configurações que irão auxiliar o administrador de rede e as equipes de segurança a obter o melhor resultado.

5 PREDIÇÃO DE ATAQUES DDOS PELA ABORDAGEM ESPA

Este capítulo apresenta os resultados da predição dos ataques DDoS usando a abordagem proposta. O objetivo geral é verificar se a abordagem proposta consegue predizer ataques DDoS nos contextos analisados, validando ou invalidando a hipótese definida neste trabalho. Como a abordagem ESPA foi projetada para operar em diferentes cenários com diferentes configurações, este trabalho avalia a abordagem em quatro casos de uso diferentes. Onde, cada caso de uso possui uma configuração diferente da abordagem ESPA. O capítulo está dividido da seguinte forma. A Seção 5.1 apresenta as bases de dados utilizadas nos estudos de caso. As Seções 5.2, 5.3, 5.4 e 5.5 detalham os quatro casos de uso da abordagem ESPA. A Seção 5.6 apresenta as vantagens e a Seção 5.7 apresenta as limitações da abordagem ESPA. Por fim, a Seção 5.8 resume o capítulo.

5.1 BASES DE DADOS AVALIADAS

Esta seção apresenta os detalhes relativos às bases de dados utilizadas ao longo do capítulo. Primeiramente, na Subseção 5.1.1, este trabalho apresenta as premissas necessárias para que a base de dados pudesse ser usada para a avaliação da abordagem ESPA. Isso acontece, pois nem todas as bases de dados presentes na literatura possuem os requisitos para que a predição dos ataques DDoS possam ser avaliadas. A Subseção 5.1.2 apresenta detalhes sobre as bases utilizadas nos casos de uso. O uso das bases de dados variou entre os casos de uso. Isso ocorre, pois cada caso de uso tem propriedades únicas. Portanto, para obter o melhor resultado e seguir os objetivos específicos de cada caso de uso foi necessário variar a utilização das bases de dados apresentadas nesta seção.

5.1.1 Premissas das Bases de Dados e da Avaliação de Desempenho

Em conformidade com a hipótese, este trabalho assumiu premissas comuns para a seleção de bases de dados e, consequentemente, para a avaliação do desempenho. Primeiramente, o ideal é que a base de dados analisada possua a documentação do momento em que ocorre o lançamento do ataque DDoS ou do início da infecção/execução dos *bots*. Essa informação é utilizada para identificar se a abordagem ESPA conseguiu identificar sinais da preparação dos ataques antes do lançamento do ataque e verificar quanto tempo antes do ataque a predição do ataque ocorreu. Caso o lançamento do ataque não esteja disponível, este trabalho usa o tempo após a infecção/execução dos *bots* para identificar quanto tempo a abordagem proposta necessitou para identificar os sinais da preparação dos ataques. A segunda premissa indica que a base de dados não pode começar diretamente na etapa do lançamento do ataque. Essa premissa é necessária, pois a abordagem ESPA identifica sinais que ocorrem antes do lançamento do ataque. Assim, dados anteriores ao ataque devem estar disponíveis.

A terceira premissa define que o cenário analisado não tenha dois ataques ocorrendo ao mesmo tempo, pois, é importante verificar se a abordagem consegue predizer ataques DDoS sem interferências de outros ataques. A quarta premissa é a existência de apenas dois tipos de tráfego, o normal e o relacionado ao ataque DDoS. Essa restrição é importante por delimitar o escopo da avaliação. Assim, nas avaliações devem existir apenas dois tipos de tráfego, o tráfego dos usuários normais, que não apresentam perigo para a segurança dos serviços. E o tráfego originado pelas *botnets* comandadas pelos atacantes. Também é importante que a documentação

das bases de dados rotulem os *bots*. Pois, através do rótulo dos *bots*, pode-se verificar se a abordagem ESPA identificou sinais da preparação do ataque DDoS antes do seu lançamento.

5.1.2 Características das Bases de Dados Avaliadas

A primeira base de dados utilizada foi desenvolvida pela Universidade Técnica Tcheca (do inglês, *Czech Technical University* - CTU) denominada CTU-13 *dataset* (Garcia et al., 2014). Essa base de dados possui 13 cenários de ações conduzidas por *botnets* combinados com dados reais da rede local da universidade. Entre as ações é oportuno destacar a comunicação entre *bots* utilizando o IRC, o envio de *spam* e o ataque DDoS. Uma importante propriedade da base de dados CTU-13 é a documentação. Todos os cenários possuem uma página web onde estão documentados os detalhes das ações dos pesquisadores, inclusive o período da infecção dos *bots* e o momento do lançamento dos ataques DDoS.

Os cenários 10 e 11, respectivamente, as capturas 51 e 52 foram frequentemente usadas nos casos de uso devido a suas propriedades que se complementam. As capturas são subsequentes, ou seja, o tráfego de rede capturado na captura 51 (cenário 10) antecedeu o tráfego de rede coletado na captura 52 (cenário 11). A captura 51 possui 8803 segundos de tráfego de dados, com dez *bots* que enviam pacotes antes ou depois do lançamento do ataque. O arquivo com a coleta do tráfego de rede possui 41 GB considerando apenas os cabeçalhos dos pacotes. Isso porque os pesquisadores não disponibilizaram a carga útil (do inglês, *payload*) dos pacotes. Ao todo foram capturados 46.997.342 pacotes. Os ataques realizados na captura 51 foram de inundação de ICMP e UDP e os *bots* estavam na rede local da vítima. A documentação indica que a infecção iniciou-se no segundo 2643 e o ataque foi lançado no segundo 5632.

A captura 52 da base de dados CTU-13 é nove vezes menor em termos de tempo de captura comparada com a captura 51, possuindo 972 segundos. Esta captura possui três *bots* que trafegam dados antes ou depois do lançamento do ataque. O arquivo com a coleta do tráfego de rede possui 555 MB, 6.336.398 pacotes e um ataque do tipo inundação de ICMP. A documentação indica que a infecção iniciou-se no segundo 527 e o ataque foi lançado no segundo 778. Ao utilizar as capturas 51 e 52 é possível avaliar a abordagem proposta frente a cenários com propriedades diferentes e desafiadoras.

Outra base de dados usada nos casos de uso é a *DDoS evaluation dataset* (CICDDoS2019) (Sharafaldin et al., 2019). Essa base de dados possui 19 cenários de ataques DDoS onde a rede vítima está conectada ao atacante pela Internet. Todos os cenários possuem o momento de início e de fim de cada ataque. Os ataques foram conduzidos no decorrer de dois dias. Dentre os 19 cenários de ataques, este trabalho utilizou o primeiro ataque DDoS realizado no primeiro dia da captura e os cenários nomeados de “Ataque UDP” e “Ataque UDP-Lag”, também executado no primeiro dia de captura. O primeiro ataque possui mais de 61 milhões de pacotes completos (cabeçalho e carga útil) e o arquivo correspondente a este ataque tem mais de 27 GB de dados referente ao tráfego malicioso e tráfego normal. Esta base de dados não indica o momento da infecção dos *bots*, mas indica o lançamento do ataque. O primeiro ataque foi lançado no segundo 1484 da captura e durou 540 segundos. Este ataque é do tipo *Portmap* que explora as portas UDP ou TCP utilizadas pelo serviço *Portmap*. Por fim, este cenário possui 2024 segundos.

Os cenários nomeados de “Ataque UDP” e “Ataque UDP-Lag” da base de dados CICDDoS2019 são subsequentes. Ou seja, no primeiro dia de ataque, o tráfego de rede capturado no cenário “Ataque UDP” antecedeu o tráfego de rede coletado no cenário “Ataque UDP-Lag”. O cenário “Ataque UDP” possui 1277 segundos de tráfego e 14.631.228 pacotes. O cenário “Ataque UDP” possui 1260 segundos de tráfego e 55.204 pacotes. Novamente, esta base de dados

não indica o início da infecção dos *bots*, mas o ataque DDoS foi lançado no segundo 600 para ambos os cenários “Ataque UDP” e “Ataque UDP-Lag”.

A terceira base de dados utilizada no Caso de uso 3 é a IoT-23 (Garcia et al., 2020). Esta base de dados possui 23 cenários de ataques DDoS em ambientes IoT. O cenário 17 contém mais de um *bot* infectado e ativo. O cenário possui 8,3 GB e 109 milhões de pacotes enviados em 24 horas. Os pesquisadores iniciaram a captura às 06:43:20, e a execução do *malware* foi às 11:43:43 do mesmo dia. Assim, a captura de tráfego pré-infecção possui tráfego legítimo e o tráfego pós-infecção contém vestígios da preparação do ataque. A documentação apresentou um problema elétrico na universidade. Portanto, este trabalho não identificou a iniciação efetiva de ataques DDoS. Este trabalho levanta a hipótese de que a falta de energia elétrica comprometeu o lançamento do ataque. Assim, não é possível quantificar quanto tempo antes do lançamento do ataque a predição ocorreu. Portanto, este trabalho avalia quanto tempo depois da execução do *malware* a proposta identificou a preparação do ataque. Os detalhes da utilização dessa base de dados estão no Caso de uso 3 (Seção 5.4).

5.2 CASO DE USO 1 - AM SUPERVISINADO E DISTRIBUÍDO

No caso de uso 1, a abordagem ESPA opera como apresentado na Figura 5.1. A abordagem utiliza agentes que são programas capazes de coletar o tráfego da rede de modo distribuído. Os agentes cooperam enviando o tráfego de rede coletado para a central de inteligência. A central de inteligência realiza a engenharia de sinais criando novas características utilizando todos os dados coletados. O aprendizado de máquina é usado para identificar os sinais da preparação de ataques DDoS antes que o ataque seja lançado. A abordagem ESPA utiliza a saída da central de inteligência para notificar os administradores sobre a possibilidade de um ataque vindouro. A Figura 5.1 apresenta as cinco etapas que a abordagem utiliza para realizar a predição dos ataques DDoS. Essas etapas foram apresentadas no capítulo anterior e especializadas para a condução dos experimentos deste caso de uso.

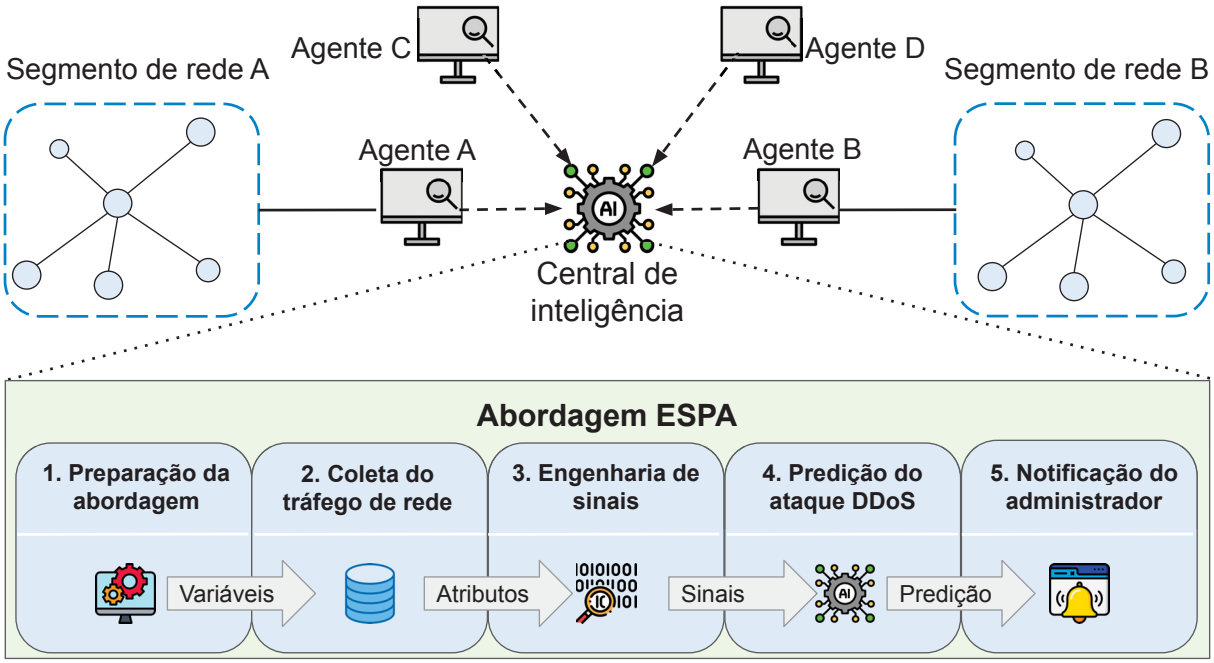


Figura 5.1: Modo de Operação da Abordagem ESPA no Caso de Uso 1

Este primeiro caso de uso possui quatro experimentos. O **objetivo geral dos experimentos é corroborar a hipótese e verificar se a abordagem ESPA consegue identificar sinais que demonstram a preparação de ataques DDoS**. Para isso, é necessário que a abordagem ESPA identifique algum sinal de preparação do ataque. Caso a hipótese seja corroborada com os Experimentos 1 e 2, **o objetivo específico dos Experimentos 3 e 4 é verificar se algoritmos de *deep learning* supervisionados conseguem melhorar os resultados obtidos nos Experimentos 1 e 2**. Para isso é necessário que a abordagem ESPA equipada com algoritmos de *deep learning* possa diminuir a quantidade de erros em comparação com os Experimentos 1 e 2 ou que os tempos de predição dos ataques nos Experimentos 3 e 4 sejam maiores quando comparados com os experimentos anteriores. Por fim, todos os experimentos deste caso de uso foram conduzidos em um dispositivo que possui um processador Intel Core I5, um disco rígido de um terabyte, oito gigabytes de memória, com um sistema operacional Linux Mint e com Python 3.10.12.

5.2.1 Definição dos Experimentos do Caso de Uso 1

Para conduzir os experimentos é necessário preparar a abordagem ESPA definindo os parâmetros apresentados na Subseção 4.2.1. A Tabela 5.1 apresenta os parâmetros usados pela abordagem ESPA no Caso de uso 1. A motivação da escolha e a apresentação destes parâmetros estão descritas ao longo desta subseção. Por fim, todos os resultados de todos os casos de uso descritos neste capítulo estão disponíveis online[1].

Tabela 5.1: Definição dos Parâmetros Usados Pela Abordagem ESPA no Caso de Uso 1

Parâmetro	**Valor**
Modo de coleta do tráfego de rede	Distribuído
Quantidade de agentes	Quatro
Localização dos agentes	Na rede vítima
Atributos do tráfego de rede	Total de pacotes trafegados na rede
Ciclos de captura	Um segundo
Indicadores estatísticos	*kurtosis*, *skewness*, S_{max}, S_{null}, S_{fold} e S_{hopf}
Algoritmos de aprendizado de máquina	Adaboost e MLP
Seleção hiperparâmetros dos algoritmos de aprendizado de máquina	Manual
Janela deslizante	20% da base de dados
Tratamento de dados faltantes	Adicionar o valor -1
Redimensionamento dos dados	Min-Max

Neste caso de uso, a abordagem ESPA foi configurada para coletar o tráfego de rede da vítima do ataque DDoS de modo distribuído, assim como representado na Figura 5.1. Para isso, a abordagem ESPA utilizou quatro agentes inseridos na rede da vítima. O nome simbólico de cada agente é Agente 1, Agente 64, Agente 128 e Agente 192. Cada agente ficou responsável por coletar os pacotes originados em um intervalo específico. A Tabela 5.2 apresenta o intervalo de atuação de cada agente. Para exemplificar, o Agente 1 ficou responsável por coletar os pacotes cujo endereço de origem do pacote estava no intervalo de 0.0.0.1 a 63.255.255.254. A definição do intervalo nos moldes apresentados na Tabela 5.2 possui dois objetivos. O primeiro objetivo é garantir que cada agente fique responsável por coletar o tráfego de rede de um segmento exclusivo. Assim, um pacote não será analisado por dois agentes. O segundo motivo é prover imparcialidade

[1] https://github.com/andersonneira/tese_resultados/

na coleta e no processamento do tráfego de rede. Como todos os agentes possuem a mesma proporção de endereços possíveis (1.073.741.824 endereços), este trabalho não adiciona vieses quanto a divisão do tráfego de rede.

Tabela 5.2: Intervalo de Atuação dos Agentes nos Experimentos do Caso de Uso 1

Atributo	Descrição
Agente 1	0.0.0.1 - 63.255.255.254
Agente 64	64.0.0.1 - 127.255.255.254
Agente 128	128.0.0.1 - 191.255.255.254
Agente 192	192.0.0.1 - 255.255.255.254

O atributo do tráfego de rede escolhido para a realização dos experimentos foi a quantidade de pacotes trafegados em um ciclo de um segundo. Este atributo foi escolhido por ser um dos atributos mais relevantes apresentados por Feng et al. (2018). Deste modo, os agentes coletam apenas um atributo do tráfego de rede e os experimentos avaliam se a abordagem funciona com um único atributo. O tamanho do ciclo de captura de um segundo foi escolhido por dois motivos. O primeiro motivo foi para estar conforme a premissa de tempo suficiente antes do ataque. Facilitando assim a escolha por bases de dados, visto que, bases de dados com poucos minutos de tráfego poderiam ser analisadas. O segundo motivo foi estar conforme a escolha feita por Salemi et al. (2021), pois os autores também escolheram a quantidade de pacotes por segundo para avaliar a proposta deles.

Kurtosis, *skewness*, S_{max}, S_{null}, S_{fold} e S_{hopf} são os indicadores estatísticos que a central de inteligência utiliza para realizar a engenharia de sinais e gerar as novas características (Subseção 4.2.3). Este caso de uso utilizou esses indicadores estatísticos por dois motivos. Primeiramente o *kurtosis*, *skewness*, S_{max} podem apresentar variações extremas nos valores observados durante a análise. Essas variações indicam mudanças na distribuição dos dados. Ou seja, inicialmente a distribuição dos dados da abordagem possui um aspecto. Porém, conforme a abordagem observa a aproximação da transição crítica, esse aspecto pode variar (Figura 4.8).

Em segundo lugar, os indicadores estatísticos S_{null}, S_{fold} e S_{hopf} podem indicar qual é o tipo da transição crítica que está se aproximando. O objetivo de escolher esses indicadores é complementar a análise dos dados. Assim, com essas informações, é possível evitar FPs e auxiliar na definição do tipo do ataque vindouro. Por fim, a engenharia de sinais foi realizada com uma janela deslizante de tamanho 20% da base de dados. Esse era o valor padrão para a biblioteca[2] que implementou os cálculos do S_{max}, S_{null}, S_{fold} e S_{hopf}.

Para os Experimentos 1 e 2, este trabalho selecionou o Adaboost como o algoritmo de aprendizado de máquina para processar as novas características geradas pelo processo de engenharia de sinais. O Adaboost é um algoritmo de aprendizado supervisionado, do tipo *shallow learning*, que executa tarefas de classificação e regressão e possui implementação disponível em Python (Pedregosa et al., 2011). A estratégia do Adaboost é treinar diferentes instâncias do mesmo algoritmo de aprendizado de máquina e combinar as saídas de todas elas para gerar uma saída (Freund e Schapire, 1997; Schapire, 2013). Deste modo, cada instância pode lidar com diferentes casos, fazendo com que as várias instâncias de um classificador possam gerar um modelo robusto (Vezhnevets e Vezhnevets, 2005). Apesar de apresentar bons resultados, o Adaboost não é o algoritmo de aprendizado de máquina mais poderoso existente atualmente, bem como não é o mais simples. Assim, o Adaboost é ideal para a execução das avaliações

[2] https://github.com/ThomasMBury/ewstools

preliminares propostas neste caso de uso. Pois o problema de predição não é trivial, porém para avaliar a proposta não é necessário utilizar o algoritmo mais poderoso da literatura.

Os Experimentos 1 e 2 testaram o Adaboost utilizando a configuração padrão definida pelos desenvolvedores da biblioteca, exceto pela quantidade de estimadores. A configuração padrão do Adaboost é utilizar a Árvore de Decisão como estimador base para criar as várias instâncias presentes no modelo. Os dois experimentos utilizaram apenas cinco estimadores, quando o padrão é 50. O objetivo dessa ação é utilizar a mesma configuração para os experimentos.

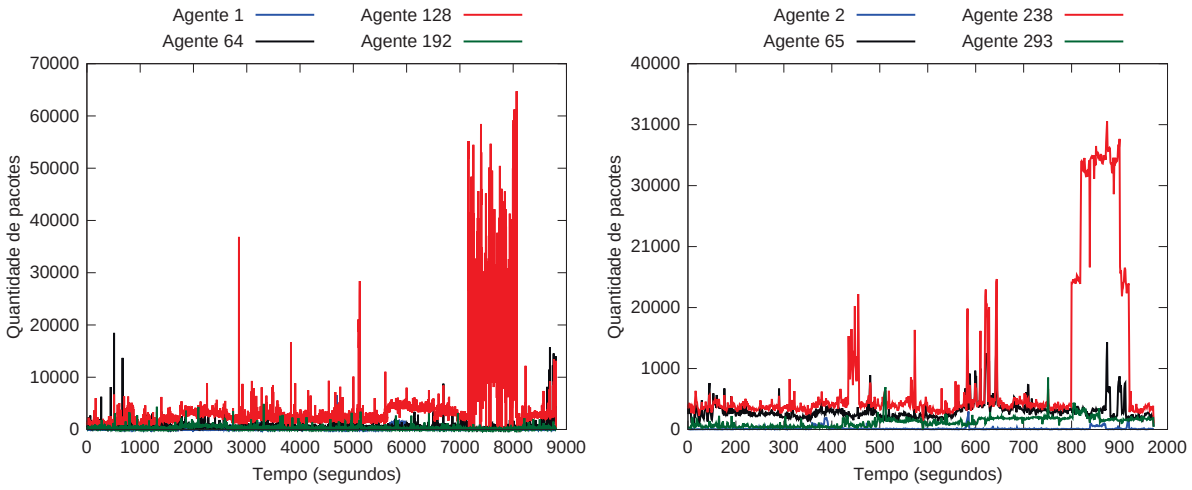
Para os Experimentos 3 e 4 o algoritmo escolhido foi a MLP. A MLP é um algoritmo de aprendizado supervisionado, do tipo *deep learning* (Wei et al., 2021; Sarker, 2021) e possui implementação disponível em Python (Pedregosa et al., 2011). A MLP é um dos primeiros algoritmos de *deep learning* a serem apresentadas na literatura (Suzuki, 2017) e é amplamente utilizada para executar tarefas de classificação e regressão (Ma et al., 2019). A MLP é uma rede neural totalmente conectada que opera com uma camada de entrada, uma ou mais camadas intermediárias, denominadas camadas ocultas e uma camada de saída. A camada de entrada recebe os sinais que serão processados pela MLP. Nas camadas ocultas, os sinais são ponderados até passarem por todas as camadas ocultas. A camada de saída recebe os resultados da última camada oculta e então emite o resultado da operação (Han et al., 2011; Sarker, 2021). Todas as camadas são conectadas por neurônios. Os neurônios conduzem o processo de aprendizagem por meio da computação matemática (Aldweesh et al., 2020). Durante o treinamento, a MLP aprende por meio do ajuste de pesos, sempre com o intuito de prover a saída correta. O algoritmo *Backpropagation* é utilizado para realizar esse ajuste, para isso a cada iteração os resultados reais são comparados com os resultados obtidos pelo modelo, e os pesos das camadas são atualizados no sentido da saída para a entrada (Han et al., 2011). Deste modo, a cada iteração do treinamento o modelo tende a se tornar mais apropriado para o problema. O processo de treinamento pode ser otimizado utilizando *Stochastic Gradient Descent*, *Adaptive Moment Estimation* e o *Limited Memory BFGS* (L-BFGS). Por fim, funções de ativação como a *Rectifed Linear Unit*, *Tanh*, *Sigmoid* e *Softmax* são utilizadas para determinar a saída do modelo (Sarker, 2021).

Os Experimentos 3 e 4 testaram a MLP utilizando as configurações padrão, exceto pela abordagem de otimização e pela quantidade de camadas ocultas. A abordagem de otimização escolhida foi a L-BFGS, pois os autores da biblioteca recomendam usá-la quando a base de dados analisada não possui vários milhares de registros, como é o caso dos experimentos. Os Experimentos 3 e 4 utilizaram três camadas ocultas, 33, 17 e 11 foram a quantidade de neurônios para a primeira, segunda e terceira camadas respectivamente. Por fim, é oportuno ressaltar que as configurações foram obtidas manualmente por meio da avaliação de diferentes combinações.

5.2.2 Experimento 1 do Caso de Uso 1

A primeira ação do Experimento 1 define a utilização das capturas 51 e 52 da base de dados CTU-13. As capturas 51 e 52 possuem ataques DDoS e são subsequentes, por esse motivo, este trabalho utilizou a captura 51 como base de treinamento e a captura 52 com base de teste. Na sequência, este trabalho segmentou o tráfego de rede nos moldes da Tabela 5.2 e contabilizou a quantidade de pacotes por segundo. A Figura 5.2 apresenta o resultado da segmentação e contabilização do tráfego de rede para a base de treinamento e a base de testes. A Figura 5.2(a) mostra que ao longo da base de treinamento o Agente 128 possui os maiores picos na quantidade de pacotes. Esse comportamento pode ser justificado, pois todos os *bots* que realizaram o ataque possuíam endereços IP presentes no intervalo do Agente 128. Contudo, existem momentos em que outros agentes capturaram mais pacotes. No segundo 511 o Agente 64 contabilizou 18.489 pacotes em um segundo. O Agente 1 contabilizou 6599 pacotes representando o pico no segundo 4688. A Figura 5.2(b) apresenta comportamento similar também para o teste. Novamente o

Agente 128 é o que mais possui picos de pacotes. O Agente 64 é o segundo agente com mais picos de pacotes trafegados. Novamente o fato dos *bots* estarem alocados em IPs no intervalo do Agente 128 é a justificativa para o Agente 128 possuir mais picos de pacotes, visto que, na captura original, todos os bots possuíam IPs no intervalo do Agente 128.



(a) Pacotes por segundo no treinamento

(b) Pacotes por segundo no teste

Figura 5.2: Quantidade de Pacotes por Segundo no Experimento 1 do Caso de Uso 1

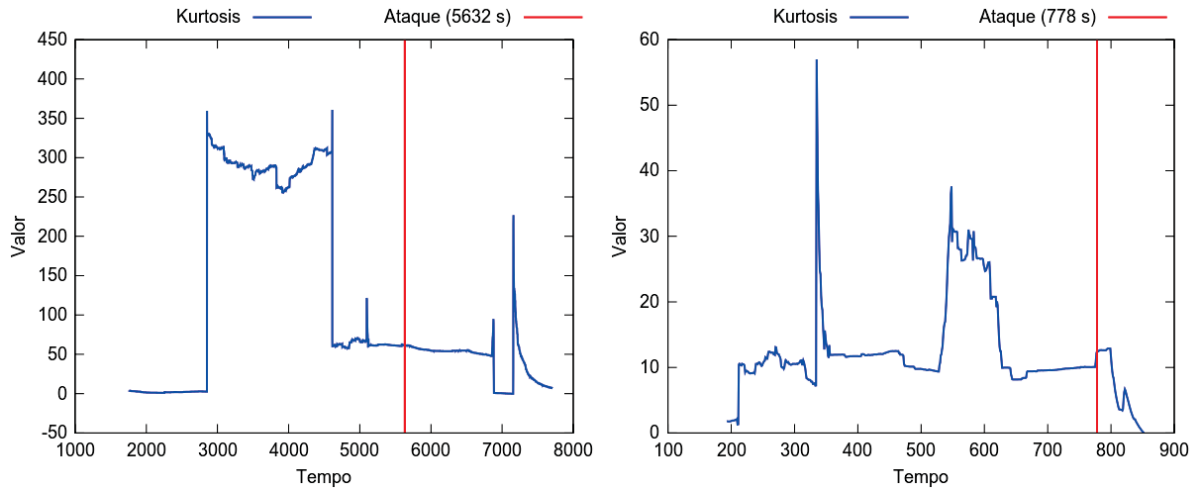
Realizar a engenharia de sinais é a ação seguinte. Deste modo, os agentes enviaram a contagem de pacotes ($f = 1$) para a central de inteligência. A central de inteligência aplica a engenharia de sinais sobre o tráfego de rede coletado para gerar as novas características. Como o tráfego de rede foi coletado por quatro agentes ($a = 4$) e a central de inteligência utilizou seis indicadores estatísticos ($i = 6$) o total de novas características é 24 ($ews = 4 \times 1 \times 6 = 24$ Subseção 4.2.3). Apenas para simplificar a apresentação e a discussão dos resultados, a Figura 5.3 apresenta a variação do *kurtosis* originado pela coleta do Agente 128 nas bases de treinamento e no teste. Contudo, todos os outros sinais estão disponíveis online[3].

A Figura 5.3(a) apresenta a variação do *kurtosis* no treinamento. A linha vermelha indica o segundo onde o ataque foi lançado pelos pesquisadores. A observação do *kurtosis* apresenta três pontos de variação intensa (picos no *kurtosis*) antes do lançamento do ataque. Entre os segundos 2853 e 2854 o valor do *kurtosis* varia de 2,603 para 304,925. Essa variação corresponde a um aumento de 11611,263%. Nos segundos 4613 e 4614 o valor do *kurtosis* vai de 307,954 para 360,741. Por fim, entre os segundos 5097 e 5100 o valor do *kurtosis* vai de 66,568 para 120,578.

A Figura 5.3(b) apresenta a variação do *kurtosis* para a base de teste. Esses dados apresentam três pontos de variação intensa (picos no *kurtosis*) antes do lançamento do ataque representado pela linha vermelha. Entre os segundos 335 e 336 o valor do *kurtosis* varia de 7,247 para 56,989. Essa variação corresponde a um aumento de 686,366%. Nos segundos 541 e 549 o teste apresenta outro pico. A variação nesse segundo pico é de 20,514 para 37,623. Por fim, outra variação que chamou a atenção deste trabalho encontra-se entre os segundos 570 e 576. Nesse intervalo o valor do *kurtosis* varia de 26,440 até 31,020.

Para treinar a central de inteligência, esse trabalho conduziu a rotulação, o tratamento de dados faltantes e o dimensionamento Min-Max nas bases de treinamento e teste. É necessário rotular as bases para ser possível utilizar o aprendizado de máquina supervisionado. O objetivo de rotular a base nos segundos apresentados na Figura 5.4 é ensinar o algoritmo de aprendizado

[3] https://github.com/andersonneira/tese_resultados/



(a) *kurtosis* por segundo no treinamento

(b) *kurtosis* por segundo no teste

Figura 5.3: *Kurtosis* por Segundo no Experimento 1 do Caso de Uso 1 Coletado pelo Agente 128 e Processado Pela Central de Inteligência

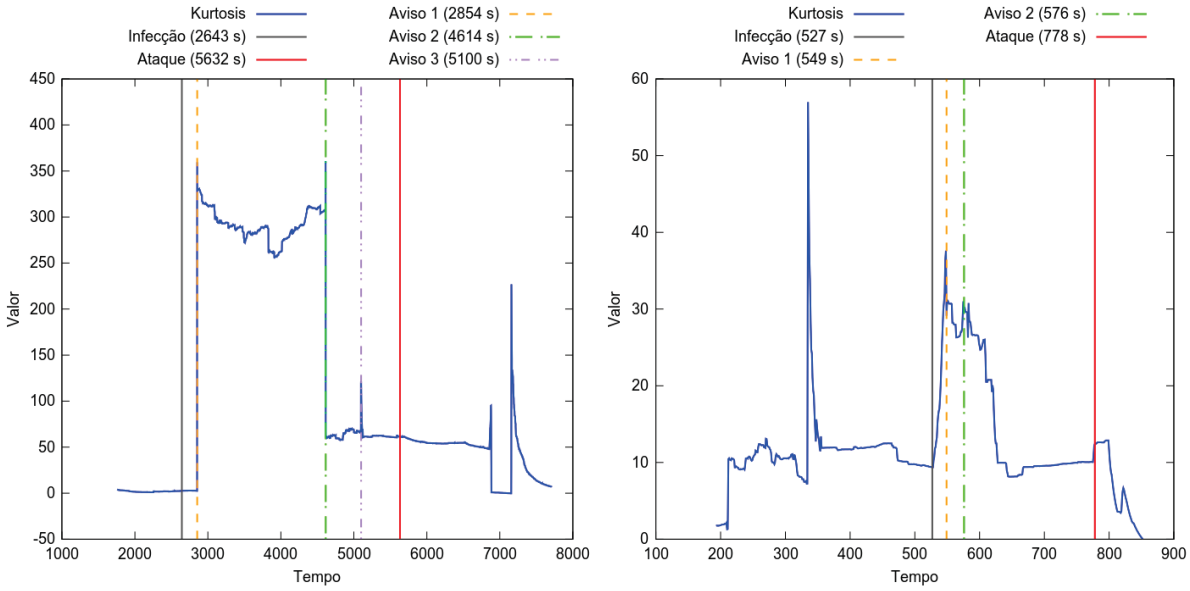
de máquina a identificar valores extremos nas bases de dados. Assim, caso o algoritmo de aprendizado de máquina identifique um possível valor extremo, a abordagem ESPA notifica os administradores de rede. Este trabalho utilizou a identificação dos valores extremos apresentados anteriormente aliados com o momento de infecção dos *bots* e a existência de tráfego malicioso para rotular as bases de dados.

A Figura 5.4 apresenta os segundos rotulados nas bases de treinamento e teste. No caso do treinamento (Figura 5.4(a)), a documentação[4] indica que a infecção dos *bots* ocorreu no segundo 2643. A partir desse instante, a base de dados começa a ter tráfego de *bots*. Deste modo, os valores extremos obtidos nos segundos 2854, 4614 e 5100 foram influenciados pelo tráfego dos *bots*, sendo então definidos como os rótulos que avisam o acontecimento de um possível ataque DDoS. É oportuno destacar que, apenas os valores extremos obtidos **antes** do lançamento do ataque e após o início da infecção, foram considerados rótulos que avisam o acontecimento de ataques DDoS. Isso ocorre, pois, o objetivo da abordagem ESPA e desta avaliação é predizer os ataques DDoS. Assim, é necessário que os sinais da preparação precedam o ataque.

O mesmo acontece com os dados de teste apresentados pela Figura 5.4(b). Neste caso, segundo a documentação da base[5], o início da infecção ocorreu no segundo 527. Isso significa que o segundo 336 não obteve influência de *bots*. Assim, esse trabalho rotulou o segundo 336 como tráfego normal, visto que a documentação da base não apresenta justificativas para indicá-lo como a preparação de um ataque DDoS. Os valores extremos obtidos nos segundos 549 e 576 foram influenciados pelo tráfego dos *bots*, sendo então definidos como os rótulos que avisam o acontecimento de um possível ataque DDoS.

O tratamento de dados faltantes foi necessário, pois, assim como apresentado na Seção 2.5, alguns indicadores estatísticos geram sinais com frequência menor. Por exemplo, com o intervalo de 20 segundos. Assim, para evitar a degradação da acurácia, este trabalho realizou a ação padrão da abordagem ESPA e adicionou o valor -1 em todos os campos sem dados. O redimensionamento de dados é uma prática comum na área de aprendizado de máquina. Este trabalho redimensionou as bases de treinamento e testes utilizando a estratégia Min-Max. Ao fim

[4] https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-51/
[5] https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-52/



(a) Rótulos no treinamento

(b) Rótulos no teste

Figura 5.4: Rótulos no Experimento 1 do Caso de Uso 1

do processo de redimensionamento, o Adaboost é treinado com a base de treinamento contendo a união dos sinais processados pela engenharia de sinais da central de inteligência.

A Figura 5.4 ainda ressalta o grande desbalanceamento presente nas bases de dados. Isso ocorre, pois a base de treinamento possui 8803 segundos de tráfego de dados e apenas três desses segundos podem ser rotulados ciclos que avisam o acontecimento de um possível ataque DDoS. O mesmo é observado na base de teste que possui 972 segundos e apenas dois ciclos que avisam o acontecimento de ataques DDoS. O problema do desbalanceamento é relevante por impactar negativamente alguns algoritmos de aprendizado de máquina. É possível que algoritmos privilegiem a classe majoritária, classificando erroneamente os ciclos que avisam o acontecimento de ataques DDoS (Castro e Braga, 2011). Por exemplo, um algoritmo de aprendizado de máquina que classifica toda a base de teste, todos os 972 segundos, como ciclos normais tem 99,79% de acurácia. A primeira vista essa acurácia parece boa, porém nesse resultado hipotético nenhum sinal da preparação dos ataques foi identificado. Assim, a abordagem ESPA não avisaria o administrador de rede sobre a ocorrência de um ataque futuro e consequentemente não iria cumprir sua função. Apesar disso ser um desafio para a abordagem, nenhuma ação para diminuir o desbalanceamento foi realizada. Pois, também era de interesse deste trabalho avaliar a capacidade da abordagem ESPA em lidar com o desbalanceamento.

A última ação do Experimento 1 é testar a central de inteligência. Para isso, o Adaboost tenta classificar cada segundo da base de testes em ciclo normal ou em ciclo malicioso. O **ciclo normal** faz referência aos segundos onde não existem valores extremos nos sinais precoces de alerta. Os **ciclos maliciosos** são os segundos que possuem valores extremos nos sinais precoces de alerta e foram previamente rotulados (Figura 5.4).

A Tabela 5.3 apresenta os resultados da classificação. É pertinente destacar a diagonal principal da matriz, onde estão alocados os acertos do modelo. Assim, a abordagem ESPA acertou a classificação de 968 segundos como ciclos normais. Porém, o mais relevante é o outro componente da diagonal principal. Pois ele apresenta que a abordagem conseguiu identificar um ciclo malicioso. O momento corretamente classificado como ciclo malicioso foi o segundo 576. O ciclo malicioso rotulado no segundo 549 foi erroneamente classificado como ciclo normal, representando um FN. Por fim, a Tabela 5.3 indica a existência de dois FPs, ou seja, a abordagem

classificou dois ciclos normais como ciclos maliciosos. A acurácia da abordagem ESPA é de 99,69% e a precisão atingiu 33% para os ciclos maliciosos, isso ocorreu, pois dois ciclos normais foram classificados como ciclos maliciosos. A precisão dos ciclos normais foi de 99,89%. O *recall* para a classe dos ciclos maliciosos é de 50% pois um dos dois ciclos maliciosos foi identificado corretamente, enquanto o *recall* para a classe dos ciclos normais é de 99,79%. Por fim, o *F1-score* dos ciclos maliciosos é de 40% e o *F1-score* dos ciclos normais é de 99,85%.

Tabela 5.3: Resultados Experimento 1 do Caso de Uso 1

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	1	2
	Ciclo Normal	1	968

5.2.3 Experimento 2 do Caso de Uso 1

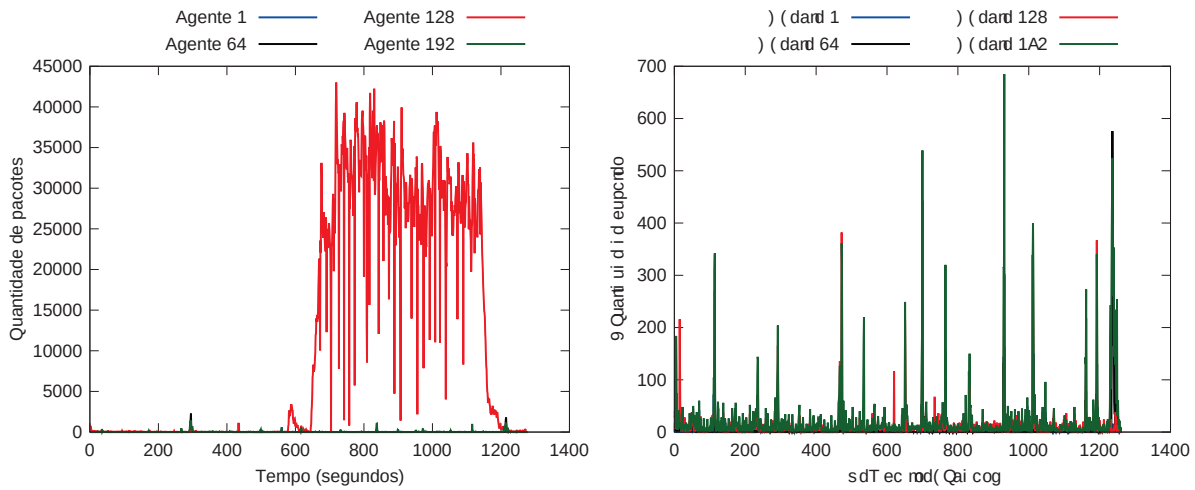
O Experimento 2 utiliza dois cenários da base de dados CICDDoS2019. Os cenários "Ataque UDP" e "Ataque UDP-Lag" são subsequentes, no primeiro dia de ataque, o tráfego de rede capturado no cenário "Ataque UDP" antecedeu o tráfego de rede coletado no cenário "Ataque UDP-Lag". Por esse motivo, este trabalho utilizou o cenário "Ataque UDP" como base de treinamento e o cenário "Ataque UDP-Lag" comoo base de teste. Apesar dos cenários selecionados disponibilizarem a carga útil dos pacotes, é oportuno reiterar que este experimento não manipulou nenhum tipo de informação presente na carga útil dos pacotes. Os ataques DDoS presentes na base foram conduzidos pelas *botnets* controladas pelos pesquisadores. A base de treinamento possui 1277 segundos de tráfego de dados, a base de teste possui 1260 segundos.

A próxima ação do Experimento 2 segmentou o tráfego de rede e extraiu a quantidade de pacotes por segundo. A Figura 5.5 apresenta o resultado da segmentação e contabilização do tráfego de rede para as bases de treinamento e teste. A Figura 5.5(a) mostra que ao longo da base de treinamento o Agente 128 possui os maiores picos na quantidade de pacotes. Novamente, esse comportamento pode ser justificado, pois todos os *bots* que realizaram o ataque possuíam endereços IP presentes no intervalo do Agente 128. O fato do Agente 128 coletar o tráfego de rede dos *bots* nos experimentos foi apenas coincidência. Contudo, existem momentos em que outros agentes capturaram mais pacotes. Por exemplo, nos segundos de 294 a 296 o Agente 64 contabilizou os maiores picos de pacotes. E o Agente 1 contabilizou 245 pacotes, representando o pico no segundo 172. A Figura 5.5(b) apresenta comportamento diferente para o teste. Apesar do Agente 128 possuir alguns picos de tráfego, o agente que mais possui picos é o Agente 192. A quantidade total de pacotes coletados pelo Agente 128 é 15020 pacotes e pelo Agente 192 é 28593.

Como no experimento anterior, a ação seguinte é o processamento do tráfego. Cada agente enviou contagem de pacotes para a central de inteligência. A central de inteligência utilizou os indicadores estatísticos *kurtosis*, *skewness*, S_{max}, S_{null}, S_{fold} e S_{hopf} para gerar 24 novas características por segundo. Novamente, apenas para simplificar a apresentação e a discussão dos resultados, a Figura 5.6 apresenta a variação do sinal *kurtosis* coletado pelo Agente 128 e processado pela central de inteligência nas bases de treinamento e no teste. Porém, todos os outros sinais estão presentes online[6].

A Figura 5.6(a) apresenta a variação do *kurtosis* no treinamento. A linha vermelha indica o segundo onde o ataque foi lançado pelos pesquisadores. É oportuno destacar que entre os

[6]https://github.com/andersonneira/tese_resultados/

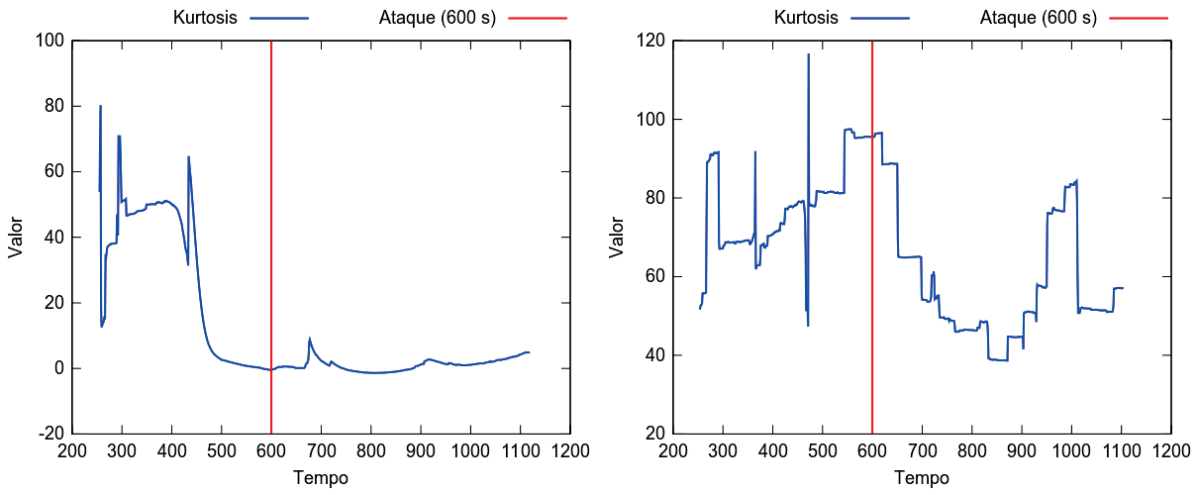


(a) Pacotes por segundo no treinamento

(b) Pacotes por segundo no teste

Figura 5.5: Quantidade de Pacotes por Segundo no Experimento 2 do Caso de Uso 1

segundos 433 e 434 o valor do *kurtosis* varia de 31,371 para 64,772. Essa variação corresponde a um aumento de 106,471%. Nos segundos 257, 267 e 293, a figura também apresenta variações, porém a intensidade da variação é menor quando comparadas ao segundo 434. A Figura 5.6(b) apresenta a variação do sinal *kurtosis* para a base de teste. Entre os segundos 471 e 472 o valor do *kurtosis* varia de 47,274 para 116,746. Essa variação corresponde a um aumento de 146,954%. O teste apresenta outros picos nos segundos 268, 365 e 544.



(a) *kurtosis* por segundo no treinamento

(b) *kurtosis* por segundo no teste

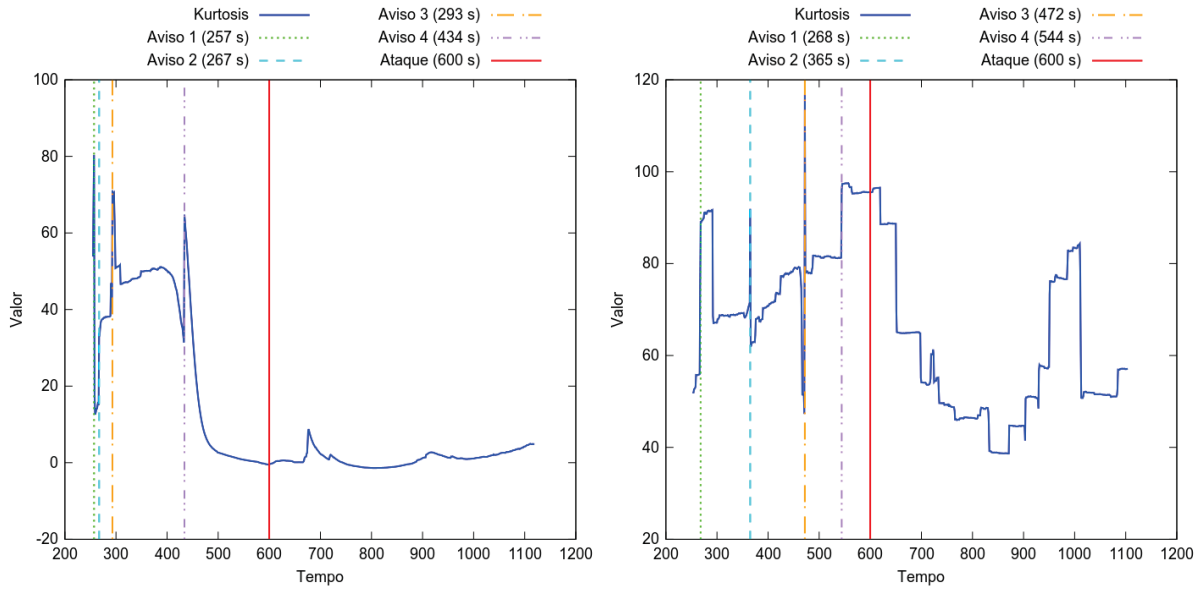
Figura 5.6: *Kurtosis* por Segundo no Experimento 2 do Caso de Uso 1

É necessário realizar o treinamento da central de inteligência. Para isso, esse trabalho conduziu a rotulação, o tratamento de dados faltantes e o dimensionamento Min-Max nas bases de treinamento e teste. Novamente, este trabalho utilizou os valores extremos apresentados anteriormente para rotular as bases de dados. A Figura 5.7 apresenta os segundos rotulados nas bases de treinamento e teste. A documentação oficial não especifica quando acontece a infecção dos *bots*. Porém, desde o começo, a base de dados e a documentação apresentam o tráfego de *bots*. No caso do treinamento (Figura 5.7(a)), valores extremos obtidos nos segundos 257, 267, 293 e 434 são os rótulos que indicam o acontecimento de um possível ataque DDoS. É

oportuno destacar que, apenas os valores extremos obtidos **antes** do lançamento do ataque foram considerados ciclos maliciosos.

O mesmo acontece com os dados de teste apresentados pela Figura 5.7(b). Neste caso, os rótulos que representam os ciclos maliciosos foram adicionados nos picos representados pelos segundos 268, 365, 472 e 544. É necessário rotular as bases para ser possível utilizar o aprendizado de máquina supervisionado. Assim, caso o algoritmo de aprendizado de máquina identifique um possível valor extremo, a abordagem ESPA notifica os administradores de rede. Como no experimento anterior, este trabalho realizou a ação padrão da abordagem ESPA e adicionou o valor -1 em todos os campos sem dados e redimensionou as bases de treinamento e testes utilizando a estratégia Min-Max. Ao fim do processo de redimensionamento, o Adaboost é treinado com a base de treinamento contendo a união do tráfego de rede coletado por todos os agentes e processados (engenharia de sinais) pela central de inteligência.

A Figura 5.7 também ressalta o grande desbalanceamento presente nas bases de dados. Isso ocorre, pois a base de treinamento possui 1277 segundos de tráfego de dados e apenas quatro desses segundos podem ser rotulados como ciclo malicioso. O mesmo é observado na base de teste que possui 1260 segundos e apenas quatro ciclos maliciosos. Assim como apresentado no Experimento 1 o desbalanceamento dos dados pode ser um problema. Por exemplo, um algoritmo de aprendizado de máquina que classifica toda a base de teste, todos os 1260 segundos, como ciclos normais tem 99,68% de acurácia. Novamente, esse resultado parece bom, mas não é. Pois a abordagem ESPA não avisaria o administrador de rede sobre a ocorrência de um ataque futuro e consequentemente não iria cumprir sua função. É de interesse deste trabalho verificar como a abordagem ESPA lidaria com o desbalanceamento, por isso, nenhuma ação foi realizada para lidar com o desbalanceamento dos dados.



(a) Rótulos no treinamento

(b) Rótulos no teste

Figura 5.7: Rótulos no Experimento 2 do Caso de Uso 1

A última ação do Experimento 2 é testar a central de inteligência. Para isso, o Adaboost tenta classificar cada segundo da base de testes em ciclo normal ou em ciclo malicioso. A Tabela 5.4 apresenta os resultados da classificação. É oportuno destacar a diagonal principal da matriz, onde estão alocados os acertos do modelo. Assim, a abordagem ESPA acertou a classificação de 1256 ciclos como ciclos normais e um ciclo malicioso corretamente. O momento corretamente classificado como ciclo malicioso foi o segundo 472. Três ciclos maliciosos

rotulados nos segundos 268, 365 e 544 foram erroneamente classificados como ciclos normais, apresentando três FNs. Diferentemente do experimento anterior, a Tabela 5.4 não indica a existência de FPs. A acurácia da abordagem é de 99,76% e a precisão para a classe dos ciclos maliciosos atingiu 100% pois não existem FPs. Para os ciclos normais a precisão foi de 99,76% pois três segundos foram classificados como ciclos normais e eram ciclos maliciosos. O *recall* para a classe dos ciclos maliciosos é de 25% pois apenas um dos quatro ciclos maliciosos foi identificado corretamente, enquanto o *recall* para a classe dos ciclos normais é de 100%. Por fim, o *F1-score* dos ciclos maliciosos é de 39,91% e o *F1-score* dos ciclos normais é de 99,87%.

Tabela 5.4: Resultados Experimento 2 do Caso de Uso 1

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	1	0
	Ciclo Normal	3	1256

5.2.4 Experimento 3 do Caso de Uso 1

A diferença entre os Experimentos 1 e 3 é o algoritmo de aprendizado de máquina presente na central de inteligência. No Experimento 1, a central de inteligência utiliza o Adaboost, um algoritmo de aprendizado de máquina do tipo *shallow learning* (definido na Subseção 5.2.1). No Experimento 3, a central de inteligência utiliza a MLP, um algoritmo de aprendizado de máquina do tipo *deep learning*. Deste modo, o Experimento 3 utiliza os mesmos dados de treinamento e teste e os mesmos rótulos do Experimento 1. Os dados de treinamento e teste estão apresentados na Figura 5.4. Como apresentado anteriormente, o objetivo de replicar o Experimento 1 modificando a central de inteligência para usar *deep learning* é verificar se algoritmos de *deep learning* podem melhorar os resultados obtidos anteriormente.

A Tabela 5.5 apresenta os resultados obtidos no Experimento 3. A primeira célula da diagonal principal indica que o experimento obteve sucesso em identificar um ciclo malicioso. O segundo corretamente classificado como ciclo malicioso foi o 549. O outro componente da diagonal principal indica que a grande maioria dos ciclos normais foram classificados corretamente. O ponto negativo desses resultados é o total de erros obtidos. A central de inteligência errou a classificação de 11 segundos, 10 foram erroneamente classificados como ciclo malicioso e um ciclo malicioso foi erroneamente classificado como ciclo normal. Os 10 ciclos normais erroneamente classificados como ciclos maliciosos foram os segundos 548, 551 a 558 e 582. A acurácia da central de inteligência é 98,87%. A precisão para a classe dos ciclos maliciosos é de 9,09%, isso ocorre, pois, das 11 vezes que a central de inteligência classificou um segundo como ciclo malicioso, apenas uma classificação estava correta. A precisão para a classe dos ciclos normais é de 99,89%, isso porque das 961 classificações como ciclo normal 960 estavam corretas. O *recall* para a classe dos ciclos maliciosos é de 50%, pois dos dois ciclos maliciosos, a central de inteligência identificou um. O *recall* para a classe dos ciclos normais é de 98,96%, pois dos 970 ciclos normais, 960 foram corretamente classificados. Por fim, o *F1-score* dos ciclos maliciosos é de 15,38% e o *F1-score* dos ciclos normais é de 99,37%.

5.2.5 Experimento 4 do Caso de Uso 1

A diferença entre os Experimentos 2 e 4 é o algoritmo de aprendizado de máquina presente na central de inteligência. No Experimento 2, a central de inteligência utiliza o Adaboost,

Tabela 5.5: Resultados Experimento 3 do Caso de Uso 1

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo malicioso	1	10
	Ciclo Normal	1	960

no Experimento 4, a central de inteligência utiliza a MLP. Deste modo, o Experimento 4 utiliza os mesmos dados e os mesmos rótulos do Experimento 2. Os dados de treinamento e teste estão apresentados na Figura 5.7 e online[7]. Como apresentado anteriormente, o objetivo de replicar o Experimento 2 modificando a central de inteligência para usar *deep learning* é verificar se algoritmos de *deep learning* podem melhorar os resultados obtidos anteriormente.

A Tabela 5.6 apresenta os resultados obtidos no Experimento 4. A primeira célula da diagonal principal indica que o experimento obteve sucesso em identificar um ciclo malicioso. O segundo corretamente classificado como ciclo malicioso foi o 365. O outro componente da diagonal principal indica que a grande maioria dos ciclos normais foram classificados corretamente. Assim como no Experimento 2, a central de inteligência errou três dos quatro ciclos maliciosos. Além disso, a central de inteligência equipada com o *deep learning* classificou dois ciclos normais como ciclo malicioso. Os dois ciclos normais erroneamente classificados como ataque foram os segundos 537 e 542. A acurácia da central de inteligência é 99,60%. A precisão para a classe dos ciclos maliciosos é de 33,33%, isso ocorre, pois, das três vezes que a central de inteligência classificou um segundo como ciclo malicioso, apenas uma classificação estava correta. A precisão para a classe dos ciclos normais é de 99,76%, isso porque das 1257 classificações como ciclo normal 1254 estavam corretas. O *recall* para a classe dos ciclos maliciosos é de 25%, pois dos quatro ciclos maliciosos, a central de inteligência identificou um. O *recall* para a classe dos ciclos normais é de 99,84%, pois dos 1256 ciclos normais, 1254 foram corretamente classificados. Por fim, o *F1-score* dos ciclos maliciosos é de 28,57% e o *F1-score* dos ciclos normais é de 99,80%.

Tabela 5.6: Resultados Experimento 4 do Caso de Uso 1

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	1	2
	Ciclo Normal	3	1254

5.2.6 Discussão dos Resultados do Caso de Uso 1

No Experimento 1, a proposta classificou corretamente um segundo como ciclo malicioso (Tabela 5.3). O segundo 576 da base de teste foi previamente rotulado como um sinal capaz de indicar um possível ataque DDoS. Assim como apresentado na Figura 5.8, esse momento antecede o ataque em 3 minutos e 22 segundos. Esse resultado é expressivo, pois apenas 49 segundos após a infecção a abordagem notificou a possibilidade de um ataque DDoS. Além disso, o pico de pacotes durante o ataque acontece no segundo 875, atingindo 25269 pacotes coletados pelo Agente 128. Assim, a abordagem ESPA proporciona para os administradores da rede 4

[7] https://github.com/andersonneira/tese_resultados/

minutos e 59 segundos antes do pico do ataque (Figura 5.8) para ser possível tomar medidas contra o ataque antecipadamente.

Para ressaltar ainda mais a importância desses resultados, este trabalho apresenta uma situação hipotética. Supondo que uma solução de detecção de ataque DDoS qualquer identificasse o ataque no exato segundo após lançamento do ataque, os administradores de rede teriam apenas 1 minuto e 37 segundos até o ataque atingir o ápice. Enquanto a abordagem ESPA proporciona 4 minutos e 59 segundos até o ápice do ataque. Portanto, nessa situação hipotética, o administrador de rede teria 208% mais tempo para lidar com o pico do ataque.

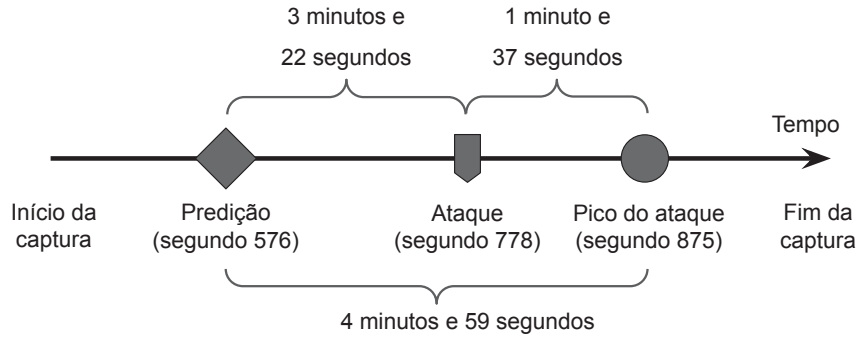


Figura 5.8: Relação Entre a Predição, o Lançamento do Ataque e o Pico do Ataque no Experimento 1 do Caso de Uso 1

Ainda sobre o Experimento 1, a abordagem ESPA classificou os segundos 577 e 578 como ciclos maliciosos. Isso significa que dois segundos normais, onde não ocorrem mudanças nos dados, foram erroneamente classificados como possíveis evidências de ataque. Esse erro de classificação é muito sério, pois esse momento não representa um sinal no *kurtosis* que justificasse um alerta de ataque. Assim, em um cenário onde não ocorrem ataques, a abordagem ESPA iria predizer um ataque inexistente, estimulando a equipe de segurança a tomar medidas desnecessárias, desperdiçando tempo e dinheiro. No caso avaliado, os segundos 577 e 578 sucedem o sinal precoce de alerta (segundo 576). Isso reduz a magnitude do erro. Pois eles poderiam reforçar o alerta de possíveis ataques, visto que, este é um cenário de ataque DDoS.

No Experimento 2, a proposta identificou um segundo como um sinal capaz de indicar um possível ataque DDoS (Tabela 5.4). Esse segundo é o 472 da base de teste. Assim, a abordagem ESPA prediz o ataque 2 minutos e 8 segundos antes dos pesquisadores lançarem o ataque (Figura 5.9). Esse resultado é muito importante, por possibilitar aos administradores de rede uma vantagem em relação ao ataque. Essa vantagem seria maior caso a abordagem ESPA pudesse identificar o ataque nos segundos 268 e 365, pois eles precedem o segundo 472. Além de aumentar a vantagem em relação ao ataque, acertar os outros três ciclos maliciosos iria aumentar a confiança na informação passada pela abordagem ESPA.

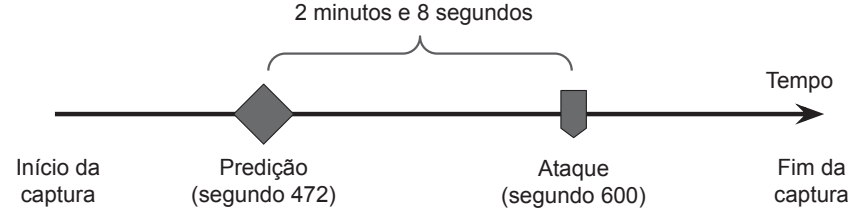


Figura 5.9: Relação Entre a Predição e o Lançamento do Ataque no Experimento 2 do Caso de Uso 1

Ao comparar os resultados obtidos nos Experimentos 1 e 3 (Tabelas 5.3 e 5.5 respectivamente) os resultados do Experimento 1 aparentam ser melhores do que dos resultados do

Experimento 3. Se a análise for realizada utilizando o total de erros, os resultados do Experimento 1 apresentam menos erros. Quando a central de inteligência utilizou o Adaboost o total de erros foi apenas 3 (dois FPs e um FN). Quando a central de inteligência utilizou a MLP o total de erros foi 11 (10 FPs e um FN). Porém, se a análise se concentrar em quando aconteceu a predição, a central de inteligência utilizando a MLP apresenta vantagem. Isso porque o ciclo malicioso identificado no Experimento 3 é anterior ao ciclo malicioso identificado no Experimento 1. Enquanto no Experimento 3 a identificação de sinais da preparação de um ataque DDoS ocorreria no segundo 549, no Experimento 1 a predição ocorreria no segundo 576. Deste modo, utilizando a MLP na central de inteligência, a abordagem ESPA poderia proporcionar 27 segundos a mais do que utilizando o Adaboost. A Figura 5.10 ressalta essa vantagem de 27 segundos, pois 3 minutos e 49 segundos antes do lançamento do ataque os administradores de rede receberam uma notificação sobre a possibilidade de acontecer um ataque DDoS. Esse tempo fica ainda mais expressivo quando comparado com o pico do ataque. Pois a abordagem ESPA antecipou o pico do ataque em 5 minutos e 26 segundos. Como citado anteriormente, o custo desses 27 segundos de vantagem foi o aumento de oito FPs.

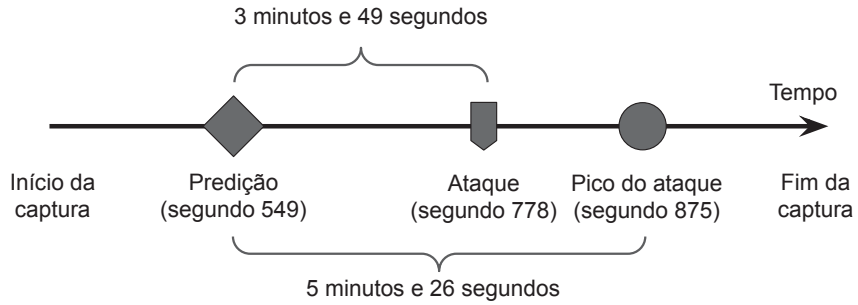


Figura 5.10: Relação Entre a Predição, o Lançamento do Ataque e o Pico do Ataque no Experimento 3 do Caso de Uso 1

Ao comparar os resultados obtidos nos Experimentos 2 e 4 (Tabelas 5.4 e 5.6 respectivamente) a diferença é que o Experimento 4 apresenta dois FPs enquanto o Experimento 2 apresenta zero FP. Como discutido anteriormente, o FP é um erro que deve ser evitado. Seguindo este prisma, os resultados da central de inteligência utilizando o Adaboost são melhores. Contudo, é oportuno realizar a análise também utilizando o momento em que a abordagem realizou a predição. No Experimento 2, a predição ocorreu no segundo 472, enquanto no Experimento 4 a predição ocorreu no segundo 365. Essa diferença entre os resultados apresenta um aumento de 1 minuto e 47 segundos do Experimento 4 em relação com a predição realizada no Experimento 2. A Figura 5.11 apresenta que esse aumento no tempo de predição proporciona aos administradores de rede 3 minutos e 55 segundos antes do lançamento do ataque. Assim como no Experimento 3, o custo para aumentar o tempo da predição é o aumento dos FPs. Contudo, o aumento dos FPs no Experimento 4 é significativamente menor do que o aumento dos FPs no Experimento 3.

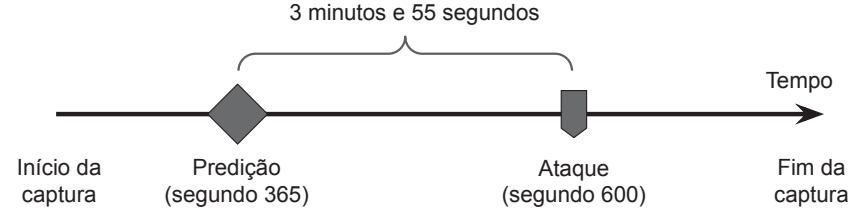


Figura 5.11: Relação Entre a Predição e o Lançamento do Ataque no Experimento 4 do Caso de Uso 1

Os resultados obtidos nos Experimentos 1, 2, 3 e 4 do Caso de uso 1 indicam que, nas condições analisadas, a abordagem ESPA identificou sinais da orquestração dos ataques antes que os atacantes lancem os ataques. Assim, esses resultados corroboram a hipótese de que a proposta pode realizar a predição de ataque DDoS e identificar sinais da preparação dos ataques DDoS. Deste modo, a abordagem ESPA viabiliza mais tempo para os administradores lidarem com os ataques DDoS, pois os administradores de rede seriam alertados sobre ataques ainda não lançados pelos atacantes. Contudo, alguns erros de predição poderiam e deveriam ser evitados. Assim, a abordagem ESPA proporciona mais tempo e apresentaria mais evidências dos possíveis ataques.

5.2.7 Comparação dos Resultados do Caso de Uso 1 com a Literatura

Esta subseção apresenta a comparação entre a abordagem ESPA e o sistema ANTE (de Neira et al., 2020). O objetivo do sistema ANTE é realizar a identificação antecipada de *botnets*. Para isso, o sistema ANTE analisa o tráfego da rede com o intuito de identificar *bots* para notificar os administradores da rede antes que as consequências do ataque gerado pela *botnet* sejam irremediáveis. O sistema ANTE captura o tráfego de rede de modo online ou off-line. Após a coleta do tráfego de rede, o sistema pré-processa os dados transformando-os em atributos que representem o comportamento dos *bots*. O sistema ANTE suporta algoritmos de aprendizado de máquina supervisionado e não supervisionado para identificar os *bots*. Dentre os resultados reportados, o sistema ANTE identificou *bots* utilizando dados de 60 segundos antes do lançamento do ataque. Apesar de não ser focado para realizar a predição de ataques DDoS, tal ação pode acontecer se o sistema ANTE notificar os administradores de rede sobre a existência de *bots* antes do lançamento do ataque.

O sistema ANTE foi testado em oito cenários onde *botnets* realizaram ataques DDoS, tentaram quebrar senhas utilizando força bruta ou tentaram roubar dados sensíveis. Dentre as oito avaliações, uma delas ocorreu no cenário 52 da CTU-13, assim como nos Experimentos 1 e 3 apresentados anteriormente. O sistema ANTE identificou evidências sobre a possibilidade da ocorrência de um ataque DDoS 60 segundos antes dos pesquisadores lançarem o ataque. Já a abordagem ESPA, utilizando *shallow learning*, conseguiu notificar os administradores de rede 3 minutos e 22 segundos antes do lançamento do ataque (Experimento 1). A versão da abordagem ESPA com *deep learning* conseguiu predizer o ataque com 3 minutos e 49 segundos de vantagem (Experimento 3). Deste modo, a abordagem ESPA proporciona mais tempo para os administradores de rede lidarem com o ataque. Outra vantagem da abordagem ESPA está relacionada ao processamento dos dados da rede. Enquanto a abordagem ESPA pode coletar os dados distribuidamente, o sistema ANTE coleta e processa o tráfego de rede centralizadamente. Assim, caso algum agente da abordagem ESPA pare de funcionar, a abordagem continua operando. Porém, caso o coletor do tráfego de rede do sistema ANTE pare de funcionar, o sistema para completamente. Uma propriedade comum entre a abordagem ESPA e o sistema ANTE é a possibilidade de usar diferentes algoritmos de aprendizado de máquina. Assim, ambos são configuráveis e funcionam em diferentes modelos de rede. O sistema ANTE possui vantagem em relação aos detalhes do ataque. Pois a versão atual da abordagem ESPA apenas notifica sobre a possibilidade do ataque, enquanto o sistema ANTE caracteriza os possíveis *bots*. Por fim, a Tabela 5.7 resume a comparação realizada nesta subseção.

5.3 CASO DE USO 2 - AM NÃO SUPERVISIONADO E CENTRALIZADO

No Caso de uso 1, este trabalho verificou a validade da hipótese ao ser possível identificar sinais da preparação dos ataques DDoS usando a teoria dos sinais precoces de alerta por meio da

Tabela 5.7: Resumo da Comparação Entre a abordagem ESPA e o sistema ANTE no Cenário 52 (CTU-13)

Sistema	Tempo predição	Coleta distribuída	Diferentes AM	Detalhes
Abordagem ESPA *(Shallow Learning)*	3 min e 22 seg	✓	✓	✗
Abordagem ESPA *(Deep Learning)*	**3 min e 49 seg**	✓	✓	✗
Sistema ANTE	1 min e 00 seg	✗	✓	✓

engenharia de sinais realizada pela abordagem ESPA. Para complementar a análise anterior e evoluir a literatura, o Caso de uso 2 visa avaliar a abordagem ESPA sobre outras circunstâncias como a diminuição da dependência de dados rotulados. Ao construir uma solução baseada em dados rotulados, a capacidade de predizer ataques DDoS pode ficar limitada a comportamentos semelhantes aos padrões de treinamento. Portanto, diminuir a dependência de dados rotulados reduz essa limitação e facilita o uso da abordagem, visto que obter dados rotulados sobre vários tipos de ataques DDoS pode ser custoso. Ao avaliar outros parâmetros da abordagem ESPA é possível verificar a capacidade de adaptação da abordagem ESPA. Para atingir esses objetivos, este caso de uso conduziu seis experimentos detalhados a seguir.

5.3.1 Definição dos Experimentos do Caso de uso 2

É necessário preparar a abordagem ESPA definindo os parâmetros apresentados anteriormente (Subseção 4.2.1). A Tabela 5.8 apresenta a configuração dos parâmetros usados pela abordagem ESPA nos experimentos deste caso de uso. A motivação da escolha e a apresentação destes parâmetros estão descritas ao longo desta subseção.

Tabela 5.8: Definição dos Parâmetros Usados Pela Abordagem ESPA no Caso de Uso 2

Parâmetro	**Valor**
Modo de coleta do tráfego de rede	Centralizado
Quantidade de agentes	Um
Localização dos agentes	Na rede vítima
Atributos do tráfego de rede	Total de pacotes trafegados na rede, quantidade de endereços de IP na origem e no destino
Ciclos de captura	Um segundo
Indicadores estatísticos	*kurtosis*, *skewness* e CV
Algoritmos de aprendizado de máquina	LSTM *Autoencoder* e *One-Class* SVM
Seleção hiperparâmetros dos algoritmos de aprendizado de máquina	Manual e Automática
Janela deslizante	5% e 10% da base de dados
Tratamento de dados faltantes	Remoção
Redimensionamento dos dados	Nenhum

Este caso de uso utiliza apenas um agente para coletar o tráfego de rede centralizadamente na rede da vítima. Assim, este trabalho avalia a abordagem ESPA com um modo de coleta do tráfego de rede diferente do usado no Caso de uso 1. A coleta do tráfego de rede centralizada pode

requerer grande poder computacional para ser executada. Contudo, para verificar a capacidade de adaptação da abordagem ESPA e reforçar a verificação da hipótese, é oportuno verificar se a abordagem proposta pode predizer ataques DDoS mesmo quando todos os dados estão concentrados, ao invés de distribuídos (Caso de uso 1).

O Caso de uso 1 avaliou a abordagem ESPA utilizando um atributo do tráfego de rede coletado em ciclos de captura de um segundo. O Caso de uso 2 também usa o ciclo de captura de um segundo. Contudo, este caso de uso avalia outros atributos além do total de pacotes trafegados na rede. A quantidade de endereços de IP na origem e no destino dos pacotes representam outros dois atributos do tráfego de rede usados neste caso de uso. Para definir a quantidade de endereços IPs de origem, contaram-se quantos endereços únicos enviaram pacotes através do campo endereço de origem do pacote IP. A quantidade de endereços de destino baseia-se na contagem de endereços IPs únicos existentes no campo de destino do pacote IP. Este trabalho escolheu esses atributos, pois a falsificação de endereços IP é uma prática comum em ataques DDoS (Jyoti e Behal, 2021). Então, a quantidade de endereços IPs que enviam pacotes antes do ataque apresenta potencial para ser utilizada pela abordagem ESPA. Pois, a preparação do ataque pode causar variações nesse atributo. Além disso, é possível que varreduras de dispositivos vulneráveis impactem na quantidade de endereços IP de destino. Portanto, a engenharia de sinais é realizada com base em três séries temporais formadas pelo total de pacotes trafegados na rede e pela quantidade de endereços de IP na origem e no destino coletados a cada segundo.

Como este caso de uso definiu a utilização de mais atributos do tráfego de rede, foi oportuno avaliar uma configuração da abordagem ESPA que gere menos características durante a engenharia de sinais. Assim, os experimentos deste caso de uso avaliaram a predição dos ataques DDoS com até três indicadores estatísticos simultaneamente. Os indicadores estatísticos são o *kurtosis*, o *skewness* e o CV. Além de variar a quantidade de indicadores estatísticos, este caso de uso variou o modo em que a engenharia de sinais fez uso deles. Assim como definido na Subseção 4.2.3, é possível executar a engenharia de sinais aplicando todos os indicadores estatísticos sobre todos os atributos do tráfego de rede. Bem como é possível aplicar apenas um indicador estatístico para um atributo do tráfego de rede. Assim, cada experimento conduzido neste caso de uso irá definir como a engenharia de sinais foi aplicada.

Neste caso de uso foram avaliados a LSTM *Autoencoder* e o *One-Class* SVM, dois algoritmos de aprendizado de máquina que independem de dados rotulados para realizar os treinamentos. A rede neural LSTM é um subtipo das redes neurais recorrentes (do inglês, *Recurrent Neural Networks* - RNNs) (Hochreiter e Schmidhuber, 1997), projetadas para lidar com entradas de dados sequenciais. Como elas utilizam uma memória interna, essas redes possuem a capacidade de conservar informações, armazenando o estado das células para uso posterior. Elas fazem isso com um conjunto de portões (do inglês, *gates*) utilizados para controlar quando as informações entram, são emitidas ou removidas das células. As LSTMs são adequadas para tratar problemas onde os dados evoluem ao longo do tempo e para realizar predições (Lindemann et al., 2021). A LSTM se diferencia das outras redes neurais por diminuir o problema de dissipação do gradiente. O algoritmo *backpropagation* é utilizado para o treinamento de redes neurais. Este algoritmo atualiza os pesos dos neurônios baseado no erro da saída do modelo. Contudo, ao longo do tempo de treinamento, é possível que o algoritmo *backpropagation* acabe dissipando o valor do erro e assim não consiga atualizar os pesos das camadas, principalmente as iniciais. A rede LSTM evita o problema de dissipação do gradiente por meio de Carrosséis de Erro Constante (CECs). Os CECs são aplicados em células especiais para criar um fluxo de erro constante. O *backpropagation* obtém acesso a essas células especiais por intermédio de um portão multiplicativo, que aprendem quando devem conceder acesso (Staudemeyer e Morris, 2019).

O *Autoencoder* é uma rede neural composta por camadas de entrada e saída, um espaço latente, uma rede neural codificadora e uma rede neural decodificadora. Esta rede neural visa compreender como codificar e decodificar os dados selecionados pelo usuário da rede neural. Para isso, o *Autoencoder* comprime os dados no espaço latente, para em seguida decodificar os dados de saída do modelo. O *Autoencoder* compara a saída processada com os dados originais do usuário. A diferença entre o predito e o real é utilizado para atualizar os pesos da rede (Nguyen et al., 2021). A principal vantagem da rede é que o treinamento pode ser conduzido sem a utilização de dados rotulados (Malhotra et al., 2015). Isso facilita a adoção da proposta em ambientes reais por diminuir o custo da aquisição dos rótulos dos dados. Assim, o administrador de rede inicia a coleta do funcionamento da rede e a abordagem ESPA identifica mudanças com comportamento dos sinais que representem a ocorrência de um ataque DDoS futuro.

Este trabalho utiliza uma rede LSTM *Autoencoder*. Assim, tanto o codificador quanto o decodificador são redes LSTM (Nguyen et al., 2021). A Figura 5.12 ilustra a arquitetura genérica da rede LSTM *Autoencoder*. A LSTM *Autoencoder* aprende a criar uma representação compacta de sinais temporais inseridos na entrada da rede neural. A partir dessa representação, a rede LSTM *Autoencoder* consegue recriar sinais temporais similares aos introduzidos na entrada sem gerar *overfitting*. No entanto, usualmente esses sinais não são exatamente iguais. Assim, ao comparar o sinal reconstruído na saída da rede neural com o sinal de entrada é possível obter uma diferença entre os valores. Neste trabalho essa diferença entre os sinais reconstruídos e os sinais originais é nomeado de custo de reconstrução. Para obter o custo de reconstrução, este trabalho utiliza o erro médio absoluto (do inglês, *Mean Absolute Error* - MAE) como o custo de reconstrução. O MAE (Fórmula 5.1) é uma média dos erros absolutos $|e_i| = |y_i - x_i|$, onde y_i é o valor reconstruído e x_i o valor original. O termo N é a quantidade de sinais observados.

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{N} = \frac{\sum_{i=1}^{n} |e_i|}{N} \tag{5.1}$$

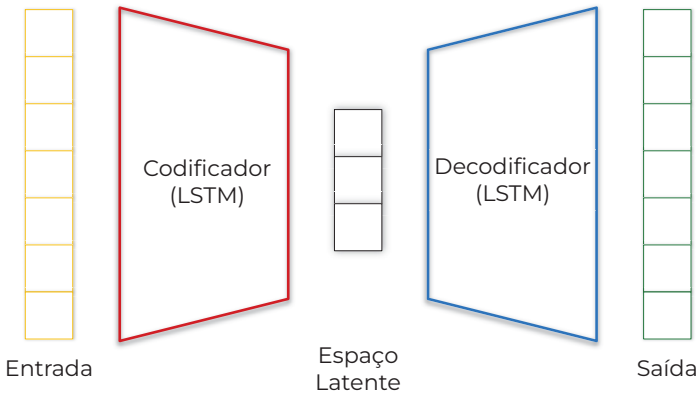


Figura 5.12: Arquitetura Genérica da Rede LSTM *Autoenconder* (adaptado de Nguyen et al. (2021))

O *One-Class* SVM é o outro algoritmo usado neste caso de uso. O *One-Class* SVM é um algoritmo de aprendizado de máquina não supervisionado que realiza a detecção de uma única classe. Este trabalho relaciona o *One-Class* SVM com o aprendizado de máquina não supervisionado, pois os resultados (predições) são obtidos sem o uso de rótulos, assim como definido na literatura (Dayan et al., 1999; Ghahramani, 2004; Hastie et al., 2009; James et al., 2023). O *One-Class* SVM funciona como uma função de decisão que envolve a maioria do tráfego normal formando uma barreira limite. Essa função de decisão é definida por meio do treinamento do modelo. Todos os pontos dentro dessa barreira fazem parte da classe dos dados

normais, e todos os pontos fora dessa barreira serão classificados como *outliers* (Amer et al., 2013). Portanto, o *One-Class* SVM é adequado para detecção de *outliers* (Devi et al., 2019) e para a predição dos ataques DDoS, pois os cenários de ataques DDoS apresentam o desbalanceamento dos dados. Este trabalho hipotetizou que o tráfego normal seria utilizado como a única classe (dentro da barreira) e a preparação do ataque ficaria como *outliers* (fora da barreira).

O *One-Class* SVM possui parâmetros que desempenham papéis fundamentais na capacidade do algoritmo de trabalhar com dados de alta dimensionalidade. O *Kernel* determina a forma pela qual os dados serão separados no espaço. Possíveis valores para o parâmetro *Kernel* são: $poly$, $linear$, rbf, $sigmoid$ e $precomputed$[8]. O parâmetro *nu* atua como um limite para a taxa de erros durante o treinamento e também como um limite inferior para a proporção de vetores de suporte. Ao ajustar o valor do parâmetro *nu*, é possível controlar a quantidade de pontos de dados considerados na preparação dos ataques DDoS durante a fase de treinamento. Esse valor deve estar no intervalo $(0, 1]$, representando a porcentagem de pontos que serão classificados como *outliers*. A configuração e o treinamento do *One-Class* SVM impactam diretamente nos resultados obtidos. Assim, ao usar o *One-Class* SVM, a abordagem ESPA proporciona às equipes de segurança uma maior autonomia para adaptar o algoritmo, desenvolvendo assim um modelo de predição adequado e ajustado para atender suas necessidades.

Diferentemente do caso de uso anterior, os experimentos deste caso de uso não usam dados rotulados para treinar a abordagem ESPA. Porém, é necessário rotular os dados para quantificar os acertos e os erros da abordagem. Deste modo, os experimentos deste caso de uso consideram como **ciclo malicioso** todos os ciclos de captura de um segundo do tráfego de rede onde os *bots* enviaram **mais de dois pacotes**. Assim, os ciclos de um segundo do tráfego de rede que representam a preparação de ataque foram impactados por ações conduzidas pelos *bots* antes do lançamento do ataque. Os **ciclos normais** são os ciclos de um segundo onde os *bots* não enviaram pacotes ou enviaram até dois pacotes.

Os Experimentos 1, 2 e 3 utilizaram 5% da base de dados como o tamanho da janela deslizante e os Experimentos 4, 5 e 6 utilizaram 10%. O objetivo da escolha desses valores foi maximizar o tempo de predição dos ataques DDoS e avaliar a abordagem ESPA frente a diferentes valores da janela deslizante de tamanho fixo. Ao final da engenharia de sinais, os primeiros ciclos de captura da base de dados não possuem os valores respectivos às novas características. Isso ocorre, pois o tamanho da janela deslizante não foi atingido (Subseção 4.2.3). Assim, a abordagem ESPA não realiza a engenharia de sinais enquanto o tamanho da janela não é atingido. Nos experimentos deste caso de uso, esses dados foram descartados. Ou seja, o tratamento para os dados faltantes foi a remoção deles em todos os experimentos. Por fim, neste caso de uso a abordagem ESPA não aplicou nenhuma técnica de redimensionamento dos dados. O objetivo dessa decisão foi avaliar o comportamento da abordagem frente a dados não padronizados.

Os ambientes onde os experimentos foram executados também variam. Os Experimentos 1, 2, e 3 utilizaram a plataforma Google Colaboratory[9]. O Google Colaboratory disponibiliza o acesso a uma máquina virtual com 12 GB de memória RAM, 107 GB de armazenamento e uma GPU NVIDIA Tesla K80. Os Experimentos 4, 5 e 6 foram executados em um dispositivo que possui um processador Intel Core I5, um disco rígido de um terabyte, oito gigabytes de memória, com um sistema operacional Linux Mint e com Python 3.10.12. Todos os resultados são passíveis de replicação usando o código disponibilizado online[10].

[8] https://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html

[9] https://colab.research.google.com

[10] https://github.com/andersonneira/tese_resultados/

5.3.2 Experimento 1 do Caso de Uso 2

O objetivo específico do primeiro experimento é verificar se a abordagem ESPA pode ser usada para predizer os ataques DDoS sendo equipada com redes LSTM *Autoencoder* configuradas autonomamente e com os demais parâmetros citados anteriormente. Isso ocorre, pois os experimentos do caso de uso anterior usaram dados rotulados para obter os resultados apresentados. Este experimento utiliza um *framework* de AutoML do tipo NAS para construir e configurar uma rede neural para realizar a predição dos ataques DDoS. Assim, poupa-se tempo nessa tarefa, adquirindo, simultaneamente, resultados competitivos frente à literatura, no que tange ao tempo de processamento e à predição de ataques DDoS.

Este experimento utilizou a captura 51 da base de dados CTU-13. Diferentemente dos experimentos do caso de uso anterior, este experimento utiliza apenas tráfego prévio ao ataque. Assim, a base de dados foi dividida em três partes. Os dados após o lançamento do ataque, iniciado no segundo 5632, foram desconsiderados da avaliação. Como a janela deslizante de tamanho fixo utilizada neste experimento é de 5% da base de dados, a base de treinamento inicia-se no segundo 441. A base de treinamento compreende 30% de toda a base de dados. Assim, a base de treinamento vai até o segundo 2641 da captura. A base de testes utilizou o tráfego restante até o momento anterior ao lançamento do ataque (do segundo 2642 até o segundo 5631). Portanto, a base de teste possui 2990 segundos a serem avaliados.

Neste experimento, a engenharia de sinais utilizou os indicadores estatísticos *kurtosis*, *skewness* e CV para gerar as novas características. Este experimento atribuiu um indicador estatístico para cada atributo do tráfego de rede (par a par). Isso foi necessário para que a execução do experimento respeitasse os limites do ambiente de execução. Portanto, o *kurtosis* foi atribuído para o atributo quantidade de IPs de origem, o *skewness* foi calculado para a quantidade de IPs de destino e o CV foi calculado para o total de pacotes. Todos os resultados estão disponíveis online[10]. A combinação de indicadores estatísticos com os atributos do tráfego de rede (engenharia de sinais) foi responsável por maximizar os resultados da predição de ataques DDoS dentre vários testes. Porém, trabalhos futuros criarão maneiras para definir essas combinações automaticamente.

Após a engenharia de sinais executou-se o Autokeras (Jin et al., 2019) para identificar a melhor rede neural do tipo LSTM *Autoencoder* para a base de treinamento. O Autokeras é um *framework* AutoML do tipo NAS que utiliza uma otimização Bayesiana para encontrar a melhor arquitetura de rede neural baseada nos dados inseridos pelo usuário. O Autokeras implementa os passos da Figura 2.15 (Capítulo 2) por meio de um algoritmo de otimização de função de aquisição estruturado em árvore e um *kernel* de rede neural. O objetivo é que o Autokeras possa analisar o espaço de busca eficientemente. Além do código do *framework* ser aberto e ter extensiva documentação, o *framework* pode operar em CPU e GPU e os resultados das análises obtidas pelos autores são competitivos em relação à literatura (Jin et al., 2019).

Após a execução do Autokeras, a central de inteligência da abordagem ESPA utiliza a rede LSTM *Autoencoder* configurada pelo Autokeras para predizer os ataques DDoS na base de teste. A entrada da rede LSTM *Autoencoder* é composta por três novas características geradas pela engenharia de sinais (o *kurtosis* da quantidade de IPs de origem, o *skewness* da quantidade de IPs de destino e o CV do total de pacotes). Deste modo, a saída é composta pela reconstrução dessas características. A abordagem ESPA calcula e armazena o custo de reconstrução (MAE) para cada uma das três novas características. Para simplificar a apresentação dos resultados, a Figura 5.13 apresenta a média do custo de reconstrução das três novas características ao longo dos 2990 segundos da base de teste. Assim como no caso de uso anterior, uma fração da base de teste apresenta um comportamento diferente do restante. Entre os segundos 3814 e 4258 o custo de reconstrução é maior que um. Isso significa que, nesses segundos (de 3814 e 4258), a rede

LSTM *Autoencoder* selecionada e configurada pelo Autokeras apresenta um MAE maior do que no restante da base.

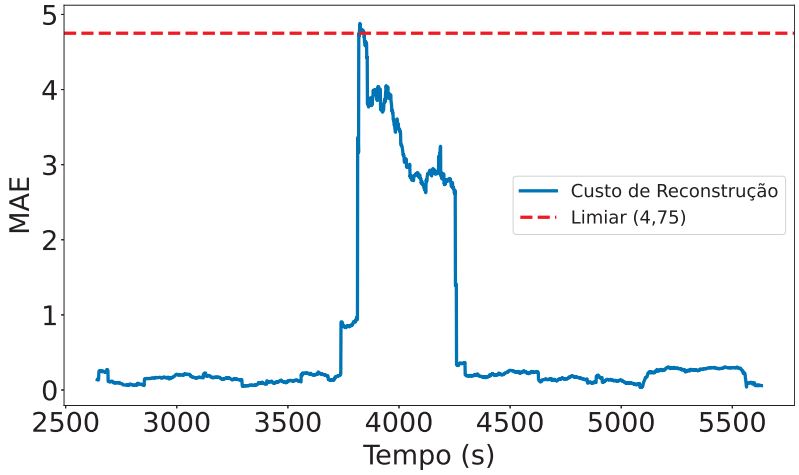


Figura 5.13: Variação do Custo de Reconstrução Médio na Base de Teste do Experimento 1 do Caso de Uso 2

Como a rede LSTM *Autoencoder* não classifica os ciclos de captura, é necessário definir algum mecanismo para automatizar a predição do ataque DDoS. O mecanismo utilizado neste experimento (e nos Experimentos 2 e 3) é o limiar de predição representado pela linha vermelha na Figura 5.13. Assim, qualquer ciclo de captura que apresente um custo de reconstrução maior do que o limiar é marcado como um ciclo malicioso e a predição do ataque DDoS ocorre. Este trabalho escolheu o valor do limiar para maximizar os resultados obtidos. Contudo, trabalhos como o de Silva et al. (2022b) e Brito et al. (2023b) avaliam modos para obter limiares adequados dinamicamente.

A Tabela 5.9 apresenta os resultados obtidos utilizando o limiar com o valor de 4,75 (linha vermelha na Figura 5.13). Neste experimento, a abordagem ESPA errou a classificação de apenas 59 dos 2990 segundos da base de teste. O momento corretamente classificado como ciclo malicioso foi o segundo 3841. A Tabela 5.9 indica a existência de 20 FPs e 39 FNs. A acurácia da central de inteligência da abordagem ESPA é de 98,03%. A precisão para a classe dos ciclos maliciosos é de 4,8% pois das 21 predições (1 VP + 20 FPs) apenas uma estava correta. Para os ciclos normais a precisão foi de 98,69% pois 39 ciclos maliciosos foram classificados como ciclos normais. O *recall* para a classe dos ciclos maliciosos é de 2,5% pois apenas um dos 40 ciclos maliciosos foram identificados corretamente, enquanto o *recall* para a classe dos ciclos normais é de 99,32% pois dos 2950 ciclos normais 2930 foram identificados corretamente. Por fim, o *F1-score* dos ciclos maliciosos é de 3,28% e o *F1-score* dos ciclos normais é de 99%.

Tabela 5.9: Resultados Experimento 1 do Caso de Uso 2

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	1	20
	Ciclo Normal	39	2930

5.3.3 Experimento 2 do Caso de Uso 2

O objetivo específico do segundo experimento deste caso de uso é avaliar a proposta em uma base de dados menor do que a utilizada no Experimento 1 deste caso de uso. Uma

base de dados menor pode impactar nos resultados obtidos devido à limitação da base de treinamento e teste. Assim, é oportuno verificar como a abordagem ESPA irá operar frente a tal situação. Para atingir esse objetivo específico, este experimento utilizou a captura 52 da base de dados CTU-13. Assim como no experimento anterior, este experimento utiliza apenas tráfego prévio ao ataque. A base de dados foi dividida em três partes. Os dados após o lançamento do ataque, iniciado no segundo 778, foram desprezados. Como no experimento anterior, 5% da base de dados foi o tamanho da janela deslizante de tamanho fixo. Como a base de dados é bem menor que a anterior, este trabalho aumentou proporcionalmente o tamanho da base de treinamento em relação ao experimento anterior. A base de treinamento compreende 37,5% de toda a base de dados. Portanto, a base de treinamento inicia-se no segundo 48 e vai até o segundo 364 da captura. Isso foi necessário para que a base de treinamento possuísse informação suficiente para que o AutoML pudesse identificar a arquitetura da rede neural ideal. A base de testes utilizou o tráfego restante até o momento anterior ao lançamento do ataque (do segundo 365 até o segundo 778). Portanto, a base de teste possui 413 segundos a serem avaliados.

Assim como no Experimento 1 deste caso de uso, a abordagem ESPA realizou a engenharia de sinais par a par com os atributos do tráfego de rede coletados. O *kurtosis* foi atribuído para o atributo quantidade de IPs de origem, o *skewness* foi calculado para a quantidade de IPs de destino e o CV foi calculado para o total de pacotes. Após a engenharia de sinais, este trabalho utilizou a base de treinamento formada pelas novas características para executar o Autokeras. A abordagem ESPA utiliza a rede LSTM *Autoencoder* configurada pelo Autokeras e base de treino para avaliar a predição dos ataques DDoS.

A Figura 5.14 apresenta o custo de reconstrução médio, das três novas características reconstruídas, ao longo dos 413 segundos da base de teste. Assim como nos resultados anteriores, uma fração da base de teste apresenta um comportamento diferente do restante. Entre os segundos 490 e 537 o custo de reconstrução é maior que dois. Isso significa que a rede LSTM *Autoencoder* selecionada e configurada pelo Autokeras apresentou dificuldade em reconstruir os dados coletados na rede e processados pela engenharia de sinais. Assim, nesses segundos (de 490 a 537) o experimento apresenta um MAE maior do que no restante da base.

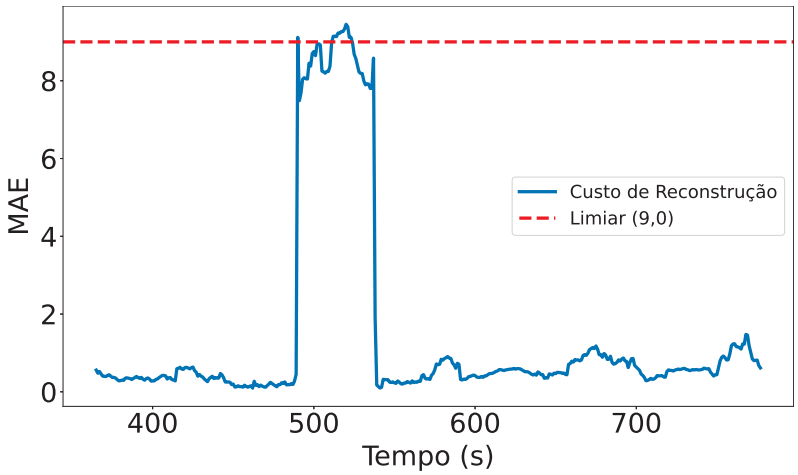


Figura 5.14: Variação do Custo de Reconstrução Médio na Base de Teste do Experimento 2 do Caso de Uso 2

A Tabela 5.10 apresenta os resultados obtidos utilizando o limiar com o valor de 9,0 (linha vermelha na Figura 5.14). Neste experimento, a abordagem errou a classificação de apenas 22 dos 413 segundos da base de teste. A Tabela 5.10 indica a existência de 11 FPs e 11 FNs. Os momentos corretamente classificados como ciclo malicioso foram os segundos 514 e 517. A acurácia da abordagem ESPA é de 94,70%. A precisão para a classe dos ciclos maliciosos é de

15,38% pois das 13 predições (2 VPs + 11 FPs) duas estavam corretas. Para os ciclos normais a precisão foi de 97,25% pois 11 ciclos maliciosos foram classificados como ciclos normais. O *recall* para a classe dos ciclos maliciosos é de 15,38% pois apenas dois dos 13 ciclos maliciosos foram identificados corretamente, enquanto o *recall* para a classe dos ciclos normais é de 97,25% pois dos 400 ciclos normais 389 foram identificados corretamente. Por fim, o *F1-score* dos ciclos maliciosos é de 15,38% e o *F1-score* dos ciclos normais é de 97,25%.

Tabela 5.10: Resultados Experimento 2 do Caso de Uso 2

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	2	11
	Ciclo Normal	11	389

5.3.4 Experimento 3 do Caso de Uso 2

O objetivo específico deste experimento é avaliar a proposta em uma base de dados com uma topologia diferente do que a utilizada nos Experimentos 2 e 3 deste caso de uso. Nos experimentos anteriores, os *bots* estavam na rede local da vítima. Para complementar a análise da abordagem ESPA usando redes neurais do tipo LSTM *Autoencoders* configurados autonomamente, este experimento define que, na base de dados avaliada, os *bots* não estejam alocados na mesma rede que a vítima.

Este experimento avaliou a base de dados CICDDoS2019, onde a rede vítima está conectada ao atacante pela Internet. Essa captura possui 19 ataques DDoS lançados pelos pesquisadores em dois dias. Neste caso, o ataque utilizado foi o primeiro ataque do primeiro dia (*Portmap*). O ataque começa no segundo 1484 da captura e possui duração de 540 segundos. O tamanho da janela deslizante de tamanho fixo compreende 5% da base de dados, assim como nos dois experimentos anteriores. O experimento utilizou 27,7% do tráfego total da captura para realizar o treinamento do modelo (do segundo 101 até o segundo 559) e o restante de tráfego prévio ao ataque para avaliação (do segundo 560 até o segundo 1483). Portanto, a base de teste possui 924 segundos.

Assim como nos experimentos anteriores deste caso de uso, a engenharia de sinais foi realizada par a par com os atributos do tráfego de rede coletados. O *kurtosis* foi atribuído para o atributo quantidade de IPs de origem, o *skewness* foi calculado para a quantidade de IPs de destino e o CV foi calculado para o total de pacotes. Após a engenharia de sinais, este trabalho utilizou a base de treinamento constituída pelas novas características geradas pela engenharia de sinais para executar o AutoML (Autokeras). A central de inteligência da abordagem ESPA utiliza a rede LSTM *Autoencoder* configurada pelo Autokeras para predizer os ataques DDoS na base de treino.

A Figura 5.15 apresenta o custo de reconstrução médio ao longo dos 924 segundos da base de teste. Assim como nos resultados anteriores, uma fração da base de teste (no começo da base) apresenta um comportamento diferente do restante. Entre os segundos 560 e 641 o custo de reconstrução é maior que quatro. Isso significa que a rede LSTM *Autoencoder* selecionada e configurada pelo Autokeras não consegue reconstruir precisamente os dados coletados na rede e processados pela engenharia de sinais. Assim, nesses segundos (de 560 e 641), o experimento apresenta um MAE maior do que no restante da base.

Usando o limiar de 8,3 (linha vermelha na Figura 5.15), a abordagem ESPA obteve os resultados apresentados na Tabela 5.11. Neste experimento, a abordagem errou a classificação de

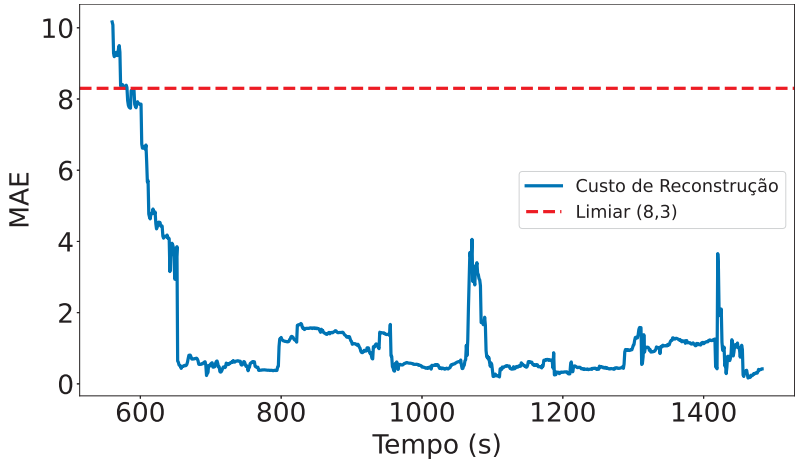


Figura 5.15: Variação do Custo de Reconstrução Médio na Base de Teste do Experimento 3 do Caso de Uso 2

133 dos 924 segundos da base de teste. Os seis ciclos maliciosos foram corretamente identificados nos segundos 571, 572, 573, 575, 578 e 580. A Tabela 5.11 indica a existência de 16 FPs e 117 FNs. A acurácia da abordagem proposta é de 85,61%. A precisão para a classe dos ciclos maliciosos é de 27,27% pois das 22 predições (6 VPs + 16 FPs) seis estavam corretas. Para os ciclos normais a precisão foi de 87,03% pois 117 ciclos maliciosos foram classificados como ciclos normais. O *recall* para a classe dos ciclos maliciosos é de 4,88% pois apenas 6 dos 123 ciclos maliciosos foram identificados corretamente, enquanto o *recall* para a classe dos ciclos normais é de 98% pois dos 801 ciclos normais 785 foram identificados corretamente. Por fim, o *F1-score* dos ciclos maliciosos é de 8,28% e o *F1-score* dos ciclos normais é de 92,19%.

Tabela 5.11: Resultados Experimento 3 do Caso de Uso 2

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	6	16
	Ciclo Normal	117	785

5.3.5 Experimento 4 do Caso de Uso 2

Este experimento utilizou o algoritmo *One-Class* SVM para compor a central de inteligência e realizar a predição dos ataques DDoS. Em comparação com a rede neural LSTM Autoencoder, o *One-Class* SVM pode consumir menos recursos para treinamento e predição dos ataques DDoS. **Assim, é o objetivo deste experimento é avaliar se o *One-Class* SVM pode ser uma alternativa ao LSTM *Autoencoder*.** Este trabalho selecionou o *One-Class* SVM devido ao fato deste algoritmo não necessitar de dados rotulados para realizar a predição de ataques DDoS (Muller et al., 2001) e poder funcionar em cenários desbalanceados (Devi et al., 2019).

Assim como no Experimento 1 deste caso de uso, este experimento usa a captura 51 da CTU-13. Assim como nos experimentos anteriores, este experimento desconsiderou o tráfego de rede após o lançamento do ataque (após o segundo 5631). *One-Class* SVM foi treinado usando um terço de toda a base de dados (até o segundo 2934). Os dados utilizados para o teste variam do segundo 2934 até 5631. Isso ocorre, pois o propósito da abordagem ESPA é predizer os ataques DDoS. Assim, a fase de teste varia até o segundo anterior ao lançamento do ataque. O

One-Class SVM usado nesse experimento usa todas as configurações padrão da biblioteca[11], exceto pelos parâmetros *nu* e $kernel$. A configuração que maximizou a acurácia e a predição dos ataques DDoS foi $kernel = poly$ e $nu = 0.34$. É oportuno ressaltar que esses hiperparâmetros foram obtidos empiricamente e o código-fonte está online[12].

Também é objetivo deste experimento utilizar a abordagem ESPA para combinar os três indicadores estatísticos com os três atributos do tráfego de rede utilizados no experimento anterior. Essa ação gera três características para cada atributo. Portanto, a central de inteligência da abordagem ESPA calcula o valor do *kurtosis*, *skewness* e CV (indicadores estatísticos) para o total de endereço IP de origem e destino e o total de pacotes para cada ciclo de tempo (atributos do tráfego de rede). Esse processo gera nove novas características, uma para cada combinação entre indicadores estatísticos e atributos do tráfego de rede. Assim, são geradas nove características distintas, onde as três primeiras correspondem ao valor de *Kurtosis*, *Skewness* e coeficiente de variação relacionada ao total de endereços IPs de origem. As três características seguintes registram esses valores em relação ao total de IPs de destino e as três últimas capturam esses valores em relação ao número total de pacotes. Por fim, também com o intuito de avaliar uma configuração alternativa em relação aos experimentos anteriores, a engenharia de sinais foi executada com uma janela deslizante de tamanho 880 (10% do tamanho total da base analisada).

Após a engenharia de sinais, a abordagem ESPA equipada com o *One-Class* SVM obteve os resultados apresentados na Tabela 5.12. Neste experimento, a abordagem errou a classificação de 82 (69 FPs e 13 FNs) dos 2698 segundos da base de teste. Os dois ciclos maliciosos foram corretamente identificados nos segundos 3770 e 3779. A acurácia da abordagem ESPA é de 96,96%. A precisão para a classe dos ciclos maliciosos é de 2,81% pois das 71 predições (2 VPs + 69 FPs) duas estavam corretas. Para os ciclos normais a precisão foi de 99,51% pois 13 ciclos maliciosos foram classificados como ciclos normais. O *recall* para a classe dos ciclos maliciosos é de 13,33% pois apenas 2 dos 15 ciclos maliciosos foram identificados corretamente, enquanto o *recall* para a classe dos ciclos normais é de 97,43% pois dos 2683 ciclos normais 2614 foram identificados corretamente. Por fim, o *F1-score* dos ciclos maliciosos é de 4,65% e o *F1-score* dos ciclos normais é de 98,46%.

Tabela 5.12: Resultados Experimento 4 do Caso de Uso 2

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	2	69
	Ciclo Normal	13	2614

5.3.6 Experimento 5 do Caso de Uso 2

Assim como no Experimento 2 deste caso de uso, **o objetivo específico deste experimento é avaliar a proposta em uma base de dados menor do que a utilizada anteriormente.** É importante analisar o comportamento da abordagem ESPA em uma base menor, pois menos dados disponíveis para o treinamento do *One-Class* SVM pode impactar nos resultados obtidos. Assim, é oportuno verificar como a abordagem ESPA irá operar frente a tal situação.

Assim como no Experimento 2 deste caso de uso, este experimento usa a captura 52 da CTU-13. Este experimento desconsiderou o tráfego de rede após o lançamento do ataque (após

[11] `https://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html`

[12] `https://github.com/andersonneira/tese_resultados/`

o segundo 778). *One-Class* SVM foi treinado usando um terço de toda a base de dados (até o segundo 324). Os dados utilizados para o teste variam dos segundos 324 até 777. Isso ocorre, pois o propósito da abordagem ESPA é predizer os ataques DDoS. A fase de teste varia até o segundo anterior ao lançamento do ataque. O *One-Class* SVM usado nesse experimento usa a configuração padrão da biblioteca[13], exceto pelos parâmetros *nu* e $kernel$. A configuração que maximizou a acurácia e a predição dos ataques DDoS foi $kernel = linear$ e $nu = 0.035$. Esses hiperparâmetros foram obtidos empiricamente e o código-fonte está online[14].

Assim como no Experimento 4, a engenharia de sinais combinou os todos os três indicadores estatísticos com todos os três atributos do tráfego de rede. Portanto, a central de inteligência da abordagem ESPA calcula o valor do *kurtosis*, *skewness* e CV (indicadores estatísticos) para o total de endereço IP de origem e destino e o total de pacotes para cada ciclo de tempo (atributos do tráfego de rede). Esse processo gera nove novas características, uma para cada combinação entre indicadores estatísticos e atributos do tráfego de rede. Assim, são geradas nove características distintas, onde as três primeiras correspondem ao valor de *Kurtosis*, *Skewness* e coeficiente de variação relacionada ao total de endereços IPs de origem. As três características seguintes registram esses valores em relação ao total de IPs de destino e as três últimas capturam esses valores em relação ao número total de pacotes. Por fim, este experimento utilizou uma janela deslizante de tamanho 10% do tamanho total da base analisada. Como a base possui 972 segundos, o tamanho da janela deslizante é 97.

A Tabela 5.13 apresenta os resultados obtidos após a engenharia de sinais e para a abordagem ESPA equipada com o *One-Class* SVM na base CTU-13 captura 52. Os resultados indicam que a abordagem errou a classificação de 29 dos 454 segundos da base de teste, sendo 19 FPs e 10 FNs. Os três ciclos maliciosos foram corretamente identificados nos segundos 576, 577 e 580. A acurácia da abordagem ESPA é de 93,61%. A precisão para a classe dos ciclos maliciosos é de 13,64% pois das 22 predições (3 VPs + 19 FPs) três estavam corretas. Para os ciclos normais a precisão foi de 97,69% pois dez ciclos maliciosos foram classificados como ciclos normais. O *recall* para a classe dos ciclos maliciosos é de 23,08% pois 3 dos 13 ciclos maliciosos foram identificados corretamente, enquanto o *recall* para a classe dos ciclos normais é de 95,69% pois dos 441 ciclos normais, 422 foram identificados corretamente. Por fim, o *F1-score* dos ciclos maliciosos é de 17,14% e o *F1-score* dos ciclos normais é de 96,68%.

Tabela 5.13: Resultados Experimento 5 do Caso de Uso 2

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	3	19
	Ciclo Normal	10	422

5.3.7 Experimento 6 do Caso de Uso 2

O objetivo específico deste experimento é avaliar a proposta em uma base de dados com uma topologia diferente do que a utilizada nos Experimentos 4 e 5 deste caso de uso. Nos experimentos anteriores, os *bots* estavam na rede local da vítima. Para complementar a análise da abordagem ESPA usando o *One-Class* SVM configurado manualmente, o objetivo

[13] https://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html

[14] https://github.com/andersonneira/tese_resultados/

específico deste experimento é avaliar a abordagem proposta onde os *bots* não estejam alocados na mesma rede que a vítima.

Este experimento utilizou a captura do tráfego de rede do primeiro ataque do primeiro dia da base de dados CICDDoS2019. O ataque começa no segundo 1484 da captura e possui duração de 540 segundos. O experimento utilizou um terço do tráfego total da captura para realizar o treinamento do modelo (até o segundo 673) e o restante de tráfego prévio ao ataque para avaliação (do segundo 674 até o segundo 1483). Portanto, a base de teste possui 810 segundos. O *One-Class* SVM usado nesse experimento usa todas as configurações padrão da biblioteca[15], exceto pelos parâmetros *nu* e $kernel$. A configuração que maximizou a acurácia e a predição dos ataques DDoS foi $kernel = poly$ e $nu = 0.13$. É oportuno ressaltar que esses hiperparâmetros foram obtidos empiricamente e o código-fonte está online[16].

Assim como nos experimentos anteriores, a engenharia de sinais combinou os três indicadores estatísticos com os três atributos do tráfego de rede. Portanto, a central de inteligência da abordagem ESPA calcula o valor do *kurtosis*, *skewness* e CV (indicadores estatísticos) para o total de endereço IP de origem e destino e o total de pacotes para cada ciclo de tempo (atributos do tráfego de rede). Esse processo gera nove novas características, uma para cada combinação entre indicadores estatísticos e atributos do tráfego de rede. Por fim, este experimento utilizou uma janela deslizante de tamanho 10% do tamanho total da base analisada. Como a base possui 2024 segundos, o tamanho da janela deslizante é 202.

A Tabela 5.14 apresenta os resultados obtidos após a engenharia de sinais e com a abordagem ESPA usando o *One-Class* SVM na predição do primeiro ataque da base CICDDoS2019. Os resultados indicam que a abordagem ESPA errou a classificação de 133 (28 FPs e 105 FNs) dos 810 segundos da base de teste. Os três ciclos maliciosos foram corretamente identificados nos segundos 917, 918 e 921. A acurácia da abordagem ESPA é de 83,58%. A precisão para a classe dos ciclos maliciosos é de 9,68% pois das 31 predições (3 VPs + 28 FPs) três estavam corretas. Para os ciclos normais a precisão foi de 86,52% pois 105 ciclos maliciosos foram classificados como ciclos normais. O *recall* para a classe dos ciclos maliciosos é de 2,78% pois 3 dos 108 ciclos maliciosos foram identificados corretamente, enquanto o *recall* para a classe dos ciclos normais é de 96,01% pois dos 702 ciclos normais, 674 foram identificados corretamente. Por fim, o *F1-score* dos ciclos maliciosos é de 4,32% e o *F1-score* dos ciclos normais é de 91,02%.

Tabela 5.14: Resultados Experimento 6 do Caso de Uso 2

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	3	28
	Ciclo Normal	105	674

5.3.8 Discussão dos Resultados do Caso de Uso 2

Este caso de uso apresentou seis diferentes experimentos com o objetivo geral de reforçar a verificação da hipótese e apresentar outros modos de uso da abordagem ESPA. Sem usar dados rotulados, a abordagem ESPA predisse os ataques DDoS em três conjuntos de dados diferentes. Os Experimentos 1, 2, 4, e 5 avaliaram a predição dos ataques DDoS onde os *bots* estavam na mesma rede da vítima e os Experimentos 3 e 6 avaliaram a predição onde a Internet conectava

[15] https://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html

[16] https://github.com/andersonneira/tese_resultados/

a vítima com os *bots*. Os resultados dos experimentos indicam que a acurácia variou entre 83,58% e 98,03%. Assim, os resultados reforçam a validade da hipótese definida neste trabalho e auxiliam a avaliar a abordagem ESPA em diferentes configurações.

A Figura 5.16 apresenta o tempo antes do lançamento do ataque que a abordagem ESPA obteve nos Experimentos 1 e 4. Estes experimentos avaliaram a captura 51 da base de dados CTU-13. Nesta base de dados, o ataque foi lançado no segundo 5632. No Experimento 1, que usa três novas características geradas pela engenharia de sinais e o LSTM *Autoencoder*, a abordagem proposta predisse o ataque 29 minutos e 51 segundos antes do lançamento do ataque (diamante na Figura 5.16). Usando nove novas características e o *One-Class* SVM (círculo preenchido na Figura 5.16), a predição do mesmo ataque ocorreu 31 minutos e 2 segundos antes do lançamento do ataque. Esses resultados indicam que a versão da abordagem ESPA que gera mais características e prediz o ataque usando o *One-Class* SVM conseguiu prover mais tempo antes do lançamento do ataque. Contudo, esta versão cometeu mais erros do que a versão da abordagem com a LSTM *Autoencoder*. Esses erros, principalmente os FPs, afetam negativamente a utilização da abordagem proposta. A versão da abordagem ESPA com o *One-Class* SVM (Experimento 4) gerou 69 alertas onde os *bots* não enviaram ou receberam mais de dois pacotes. Já a versão da abordagem ESPA com o LSTM *Autoencoder* (Experimento 1) gerou 20.

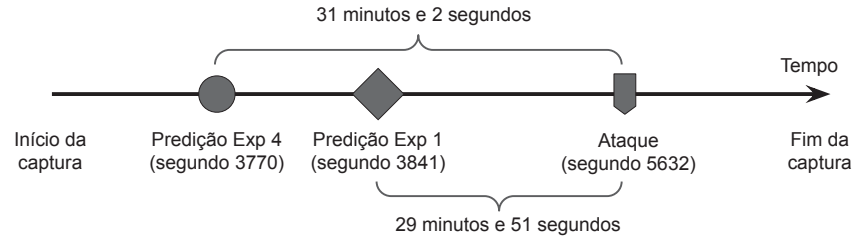


Figura 5.16: Predição do Ataque DDoS nos Experimentos 1 e 4 (CTU-13 Captura 51)

A Figura 5.17 apresenta o tempo de predição para os Experimentos 2 e 5, pois eles avaliaram a mesma base de dados. Estes experimentos realizaram a predição do ataque DDoS em uma base de dados com menos dados disponíveis para treinar as soluções quando comparados aos dados disponíveis nos Experimentos 1 e 4. Isso impacta tanto o tempo de predição quanto as métricas de avaliação (acurácia, precisão, *recall* e *F1-score*). A abordagem ESPA equipada com a rede neural LSTM *Autoencoder* (Experimento 2) predisse o ataque 4 minutos e 24 segundos antes do lançamento do ataque. Enquanto a versão da abordagem equipada com o *One-Class* SVM (Experimento 5) predisse o ataque com 3 minutos e 22 segundos. Assim, diferentemente dos Experimentos 1 e 4, a versão da abordagem ESPA equipada com o *One-Class* SVM predisse o ataque depois da versão equipada com o LSTM *Autoencoder*. A acurácia da abordagem ESPA com *One-Class* SVM foi de 93,61% com 19 FPs. A versão da abordagem ESPA equipada com a rede LSTM *Autoencoder* obteve uma acurácia de 94,7% com 11 FPs.

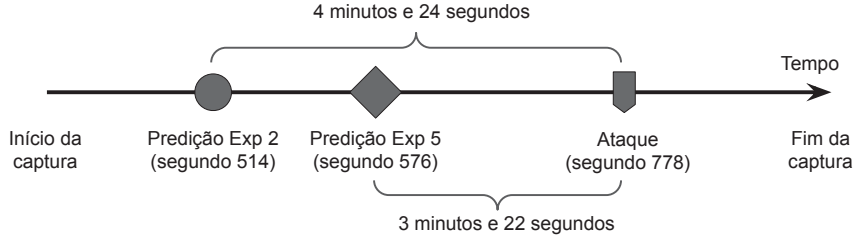


Figura 5.17: Predição do Ataque DDoS nos Experimentos 2 e 5 (CTU-13 Captura 52)

A Figura 5.18 apresenta os tempos de predição obtidos nos Experimentos 3 e 6. A base de dados avaliada nestes experimentos é maior do que a captura 52 da CTU-13 (Experimentos 2 e 5) e é menor do que a captura 51 da CTU-13 (Experimentos 1 e 4). A versão da abordagem ESPA equipada com a rede LSTM *Autoencoder* (Experimento 3) superou a versão equipada com o *One-Class* SVM (Experimento 6) em tempo de predição e nas métricas de avaliação. Enquanto a primeira versão predisse o ataque com 15 minutos e 23 segundos, a versão com o *One-Class* SVM predisse com 9 minutos e 26 segundos. A acurácia da abordagem com a rede neural LSTM *Autoencoder* obteve uma acurácia de 85,61% e 16 FPs. Enquanto a versão com o *One-Class* SVM obteve 83,58% de acurácia com 28 FPs.

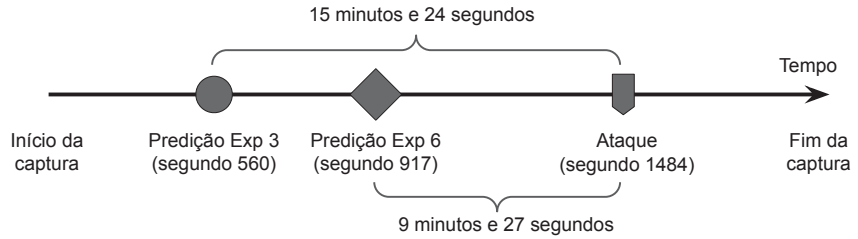


Figura 5.18: Predição do Ataque DDoS nos Experimentos 3 e 6 (CICDDoS2019)

Os resultados indicam que a versão da abordagem ESPA equipada com a rede neural LSTM *Autoencoder* configurada autonomamente pelo AutoML é mais acurada do que a versão com o *One-Class* SVM. Em um dos três experimentos, a versão com o *One-Class* SVM predisse o ataque antes. Contudo, os erros de predição, principalmente os FPs, representam pontos de atenção para essa versão da abordagem ESPA. Este trabalho hipotetiza que a abordagem ESPA equipada com a rede neural LSTM *Autoencoder* obteve menos erros em relação à versão com o *One-Class* SVM devido a autoconfiguração do modelo. O *One-Class* SVM foi configurado manualmente. Assim, é possível existir uma combinação de hiperparâmetros que melhore os resultados obtidos. Principalmente quando comparado com a rede neural configurada autonomamente pelo AutoML. Pois a configuração identificada pelo AutoML extrai toda a capacidade do *deep larning* e favorece a versão da abordagem ESPA configurada com a rede neural LSTM *Autoencoder*. Atualmente, pesquisas como a de Araújo et al. (2022) visam reconstruir os modelos sempre que necessário. Assim, a abordagem ESPA usa o sobreajuste da rede LSTM *Autoencoder* como uma virtude ao invés de uma limitação.

Mesmo a versão da abordagem ESPA equipada com a rede LSTM *Autoencoder* sendo vantajosa, esta também possui limitações. Em relação ao uso do limiar nos Experimentos 1, 2 e 3 o mais importante é que a rede LSTM *Autoencoder* treinada pelo AutoML conseguiu apresentar valores de erros de reconstrução em momentos onde *bots* trafegavam dados. Assim, mesmo definindo manualmente o limiar, a hipótese deste trabalho pode ser verificada também com esses resultados. Para os objetivos atuais deste trabalho, esse é o aspecto mais importante. Além disso, o tempo de seleção da arquitetura da rede neural e o seu treinamento são pontos de atenção. Usando o Google Colaboratory como ambiente de testes, a abordagem ESPA utilizou aproximadamente 42, 34 e 28 minutos para configurar e treinar as redes neurais LSTM *Autoencoder* nos Experimentos 1, 2 e 3 respectivamente. Este tempo foi investido pelo AutoML para selecionar as redes neurais que obtiveram os resultados apresentados. A literatura indica que esse processo pode levar mais tempo. Lam e Abbas (2020) citam que os *frameworks* NASNet e MNasNet podem demandar entre 3 e 24 horas para identificar as arquiteturas ideais. Além disso, é plausível hipotetizar que especialistas demandem entre 28 e 42 minutos para identificar uma arquitetura de rede neural adequada ao problema. Assim, os resultados obtidos indicam que

a proposta pode demandar menos tempo de processamento que a literatura e pode competir com especialistas humanos.

O *One-Class* SVM não necessita de limiar de predição, isso facilita o uso dessa versão da abordagem ESPA. Contudo, os resultados do *One-Class* são mais difíceis de interpretar em comparação ao LSTM *Autoencoder*. O *One-Class* SVM pode lidar com dados de alta dimensionalidade. Representar graficamente isso pode ser um desafio. Enquanto isso, com o custo de reconstrução obtido com o LSTM *Autoencoder* é possível representar graficamente os resultados. Os gráficos representados nas Figuras 5.13, 5.14 e 5.15 ilustram isso. Além de usar apenas uma linha para representar a média do custo de reconstrução (MAE), é possível obter a diferença absoluta entre o real e o predito para todas as características. Como nos Experimentos 1, 2 e 3 a engenharia de sinais gerou três novas características, é possível gerar um gráfico com três linhas. Onde cada linha representa a diferença absoluta para cada nova característica. Isso facilita a interpretação dos resultados em comparação com a versão da abordagem ESPA equipada com o *One-Class* SVM.

5.3.9 Comparação dos Resultados do Caso de uso 2 com a Literatura

A Tabela 5.15 resume todos os resultados obtidos nos Casos de uso 1 e 2 e os compara com o sistema ANTE e com o trabalho de Rahal et al. (2020). O sistema ANTE e o trabalho de Rahal et al. (2020) realizaram a predição do ataque na captura 51 da base de dados CTU-13. O sistema ANTE predisse com até 1 minuto de antecedência e o trabalho de Rahal et al. (2020) predisse o ataque com 5 minutos e 41 segundos. Os Experimentos 1 e 4 do Caso de uso 2 superam o tempo de predição de ambos os trabalhos. O melhor tempo de predição para a captura 51 foi de 31 minutos e 2 segundos obtido no Experimento 4. Como discutido na subseção anterior, a maior acurácia para a captura 51 foi obtida no Experimento 1 deste caso de uso.

É oportuno destacar que os resultados obtidos no Caso de uso 2 também são competitivos frente ao Caso de uso 1. As duas versões da abordagem ESPA do Caso de uso 2 que avaliaram a captura 52 da base de dados CTU-13 superaram as versões do Caso de uso 1 em questão de tempo de predição. A abordagem ESPA equipada com o LSTM *Autoencoder* foi a versão da abordagem que maximizou o tempo de predição na captura 52. A versão da abordagem ESPA usando o Adaboost obtida no Experimento 1 do Caso de uso 1 ainda é a versão da abordagem proposta que maximiza acurácia para a captura 52. O Adaboost maximizar a acurácia pode ser explicado pelo fato deste ter sido treinado usando dados rotulados. Por outro lado, tanto o *One-Class* SVM quanto o LSTM *Autoencoder* não usam dados rotulados durante o treinamento. Em relação ao cenário *Portmap* da base de dados CICDDoS2019 a versão da abordagem ESPA que maximiza o tempo de predição é a equipada com o LSTM *Autoencoder* configurado autonomamente. O tempo de predição de 15 minutos e 23 segundos e a acurácia de 85,61% superam a versão da abordagem ESPA com o *One-Class* SVM e o sistema ANTE.

5.4 CASO DE USO 3 - CLUSTERIZAÇÃO, EXPLICABILIDADE E AUTOCOFIGURAÇÃO

Nos Casos de uso 1 e 2, a hipótese definida neste trabalho foi verificada, pois a abordagem proposta predisse os ataques DDoS usando novas características geradas pela engenharia de sinais. Este caso de uso também tem o objetivo geral de reforçar a verificação da hipótese. Contudo, este caso de uso difere-se dos anteriores ao apresentar um modo para utilizar a abordagem ESPA focada na explicabilidade dos resultados sem a utilização de nenhum tipo de rótulo dos dados. Além disso, este caso de uso avalia a funcionalidade de selecionar autonomamente o algoritmo de aprendizado de máquina não supervisionado que irá maximizar a acurácia da predição dos ataques

Tabela 5.15: Resumo da Comparação Entre a abordagem ESPA e a Literatura no Caso de Uso 2

Experimento	Base de dados	Referência	Tempo predição	Acurácia
Experimento 1 (**Caso de uso 1**)	Cap. 52 (CTU-13)	ESPA *(Adaboost)*	3 min e 22 seg	**99,69%**
Experimento 2 (**Caso de uso 1**)	UDP-Lag (CICDDoS2019)	ESPA *(Adaboost)*	2 min e 8 seg	**99,76%**
Experimento 3 (**Caso de uso 1**)	Cap. 52 (CTU-13)	ESPA *(MLP)*	3 min e 49 seg	98,87%
Experimento 4 (**Caso de uso 1**)	UDP-Lag (CICDDoS2019)	ESPA *(MLP)*	**3 min e 55 seg**	99,60%
Experimento 1 (**Caso de uso 2**)	Cap. 51 (CTU-13)	ESPA *(LSTM Autoencoder)*	29 min e 51 seg	**98,03%**
Experimento 2 (**Caso de uso 2**)	Cap. 52 (CTU-13)	ESPA *(LSTM Autoencoder)*	**4 min e 24 seg**	94,7%
Experimento 3 (**Caso de uso 2**)	*Portmap* (CICDDoS2019)	ESPA *(LSTM Autoencoder)*	**15 min e 23 seg**	**85,61%**
Experimento 4 (**Caso de uso 2**)	Cap. 51 (CTU-13)	ESPA *(One-Class SVM)*	**31 min e 2 seg**	96,96%
Experimento 5 (**Caso de uso 2**)	Cap. 52 (CTU-13)	ESPA *(One-Class SVM)*	3 min e 22 seg	93,61%
Experimento 6 (**Caso de uso 2**)	*Portmap* (CICDDoS2019)	ESPA *(One-Class SVM)*	9 min e 26 seg	83,58%
-	Capturas 51 e 52 (CTU-13) e *Portmap* (CICDDoS2019)	Sistema ANTE	1 min e 00 seg	-
-	Capturas 51 (CTU-13))	Rahal et al. (2020)	5 min e 41 seg	-

DDoS (Seção 4.3). Por fim, este caso de uso começa a exploração da seleção de características sem o uso de dados rotulados.

5.4.1 Definição dos Experimentos do Caso de uso 3

O Caso de uso 3 possui quatro experimentos que avaliam a predição de ataques DDoS para diferentes configurações da abordagem ESPA. Cada experimento possui três avaliações com diferentes objetivos. Em cada avaliação a abordagem ESPA é submetida a variação dos parâmetros. A Tabela 5.16 apresenta todos os parâmetros usados pela abordagem ESPA. Neste caso de uso, a abordagem ESPA coletou centralizadamente o tráfego de rede da vítima do ataque DDoS usando um agente. Essa configuração é a mesma para todas as avaliações dos quatro experimentos. Todos os experimentos deste caso de uso foram conduzidos em um dispositivo que possui um processador Intel Core I5, um disco rígido de um terabyte, oito gigabytes de memória, com um sistema operacional Linux Mint e com Python 3.10.12 e os resultados estão disponíveis online[17].

[17] https://github.com/andersonneira/tese_resultados/

Tabela 5.16: Definição dos Parâmetros Usados Pela Abordagem ESPA no Caso de Uso 3

Parâmetro	**Valor**
Modo de coleta do tráfego de rede	Centralizado
Quantidade de agentes	Um
Localização dos agentes	Na rede vítima
Atributos do tráfego de rede	51 atributos baseados nos protocolos das camadas de Enlace, Rede e Transporte do modelo TCP/IP
Ciclos de captura	Um segundo
Indicadores estatísticos	*kurtosis*, *skewness*, CV, AC-1, AC-2 e AC-3
Algoritmos de aprendizado de máquina	K-*Means, Birch, Bisecting K-Means, Gaussian Mixture, Agglomerative Clustering*, *SOM, DBSCAN, OPTICS e HDBSCAN*
Seleção dos algoritmos de aprendizado de máquina	Manual e Automática
Janela deslizante	5% e 10% da base de dados
Tratamento de dados faltantes	Nenhum
Redimensionamento dos dados	Nenhum

O primeiro diferencial deste caso de uso é a explicabilidade dos resultados e a predição dos ataques DDoS sem o uso de dados rotulados. A explicabilidade é obtida ao expressar visualmente, em forma de gráficos de três dimensões, os resultados da clusterização dos dados trafegados na rede dos usuários. Isso permite a compreensão da predição dos ataques DDoS e propicia a tomada de decisão por parte dos administradores. Além disso, a abordagem proposta facilita a adoção em ambientes reais, ao evitar o uso de dados rotulados para a predição de ataques DDoS. Portanto, a abordagem ESPA não fica limitada a predizer ataques DDoS similares aos ataques que foi treinada para isso. Assim, a abordagem ESPA pode predizer ataques até então desconhecidos como os ataques dia zero (do inglês, *zero-day attacks*).

Além de focar na explicabilidade, as avaliações deste caso de uso realizam a seleção autônoma da técnica de aprendizado de máquina não supervisionada ideal para o contexto analisado. Para isso, a abordagem ESPA foi equipada com o Algoritmo 3 (Seção 4.3). No Caso de uso 2, a abordagem ESPA foi equipada com o *framework* Autokeras para identificar a rede neural que maximiza a predição dos ataques DDoS. No Caso de uso 3 a abordagem proposta foi equipada com um algoritmo desenvolvido especialmente para a abordagem ESPA (Algoritmo 3). Esse algoritmo analisa um conjunto de dados para escolher a técnica ideal para realizar a predição dos ataques DDoS. Os detalhes do Algoritmo 3 estão descritos na Seção 4.3.

A Tabela 5.17 apresenta todos os 51 atributos usados neste caso de uso para a avaliação da seleção de atributos do tráfego de rede sem o uso de dados rotulados. Esses atributos são baseados nos campos dos protocolos das camadas de enlace, rede e transporte do modelo TCP/IP. Os atributos de 1 a 3 são obtidos por meio da análise dos protocolos da camada de enlace como o Ethernet. Os atributos entre 4 e 17 são obtidos do protocolo IP (camada de rede). Os atributos entre 18 e 36 são extraídos dos protocolos TCP e UDP (camada de transporte) (Kurose e Ross, 2014). Os atributos entre 37 e 41 fazem referência ao tempo de chegada de pacotes e os atributos entre 42 e 51 especializam a análise do tempo de chegada entre pacotes para o protocolo TCP. É oportuno destacar que este caso de uso não utiliza dados da carga útil dos pacotes, apenas dados dos cabeçalhos foram considerados neste caso de uso.

Na Tabela 5.17, os atributos quantidade de IPs de origem, quantidade de IPs de destino e o total de pacotes foram destacados por serem os mesmos utilizados no caso de uso anterior. A engenharia de sinais de algumas avaliações dos experimentos usam esses três atributos para a predição dos ataques DDoS. É importante destacar que, esses atributos foram obtidos por unidade de tempo (ciclo de captura). Em geral, o tempo usado nos experimentos é um segundo. Contudo, cada experimento detalha o ciclo de captura utilizado, bem como o tamanho da janela deslizante (5% ou 10% da base de dados) utilizado na engenharia de sinais. O objetivo de não predefinir o ciclo de captura e o tamanho da janela deslizante é poder avaliar a abordagem ESPA em casos diferentes respeitando as características inerentes às bases de dados. Como este caso de uso aplica a seleção de características e a seleção dos algoritmos de aprendizado de máquina, nenhuma ação adicional foi realizada com intuito de tratar ou de redimensionar os dados durante os experimentos.

Tabela 5.17: Definição dos Atributos do Tráfego de Rede Usados Pela Abordagem ESPA no Caso de Uso 3

Atributo	**Descrição**
1. total_mac_origem	Total de endereços MAC de origem únicos
2. total_mac_destino	Total de endereços MAC de destino únicos
3. versao_ip	Versão mais frequente do protocolo que estará na carga útil
4. maior_pacote	Tamanho do maior pacote
5. menor_pacote	Tamanho do menor pacote
6. soma_pacotes	Soma do tamanho dos pacotes
7. total_pacotes_icmp	Total de pacotes do tipo ICMP
8. total_pacotes_udp	Total de pacotes do tipo UDP
9. total_pacotes_tcp	Total de pacotes do tipo TCP
10. total_ips_origem	Total de endereços IP de origem únicos
11. total_ips_destino	Total de endereços IP de destino únicos
12. maior_ttl	Maior *time to live* dos pacotes
13. menor_ttl	Menor *time to live* dos pacotes
14. soma_ttl	Soma do *time to live* dos pacotes
15. std_ttl	Desvio padrão do *time to live* dos pacotes
16. media_ttl	Média do *time to live* dos pacotes
17. total_pacotes	Total de pacotes trafegados na rede
18. porta_origem_mais_frequente	Porta de origem mais comum encontrada nos pacotes
19. porta_destino_mais_frequente	Porta de destino mais comum encontrada nos pacotes
20. total_flags_tcp	Total de *flags* TCP únicas
21. total_tcp_flags_fin	Total de *flags* TCP FIN
22. total_tcp_flags_syn	Total de *flags* TCP SYN
23. total_tcp_flags_reset	Total de *flags* TCP RST
24. total_tcp_flags_push	Total de *flags* TCP PUSH
25. total_tcp_flags_ack	Total de *flags* TCP ACK
26. total_tcp_flags_urg	Total de *flags* TCP URG
27. maior_tcp_window_size_value	Maior valor do campo TCP *window size*
28. menor_tcp_window_size_value	Menor valor do campo TCP *window size*
29. soma_tcp_window_size_value	Soma do valor do campo TCP *window size*
30. std_tcp_window_size_value	Desvio padrão do valor do campo TCP *window size*
31. media_tcp_window_size_value	Média do valor do campo TCP *window size*

32. maior_tcp_seq_num	Maior valor do campo TCP *Sequence Number*
33. menor_tcp_seq_num	Menor valor do campo TCP *Sequence Number*
34. soma_tcp_seq_num	Soma do valor do campo TCP *Sequence Number*
35. std_tcp_seq_num	Desvio padrão do valor do campo TCP *Sequence Number*
36. media_tcp_seq_num	Média do valor do campo TCP *Sequence Number*
37. maior_time_delta	Maior tempo de chegada entre pacotes
38. menor_ttime_delta	Menor tempo de chegada entre pacotes
39. soma_time_delta	Soma do tempo de chegada entre pacotes
40. std_time_delta	Desvio padrão do tempo de chegada entre pacotes
41. media_time_delta	Média do tempo de chegada entre pacotes
42. maior_tcp_time_delta	Maior tempo de chegada entre pacotes TCP do mesmo fluxo
43. menor_tcp_time_delta	Menor tempo de chegada entre pacotes TCP do mesmo fluxo
44. soma_tcp_time_delta	Soma do tempo de chegada entre pacotes TCP do mesmo fluxo
45. std_tcp_time_delta	Desvio padrão do tempo de chegada entre pacotes TCP do mesmo fluxo
46. mean_tcp_time_delta	Médio do tempo de chegada entre pacotes TCP do mesmo fluxo
47. maior_tcp_time_relative	Maior tempo de chegada desde o primeiro pacote TCP do mesmo fluxo
48. menor_tcp_time_relative	Menor tempo de chegada desde o primeiro pacote TCP do mesmo fluxo
49. soma_tcp_time_relative	Soma do tempo de chegada desde o primeiro pacote TCP do mesmo fluxo
50. std_tcp_time_relative	Desvio padrão do tempo de chegada desde o primeiro pacote TCP do mesmo fluxo
51. mean_tcp_time_relative	Média do tempo de chegada desde o primeiro pacote TCP do mesmo fluxo

A exploração da redução de dimensionalidade sem o uso de dados rotulados também é uma definição dos experimentos deste caso de uso. O intuito dessa exploração é diminuir o total de erros obtidos anteriormente utilizando diferentes características. Nos casos de uso anteriores, a seleção de características foi realizada manualmente. Portanto, em relação aos casos de uso anteriores, o Caso de uso 3 é o único que lida com a seleção de características automatizadas. Contudo, esse tema tem grande potencial para melhorar a predição dos ataques DDoS. Assim, o Caso de uso 4 e trabalhos futuros explorarão outros algoritmos e outras abordagens para a seleção de características.

O algoritmo de redução de dimensionalidade usado neste caso de uso foi o *FastICA*. O *FastICA* é uma implementação computacionalmente eficiente do *Independent component analysis* (ICA) (Hyvärinen e Oja, 2000). O ICA objetiva encontrar uma nova representação dos dados, buscando tornar os componentes (novas características) estatisticamente independentes. Essa representação visa capturar a estrutura essencial dos dados em diversas aplicações, como extração de características e separação de sinais (Hyvärinen e Oja, 2000). Em outras palavras, o propósito

do ICA é identificar sinais provenientes de fontes distintas em meio a sinais combinados (Li et al., 2014). Classicamente, o ICA é utilizado para separar sinais sobrepostos em diversas áreas como telecomunicações, processamento de sinais biomédicos, imagens e áudio (Hyvärinen e Oja, 2000). Contudo, o ICA também pode ser utilizado para a redução de dimensionalidade sem o uso dos rótulos reais. A biblioteca Scikit-learn apresenta uma implementação do algoritmo *FastICA* para Python[18]. Ao usar o FastICA, esse trabalho tem o objetivo de reduzir a dimensionalidade das novas características, encontrando a estrutura essencial dos dados. Ao reduzir a dimensionalidade dos dados, novas características são geradas para substituir as originais.

A última definição dos experimentos deste caso de uso é a definição de ciclos maliciosos e ciclos normais. Assim como no caso de uso anterior, os experimentos deste caso de uso não usam dados rotulados para treinar a central de inteligência da abordagem ESPA. Porém, é necessário rotular os dados para quantificar os acertos e os erros da abordagem ESPA. Este caso de uso leva o processo de rotulação ao extremo. Pois, os experimentos deste caso de uso consideram como **ciclo malicioso** todos os ciclos de captura onde os ***bots* enviaram ao menos um pacote**. Assim, os ciclos de um segundo do tráfego de rede que representam a preparação de ataque foram impactados por ações conduzidas pelos *bots* antes do lançamento do ataque. Por fim, os **ciclos normais** são os segundos do tráfego de rede onde os ***bots* não enviam pacotes**.

5.4.2 Experimento 1 do Caso de Uso 3

O Experimento 1 realizou três avaliações diferentes, cada qual com um objetivo específico, além de reforçar a verificação da hipótese definida neste trabalho. Todas elas utilizam o tráfego de rede coletado na captura 51 disponibilizado pela base de dados CTU-13. Por esse motivo, todas as avaliações foram agrupadas neste primeiro experimento. Os pesquisadores infectaram os *bots* no segundo 2643 e lançaram os ataques no segundo 5632 da captura.

A engenharia de sinais foi realizada de dois modos diferentes. A abordagem ESPA realizou a engenharia de sinais par a par com as séries temporais dos atributos do tráfego de rede coletados em ciclos de um segundo para as avaliações 1 e 2. O *kurtosis* foi atribuído para o atributo quantidade de IPs de origem, o *skewness* foi calculado para a quantidade de IPs de destino e o CV foi calculado para o total de pacotes. Na avaliação 3 (Sub-subseção 5.4.2.3) a engenharia de sinais foi realizada para todas as séries temporais dos 51 atributos apresentados na Tabela 4.2 (Seção 4.3) usando seis indicadores estatísticos. Sendo eles: o *skewness*, o *kurtosis*, o CV, a AC-1, AC-2 e AC-3. Assim, 306 novas características foram geradas pela engenharia de sinais da abordagem ESPA. Para diferenciar do Experimento 1 do Caso de uso 2, este experimento utilizou 10% da base de dados como o tamanho da janela deslizante de tamanho fixo. A seguir, cada avaliação deste caso de uso apresenta os objetivos específicos e os resultados obtidos.

5.4.2.1 Avaliação 1 do Experimento 1

O objetivo desta avaliação é verificar se a abordagem ESPA pode predizer ataques DDoS usando uma técnica de aprendizado de máquina não supervisionada, focando na explicabilidade dos resultados. O *K-means* foi escolhido e configurado manualmente para predizer os ataques DDoS. O *K-means* foi configurado usando todas as opções padrão da biblioteca Scikit-learn, exceto pela quantidade de clusters. Neste caso, o *K-means* foi configurado para obter dois clusters (1º algoritmo da Tabela 4.2 - Seção 4.3). Pois era esperado que um cluster fosse composto pelos ciclos maliciosos e outro cluster fosse composto por ciclos normais.

[18] `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.FastICA.html`

A Figura 5.19(a) apresenta o resultado da abordagem ESPA entre os segundos 3294 e 3794 da base de dados. A figura apresenta a existência de dois grupos de ciclo de tempo de um segundo. Ambos os grupos possuem ciclos normais e ciclos maliciosos. A Figura 5.19(b) apresenta os resultados do *K-means* considerando o grupo com mais ciclos e com os ciclos mais antigos, próximos do segundo 3294, como o grupo dos ciclos normais (Grupo 1). O grupo dos ciclos maliciosos, Grupo 2, é formado pelo grupo que possui menos ciclos e os mais recentes. O Grupo 1 possui 413 ciclos normais corretamente identificados e 32 ciclos maliciosos erroneamente considerados normais. O Grupo 2 agrega 10 ciclos maliciosos corretamente identificados como maliciosos e 45 ciclos normais erroneamente considerados maliciosos.

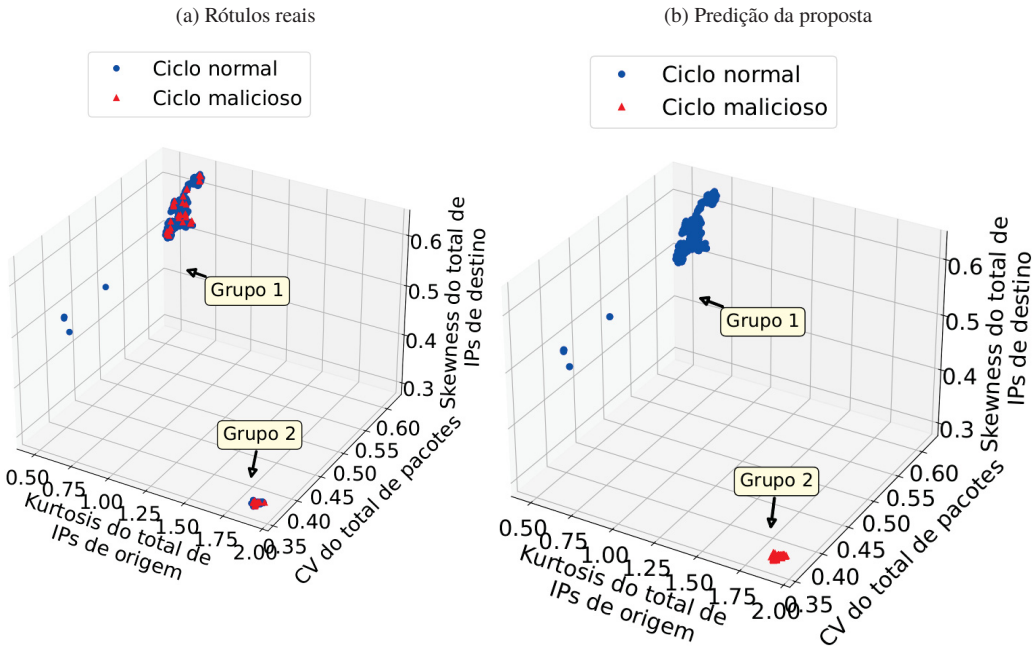


Figura 5.19: Predição de Ataques DDoS no Experimento 1 do Caso de Uso 3

A Tabela 5.18 apresenta os resultados desta avaliação. O *K-means* configurado manualmente obteve uma acurácia de 84,6%. A precisão para a classe dos ciclos maliciosos é de 18,18% pois das 55 predições (10 VPs + 45 FPs) 10 estavam corretas. Para os ciclos normais a precisão foi de 92,80% pois dos 445 ciclos classificados como normais, 413 foram classificados corretamente. O *recall* para a classe dos ciclos maliciosos é de 23,80% pois 10 dos 42 ciclos maliciosos foram identificados corretamente, enquanto o *recall* para a classe dos ciclos normais é de 90,17% pois dos 458 ciclos normais, 413 foram identificados corretamente. Por fim, o *F1-score* dos ciclos maliciosos é de 20,62% e o *F1-score* dos ciclos normais é de 91,47%.

Tabela 5.18: Resultados da Avaliação 1 do Experimento 1 do Caso de Uso 3

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	10	45
	Ciclo Normal	32	413

5.4.2.2 Avaliação 2 do Experimento 1

Esta avaliação verifica se a abordagem ESPA equipada com o Algoritmo 3 (Seção 4.3) pode selecionar um algoritmo de aprendizado de máquina não supervisionado autonomamente. Na avaliação anterior, a abordagem ESPA conseguiu predizer o ataque DDoS utilizando o algoritmo de aprendizado de máquina configurado manualmente. Nesta avaliação, é esperado que a abordagem ESPA possa identificar um algoritmo capaz de predizer ataques DDoS superando ou, no mínimo, mantendo os mesmos resultados em termos de métricas de avaliação (acurácia, precisão e *recall*) da avaliação anterior.

Ao executar a versão padrão do Algoritmo 3 com os mesmos dados utilizados na avaliação anterior, entre os segundos 3294 e 3794, a abordagem ESPA obteve como resultado quatro modelos sugeridos. Eles são: *Bisecting K-Means* com dois clusters e *bisecting_strategy* = *"biggest_inertia"* (16º na Tabela 4.2), *Bisecting K-Means* com dois clusters e *bisecting_strategy* = *"largest_cluster"* (20º na Tabela 4.2), *Birch* com dois clusters e *threshold*=0,4 (12º na Tabela 4.2) e o *Birch* com dois clusters e *threshold*=0,3 (11º na Tabela 4.2). Aplicando qualquer um dos modelos aos dados da avaliação anterior, a abordagem ESPA reproduz autonomamente os resultados obtidos na avaliação anterior. Assim, os grupos formados por qualquer um dos quatro modelos são os mesmos apresentados na Figura 5.19(b) e a matriz de confusão apresentada pela Tabela 5.18 também ilustra os acertos e erros obtidos nesta avaliação. Portanto, para este conjunto de dados, a abordagem ESPA equipada com o Algoritmo 3 foi capaz de autonomamente identificar quatro modelos de aprendizado de máquina que predizem o ataque DDoS com a mesma acurácia do modelo selecionado manualmente.

5.4.2.3 Avaliação 3 do Experimento 1

A última avaliação deste experimento explora a redução de dimensionalidade sem o uso de dados rotulados juntamente com a seleção autônoma dos algoritmos de aprendizado de máquina não supervisionado. Assim como citado anteriormente, o objetivo de diminuir a dimensionalidade dos dados nesta avaliação é diminuir o total de erros obtidos anteriormente utilizando diferentes atributos do tráfego de rede. O algoritmo de redução de dimensionalidade utilizado neste trabalho foi o *FastICA* (apresentado na Subseção 5.4.1). Nesta avaliação, a engenharia de sinais aplicou seis indicadores estatísticos (o *skewness*, o *kurtosis*, o CV, a AC-1, a AC-2 e a AC-3) sobre 51 atributos do tráfego de rede (Tabela 5.17). Esse processo originou 306 novas características que o algoritmo *FastICA* recebeu como entrada para realizar a seleção de características. O algoritmo *FastICA* necessita que a quantidade de sinais a ser decomposta seja informada. O valor escolhido nesta avaliação foi quatro, pois este maximizou os resultados obtidos. Portanto, o algoritmo *FastICA* recebe como entradas as 306 novas características e decompõe os dados em quatro sinais que serão usados para predizer os ataques DDoS.

Após a decomposição das características, esta avaliação aplicou a abordagem ESPA equipada com o algoritmo de seleção de algoritmos de aprendizado de máquina não supervisionado (Algoritmo 3). Contudo, visando avaliar uma versão não padrão do Algoritmo 3, este caso de uso utilizou o Algoritmo 3 apenas com o índice CDbw ao invés da média dos cinco índices. Além disso, para proporcionar uma comparação justa com as avaliações anteriores, o conjunto de dados analisados nesta avaliação foi o mesmo das Avaliações 1 e 2 deste experimento. Assim, esta avaliação utiliza os dados entre os segundos 3294 e 3794 da captura 51 da base de dados CTU-13. A abordagem ESPA equipada com o algoritmo de seleção de algoritmos de aprendizado de máquina não supervisionado selecionou o algoritmo *Agglomerative Clustering* com dois clusters e o atributo $linkage$ = "$average$" (modelo 29 da Tabela 4.2).

A Tabela 5.19 apresenta os resultados da clusterização usando o *Agglomerative Clustering* selecionado autonomamente pela abordagem ESPA. A acurácia obtida foi de 91,40%, um aumento de 8,03% em relação às acurácias das avaliações anteriores. A precisão para a classe dos ciclos maliciosos é de 40% pois das 5 predições (2 VPs + 3 FPs) 2 estavam corretas. Para os ciclos normais a precisão foi de 91,91% pois 40 ciclos maliciosos foram incorretamente classificados como ciclos normais. O *recall* para a classe dos ciclos maliciosos é de 4,76% pois 2 dos 42 ciclos maliciosos foram identificados corretamente, enquanto o *recall* para a classe dos ciclos normais é de 99,34% pois dos 458 ciclos normais, 455 foram identificados corretamente. Por fim, o *F1-score* dos ciclos maliciosos é de 8,51% e o *F1-score* dos ciclos normais é de 95,48%.

Tabela 5.19: Resultados da Avaliação 3 do Experimento 1 do Caso de Uso 3

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	2	3
	Ciclo Normal	40	455

5.4.3 Experimento 2 do Caso de Uso 3

O Experimento 2 usou o tráfego de rede da captura 52 da base de dados CTU-13. Os pesquisadores infectaram os *bots* no segundo 527 e lançaram o ataque no segundo 778 da captura. Assim como no experimento anterior, a engenharia de sinais utilizou os dados da captura 52 de modos diferentes. Para as avaliações 1 e 2, a abordagem ESPA realizou a engenharia de sinais aplicando o *kurtosis* para o atributo quantidade de IPs de origem, o *skewness* foi calculado para a quantidade de IPs de destino e o CV foi calculado para o total de pacotes (par a par). Na avaliação 3 (Sub-subseção 5.4.3.3) a engenharia de sinais foi realizada para todas as séries temporais dos 51 atributos apresentados na Tabela 4.2 usando o *skewness*, o *kurtosis*, o CV, a AC-1, a AC-2 e a AC-3. Assim, 306 novas características foram geradas pela engenharia de sinais da abordagem ESPA. Este experimento utilizou 5% da base de dados como o tamanho da janela deslizante de tamanho fixo visando avaliar um tamanho de janela diferente do Experimento 1 do Caso de uso 3. Além disso, a base de dados é menor do que a usada no caso de uso anterior, portanto, avaliar uma janela menor auxilia na condução dos experimentos. A seguir, cada uma das avaliações deste caso de uso apresenta os objetivos específicos e os resultados obtidos.

5.4.3.1 Avaliação 1 do Experimento 2

O objetivo desta avaliação é verificar se a abordagem ESPA pode predizer ataques DDoS usando o *K-means* em uma base de dados menor, focando na explicabilidade dos resultados. Assim como na primeira avaliação do experimento anterior, o *K-means* foi configurado manualmente para predizer os ataques DDoS. O *K-means* foi configurado para obter dois clusters e manteve o restante das definições sendo o padrão (1º algoritmo da Tabela 4.2).

A Figura 5.20(a) apresenta o resultado da aplicação da abordagem usando a captura 52 no intervalo entre os segundos 48 e 542. Essa figura apresenta a variação do *Kurtosis* para a quantidade de IPs de origem, o *Skewness* para a quantidade de IPs de destino e o CV para o total de pacotes medidos a cada ciclo de captura de um segundo. A abordagem ESPA identificou mudança no comportamento dos sinais precoces de alerta e isso acarretou a existência de dois grupos na visualização dos dados. O Grupo 1 possui apenas ciclos normais, onde não existem *bots* enviando pacotes. O Grupo 2 possui 48 ciclos onde 38 são normais e 10 maliciosos, onde

existe comunicação dos *bots*. Por fim, destaca-se que o Grupo 2 começou a ser formado no fim do processo de inicialização das máquinas virtuais que hospedavam os *bots*.

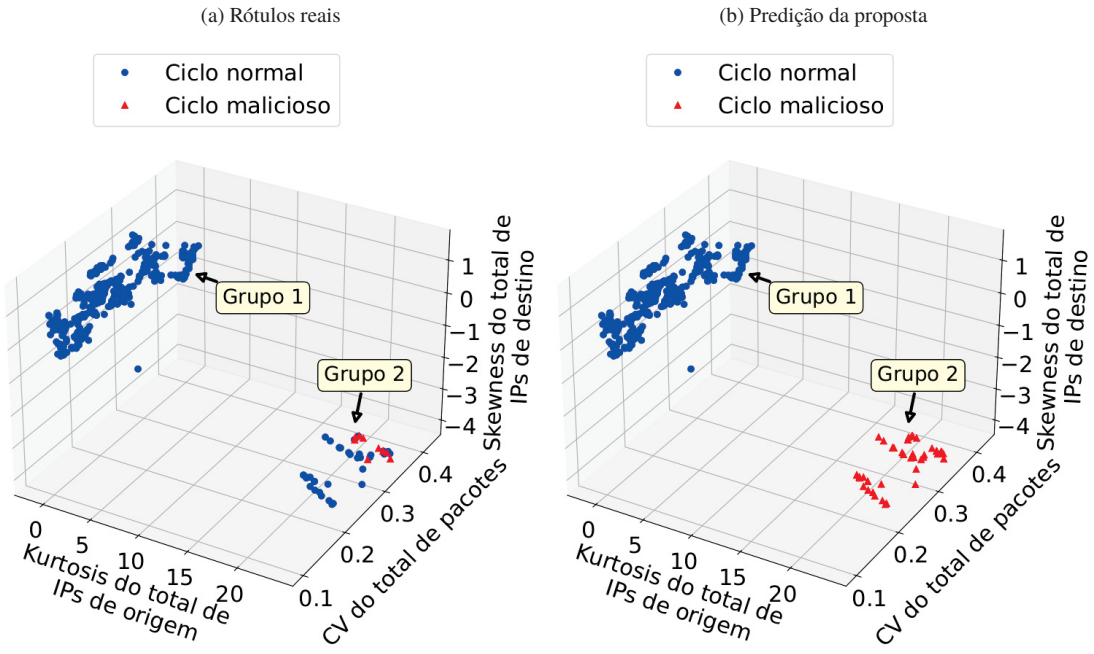


Figura 5.20: Predição de Ataques DDoS no Experimento 2 do Caso de Uso 3

O *K-means* aplicado sobre os resultados da abordagem ESPA (Figura 5.20(b)) identificou dois grupos e os resultados estão dispostos na Tabela 5.20. O Grupo 2 possui a menor quantidade de ciclos e os mais recentes (próximos ao segundo 542). Considerando o Grupo 2 como ciclos maliciosos, a abordagem ESPA distingue sinais da preparação de um ataque DDoS e consequentemente a abordagem ESPA pode automatizar a predição do ataque. Considerando os ciclos do Grupo 1 como normais, a proposta classifica corretamente 494 ciclos. Tomando os ciclos do Grupo 2 como maliciosos, a proposta identifica corretamente 10 ciclos maliciosos e rotula incorretamente 38 ciclos normais.

Nesta avaliação, a abordagem ESPA obteve uma acurácia de 92,32%. A precisão para a classe dos ciclos maliciosos foi de 20,83%, pois dos 48 (10 VP + 38 FP) ciclos identificados como maliciosos 10 estavam corretos. A precisão para os ciclos normais foi de 100% pois o Grupo 1 possui apenas ciclos normais. O *recall* para a classe dos ciclos maliciosos foi de 100% pois todos os ciclos maliciosos foram identificados. O *recall* para a classe ciclos normais foi de 92,16% pois dos 485 ciclos normais, 447 foram corretamente identificados. Por fim, o *F1-score* dos ciclos maliciosos é de 34,48% e o *F1-score* dos ciclos normais é de 95,92%.

Tabela 5.20: Resultados da Avaliação 1 do Experimento 2 do Caso de Uso 3

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	10	38
	Ciclo Normal	0	447

5.4.3.2 Avaliação 2 do Experimento 2

Assim como no experimento anterior, **o objetivo da Avaliação 2 é verificar se a abordagem ESPA equipada com o Algoritmo 3 pode selecionar um modelo de aprendizado de máquina não supervisionado que maximize a predição do ataque DDoS para o conjunto de dados analisados.** Para fins de comparação, o mesmo conjunto de dados utilizados na Avaliação 1 deste experimento foi utilizado nesta avaliação. O esperado é que a abordagem ESPA selecione um modelo que obtenha resultados similares com os resultados obtidos manualmente (Sub-subseção 5.4.3.1). Pois as novas características resultantes da engenharia de sinais proporcionaram uma predição com acurácia superior a 92%. Assim, obter o mesmo resultado autonomamente e sem a configuração manual seria um excelente resultado para esta avaliação.

A abordagem ESPA equipada com a configuração padrão do Algoritmo 3 identificou 29 modelos de algoritmos que obtiveram a maior média dos índices para avaliar o resultado do aprendizado de máquina não supervisionado. Os oito modelos que **não** foram sugeridos como os ideais foram os modelos 15, 26, 27, 32, 33, 34, 35 e 36 da Tabela 4.2. Aplicando qualquer um dos 29 modelos sugeridos pela abordagem ESPA, é possível obter o mesmo resultado apresentado pela Figura 5.20(b) e pela Tabela 5.20. Portanto, a acurácia, precisão, *recall* e *F1-score* são os mesmos obtidos na avaliação anterior. Isso ocorreu, pois a engenharia de sinais conseguiu dividir os ciclos maliciosos dos ciclos normais excepcionalmente. Assim, é plausível que vários algoritmos de aprendizado de máquina não supervisionados apresentem os mesmos resultados.

5.4.3.3 Avaliação 3 do Experimento 2

Assim como a Avaliação 3 do experimento anterior, esta avaliação explora a redução de dimensionalidade sem o uso de dados rotulados aliado com a seleção autônoma dos algoritmos de aprendizado de máquina não supervisionado. Nesta avaliação, o *FastICA* recebeu como entrada as 306 novas características originadas pela engenharia de sinais realizada pela abordagem ESPA. Essas características foram geradas ao processar as séries temporais dos 51 atributos da Tabela 5.17 sob a perspectiva de seis indicadores estatísticos (o *skewness*, o *kurtosis*, o CV, a AC-1, a AC-2 e a AC-3). O algoritmo *FastICA* necessita que a quantidade de sinais a ser decomposta seja informada. O valor escolhido nesta avaliação foi três, pois este maximizou os resultados obtidos. Portanto, o algoritmo *FastICA* recebe como entradas as 306 novas características e decompõe os dados em três sinais que serão usados para predizer os ataques.

Após a decomposição das características, esta avaliação aplicou a abordagem ESPA equipada com o algoritmo de seleção de algoritmos de aprendizado de máquina não supervisionado (Algoritmo 3.) Contudo, visando investigar uma versão alternativa do Algoritmo 3, esta avaliação utilizou o Algoritmo 3 apenas com o índice CDbw ao invés da média dos cinco índices. Além disso, para proporcionar uma comparação justa com as avaliações anteriores, o conjunto de dados analisados nesta avaliação foi o mesmo das Avaliações 1 e 2 deste caso uso. Assim, esta avaliação utiliza os dados entre os segundos 48 e 542 da captura 52 da base de dados CTU-13.

A abordagem ESPA equipada com o algoritmo de seleção de algoritmos de aprendizado de máquina não supervisionado identificou 15 modelos candidatos a serem os ideais para a base de dados analisada. Os modelos são: *K-means* com as configurações 1 a 10 da Tabela 4.2, o *Birch* com dois clusters e $threshold = 0.7$ (número 15 da Tabela 4.2), o *Agglomerative Clustering* com as configurações número 28, 29 e 31 da Tabela 4.2) e o *MeanShift* com a configuração padrão (modelo 37 da Tabela 4.2). O modelo *MeanShift* com a configuração padrão (modelo 37 da Tabela 4.2) foi o escolhido para realizar a predição do ataque DDoS. Ao contrário do observado na Avaliação 3 do experimento anterior, os resultados obtidos nesta avaliação foram os mesmos

observados na Avaliação 1 e 2 deste caso de uso. Portanto, a decomposição de características, neste caso, não aumentou nem diminuiu os acertos e erros apresentados na Tabela 5.20.

5.4.4 Experimento 3 do Caso de Uso 3

O Experimento 3 utilizou a base de dados CICDDoS2019. Nesta base, os pesquisadores lançaram o ataque no segundo 1484 da captura. Assim como nos experimentos anteriores, a engenharia de sinais utilizou os dados de dois modos diferentes. Para as avaliações 1 e 2 a abordagem ESPA realizou a engenharia de sinais aplicando o *kurtosis* para o atributo quantidade de IPs de origem, o *skewness* foi calculado para a quantidade de IPs de destino e o CV foi calculado para o total de pacotes (par a par). Na avaliação 3 (Sub-subseção 5.4.4.3) a engenharia de sinais foi realizada para todas as séries temporais dos 51 atributos apresentados na Tabela 4.2 usando o *skewness*, o *kurtosis*, o CV, a AC-1, AC-2 e AC-3. Assim, 306 novas características foram geradas pela engenharia de sinais da abordagem ESPA. Este experimento utilizou 5% da base de dados como o tamanho da janela deslizante de tamanho fixo visando avaliar um tamanho de janela diferente do Experimento 6 do Caso de Uso 2 (Subseção 5.3.7). Além disso, a base de dados é menor do que a usada no primeiro caso de uso, portanto, avaliar uma janela auxilia na condução dos experimentos. A seguir, cada uma das avaliações deste caso de uso apresenta os objetivos específicos e os resultados obtidos.

5.4.4.1 Avaliação 1 do Experimento 3

O objetivo desta avaliação é verificar se a abordagem ESPA pode predizer ataques DDoS usando o *K-means* em um cenário onde os *bots* e a vítima estão conectados pela Internet, focando na explicabilidade dos resultados. Este experimento é importante, pois, nos experimentos anteriores, os *bots* estavam na mesma rede que a vítima. Portanto, a abordagem proposta é avaliada em um cenário diferente e desafiador. Assim como na primeira avaliação dos experimentos anteriores, o *K-means* foi escolhido configurado manualmente para predizer os ataques DDoS. Para manter o padrão na utilização do *K-means*, a configuração foi a mesma dos experimentos anteriores. O *K-means* foi configurado para identificar dois clusters e o restante das configurações seguem as definições padrão da biblioteca (Modelo 1 da Tabela 3).

A Figura 5.21(a) apresenta o resultado da abordagem ESPA aplicada entre os segundos 101 e 159. Apesar de não ser tão claro quanto nos experimentos anteriores, o resultado da abordagem ESPA apresenta o mesmo comportamento dos experimentos anteriores. Existem dois grupos onde alguns ciclos normais e maliciosos se combinam. O Grupo 1 da Figura 5.21(a) possui grande concentração de ciclos normais e os dados mais antigos (próximo do segundo 101). O Grupo 2 da Figura 5.21(a) possui vários ciclos maliciosos e os dados mais recentes (próximo do segundo 159).

O *K-means* aplicado sobre os resultados (Figura 5.21(b)) da abordagem ESPA reforça a separação dos dados. Como nos experimentos anteriores, a Tabela 5.21 apresenta a matriz de confusão considerando o menor e mais recente grupo como malicioso. A predição dos ataques DDoS foi realizada com uma acurácia de 70,69%. A precisão referente a classe dos ciclos maliciosos é de 53,33%, pois dos 15 ciclos identificados como maliciosos, 8 estavam corretos. A precisão para a classe dos ciclos normais é de 76,74%, pois dos 43 ciclos marcados como normais, 33 estavam corretos. O *recall* para os ciclos maliciosos é de 44,44%, pois, dos 18 ciclos maliciosos, 8 foram identificados corretamente. O *recall* para os ciclos normais é de 82,50%, pois, dos 40 ciclos normais, 33 foram identificados corretamente. Por fim, o *F1-score* dos ciclos maliciosos é de 48,48% e o *F1-score* dos ciclos normais é de 79,52%.

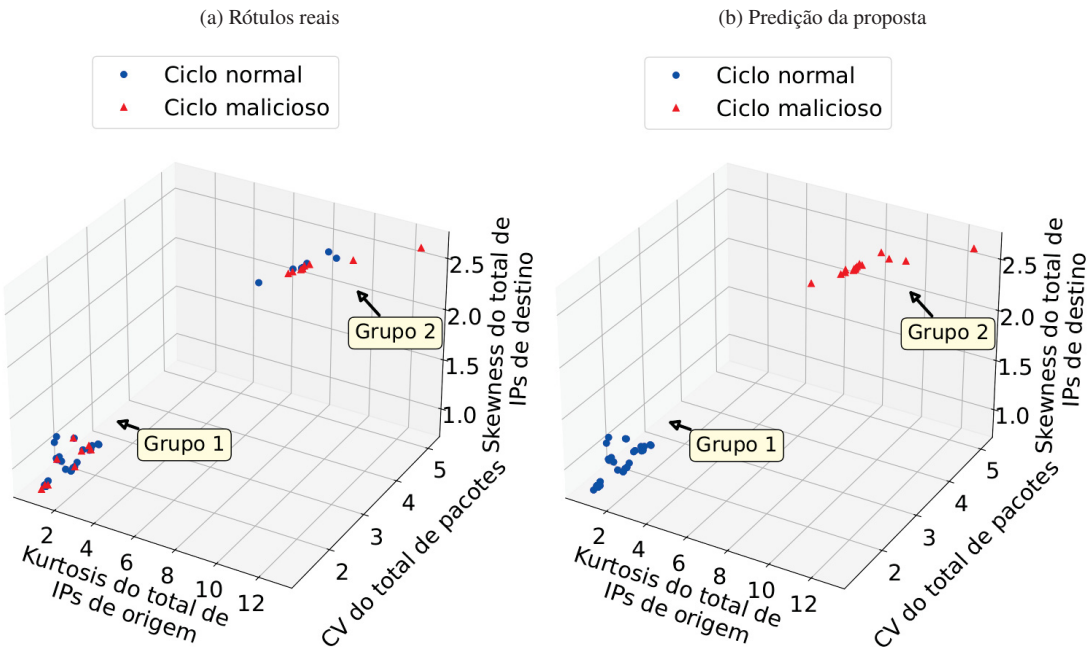


Figura 5.21: Predição de Ataques DDoS no Experimento 3 do Caso de Uso 3

Tabela 5.21: Resultados da Avaliação 1 do Experimento 3 do Caso de Uso 3

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	8	7
	Ciclo Normal	10	33

5.4.4.2 Avaliação 2 do Experimento 3

O objetivo da segunda avaliação deste experimento é verificar se a abordagem ESPA equipada com o Algoritmo 3 pode selecionar um modelo de aprendizado de máquina não supervisionado ideal para o conjunto de dados analisados. O mesmo conjunto de dados utilizados na Avaliação 1 deste caso de uso foi utilizado nesta avaliação para fins de comparação. O esperado é que a abordagem ESPA selecione um modelo que obtenha resultados melhores ou similares que os resultados da avaliação anterior.

A abordagem ESPA equipada com a configuração padrão do Algoritmo 3 identificou 21 modelos de algoritmos que obtiveram a maior média dos índices para avaliar o resultado do aprendizado de máquina não supervisionado. Os 16 modelos que **não** foram sugeridos como os ideais foram os modelos 11 a 15, 26 e 28 a 37 da Tabela 4.2. Aplicando qualquer um dos modelos sugeridos pela abordagem ESPA, é possível obter o mesmo resultado apresentado pela Figura 5.21 e pela Tabela 5.21. Portanto, a acurácia, precisão, *recall* e *F1-score* são os mesmos obtidos na avaliação anterior deste experimento. Contudo, 21 modelos poderiam obter os mesmos resultados sem conhecimentos prévios da base de dados e sem a interação humana.

5.4.4.3 Avaliação 3 do Experimento 3

A última avaliação deste experimento explora a redução de dimensionalidade sem o uso de dados rotulados aliado com a seleção autônoma dos algoritmos de aprendizado de máquina não supervisionado. Assim como citado anteriormente, o objetivo de diminuir a

dimensionalidade dos dados nesta avaliação é diminuir o total de erros obtidos anteriormente utilizando diferentes características. O *FastICA* recebeu como entrada 306 novas características originadas pela engenharia de sinais realizada pela abordagem ESPA. O algoritmo *FastICA* necessita que a quantidade de sinais a ser decomposta seja informada. O valor escolhido nesta avaliação foi cinco, pois este maximizou os resultados obtidos. Portanto, o algoritmo *FastICA* recebe como entradas as 306 novas características e decompõe os dados em cinco sinais que serão usados para predizer os ataques DDoS.

Após a decomposição das características, esta avaliação aplicou a abordagem ESPA equipada com o algoritmo de seleção de algoritmos de aprendizado de máquina não supervisionado (Algoritmo 3.) Assim como nas Avaliações 3 dos experimentos anteriores, esta avaliação utilizou o Algoritmo 3 apenas com o índice CDbw ao invés da média dos cinco índices (configuração padrão). Além disso, para proporcionar uma comparação justa com as avaliações anteriores, o conjunto de dados analisados nesta avaliação foi o mesmo das Avaliações 1 e 2 deste experimento. Assim, esta avaliação utiliza os dados entre os segundos 101 e 159 da base de dados CICDDoS2019.

Como resultado, a abordagem ESPA equipada com o algoritmo de seleção de algoritmos de aprendizado de máquina não supervisionado (Algoritmo 3) identificou sete modelos como os ideais para a base de dados analisados. Os modelos selecionados são as cinco configurações do algoritmo *Birch* apresentadas na Tabela 4.2 (modelos de 11 a 15) e duas versões alternativas do Agglomerative Clustering (números 30 e 31 da Tabela 4.2). A Tabela 5.22 apresenta os resultados da clusterização usando o modelo número 13 (*Birch* com dois clusters e $threshold = 0,5$ que é o valor padrão) selecionado autonomamente pela abordagem ESPA. A acurácia obtida foi de 72,41%. Um aumento de 2,43% em comparação aos resultados obtidos nas Avaliações 1 e 2 deste experimento. A precisão para a classe dos ciclos maliciosos é de 100% pois das 2 predições (2 VPs + 0 FPs) todas estavam corretas. Para os ciclos normais a precisão foi de 71,42% pois 16 ciclos maliciosos foram incorretamente classificados como ciclos normais. O *recall* para a classe dos ciclos maliciosos é de 11,11% pois 2 dos 18 ciclos maliciosos foram identificados corretamente, enquanto o *recall* para a classe dos ciclos normais é de 100% pois todos os ciclos normais foram identificados corretamente. Por fim, o *F1-score* dos ciclos maliciosos é de 20% e o *F1-score* dos ciclos normais é de 83,33%.

Tabela 5.22: Resultados da Avaliação 3 do Experimento 3 do Caso de Uso 3

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	2	0
	Ciclo Normal	16	40

5.4.5 Experimento 4 do Caso de Uso 3

O Experimento 4 utilizou a base de dados IoT-23 que possui 23 cenários de ataques DDoS em ambientes IoT. No cenário 17, os pesquisadores iniciaram a captura às 06:43:20 e iniciaram a execução do *malware* às 11:43:43 do mesmo dia. Assim, o tráfego anterior a infecção é normal e o tráfego posterior contém traços da preparação do ataque. O principal diferencial deste experimento para os outros é a base de dados e como ela é utilizada neste experimento. Essa base compreende uma rede IoT. Além disso, essa base de dados possui 24 horas de tráfego de rede.

Devido às características da base de dados IoT-23, a engenharia de sinais foi realizada por minuto e não por segundo como nos experimentos anteriores. Além do ciclo de captura ser um

minuto, a engenharia de sinais utilizou um atributo do tráfego de rede diferente dos experimentos anteriores. Este experimento utilizou o maior tamanho de pacote em vez do total de pacotes enviados. Este atributo mede o tamanho do maior pacote em cada ciclo de captura. Assim, a combinação par a par entre atributos e indicadores é a seguinte: *kurtosis* da série temporal do maior pacote, *skewness* da série temporal do total de IPs de destino e CV da série temporal do total dos IPs de origem. Essas configurações foram feitas empiricamente, sendo aplicadas às Avaliações 1 e 2. A Avaliação 3 aplica engenharia de sinais usando o *kurtosis*, o *skewness*, o CV, a AC-1, a AC-2 e a AC-3 sobre todas as 51 séries temporais referentes aos atributos apresentados na Tabela 5.17. Posteriormente, a Avaliação 3 realiza a seleção de características assim como foi realizado nas Avaliações 3 dos experimentos anteriores.

5.4.5.1 Avaliação 1 do Experimento 4

O objetivo desta avaliação é verificar se a abordagem ESPA pode predizer ataques DDoS em redes IoT focando na explicabilidade. Esta avaliação também utiliza o *K-means* configurado manualmente. O *K-means* foi configurado para obter dois clusters e manter o restante das definições padrão da biblioteca. Esta avaliação analisou a abordagem ESPA equipada como o *K-means* entre os minutos 241 e 321 da captura. Como a infecção dos *bots* ocorreu no minuto 301, o período analisado refere-se aos 60 minutos antes e 20 minutos depois da execução do *malware*. Este trabalho escolheu apenas 20 minutos para reforçar que a abordagem proposta consegue rapidamente identificar sinais da preparação do ataque frente a uma quantidade razoável de tráfego de rede normal (60 minutos antes da infecção) sem conhecimento prévio da *botnet* ou do ataque DDoS.

Apesar de não ser possível quantificar quanto tempo antes do lançamento do ataque a predição ocorreu devido aos detalhes da base de dados, este trabalho avalia quanto tempo depois da execução do *malware* a proposta identificou a preparação do ataque. A Figura 5.22(a) apresenta o resultado da abordagem ESPA aplicada sobre ciclos capturados até o minuto 321, e considera 60 minutos antes da infecção e apenas 20 após o início da execução do *malware*. Assim como nos experimentos anteriores, o resultado apresenta a existência de dois grupos de dados. O Grupo 1 possui os ciclos mais antigos e apenas ciclos normais. O Grupo 2 possui os ciclos mais novos (próximo do minuto 321) e somente ciclos maliciosos. Notavelmente, o *K-means* consegue separar perfeitamente os dois grupos. Assim, todas as métricas, acurácia, precisão, *F1-score* para ambas as classes, atingem 100%. A Tabela 5.23 apresenta a matriz de confusão para os resultados obtidos nesta avaliação.

Tabela 5.23: Resultados da Avaliação 1 do Experimento 4 do Caso de Uso 3

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	20	0
	Ciclo Normal	0	60

5.4.5.2 Avaliação 2 do Experimento 4

O objetivo da Avaliação 2 é verificar se a abordagem ESPA equipada com o Algoritmo 3 pode selecionar um modelo de aprendizado de máquina não supervisionado ideal para predizer ataques DDoS em redes IoT. Para fins de comparação, o mesmo conjunto de dados utilizados na Avaliação 1 deste experimento foi utilizado nesta avaliação. Como o resultado anterior atingiu

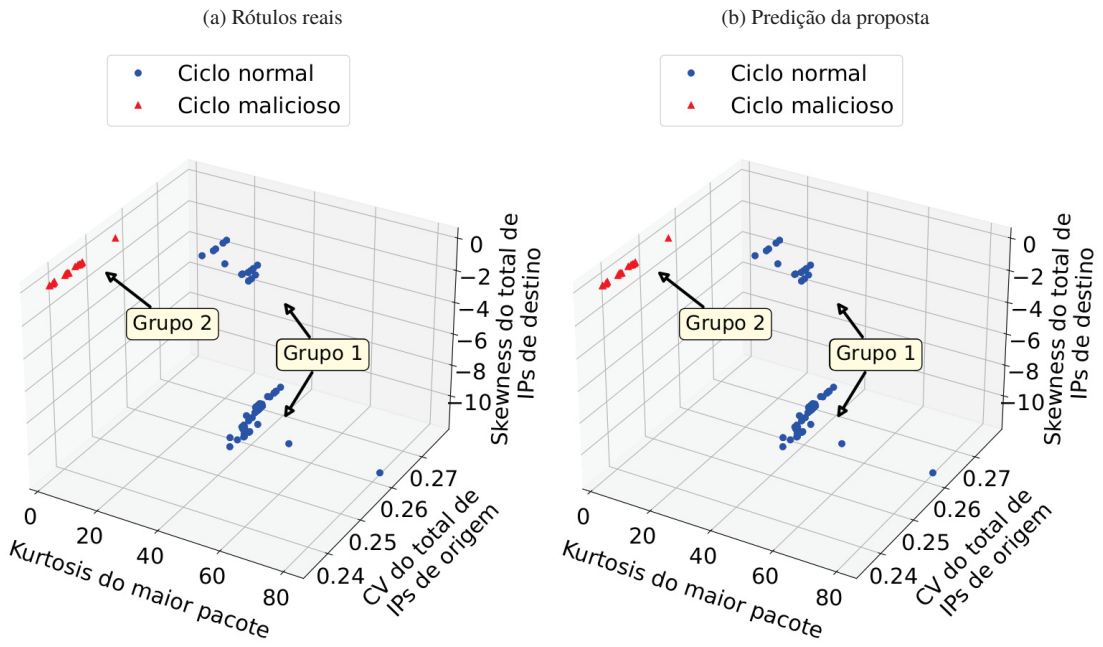


Figura 5.22: Predição de Ataques DDoS no Experimento 4 do Caso de Uso 3

100% de acurácia (Sub-subseção 5.4.5.1), o melhor resultado possível é a abordagem ESPA equipada com o Algoritmo 3 selecionar um algoritmo que mantenha a predição com 100% de acurácia. Assim, obter o mesmo resultado autonomamente, sem a configuração manual, seria um excelente resultado para esta avaliação.

A abordagem ESPA equipada com a configuração padrão do Algoritmo 3 identificou 21 modelos de algoritmos que obtiveram a maior média dos índices para avaliar o resultado do aprendizado de máquina não supervisionado. Os 21 modelos sugeridos como os ideais foram os modelos números 1 a 11, 16, 18, 19, 21 a 25, 28 e o 32 da Tabela 4.2. Aplicando qualquer um dos modelos sugeridos pela abordagem ESPA, é possível obter o mesmo resultado apresentado pela Figura 5.22(b) e pela Tabela 5.23. Portanto, a acurácia, precisão, *recall* e *F1-score* atingiram os 100%, assim como na avaliação anterior. Isso ocorreu, pois a engenharia de sinais conseguiu dividir os ciclos maliciosos dos ciclos normais excepcionalmente. Assim, é plausível que vários algoritmos de aprendizado de máquina não supervisionados apresentem os mesmos resultados.

5.4.5.3 Avaliação 3 do Experimento 4

O objetivo desta avaliação é explorar a redução de dimensionalidade sem o uso de dados rotulados aliado com a seleção autônoma dos algoritmos de aprendizado de máquina não supervisionado em redes IoT. O *FastICA* recebeu como entrada as 306 novas características originadas pela engenharia de sinais realizada pela abordagem ESPA. O algoritmo *FastICA* necessita que a quantidade de sinais a ser decomposta seja informada. O valor escolhido nesta avaliação foi cinco, pois este maximizou os resultados obtidos.

Após a decomposição das características, esta avaliação aplicou a abordagem ESPA equipada com o algoritmo de seleção de algoritmos de aprendizado de máquina não supervisionado (Algoritmo 3.) Esta avaliação utilizou o Algoritmo 3 apenas com o índice CDbw ao invés da média dos cinco índices. Além disso, para proporcionar uma comparação justa com as avaliações anteriores, o conjunto de dados analisados nesta avaliação foi o mesmo das Avaliações 1 e 2 deste caso de uso. Assim como na Avaliação 2, não é possível melhorar os resultados obtidos,

uma vez que eles atingiram a marca de 100%. Contudo, esta análise é relevante para verificar que diferentes características poderiam apresentar um resultado próximo de 100%. Além disso, é relevante verificar se a abordagem proposta também pode selecionar um modelo adequado para esses dados.

A abordagem ESPA equipada com o algoritmo de seleção de algoritmos de aprendizado de máquina não supervisionado identificou dez modelos candidatos a serem os ideais para a base de dados analisada. Os modelos são variações *K-means* com as configurações números 1 a 10 da Tabela 4.2. O modelo *K-means* com dois clusters e o restante das configurações sendo o padrão da biblioteca (modelo 1 da Tabela 4.2) foi o escolhido. Este modelo foi escolhido por ser o mesmo utilizado nas Avaliações 1 e 2 deste experimento. Usando o modelo selecionado e as cinco novas características, a abordagem proposta conseguiu atingir os mesmos resultados apresentados na Tabela 5.23. Ou seja, a predição do ataque DDoS nessa avaliação atingiu 100% em todas as métricas de avaliação (acurácia, precisão, *recall* e *F1-score*). Portanto, a abordagem ESPA conseguiu identificar sinais da preparação de um ataque DDoS estando equipada com engenharia de sinais que gerou em 306 sinais, com a seleção de características não supervisionada e o algoritmo de seleção de algoritmos de aprendizado de máquina não supervisionado.

5.4.6 Discussão dos Resultados do Caso de Uso 3

É oportuno ressaltar que todos os resultados apresentados no Caso de uso 3 foram obtidos sem a utilização de dados rotulados. Os rótulos foram usados apenas para quantificar as métricas de avaliação. Isso aumenta a dificuldade de realizar a predição dos ataques DDoS, mas reforça a importância dos resultados obtidos. A abordagem ESPA se qualifica para predizer ataques DDoS de dia zero, pois a abordagem proposta obteve todos os resultados sem conhecimento prévio das redes e dos ataques. Além dos ataques DDoS de dia zero, todas as bases de dados avaliadas neste caso de uso são desbalanceadas. Isso significa que a quantidade de tráfego normal é maior do que a quantidade de tráfego malicioso. O desbalanceamento das classes foi apresentado também nos outros casos de uso. Contudo, ao rotular os ciclos maliciosos com apenas um pacote originado por *bots*, era um grande desafio obter altas acurácias como visto nos resultados.

Em relação à Avaliação 1 do Experimento 1, a abordagem ESPA proporcionou a predição do primeiro ataque em um cenário com mais tempo de captura que os Experimentos 2 e 3 e com diferentes ataques DDoS. A documentação indica que a infecção dos *bots* ocorreu 49 minutos e 49 segundos antes do lançamento do ataque. A abordagem ESPA identificou sinais da preparação do ataque 30 minutos e 38 segundos antes de seu lançamento e 19 minutos e 11 segundos após as infecções. A Figura 5.23 ilustra a relação entre o momento da infecção, da predição e do lançamento do ataque.

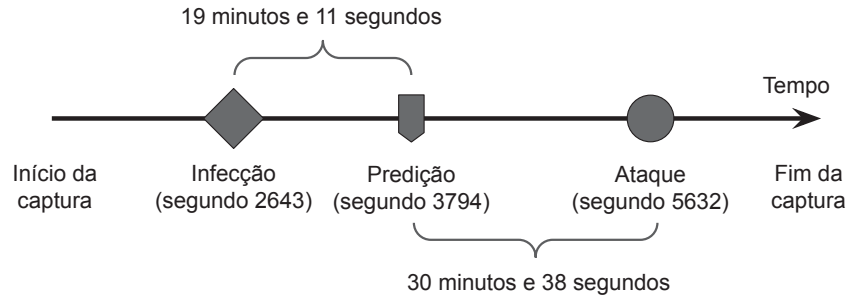


Figura 5.23: Predição do Ataque DDoS na Avaliação 1 do Experimento 1 do Caso de Uso 3

A predição do ataque DDoS com a acurácia de 84,6% e com mais de 30 minutos de antecedência obtida na Avaliação 1 do Experimento 1 deste caso de uso só foi possível

graças à engenharia de sinais realizada pela abordagem ESPA. A Figura 5.24 apresenta a comparação do resultado da abordagem com os dados originais. Na Figura 5.24(a), o resultado da abordagem ESPA possibilita a separação dos ciclos maliciosos dos ciclos normais. Enquanto na Figura 5.24(b), os atacantes conseguem esconder seu comportamento entre o tráfego de rede normal. Aplicar o *K-means* nos dados originais não separa a preparação do ataque do tráfego normal. Portanto, o resultado não suporta nenhum tipo de conclusão. Enquanto com a proposta deste trabalho é possível identificar corretamente 10 ciclos maliciosos. Assim, os resultados apresentados no Caso de uso 3 reforçam o mérito da proposta.

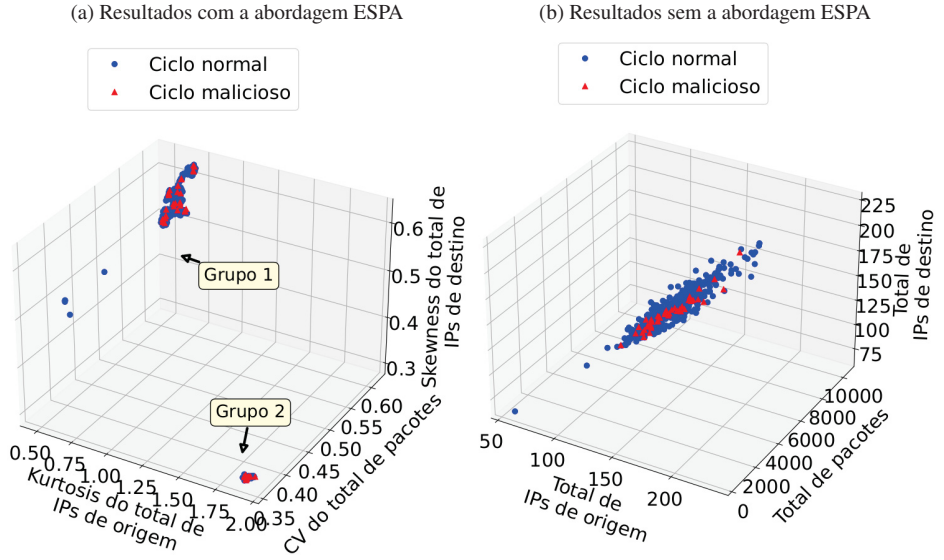


Figura 5.24: Comparação dos Resultados Obtidos pela Abordagem com os Dados Originais na Avaliação 1 do Experimento 1

Outro importante resultado obtido na Avaliação 1 do Experimento 1 é que a predição pode ser explicada com a Figura 5.24(a). O *K-means* identificou dois grupos de dados, o Grupo 1 é formado majoritariamente por ciclos mais antigos e normais. O Grupo 2 possui os ciclos mais recentes e concentra vários ciclos maliciosos, onde os *bots* enviam pacotes. A Figura 5.25 relaciona a teoria dos sinais precoces de alerta com os resultados obtidos pela abordagem ESPA. A Figura 5.25(a) exibe as interpretações relacionadas ao indicador estatístico *kurtosis*. A Figura 5.25(a) indica que o *kurtosis* para o Grupo 1 fica próximo de 0,5. Enquanto o *kurtosis* para o Grupo 2 atinge valores próximos a 1,85. A interpretação do *kurtosis* indica que a série temporal dos endereços IPs de origem no Grupo 1 tem propriedades mais simétricas do que observada no Grupo 2. Essa mudança na interpretação do *kurtosis* indica a aproximação de uma transição crítica (detalhes na Subseção 4.2.4). Portanto, essa indicação de aproximação da transição crítica é utilizada para predizer o ataque DDoS.

O *skewness* apresentado na Figura 5.25(b) reforça a aproximação da transição crítica. A interpretação do *skewness* para o Grupo 2 indica que a série temporal dos endereços IPs de origem possui propriedades mais simétricas do que as observadas no Grupo 1. Observando a Figura 5.25(b) é possível identificar que o *kurtosis* para o Grupo 1 fica próximo de 0,6. Enquanto o *kurtosis* para o Grupo 2 atinge valores próximos a 0,3. Mesmo que a diferença entre o valor do *skewness* para os grupos seja pequena, é o suficiente para auxiliar na predição dos ataques. Tanto visualmente, nas Figuras 5.19(b) e 5.25, quanto automaticamente para a central de inteligência equipada com o *K-means*. Por fim, também é importante citar que o CV é o sinal precoce de

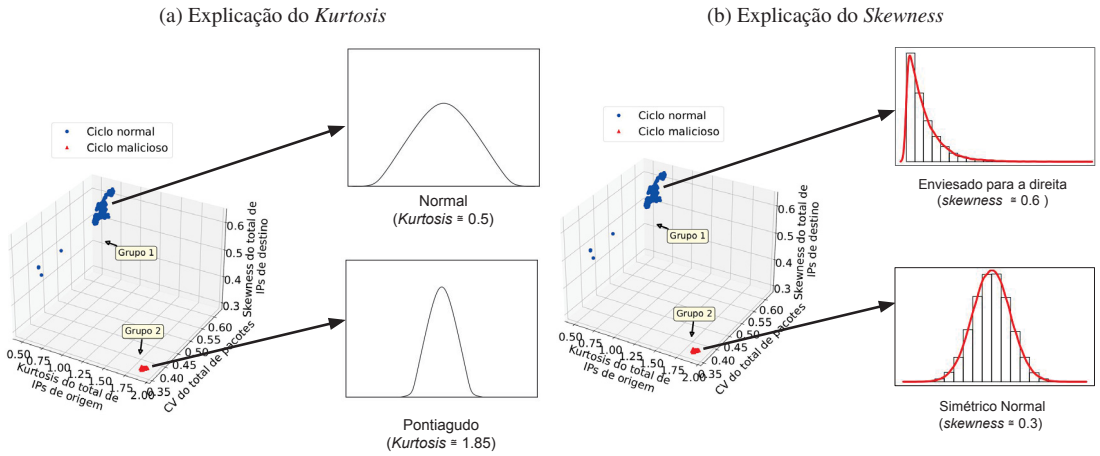


Figura 5.25: Explicação da Engenharia de Sinais na Avaliação 1 do Experimento 1 do Caso de Uso 3

alerta que menos impacta nestes resultados. Contudo, a variação do CV favorece a interpretação dos resultados e auxilia a central de inteligência a predizer o ataque DDoS.

Na Avaliação 1 do Experimento 2, a proposta possibilitou a predição do ataque DDoS 3 minutos e 56 segundos antes do lançamento do ataque. Esse resultado é relevante, pois a base possui cerca de 16 minutos de captura. A proposta ESPA obteve esses resultados apenas 15 segundos após o início da infecção. A Figura 5.26 ilustra o tempo de predição obtido na Avaliação 1 do Experimento 2 deste caso de uso. A acurácia obtida nessa predição atingiu os 92,32%. Diferentemente do experimento anterior, o *K-means* identificou que o Grupo 1 é formado somente por ciclos normais. O Grupo 2 possui os ciclos mais recentes e concentra todos os ciclos maliciosos, onde os *bots* enviam e/ou recebem pacotes e alguns ciclos normais.

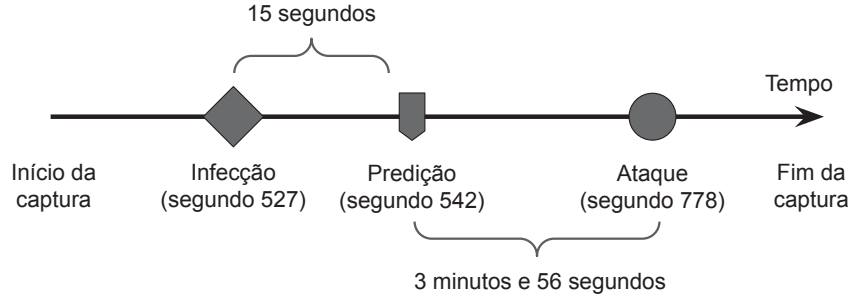


Figura 5.26: Predição do Ataque DDoS na Avaliação 1 do Experimento 2 do Caso de Uso 3

Como citado anteriormente, a engenharia de sinais realizada pela abordagem ESPA é o aspecto chave para a predição dos ataques DDoS. A Figura 5.27 reforça essa conclusão. A Figura 5.27(a) apresenta o resultado da aplicação da abordagem ESPA sobre o tráfego de rede disponibilizado na captura 52 da base de dados CTU-13. A Figura 5.27(b) apresenta os dados originais. Utilizando somente o tráfego de rede original, sem a aplicação da engenharia de sinais (Figura 5.27(b)), a abordagem ESPA não conseguiria separar o tráfego de rede que possui ações dos *bots* (ciclos maliciosos) do tráfego de rede originado por usuários legítimos (ciclos normais). Assim, não é possível verificar os sinais da preparação do ataque DDoS e não é possível predizê-lo. A mesma conclusão pode ser aplicada às Avaliações 1 dos Experimentos 3 e 4 (Figura 5.28). Isso pode ser verificado nas Figuras 5.28(a) e 5.28(b). Pois o tráfego de preparação do ataque (ciclos maliciosos) está envolto pelo tráfego normal (ciclos normais).

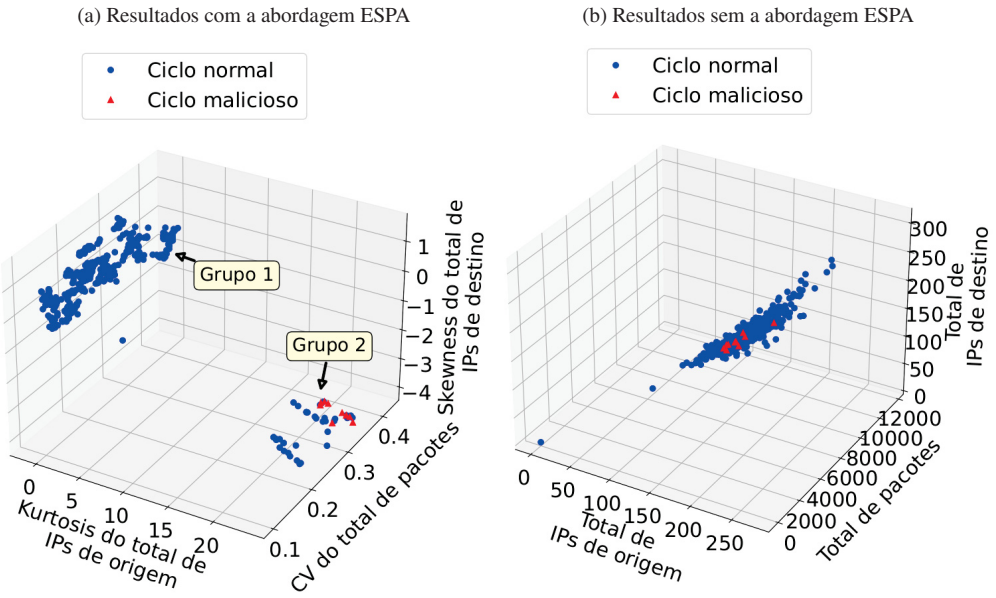


Figura 5.27: Comparação dos Resultados Obtidos pela Abordagem com os Dados Originais na Avaliação 1 do Experimento 2

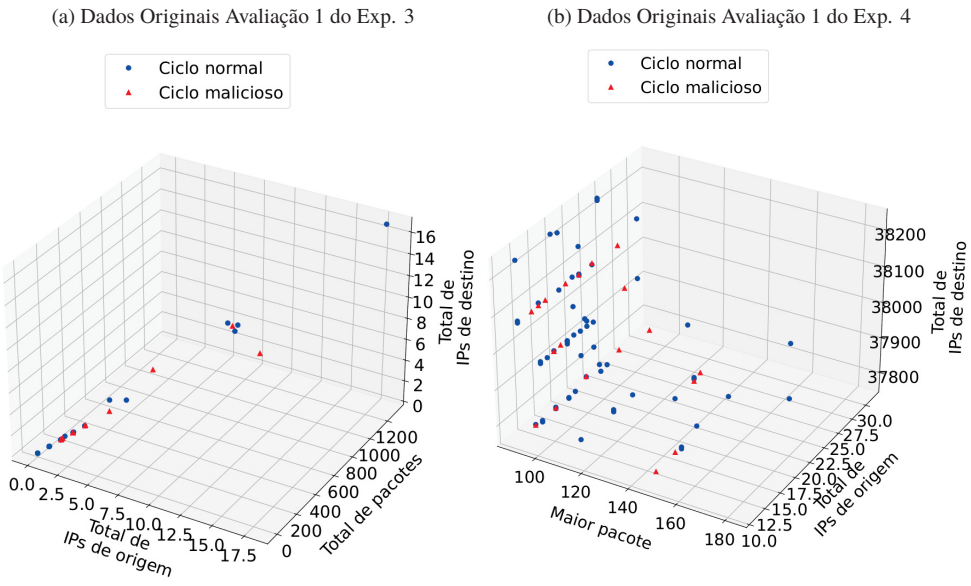


Figura 5.28: Representação dos Dados sem a Utilização da Abordagem ESPA nas Avaliações 1 dos Experimentos 3 e 4

Outro importante resultado obtido na Avaliação 1 do Experimento 2 é a explicabilidade obtida com a Figura 5.29(a). A interpretação do *kurtosis* (Figura 5.29(a)) indica que a série temporal dos endereços IPs de origem no Grupo 1 possui propriedades mais simétricas do que observada no Grupo 2. Isso ocorre, pois o valor do *kurtosis* no Grupo 1 é próximo de zero, para o Grupo 2, o valor é próximo a 20. Essa mudança na interpretação do *kurtosis* indica a aproximação de uma transição crítica, que este trabalho utiliza para identificar a aproximação de um ataque DDoS. O *skewness* apresentado na Figura 5.29(b) reforça a aproximação da transição crítica. A interpretação do *skewness* para o Grupo 1 indica que a série temporal dos endereços IPs de origem possui propriedades mais simétricas do que as observadas no Grupo 2. Para o Grupo 1, o valor do *skewness* é próximo de 0. Para o Grupo 2, *skewness* atinge valores próximos de −4. Portanto, essas variações possibilitam a predição do ataque DDoS e auxiliam no entendimento dos resultados.

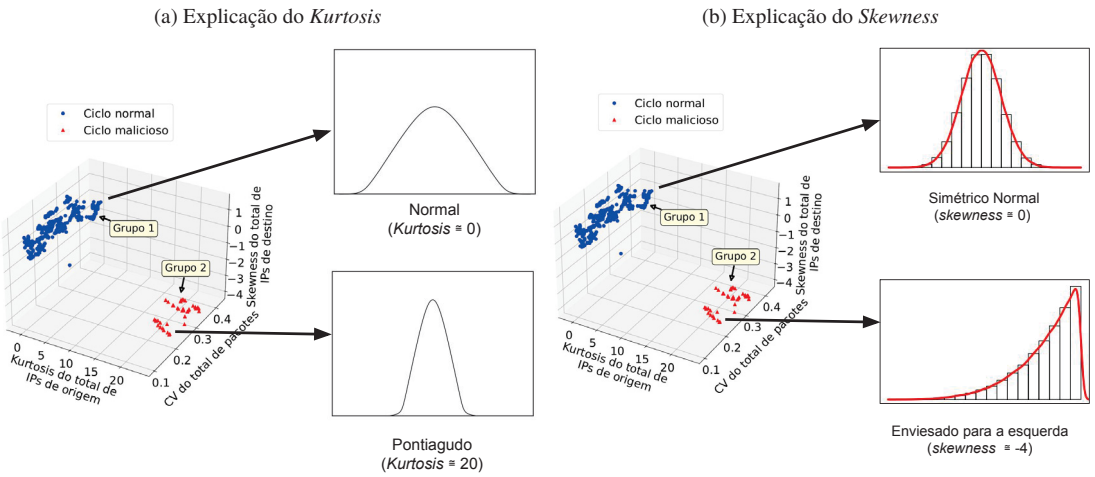


Figura 5.29: Explicação da Engenharia de Sinais na Avaliação 1 do Experimento 2 do Caso de Uso 3

A Avaliação 1 do Experimento 3 apresentou resultados inferiores comparados aos anteriores. Contudo, a abordagem ESPA identificou o mesmo comportamento nas Avaliações 1 dos experimentos anteriores. Antes do lançamento do ataque, as novas características geradas pela abordagem apresentam uma mudança simbolizada pela existência de dois grupos. Em todos os resultados das Avaliações 1, um grupo é majoritariamente composto por ciclos de captura de dados onde os *bots* não atuam, e o outro é formado por segundos com e sem a atuação dos *bots*. Mesmo que a divisão dos dois grupos não fosse tão clara quanto nos outros, a proposta deste trabalho proporcionou a predição do ataque DDoS com 22 minutos e 05 segundos antes de seu lançamento (Figura 5.30) com acurácia de 70,69%.

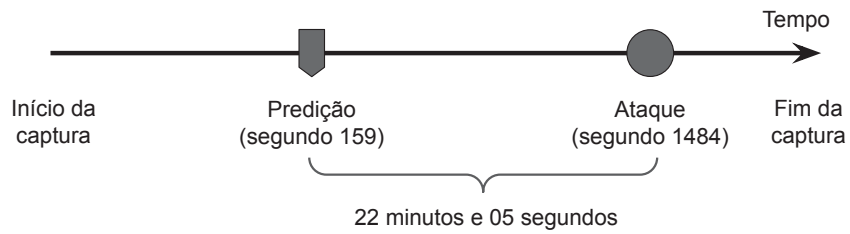


Figura 5.30: Predição do Ataque DDoS na Avaliação 1 do Experimento 3 do Caso de Uso 3

Os resultados obtidos na Análise 1 do Experimento 4 indicam uma acurácia de 100% na predição de um ataque DDoS. A alta acurácia ocorreu porque a execução do *malware* impactou o

tráfego da rede. O novo comportamento do tráfego de rede induziu alterações nos sinais precoces de alerta, gerando o novo cluster malicioso (Figura 5.22 da Sub-subseção 5.4.5.2). Embora não tenha sido possível mensurar quanto tempo antes do lançamento do ataque a predição ocorreu, os resultados obtidos na Análise 1 do Experimento 4 representam um excelente exemplo de quão importante é a predição do ataque. A predição no conjunto de dados IoT ocorreu apenas 20 minutos após a infecção do *malware*. Portanto, a proposta proporcionou mais tempo para lidar com ataques DDoS.

Os resultados obtidos pela abordagem ESPA equipada com o Algoritmo 3 (Avaliações 2 dos Experimentos 1, 2, 3 e 4) ficaram dentro do esperado. Mesmo com mais de 30 opções, em todas as Avaliações 2 (Experimentos 1, 2, 3 e 4), o modelo de aprendizado de máquina não supervisionado selecionado pela abordagem ESPA atingiu o mesmo resultado da configuração manual. Assim, a abordagem ESPA equipada com o Algoritmo 3 pode ser um importante aliado no combate aos ataques DDoS. Isso ocorre, pois, diferentemente da literatura (de Souto et al., 2008; Sáez e Corchado, 2019; Poulakis, 2020), a abordagem ESPA sugeriu vários modelos que poderiam resolver o problema similarmente ao modelo definido manualmente, sem treinamentos prévios, sem o uso de rótulos e sem a interação humana. Deste modo, o administrador da rede poderia equipar a abordagem ESPA com o algoritmo que melhor se adapta às necessidades do momento. Assim, a abordagem ESPA equipada com o Algoritmo 3 fica independente de treinamentos e pode funcionar adequadamente até mesmo em bases de dados diferentes das avaliadas neste trabalho. Contudo, esse resultado demanda análises e experimentações complementares. Uma vez que os dados analisados são oriundos da engenharia de sinais. Isso pode ter simplificado a seleção dos algoritmos.

A seleção de características sem o uso de dados rotulados aliada com a abordagem ESPA equipada com o Algoritmo 3 também apresentou resultados notáveis. Na Avaliação 3 do Experimento 1 (CTU-13 - Captura 51) deste caso de uso, ao aplicar a seleção de características, a abordagem ESPA apresentou uma diminuição na quantidade de erros totais. A Tabela 5.19 (Sub-subseção 5.4.2.3) apresenta os resultados da clusterização usando o Agglomerative Clustering (modelo 29 da Tabela 4.2) selecionado autonomamente pela abordagem ESPA. Na Avaliação 1 do Experimento 1, o total de erros (FP+FN) foi 77. Enquanto na Avaliação 3 do Experimento 1 o total de erros foi de 43, uma diminuição de 44,15% dos erros totais. Apesar da quantidade de erros totais ter diminuído, o total de VP diminuiu de 10 para 2. Mesmo com esse aumento, a acurácia obtida na Avaliação 3 do Experimento 1 foi 8,03% superior à acurácia obtida na Avaliação 1, saindo de 84,6% para 91,4%.

Outro importante resultado obtido com seleção de características aliada com a seleção de algoritmos de aprendizado não supervisionado foi observado na Avaliação 3 do Experimento 3 (CICDDoS2019). Neste caso, a abordagem ESPA apresentou um aumento de 2,43% na acurácia em comparação aos resultados das Avaliações 1 e 2 do Experimento 3. A acurácia era de 70,69% e aumentou para 72,41%. Esse resultado é extremamente relevante, por manter a predição do ataque DDoS e diminuir a quantidade de erros gerais obtidos pela abordagem ESPA. Além disso, essa versão da abordagem proposta zerou os FPs. Isso significa que a clusterização obtida por essa versão da abordagem identificou um cluster somente com o ciclos maliciosos. Isso aumentaria a confiança na abordagem e facilitaria a utilização dos resultados.

Em termos de acurácia, era impossível melhorar a acurácia da Avaliação 1 do Experimento 4 (IoT-23), pois este possuía o valor máximo (100%). Contudo, a seleção de características aliadas a seleção do modelo de aprendizado de máquina da abordagem ESPA (Avaliação 3 do Experimento 4) obteve o mesmo resultado de 100% de acurácia na predição dos ataques DDoS. Na avaliação 1 do Experimento 2 (CTU-13 - Captura 52), a acurácia que a abordagem ESPA obteve foi de 92,32%. Neste caso, todos os ciclos maliciosos foram segregados em um grupo

que também possuía ciclos normais. Melhorar esse resultado era especialmente desafiador. Tão logo, o melhor resultado obtido foi atingir a predição do ataque com a mesma acurácia anterior (92,32% na Avaliação 3 do Experimento 2). Assim, para esses dois experimentos, a abordagem ESPA equipada com o Algoritmo 3 e com seleção de características sem o uso de dados não melhorou nem piorou os resultados.

O tempo gasto para a abordagem ESPA equipada com o Algoritmo 3 executar é outro ponto a ser discutido. O maior tempo gasto para executar todas as avaliações foi de 4 minutos e 45 segundos. Esse tempo é menor do que os 42 minutos gastos no Experimento 1 do Caso de uso 2 (Subseção 5.3.2). O gasto de tempo menor pode ser justificado pelo menor espaço de busca. Pois a abordagem ESPA equipada avalia 37 modelos diferentes, enquanto o Autokeras avalia vários outros modelos. Mesmo assim, os resultados obtidos neste caso de uso são competitivos frente aos resultados obtidos anteriormente, demandando menos tempo para serem obtidos.

5.4.7 Comparação dos Resultados do Caso de uso 3 com a Literatura

A Tabela 5.24 apresenta os resultados dos casos de uso anteriores, os principais resultados do Caso de uso 3 e trabalhos comparáveis presentes na literatura. Primeiramente, é oportuno destacar que a abordagem proposta aumenta o tempo de predição em relação à literatura. Em Rahal et al. (2020), os autores predizem o ataque DDoS na captura 51 do CTU-13 com 5 minutos e 41 segundos de antecedência, enquanto a proposta ESPA melhora os resultados predizendo o mesmo ataque com 30 minutos de antecedência. Além disso, o Rahal et al. (2020) utiliza o algoritmo SOM configurado manualmente. Neste caso de uso, os resultados indicam a capacidade da abordagem ESPA de selecionar um algoritmo adequado para o conjunto de dados considerado. Essa característica também evolui o trabalho de Rahal et al. (2020).

Tabela 5.24: Resumo da Comparação Entre a abordagem ESPA e a Literatura no Caso de Uso 3

Experimento	**Base de dados**	**Referência**	**Tempo predição**	**Acurácia**
Experimento 1 (**Caso de uso 1**)	Cap. 52 (CTU-13)	ESPA *(Adaboost)*	3 min e 22 seg	**99,69**%
Experimento 2 (**Caso de uso 1**)	UDP-Lag (CICDDoS2019)	ESPA *(Adaboost)*	2 min e 8 seg	**99,76%**
Experimento 3 (**Caso de uso 1**)	Cap. 52 (CTU-13)	ESPA *(MLP)*	3 min e 49 seg	98,87%
Experimento 4 (**Caso de uso 1**)	UDP-Lag (CICDDoS2019)	ESPA *(MLP)*	**3 min e 55 seg**	99,60%
Experimento 1 (**Caso de uso 2**)	Cap. 51 (CTU-13)	ESPA *(LSTM Autoencoder)*	29 min e 51 seg	**98,03**%
Experimento 2 (**Caso de uso 2**)	Cap. 52 (CTU-13)	ESPA *(LSTM Autoencoder)*	**4 min e 24 seg**	94,7%
Experimento 3 (**Caso de uso 2**)	*Portmap* (CICDDoS2019)	ESPA *(LSTM Autoencoder)*	15 min e 23 seg	**85,61%**
Experimento 4 (**Caso de uso 2**)	Cap. 51 (CTU-13)	ESPA *(One-Class SVM)*	**31 min e 2 seg**	96,96%
Experimento 5 (**Caso de uso 2**)	Cap. 52 (CTU-13)	ESPA *(One-Class SVM)*	3 min e 22 seg	93,61%
Experimento 6 (**Caso de uso 2**)	*Portmap* (CICDDoS2019)	ESPA *(One-Class SVM)*	9 min e 26 seg	83,58%

Experimento 1 (**Caso de uso 3**)	Cap. 51 (CTU-13)	ESPA *(Agglomerative Clustering)*	30 min e 38 seg	91,4%
Experimento 2 (**Caso de uso 3**)	Cap. 52 (CTU-13)	ESPA *(K-means)*	3 min e 56 seg	92,32%
Experimento 3 (**Caso de uso 3**)	*Portmap* (CICDDoS2019)	ESPA *(Birch)*	**22 min e 05 seg**	72,41%
Experimento 4 (**Caso de uso 3**)	Cap. 17 (IoT-23)	ESPA *(K-means)*	20 min*	**100%**
-	Capturas 51 e 52 (CTU-13) e *Portmap* (CICDDoS2019)	Sistema ANTE	1 min e 00 seg	-
-	Capturas 51 (CTU-13))	Rahal et al. (2020)	5 min e 41 seg	-

*Após a infecção

A versão da abordagem ESPA equipada com técnica de aprendizado de máquina não supervisionado também evolui trabalhos anteriores, melhorando o tempo de predição. Na captura 52 da CTU-13, o melhor tempo de predição no Caso de uso 1 foi de 3 minutos e 49 segundos, enquanto a atual proposta proporcionou a predição do ataque com 3 minutos e 56 segundos de antecedência (Tabela 5.24). O Experimento 3 do Caso de uso 3 supera em tempo de predição os Experimentos 3 e 6 do Caso de uso 2. Nesses experimentos, a base de dados avaliada foi o ataque *Portmap* da base de dados CICDDoS2019. No Caso de uso 2, a versão da abordagem ESPA que maximizou o tempo de predição foi a equipada com a rede LSTM *Autoencoder*. Neste caso, a predição ocorreu 15 minutos e 23 segundos antes do lançamento do ataque. Já a versão da abordagem ESPA equipada com o algoritmo *Birch* conseguiu predizer o ataque com 22 minutos e 05 segundos antes do lançamento do ataque.

Apesar de aumentar o tempo de predição dos ataques DDoS em relação aos casos de uso anteriores, os resultados obtidos neste caso de uso indicam acurácias menores. No primeiro caso de uso, dados rotulados foram utilizados para treinar o modelo de aprendizado de máquina que a abordagem ESPA utilizou. Como existe esse conhecimento prévio, é plausível que a abordagem ESPA (Caso de uso 1) supere as versões da abordagem ESPA dos Casos de uso 2 e 3. Mesmo que a acurácia obtida nos Casos de uso 2 e 3 sejam menores do que as obtidas anteriormente (Tabela 5.24), os resultados dos Casos de uso 2 e 3 são muito relevantes. Visto que a abordagem ESPA desses casos de uso não utiliza rótulos para realizar a predição. Portanto, a abordagem ESPA pode realizar a difícil tarefa de predizer ataques DDoS sem a utilização de dados rotulados e ainda assim obter resultados competitivos.

5.5 CASO DE USO 4 - *ENSEMBLE* DE AM

A verificação da hipótese foi obtida e reforçada nos casos de usos anteriores mediante a predição dos ataques DDoS utilizando diferentes versões da abordagem ESPA. O Caso de uso 4 foi planejado para concluir a presente tese, reforçando a verificação da hipótese e avaliando uma nova versão da abordagem ESPA. Este caso de uso mantém a configuração da engenharia de sinais definida no caso de uso anterior. O diferencial deste caso de uso encontra-se na avaliação do PCA para seleção de características sem a utilização de rótulos e na utilização da central de inteligência com um *ensemble* de identificadores de *outliers*. O objetivo destas definições

é melhorar a acurácia da abordagem ESPA, realizar as predições com mais certeza e sem a interferência de ruídos nos dados. Por fim, os detalhes são descritos ao longo da seção.

5.5.1 Definição dos Experimentos do Caso de uso 4

O Caso de uso 4 possui três experimentos que avaliam a predição de ataques DDoS em bases de dados diferentes. A Tabela 5.25 apresenta todos os parâmetros usados pela abordagem ESPA nos experimentos. As definições relacionadas com a engenharia de sinais são as mesmas do Caso de uso 3 (Seção 5.4). Assim, a abordagem ESPA coletou centralizadamente, e a cada um segundo, 51 atributos do tráfego de rede da vítima do ataque DDoS usando um agente (Tabela 5.17). A central de inteligência aplicou seis indicadores estatísticos (*kurtosis*, *skewness*, CV, AC-1, AC-2 e AC-3) sobre todos os 51 atributos para criar 306 novas características por segundo usando uma janela deslizante de 10% da base de dados. Essa configuração é a mesma para todos os três experimentos. Por fim, todos os experimentos foram conduzidos em um dispositivo que possui um processador Intel Core I5, um disco rígido de um terabyte, oito gigabytes de memória, com um sistema operacional Linux Mint e com Python 3.10.12 e os resultados estão disponíveis online[19].

Tabela 5.25: Definição dos Parâmetros Usados Pela Abordagem ESPA no Caso de Uso 4

Parâmetro	**Valor**
Modo de coleta do tráfego de rede	Centralizado
Quantidade de agentes	Um
Localização dos agentes	Na rede vítima
Atributos do tráfego de rede	51 atributos baseados nos protocolos das camadas de Enlace, Rede e Transporte do modelo TCP/IP
Ciclos de captura	Um segundo
Indicadores estatísticos	*kurtosis*, *skewness*, CV, AC-1, AC-2 e AC-3
Algoritmos de aprendizado de máquina	*One-Class* SVM, *One-Class* SVM com SGD e o *Elliptic Envelope*
Seleção dos algoritmos de aprendizado de máquina	Manual
Janela deslizante	10% da base de dados
Tratamento de dados faltantes	Nenhum
Redimensionamento dos dados	Nenhum

A definição dos ciclos maliciosos e normais deste caso de uso é mesma utilizada no Caso de uso 2. Para ser considerado um **ciclo malicioso**, os *bots* precisam **enviar mais de dois pacotes**. Caso o ciclo de captura de um segundo **não possua mais de dois pacotes** enviados pelos bots, estes foram rotulados como **ciclos normais**. Por fim, é importante reiterar que neste caso de uso os rótulos são utilizados apenas para avaliar a assertividade da abordagem ESPA. Portanto, os rótulos não influenciam na geração dos modelos e tão pouco na predição dos ataques DDoS.

O primeiro diferencial deste caso de uso é a técnica de seleção de características. No caso de uso anterior, a técnica utilizada foi o *FastICA*. Neste caso de uso a técnica de seleção

[19] https://github.com/andersonneira/tese_resultados/

de características utilizadas foi o PCA. Dado um espaço n-dimensional, como uma matriz que representa os dados avaliados, o PCA encontra um subconjunto da matriz que maximiza a variância dos dados (Wold et al., 1987; Martinez e Kak, 2001). Ou seja, o PCA reduz a dimensionalidade dos dados mantendo as principais características dos dados originais (Maćkiewicz e Ratajczak, 1993). O PCA foi escolhido para este caso de uso por selecionar características sem a utilização de dados rotulados, assim como o *FastICA* (Caso de uso 3). Por fim, nos três experimentos deste caso de uso, o PCA selecionou os sete principais componentes para formar a base de dados de avaliação. Este trabalho usou sete componentes para manter a uniformidade dos experimentos.

O segundo diferencial deste caso de uso é a utilização de um *ensemble* de identificadores de *outliers* para a predição dos ataques DDoS. No Caso de uso 2 (Seção 5.3), a central de inteligência foi equipada com um modelo baseado no algoritmo *One-class* SVM. Neste caso de uso, a central de inteligência é equipada com 105 modelos baseados nos algoritmos *One-class* SVM, *One-class* SVM com Gradiente Descendente Estocástico (do inglês, *Stochastic Gradient Descent* - SGD) e o *Elliptic Envelope*. O objetivo de utilizar 105 modelos é combiná-los para predizer os ataques DDoS. A combinação dos modelos torna a central de inteligência ainda mais assertiva nas predições, evitando que tendências errôneas impactem negativamente os resultados.

O *One-class* SVM com SGD implementa uma versão do One-Class SVM usando o gradiente descendente estocástico. O objetivo é fazer com que o *One-class* SVM com SGD consuma menos recursos computacionais que o *One-class* SVM padrão[20], afetando o menos possível a assertividade dos modelos[21]. O *Elliptic Envelope* assume que os dados regulares seguem uma distribuição conhecida, como a distribuição gaussiana. Com base nessa suposição, o *Elliptic Envelope* modela o formato dos *inliners*, que representam o tráfego normal, e as observações suficientemente distantes são consideradas como *outliers*[22], que representam o tráfego de rede impactado pelos atacantes.

Dos 105 modelos, 30 são baseados no *One-class* SVM, 60 são baseados no *One-class* SVM com SGD e 15 no *Elliptic Envelope*. Para gerar os 30 modelos do *One-class* SVM, este trabalho variou os parâmetros *kernel* e *nu*. O parâmetro *kernel* assumiu os valores rbf e $sigmoid$. Para cada *kernel*, o parâmetro *nu* recebeu valores entre [0,05 e 0,20), incrementados em 0,01. Assim, são 15 modelos onde o *One-class* SVM recebe o *kernel* rbf e o *nu* varia entre [0,05 e 0,20) e outros 15 modelos onde o *One-class* SVM foi configurado com o *kernel* $sigmoid$ e o *nu* varia entre [0,05 e 0,20). Esses parâmetros foram definidos visando fornecer diversidade ao *ensemble* de identificadores de *outliers* e para lidar com o grande desbalanceamento dos dados, inerente ao problema da predição dos ataques DDoS.

O *One-class* SVM com SGD possui mais modelos, pois a tendência é que ele possua uma assertividade próxima a do *One-class* SVM tradicional, consumindo menos recursos. Este trabalho variou dois parâmetros do *One-class* SVM com SGD. O primeiro é o parâmetro *nu* que variou da mesma forma do parâmetro *nu* do *One-class* SVM padrão. Portanto, o parâmetro *nu* variou entre [0,05 e 0,20) com incrementos de 0,01. O segundo parâmetro variado foi a taxa de aprendizado. A taxa de aprendizado pode variar entre quatro opções, sendo elas: *constant*, *optimal*, *invscaling* e *adaptive*. Este caso de uso utilizou as quatro opções da taxa de aprendizado variando o parâmetro *nu* entre [0,05 e 0,20). Portanto, cada taxa de aprendizado possui 15 modelos, totalizando 60 modelos. Por fim, o *Elliptic Envelope* possui 15 modelos, pois apenas um atributo foi variado. Neste caso, o atributo *contamination* variou entre [0,05 e 0,20) para criar os 15 modelos.

[20]https://scikit-learn.org/stable/modules/sgd.html#online-one-class-svm

[21]https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDOneClassSVM.html

[22]https://scikit-learn.org/stable/modules/outlier_detection.html#outlier-detection

A Figura 5.31 ilustra o processo de combinação dos 105 modelos identificadores de *outliers*. Conforme os agentes coletam o tráfego de rede, a central de inteligência realiza a engenharia de sinais sobre os atributos do tráfego de rede a cada um segundo. Diferentemente do Caso de Uso 2, a central de inteligência não analisa (rotula/identifica os *outliers*) imediatamente após a execução da engenharia de sinais. A central de inteligência armazena temporariamente 30 ciclos de captura (30 segundos) para aplicar os 105 modelos identificadores de *outliers* sobre os 30 ciclos simultaneamente. O objetivo de armazenar 30 segundos é evitar que a predição do ataque DDoS seja excessivamente atrasada. Isso, pois somente após a coleta e a execução da engenharia de sinais dos 30 segundos é que a predição aconteceria. Armazenar temporariamente 30 ciclos permite a versão da abordagem ESPA deste caso de uso não necessitar de uma fase de treinamento dos modelos, e isso é outro importante diferencial. A cada 30 segundos, todos os modelos analisam os dados e identificam os *outliers*. Isso é possível graças à função "fit_predict" que os identificadores de *outliers* implementados pela biblioteca Scikit-Learn possuem. Portanto, a cada conjunto de 30 segundos, a central de inteligência aplica o "fit_predict" dos 105 modelos para analisar os dados, identificar os *outliers* e predizer os ataques DDoS.

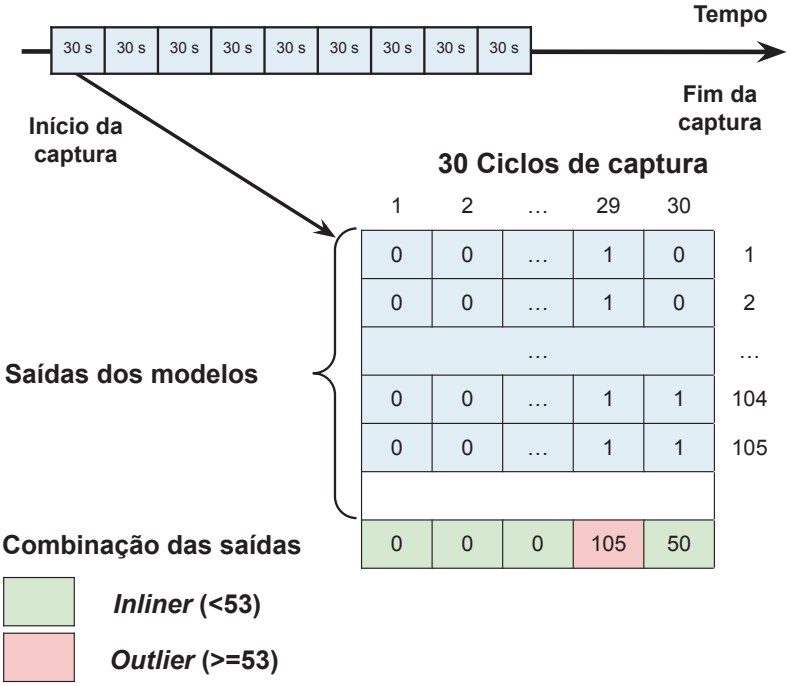


Figura 5.31: Combinação dos Resultados das Predições dos 105 Modelos Identificadores de *Outliers*

Após aplicar os modelos, a central de inteligência combina todas as saídas. As saídas dos modelos são "0" para *inliners* e "1" para *outliers*. Para que um ciclo de captura seja indicado como *outlier*, e indique a preparação de um ataque DDoS, a maioria dos modelos deve concordar que o mesmo ciclo é um *outlier*. Como são 105 modelos, no mínimo 53 devem indicar que o ciclo analisado é um *outlier*. Na Figura 5.31, as saídas dos 105 modelos são combinadas na última linha da tabela ilustrativa (linha verde e vermelha). Os zeros verificados na linha da combinação das saídas indicam que nenhum modelo marcou os ciclos de captura como *outliers*. O valor 105 indica que todos os modelos rotularam o mesmo ciclo como malicioso (*outlier*). Como a maioria concordou, esse segundo gera um alerta indicando a preparação do ataque DDoS. O último ciclo de captura não geraria um alerta, pois a maioria dos modelos concordou que o ciclo é um *inliner* e não um *outlier*, mesmo quando 50 modelos indicam o ciclo de captura como *outlier*. Por fim, este caso de uso projetou a abordagem para repetir esse processo a cada 30 segundos até o lançamento do ataque ou enquanto a abordagem estiver operando.

5.5.2 Experimento 1 do Caso de Uso 4

O Experimento 1 utiliza a captura 51 da base de dados CTU-13, a mesma utilizada nos Experimentos 1 e 4 do Caso de uso 2. **O objetivo é verificar se a versão da abordagem ESPA proposta neste caso de uso poderia atuar em um cenário onde os *bots* estão na mesma rede vítima do ataque.** Como citado anteriormente, a engenharia de sinais deste experimento é realizada com uma janela deslizante de 10% da base de dados (880 segundos), com 51 atributos do tráfego de rede (Tabela 5.17) e seis indicadores estatísticos. Como os modelos não necessitam de treinamento, a execução do experimento consistiu em aplicar a abordagem ESPA do segundo 0 até o segundo 5632, antes do lançamento do ataque. Devido aos 880 segundos da janela deslizante, o experimento aplicou os 105 modelos de detecção de *outliers* entre os segundos 880 e 5632 (lançamento do ataque).

A Tabela 5.26 apresenta os resultados do Experimento 1 do Caso de uso 4. Dentre os 4752 ciclos, a abordagem ESPA equipada com 105 modelos identificadores de *outliers* identificou corretamente dois ciclos maliciosos. O primeiro segundo corretamente identificado como malicioso por pelo menos 53 modelos foi o segundo 2648. Em outros 92 casos, a maioria dos modelos concordou erroneamente que esses ciclos de captura eram maliciosos. Dos 52 ciclos maliciosos existentes na base, 50 não foram identificados. Um total de 4608 ciclos normais foram corretamente identificados como ciclos normais. Esses resultados indicam uma acurácia de 97,01%, a precisão para classe dos ciclos maliciosos é de 2,13% e a precisão para a classe dos normais é de 98,93%. O *recall* para os ciclos maliciosos é de 3,85% e o *recall* para a classe dos ciclos normais é 98,04%. Por fim, o *F1-score* para os ciclos maliciosos é de 2,74% e de 98,48% para os ciclos normais.

Tabela 5.26: Resultados do Experimento 1 do Caso de Uso 4

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	2	92
	Ciclo Normal	50	4608

5.5.3 Experimento 2 do Caso de Uso 4

O Experimento 2 replica o experimento anterior usando a captura 52 da base de dados CTU-13, a mesma utilizada nos Experimentos 2 e 5 do Caso de uso 2. **O objetivo é verificar se a versão da abordagem ESPA proposta neste caso de uso poderia atuar em um cenário menor do que o anterior.** Por ser menor, com menos tráfego de rede, o desafio de identificar os *outliers* torna-se ainda mais complexo. A engenharia de sinais deste experimento foi realizada como a do Experimento 1. A janela deslizante também possui 10% da base de dados (97 segundos), os agentes coletam 51 atributos do tráfego de rede (Tabela 5.17) e seis indicadores estatísticos. Devido aos 97 segundos da janela deslizante, o experimento aplicou os 105 modelos de detecção de *outliers* entre os segundos 97 e 778 (lançamento do ataque).

A Tabela 5.27 apresenta os resultados do Experimento 2 deste caso de uso. Dos 681 ciclos, a abordagem ESPA equipada com o *ensemble* dos 105 modelos identificadores de *outliers* identificou corretamente apenas um ciclo malicioso no segundo 512. Neste experimento, a abordagem ESPA produziu 16 FPs, 12 FN, e 652 VN. Esses resultados indicam uma acurácia de 95,89%. A precisão para classe dos ciclos maliciosos é de 5,88% e a precisão para a classe dos normais é de 98,19%. O *recall* para os ciclos maliciosos é de 7,69% e o *recall* para a classe dos

ciclos normais é 97,60%. Por fim, o *F1-score* para os ciclos maliciosos é de 6,67% e de 97,90% para os ciclos normais.

Tabela 5.27: Resultados do Experimento 2 do Caso de Uso 4

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	1	16
	Ciclo Normal	12	652

5.5.4 Experimento 3 do Caso de Uso 4

O Experimento 3 aplica a abordagem ESPA equipada com 105 modelos identificadores de *outliers* no primeiro ataque da base de dados CICDDoS2019, utilizada nos Experimentos 3 e 6 do Caso de uso 2. **O objetivo é verificar se a versão da abordagem ESPA proposta neste caso de uso poderia atuar em um cenário onde os *bots* estão ligados à vítima pela Internet.** Assim como nos experimentos anteriores, a janela deslizante também possui 10% da base de dados (202 segundos), os agentes coletam 51 atributos do tráfego de rede (Tabela 5.17) e seis indicadores estatísticos. Devido aos 202 segundos da janela deslizante, a abordagem ESPA aplicou o *ensemble* de detecção de *outliers* entre os segundo 202 e 1484 (lançamento do ataque).

A Tabela 5.28 apresenta os resultados deste experimento. Dos 1282 ciclos, a abordagem ESPA equipada com o *ensemble* dos modelos identificadores de *outliers* identificou corretamente cinco ciclos maliciosos, o primeiro deles no segundo 916. A abordagem ESPA produziu 24 FPs, 174 FN, e 1079 VN. Esses resultados indicam uma acurácia de 84,56%. A precisão para classe dos ciclos maliciosos é de 17,24% e a precisão para a classe dos normais é de 86,11%. O *recall* para os ciclos maliciosos é de 2,79% e o *recall* para a classe dos ciclos normais é 97,82%. Por fim, o *F1-score* para os ciclos maliciosos é de 4,81% e de 91,60% para os ciclos normais.

Tabela 5.28: Resultados do Experimento 3 do Caso de Uso 4

Matriz de confusão		Classe real	
		Ciclo Malicioso	Ciclo Normal
Classe hipotética	Ciclo Malicioso	5	24
	Ciclo Normal	174	1079

5.5.5 Discussão dos Resultados do Caso de uso 4

O primeiro ponto de discussão é o tempo de predição antes do lançamento do ataque DDoS. No Experimento 1, o primeiro ciclo de captura corretamente identificado malicioso do ataque DDoS ocorreu no segundo 2648. Porém, neste caso de uso, não é possível utilizar esse segundo para calcular o tempo de predição. Pois a identificação dos *outliers* aconteceu apenas após o fechamento do conjunto de 30 segundos onde o segundo 2648 foi identificado. Assim, o conjunto de 30 segundos foi finalizado dois segundos após a coleta do segundo 2648. Portanto, este trabalho considera que o administrador de rede seria notificado no segundo 2650 da base de dados. Logo, a predição do ataque DDoS no Experimento 1 ocorreu 49 minutos e 42 segundos antes do lançamento do ataque. A Figura 5.32 ilustra o tempo de predição no Experimento 1.

No Experimento 2, o primeiro ciclo de captura acertadamente definido como ciclo malicioso foi o segundo 512. Contudo, a predição ocorreria somente cinco segundo após a

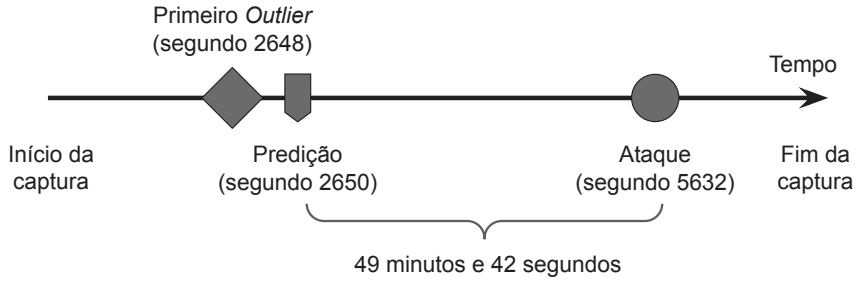


Figura 5.32: Predição do Ataque DDoS no Experimento 1 do Caso de Uso 4

identificação do segundo 512 como *outlier*. Pois é o momento em que fecha os 30 segundos da análise. Logo, a abordagem ESPA realizou a predição do ataque DDoS 4 minutos e 21 segundos antes do lançamento do ataque, conforme apresentado na Figura 5.33. A Figura 5.34 ilustra o tempo de predição no Experimento 3 (CICDDoS2019). A primeira predição correta ocorreu no segundo 916 e o fechamento dos 30 segundos de análise ocorreu seis segundos após a predição correta. Logo, no Experimento 3, a predição do ataque DDoS ocorreu 9 minutos e 22 segundos antes do lançamento do ataque.

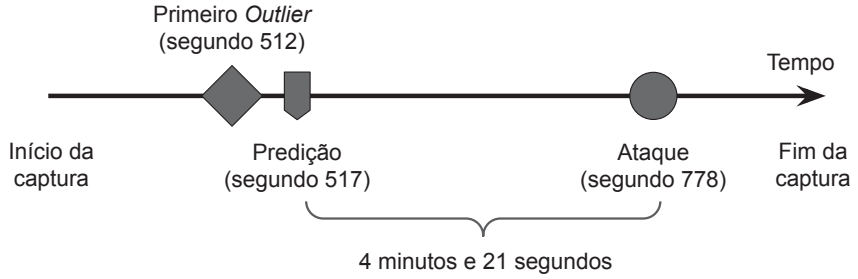


Figura 5.33: Predição do Ataque DDoS no Experimento 2 do Caso de Uso 4

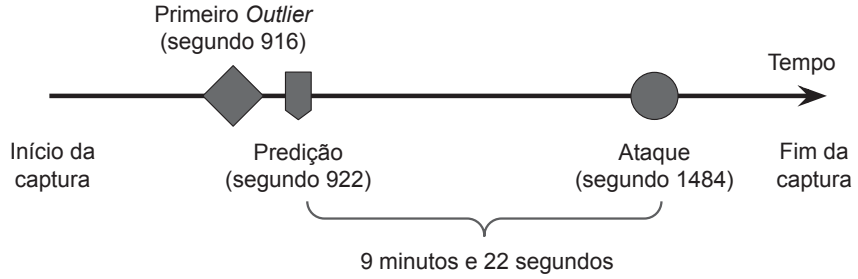


Figura 5.34: Predição do Ataque DDoS no Experimento 3 do Caso de Uso 4

Os resultados apontam que o fato da abordagem ESPA acumular 30 ciclos de captura (30 segundos) não acarreta grandes atrasos na predição. O tempo de atraso na predição foi de 2, 5 e 6 segundos nos Experimentos 1, 2 e 3, respectivamente. Mesmo que a predição atrasasse em 30 segundos, o tempo de predição verificado nos resultados ainda seria competitivo frente aos outros casos de uso e a literatura. Além disso, o armazenamento temporário de 30 ciclos de captura permite a predição sem uma fase específica para treinar o modelo, como no Caso de uso 2. Assim, a abordagem ESPA pode utilizar a função "fit_predict" dos identificadores de *outliers* para construir o *ensemble* de identificadores de *outliers*. Portanto, a versão da abordagem ESPA apresentada neste caso de uso **não** dependeu de treinamento e nem de dados rotulados para obter acurácias entre 84% e 97%, sem impactar negativamente o tempo de predição dos ataques DDoS.

Armazenar mais ciclos de captura como, por exemplo, 60 ou 100 geraria um compromisso (do inglês, *trade-off*) em relação ao atraso de predição e a melhoria dos resultados. Analisar grupos com mais ciclos de capturas poderia tornar o *ensemble* de identificadores de *outliers* mais acurado, diminuindo os FPs. Contudo, em cenários onde o ataque DDoS é lançado pouco tempo depois da infecção dos *bots*, como a base de dados utilizada no Experimento 2, é possível que esse atraso impacte negativamente na experiência do administrador de rede. Assim, o administrador de rede teria menos tempo para evitar os danos causados pelo ataque DDoS. Portanto, para os experimentos deste caso de uso, avaliar grupos de 30 segundos do tráfego de rede apresentou resultados competitivos frente aos outros casos de uso.

A abordagem ESPA equipada com o *ensemble* de identificadores de *outliers* tende a apresentar resultados confiáveis devido à colaboração dos 105 modelos. Pois é necessário que a maioria dos modelos identifique o mesmo ciclo de captura como malicioso para a predição ocorrer. Assim, para que possíveis ruídos nos dados impactem negativamente a predição dos ataques DDoS, é necessário que a maioria dos modelos seja impactada. Como eles possuem configurações variadas, nem todos os modelos irão identificar possíveis ruídos como *outliers*. Essa particularidade faz com que a abordagem possa ser menos sensível a ruídos que não representem sinais da preparação dos ataques DDoS e apresente predições confiáveis.

Os Experimentos 2 e 3 foram especialmente complexos para a abordagem ESPA equipada com o *ensemble* de identificadores de *outliers*. No Experimento 2, a abordagem ESPA gerou corretamente uma notificação da aproximação de um ataque DDoS. Isso é o suficiente para indicar o sucesso da abordagem ESPA. Contudo, outras notificações corretas poderiam reforçar a predição do ataque DDoS. Uma das causas dessa dificuldade é a escassez dos ciclos maliciosos. Nesta base de dados, os *bots* enviam mais de dois pacotes em apenas 13 segundos. Em relação ao Experimento 3, o resultado da engenharia de sinais não foi suficiente para que a abordagem ESPA pudesse identificar mais ciclos maliciosos. Isso fez com que a acurácia desse experimento fosse inferior à acurácia dos Experimentos 1 e 2. Contudo, no Experimento 3, foram identificados corretamente 5 ciclos maliciosos. Isso reforçaria a predição dos ataques DDoS. Portanto, a não identificação de vários ciclos maliciosos impacta menos negativamente a utilização da abordagem ESPA do que a geração de muitos falsos positivos.

Outra importante discussão sobre a abordagem ESPA é a sua capacidade de aprendizagem online. A aprendizagem online ocorre em soluções que possuem a capacidade de se adaptar frente a novas observações (Bitit et al., 2024). A abordagem ESPA tem o poder de adaptação frente a mudanças no tráfego de rede como uma característica intrínseca. Como a abordagem não depende de treinamento prévio nem de dados rotulados, ela pode adaptar-se a ponto de predizer ataques desconhecidos (zero-day attacks). Além disso, a abordagem ESPA aplica o PCA a cada conjunto de 30 segundos. Assim, a cada conjunto de 30 segundos, as características selecionadas pela abordagem ESPA podem ser diferentes. Isso faz com que a abordagem ESPA sempre utilize as melhores características para analisar o tráfego de rede e predizer os ataques DDoS. Portanto, a abordagem ESPA utiliza sempre as características que maximizam a predição dos ataques DDoS mesmo que os sinais da preparação dos ataques DDoS se manifestem em diferentes características ao longo da preparação do ataque.

A acurácia, a precisão e o *recall* para os ciclos normais obtidos nos experimentos são excelentes, pois se aproximam do teto de 100% sem o uso de dados rotulados nem treinamentos (Tabelas 5.26, 5.27 e 5.28). Contudo, a precisão e o *recall* para a classe dos ciclos maliciosos apresentam valores baixos em todos os experimentos. Ao menos três motivos explicam esses resultados. Primeiramente, a versão da abordagem ESPA apresentada neste caso de uso foi projetada para somente identificar os ciclos maliciosos quando um ciclo for identificado como *outlier* pela maioria dos modelos. Isso aumenta a confiança em relação às predições. Contudo,

isso faz com que a ocorrência de predições diminua. Com menos predições, menos ciclos maliciosos são identificados. Portanto, os valores relativamente baixos para a precisão e o *recall* não representam um problema, e sim refletem a estratégia utilizada no projeto da proposta para atingir os objetivos. Em segundo lugar, o fato da abordagem ESPA não depender de treinamento também auxilia na diminuição da precisão e do *recall* para a classe dos ciclos maliciosos. Novamente, essa decisão de projeto simplifica a utilização da técnica e, mesmo que menos VPs sejam gerados, o resultado geral da abordagem ESPA ainda é relevante e competitivo. Por fim, o desbalanceamento dos dados também impacta para diminuir a precisão e o *recall*. Pois são pouquíssimos os ciclos maliciosos em relação ao tráfego normal em todos os três experimentos. Assim, qualquer variação nas identificações impacta muito nos resultados obtidos.

O limiar de predição é outro ponto de discussão. O limiar de 50% + 1 modelo foi utilizado neste caso de uso para indicar que a maioria dos modelos deve obter as mesmas saídas para uma decisão ser tomada. Contudo, o administrador de rede e as equipes de segurança podem adequar esse limiar segundo suas necessidades. Caso o limiar seja maior, a abordagem proposta ficaria mais rígida e necessitaria de mais modelos para indicar que um ciclo de captura realmente é um *outlier*. Consequentemente, a tendência é que a central de inteligência gere menos alertas, aumentando os falsos negativos (*outliers* que não foram identificados). Por outro lado, diminuir o limiar faria com que menos modelos devessem concordar com a identificação do *outlier*. A tendência é que essa decisão force a geração de mais alertas, por ser mais fácil que os modelos concordem. Contudo, nem sempre a concordância de alguns modelos indica que a identificação dos *outliers* está correta. Portanto, é possível que mais falsos positivos sejam gerados (ciclos de capturas normais sendo indicadas como maliciosas).

A definição do parâmetro *nu* entre $[0,05$ e $0,20)$ é outro ponto de discussão. O parâmetro *nu* atua como um limite para a taxa de erros durante o treinamento. Ao ajustar o valor do parâmetro *nu*, é possível controlar a quantidade de pontos de dados considerados preparação dos ataques DDoS durante a fase de treinamento. Considerar modelos com o parâmetro *nu* menor que $[0,05$ pode tornar os modelos mais rígidos e fazer com que apenas os *outliers* extremamente discrepantes fossem identificados. Isso faria com que a central de inteligência gerasse menos alertas. Além disso, utilizar valores maiores que $0,20)$ poderia contribuir negativamente para a geração de falsos positivos. Como o problema da predição dos ataques DDoS é extremamente desbalanceado, os valores de *nu* entre $[0,05$ e $0,20)$ refletem isso. Portanto, os resultados obtidos indicam que os valores são adequados para a condução dos experimentos deste caso de uso.

5.5.6 Comparação dos Resultados do Caso de Uso 4 com a Literatura

A Tabela 5.29 ilustra a comparação dos resultados deste caso de uso com os casos de uso anteriores e com a literatura. É importante ressaltar que o *ensemble* de identificadores de *outliers* não demanda dados rotulados para obter os resultados apresentados neste caso de uso. Mesmo não utilizando dados rotulados, os resultados deste caso de uso aproximaram-se do Caso de uso 1, onde a abordagem ESPA foi equipada com técnicas de aprendizado de máquina supervisionado. No caso de uso 1, a melhor acurácia obtida na análise da captura 52 da base de dados CTU-13 foi de 99,69%. Enquanto no Experimento 2 do Caso de uso 4, a abordagem ESPA equipada com o *ensemble* de identificadores de *outliers* obteve 95,89% sem a utilização de dados rotulados. Além disso, o tempo de predição do Caso de uso 4 superou o melhor tempo de predição da captura 52 no Caso de uso 1. No Caso de uso 4, a abordagem predisse o ataque com 4 minutos e 21 segundos, e no Caso de uso 1, a abordagem ESPA predisse o ataque com 3 minutos e 55 segundos.

Outro importante resultado deste caso de uso em comparação com os demais é o tempo de predição na captura 51 da base de dados CTU-13. Novamente, sem a utilização de dados

rotulados, a abordagem ESPA equipada com o *ensemble* de identificadores de *outliers* superou o tempo de predição de todos os casos de uso anteriores na captura 51. Além do maior tempo de predição, a acurácia também é competitiva. A melhor acurácia obtida nos casos de uso para a captura 51 é de 98,03%. A acurácia da abordagem ESPA equipada com o *ensemble* de identificadores de *outliers* é de 97,01%. Uma pequena diferença de cerca de 1%. Contudo, a versão da abordagem ESPA do Caso de uso 4 não possui fase de treinamento. Isso é uma grande vantagem frente à versão da abordagem ESPA do Caso de uso 2. Pois o principal ponto de atenção da abordagem ESPA equipada com o LSTM *Autoencoder* configurada autonomamente pelo AutoML é o tempo de treinamento. Assim, a versão da abordagem ESPA do Caso de uso 4 propicia uma solução competitiva em relação às métricas de avaliação sem consumir tempo de treinamento e dados rotulados.

Tabela 5.29: Resumo da Comparação Entre a abordagem ESPA e a Literatura no Caso de Uso 4

Experimento	**Base de dados**	**Referência**	**Tempo predição**	**Acurácia**
Experimento 1 (**Caso de uso 1**)	Cap. 52 (CTU-13)	ESPA *(Adaboost)*	3 min e 22 seg	**99,69**%
Experimento 2 (**Caso de uso 1**)	UDP-Lag (CICDDoS2019)	ESPA *(Adaboost)*	2 min e 8 seg	**99,76%**
Experimento 3 (**Caso de uso 1**)	Cap. 52 (CTU-13)	ESPA *(MLP)*	3 min e 49 seg	98,87%
Experimento 4 (**Caso de uso 1**)	UDP-Lag (CICDDoS2019)	ESPA *(MLP)*	**3 min e 55 seg**	99,60%
Experimento 1 (**Caso de uso 2**)	Cap. 51 (CTU-13)	ESPA *(LSTM Autoencoder)*	29 min e 51 seg	**98,03**%
Experimento 2 (**Caso de uso 2**)	Cap. 52 (CTU-13)	ESPA *(LSTM Autoencoder)*	**4 min e 24 seg**	94,7%
Experimento 3 (**Caso de uso 2**)	*Portmap* (CICDDoS2019)	ESPA *(LSTM Autoencoder)*	15 min e 23 seg	**85,61%**
Experimento 4 (**Caso de uso 2**)	Cap. 51 (CTU-13)	ESPA *(One-Class SVM)*	31 min e 2 seg	96,96%
Experimento 5 (**Caso de uso 2**)	Cap. 52 (CTU-13)	ESPA *(One-Class SVM)*	3 min e 22 seg	93,61%
Experimento 6 (**Caso de uso 2**)	*Portmap* (CICDDoS2019)	ESPA *(One-Class SVM)*	9 min e 26 seg	83,58%
Experimento 1 (**Caso de uso 3**)	Cap. 51 (CTU-13)	ESPA *(Agglomerative Clustering)*	30 min e 38 seg	91,4%
Experimento 2 (**Caso de uso 3**)	Cap. 52 (CTU-13)	ESPA *(K-means)*	3 min e 56 seg	92,32%
Experimento 3 (**Caso de uso 3**)	*Portmap* (CICDDoS2019)	ESPA *(Birch)*	**22 min e 05 seg**	72,41%
Experimento 4 (**Caso de uso 3**)	Cap. 17 (IoT-23)	ESPA *(K-means)*	20 min*	**100%**
Experimento 1 (**Caso de uso 4**)	Cap. 51 (CTU-13)	ESPA *(Ensemble de outliers)*	**49 min e 42 seg**	97,01%
Experimento 2 (**Caso de uso 4**)	Cap. 52 (CTU-13)	ESPA *(Ensemble de outliers)*	4 min e 21 seg	95,89%

Experimento 3 (**Caso de uso 4**)	(CICDDoS2019)	ESPA *(Ensemble de outliers)*	9 min e 22 seg	84,56%
-	Capturas 51 e 52 (CTU-13) e *Portmap* (CICDDoS2019)	Sistema ANTE	1 min e 00 seg	-
-	Capturas 51 (CTU-13))	Rahal et al. (2020)	5 min e 41 seg	-

*Após a infecção

5.6 VANTAGENS PROPORCIONADAS PELA ABORDAGEM ESPA

Esta seção apresenta as vantagens proporcionadas pela abordagem ESPA. A literatura apresenta estudos cujo foco é identificar o ataque após o lançamento do mesmo ataque. Visto que é nesse momento onde é possível observar as consequências do ataque, como o consumo demasiado de recursos computacionais ou de largura de banda. Nesse momento, pode ser difícil evitar os danos causados pelo ataque (Zargar et al., 2013; Gupta e Dahiya, 2021). Em comparação com as soluções de detecção de ataque DDoS, as soluções de predição operam em fases anteriores ao lançamento do ataque. Assim, é possível identificar sinais da preparação de ataque futuros mesmo quando o volume de dados referente a preparação do ataque é pequeno (Abdlhamed et al., 2017). Isso ressalta a importância deste trabalho e da linha de pesquisa sobre a predição dos ataques DDoS. Uma vez que os administradores de rede são notificados sobre a ocorrência de ataques DDoS futuros, eles podem agir para evitar que os ataques causem prejuízos.

Como apresentado nos capítulos anteriores, a interrupção dos sistemas, o consumo de recursos e a extorsão são modos que os ataques DDoS podem causar prejuízos. Para ilustrar a importância de evitar os prejuízos causados pelos ataques DDoS, esta seção apresenta o tempo que algumas das principais ferramentas comerciais demandam para detectar e mitigar os ataques DDoS (Tabela 5.30). A pedido da empresa de cibersegurança Link11, a companhia de pesquisa de mercado Frost & Sullivan realizou uma comparação com o tempo de detecção e mitigação entre algumas das principais ferramentas do mercado para a detecção e mitigação dos ataques DDoS. Além da ferramenta da empresa Link11, foram avaliadas as ferramentas das empresas Netscout (Arbor DDoS Protection Solution), Cloudflare (Magic Transit) Imperva (Incapsula) e Akamai (Prolexic). Essas ferramentas foram avaliadas frente a sete diferentes ataques DDoS e todos os resultados estão presentes na Tabela 5.30.

É oportuno destacar que algumas soluções falham ao detectar e mitigar ataques DDoS. Por exemplo, as ferramentas das empresas Cloudflare, Imperva e Akamai não mitigaram o ataque de inundação do tipo HTTPS GET. Além disso, é pertinente destacar a discrepância entre o tempo de detecção e mitigação que as soluções apresentam em alguns casos. Por exemplo, no ataque do tipo TCP SYN-ACK a ferramenta que primeiro detectou e mitigou o ataque utilizou 12 segundos. Enquanto a ferramenta que consumiu mais tempo levou 8 minutos para lidar com o ataque. Os resultados indicam que a ferramenta da empresa Link11 demanda, em média, 21 segundos para detectar e mitigar os ataques DDoS. Por fim, o tempo médio das soluções para detectar e mitigar os ataques DDoS é de 2 minutos e 14 segundos (Frost e Sullivan, 2022).

A Tabela 5.31 complementa a análise anterior ao apresentar o prejuízo que os ataques DDoS podem causar em relação aos diferentes tempos de mitigação dos ataques DDoS (Corero, 2021). O ataque DDoS não causará prejuízos caso a mitigação do ataque aconteça imediatamente após o lançamento do ataque. Isso ocorre, pois os usuários não foram afetados pelo ataque.

Tabela 5.30: Tempo Médio de Detecção e Mitigação dos Ataques DDoS (adaptado de Frost e Sullivan (2022))

Ataques DDoS	Link11	Arbor	Cloudflare	Imperva	Akamai
Inundação UDP (Esgotamento de largura de banda)	21 seg.	4 min.	falhou	1 min.	>1 min.
Amplificação de DNS (Esgotamento de largura de banda)	18 seg.	4 min.	4 min.	1 min.	>1 min.
Ataque de pacote malformado (Esgotamento de recursos)	0 seg.	4 min.	4 min.	2 min.	>1 min.
TCP SYN (Esgotamento de recursos)	0 seg.	4 min.	2 min.	1 min.	5 min.
TCP SYN-ACK (Esgotamento de recursos)	12 seg.	4 min.	>1 min.	2 min.	8 min.
TCP RST (Esgotamento de recursos)	80 seg.	fail	falhou	1 min.	5 min.
Inundação HTTPS GET (Esgotamento de recursos)	18 seg.	4 min.	falhou	falhou	falhou
Tempo médio para mitigar os ataques	**21 seg.**	240 seg.	150 seg.	80 seg	180 seg.

Ataques DDoS podem ser mitigados na ordem de milissegundos caso os vetores de ataques sejam conhecidos pelos administradores de rede e pelas equipes de segurança. Ao mitigar em menos de 0 segundos completos, o serviço não é comprometido e os usuários não deixam de usá-lo. Caso a mitigação dos ataques DDoS demore mais de 3 segundos para acontecer, o custo pode variar entre 8 e 240 mil dólares para empresas com 0,1% de participação no mercado. Caso a empresa possua 1% de participação no mercado e o tempo de mitigação supere os 3 segundoso o prejuízo pode chegar a ordem de 240 milhões de dólares (Corero, 2021). Portanto, ao predizer os ataques DDoS, a abordagem ESPA provê a oportunidade das equipes de segurança evitarem os prejuízos causados pelos ataques DDoS.

Tabela 5.31: Impactos Financeiros Causados por Ataques DDoS (adaptado de Corero (2021))

Definição	**Tempo para mitigar**	**Custo de Impacto (0,1% de participação de mercado)**
Esse tempo de mitigação é curto o suficiente para não haver impacto perceptível no alvo	Imediato	$ 0,00
Vetores de ataque conhecidos que são imediatamente mitigados e não causam incidentes	>0 segundos	$ 0,00
Inclui tempo para redirecionamento para centros de depuração, análise e mitigação	3+ segundos	≈ $ 8.000
Inclui tempo para redirecionamento para centros de depuração, análise e mitigação	10+ segundos	≈ $ 36.000
Inclui tempo para redirecionamento para centros de depuração, análise e mitigação	18+ segundos	≈ $ 68.000
Inclui tempo para redirecionamento para centros de depuração, análise e mitigação	60+ segundos	≈ $ 240.000

Outra importante contribuição está relacionada com a explicabilidade dos resultados. Prover aos computadores a oportunidade de automatizar uma tarefa é muito importante para

promover a celeridade dos processos. Por exemplo, a partir da predição do ataque DDoS é possível que medidas de segurança sejam acionadas automaticamente. Contudo, é importante que os administradores de rede e as equipes de segurança compreendam os motivos que indicaram a predição dos ataques DDoS. Por esse motivo, a explicabilidade dos resultados foi uma preocupação ao longo deste trabalho. Desde os primeiros resultados é possível verificar mudanças no comportamento das novas características geradas pela engenharia de sinais e associadas à preparação dos ataques DDoS. As Figuras 5.13 e 5.20 são exemplos de gráficos em duas ou três dimensões apresentadas com o intuito de explicar a predição dos ataques DDoS. Assim, os administradores de rede têm a oportunidade de analisar os resultados e tomar as ações cabíveis com base nos gráficos gerados pela abordagem ESPA.

A distribuição da coleta do tráfego da rede é uma importante propriedade da abordagem ESPA. Essa propriedade possui três principais vantagens, sendo elas a flexibilidade na alocação de hardware, a operação contínua da abordagem mesmo que agentes deixem de operar e a diminuição de atrasos causados por gargalos na coleta. Todos os meses milhares de gigabytes de dados são criados, e a tendência é que essa quantidade de dados aumente (Barnett et al., 2018; Domo, 2023). A abordagem ESPA está preparada para a realidade da análise de grandes volumes de dados, ao ser adaptável a diferentes cenários. Por exemplo, o usuário pode preparar a abordagem ESPA para coletar apenas um ou vários atributos do tráfego de rede e transformá-los em apenas um sinal ou em vários sinais. Assim, a abordagem proposta realiza a predição de ataques com poder de processamento reduzido, ou consumindo mais recursos computacionais. Essa flexibilidade auxilia na adoção da abordagem em diferentes casos. Com a distribuição dos agentes, caso um agente pare de operar, a abordagem continua operando. O segmento de rede monitorado pelo agente não será analisado pela abordagem ESPA, porém os outros agentes e a central de inteligência continuam realizando suas tarefas. Ao descentralizar a coleta do tráfego de rede, a abordagem ESPA diminui a possibilidade da ocorrência de atrasos na coleta do tráfego da rede. Pois cada agente tem a responsabilidade de coletar apenas uma parte dos dados. Mesmo que o volume do tráfego da rede aumente, os agentes vão cooperar para manter a abordagem ESPA realizando as predições.

A abordagem ESPA também contribui com a literatura de predição de ataques apresentando uma abordagem projetada para a cooperação entre agentes distribuídos na rede (Seção 5.2). Com o processamento distribuído, os agentes analisam porções dos dados disponíveis em cada segmento de rede. Caso a predição acontecesse distribuidamente, sinais sutis poderiam passar despercebidos pela abordagem, visto que, a união dos sinais poderia representar uma evidência do possível ataque. O trabalho de Wang e Zhang (2017) é utilizado como inspiração. Eles propõem a análise de texto de redes sociais para predizer ataques DDoS. Para isso, eles coletam e analisam os textos em busca de evidências para predizer ataques. Deste modo, graças a união dos textos é possível inferir o ataque. Quando a abordagem ESPA distribui o processamento dos dados, é plausível hipotetizar que relações importantes para a identificação dos ataques sejam perdidas. A cooperação entre os agentes é utilizada para evitar a perda dessas importantes relações entre os dados. Assim, os agentes enviam apenas os sinais processados para a central de inteligência. Os sinais representam uma ínfima porção de dados quando comparado com o volume total de dados trafegados nas redes. Deste modo, a abordagem aproveita as vantagens do processamento distribuído sem perder importantes relações entre os dados e sem onerar a rede.

Uma importante vantagem proporcionada pela versão da abordagem ESPA definida no Caso de uso 4 é a independência de dados rotulados e de treinamento. Isso foi possível com a criação de um *ensemble* de identificadores de *outliers*. O *ensemble* foi fundamentado em 105 modelos baseados em três algoritmos. Os três algoritmos não utilizam dados rotulados para realizar a identificação de *outliers*. Além disso, eles possuem a função “fit_predict”.

Isso proporcionou a análise de conjuntos de dados sem a necessidade de um treinamento. A combinação de todos os modelos aliada com a engenharia de sinais auxiliam a abordagem ESPA a evitar a geração de notificações incorretas. Portanto, a versão da abordagem ESPA apresentada no Caso de uso 4 facilita a adoção em ambientes reais por não demandar rótulos e treinamentos.

A adaptação a diferentes cenários é outro ponto forte da abordagem ESPA. O projeto da abordagem ESPA permite a definição de vários parâmetros que possibilitam o uso da abordagem em vários casos. O livre posicionamento dos agentes aliados com a escolha dos atributos do tráfego de rede proporcionam a adoção da abordagem em contextos de rede diferentes. Além disso, os resultados indicam que a abordagem proposta pode beneficiar-se de *frameworks* AutoML. Isso permite utilizar o aprendizado de máquina ao máximo adaptando-se a diferentes cenários. Os resultados também mostraram ser possível predizer ataques DDoS em redes internas, na Internet e em redes IoT sem o uso de dados rotulados. Por fim, as bases de dados avaliadas possuem propriedades distintas, como tipos de ataques, quantidades de *bots* e tamanho da base.

A abordagem ESPA foi projetada para respeitar a privacidade dos usuários. Nos últimos anos, leis como GDPR, CCPA e a LGPD buscam estabelecer regras e penalidades para o uso de informações sensíveis. O intuito dessas leis é fomentar e garantir o direito à privacidade dos usuários. A motivação dessas leis é evitar que os dados dos usuários sejam utilizados indiscriminadamente para qualquer fim. Desta forma, para evitar que a abordagem ESPA infrinja a lei e desrespeite a privacidade dos usuários, a proposta não utiliza informações presentes na carga útil dos pacotes. Assim, a abordagem ESPA limita-se a manipular os dados presentes no cabeçalho dos pacotes para criar as características necessárias.

5.7 LIMITAÇÕES DA ABORDAGEM ESPA

Apesar das vantagens apresentadas anteriormente, a abordagem ESPA possui limitações que devem ser discutidas. A abordagem ESPA baseia-se na existência de mudanças nos sinais precoces de alerta (indicadores estatísticos) para criar evidências de ataques DDoS. A principal desvantagem desse tipo de sistema é a tendência de gerar FPs. Neste trabalho, o FP é quando a abordagem emite um alerta com a possibilidade da ocorrência de um ataque futuro e o ataque não acontece. Essa condição se agrava ainda mais no contexto aplicado, pois a ocorrência de um FP pode levar a equipe de segurança a executar ações sem necessidade, culminando em gasto de tempo e dinheiro desnecessariamente. Esses FPs podem ser originados devido a ruído nos dados causados por usuários que não possuem a intenção de conduzir ataques ou pelo desbalanceamento inerente ao problema. Um dos principais objetivos da cooperação entre os agentes acontecer por intermédio da central de inteligência (Caso de uso 1) é a possibilidade da central de inteligência diminuir a taxa de FPs. Além da cooperação, a utilização do AutoML (Caso de uso 2 e 3) e a seleção de características tem o objetivo de encontrar as melhores condições para a abordagem ESPA operar. Assim, a abordagem ESPA visa minimizar os erros, principalmente os FPs.

Por outro lado, os FPs também podem ser um sinal de que toda a operação para mitigar o ataque funcionou. Caso a abordagem consiga predizer corretamente o ataque e a equipe de segurança puder mitigá-lo, a vítima não receberá o ataque. Deste modo, um FP foi caracterizado, visto que a abordagem notificou sobre um ataque que não ocorreu. Porém, os prejuízos foram evitados. Portanto, a relação entre FPs e a qualidade da abordagem devem sempre ser tratadas com cautela.

O treinamento da central de inteligência com dados rotulados pode ser uma limitação da abordagem ESPA. Para atingir os resultados apresentados no Caso de uso 1 (Seção 5.2), a central de inteligência foi treinada com os dados de ataque conduzidos anteriormente. O objetivo desse treinamento foi proporcionar à central de inteligência fundamentos para que ela

conseguisse identificar sinais da preparação dos ataques. A principal limitação do treinamento com dados rotulados é a dificuldade em conseguir reproduzir essa ação em ambientes reais. Essa limitação ocorre, pois, no melhor caso, o treinamento limitar-se-ia aos tipos de ataques existentes. Assim, caso um novo tipo de ataque ocorra, é possível que a abordagem não consiga predizer o ataque. Essa limitação inspirou as configurações alternativas da abordagem ESPA apresentadas e avaliadas nos Casos de uso 2, 3 e 4 (Seções 5.3, 5.4 e 5.5).

Apesar da possibilidade de adaptação a diferentes cenários ser uma vantagem, ela dificulta a operacionalização da abordagem Um exemplo é a dificuldade em selecionar o algoritmo de aprendizado de máquina correto para alocar na central de inteligência. O uso do AutoML é uma forma de diminuir a interação humana durante a preparação da abordagem ESPA (Seção 5.3). Os resultados apontam que o aprendizado de máquina autônomo pode obter acurácias superiores a 85% ao escolher a arquitetura da rede neural que melhor se adequa ao contexto analisado. Contudo, a seleção autônoma adiciona outra dificuldade, o tempo de execução da seleção. Nos experimentos que utilizaram o AutoML realizados na Seção 5.3, o tempo necessário para configurar e treinar as redes neurais LSTM *Autoencoder* foi de 42, 34 e 28 minutos, mesmo usando apenas três características geradas pela engenharia de sinais. Apesar do elevado tempo gasto, a literatura indica que esse processo pode levar mais tempo. Lam e Abbas (2020) executaram experimentos com AutoML que consumiram entre 3 e 24 horas. Portanto, mesmo consumindo mais tempo que o ideal, a proposta requer menos tempo de processamento que alguns trabalhos presentes na literatura.

Outra limitação decorrente da necessidade da interação humana é a possibilidade da central de inteligência receber muitos dados do tráfego de rede e atrasar a predição. Como apresentado na Subseção 4.2.3, conforme aumenta a quantidade de agentes, a quantidade de atributos ou a quantidade de indicadores estatísticos, aumenta também a quantidade de informações que a central de inteligência processará. Assim, a engenharia de sinais e a predição dos ataques podem atrasar devido à quantidade exorbitante de dados. Causando um atraso na produção da identificação dos sinais da preparação dos ataques DDoS. Diminuindo o tempo que os administradores de rede teriam para se preparar para o ataque DDoS.

5.8 RESUMO

Este capítulo apresentou os casos de uso realizados para avaliar o desempenho da abordagem ESPA. Cada caso de uso avalia uma configuração diferente da abordagem proposta. O Caso de uso 1 avaliou a abordagem ESPA configurada para coletar os dados de modo distribuído, usando quatro agentes, executando a engenharia de sinais com seis indicadores estatísticos sobre um atributo do tráfego de rede. A central de inteligência foi equipada com algoritmos de aprendizado de máquina supervisionados. Nos quatro experimentos, a abordagem proposta predisse ataques DDoS com no mínimo de 2 minutos e 8 segundos antes do lançamento do ataque e com acurácia mínima de 98,87%. O Caso de uso 2 avaliou a abordagem ESPA coletando os dados centralizadamente, executando a engenharia de sinais com três indicadores estatísticos sobre três atributos do tráfego de rede. A central de inteligência foi equipada com algoritmos de aprendizado de máquina não supervisionados (LSTM *Autoencoder* e *One-Class* SVM). Nos seis experimentos, a abordagem proposta predisse ataques DDoS com no mínimo de 4 minutos e 24 segundos antes do lançamento do ataque e com acurácia mínima de 83,58%. O Caso de uso 3 avaliou a abordagem ESPA configurada para coletar os dados de modo centralizado, usando um agente, executando a engenharia de sinais com seis indicadores estatísticos sobre 51 atributos do tráfego de rede. A central de inteligência foi equipada com algoritmos de aprendizado de máquina não supervisionados. Nos quatro experimentos, a abordagem proposta predisse ataques

DDoS com no mínimo de 3 minutos e 56 segundos de vantagem e com acurácia mínima de 72,41%. As predições dos ataques DDoS ocorreram em uma rede IoT, em redes convencionais, onde os *bots* estão na mesma rede que a vítima e em cenários onde a Internet conectava os *bots* com a vítima. A abordagem ESPA obteve os resultados do Caso de uso 3 sem conhecimento prévio de ataque e em cenários com mais dados normais do que maliciosos. Isso fez com que a solução proposta pudesse predizer ataques DDoS de dia zero em um cenário desbalanceado. O Caso de uso 4 seguiu evoluindo a literatura ao propor um *ensemble* de detectores de *outliers* que não necessita de uma fase de treinamento para predizer os ataques DDoS. O objetivo é fazer a abordagem proposta produzir predições mais confiáveis. Os resultados indicam que a acurácia variou entre 84,56% e 97,01%. O tempo de predição na captura 51 da base de dados CTU-13 superou o tempo de predição de todos os experimentos dos outros casos de uso, atingindo 49 minutos e 42 segundos. Desta forma, verificou-se nos casos de uso como diferentes versões da abordagem ESPA pode predizer os ataques DDoS.

6 CONCLUSÃO

Este capítulo apresenta a conclusão do trabalho. O objetivo é ressaltar as principais contribuições e produção bibliográfica derivada deste trabalho. Além disso, esse capítulo destaca sugestões sobre a evolução da proposta. Assim, o capítulo está dividido da seguinte forma. A Seção 6.1 resume as contribuições obtidas. A Seção 6.2 apresenta os trabalhos futuros. Por fim, a Seção 6.3 apresenta a produção bibliográfica e as atividades complementares realizadas ao longo do curso.

6.1 CONTRIBUIÇÕES

A predição de ataques – ao contrário da detecção reativa de ataques–, é um mecanismo de defesa proativo que vem chamando atenção na literatura. As técnicas de predição destacam evidências de um ataque DDoS que ainda não aconteceu. O objetivo é fornecer tempo suficiente para os administradores lidarem com os ataques. A primeira contribuição deste trabalho foi apresentar o que há de mais moderno na predição de ataques DDoS. Este trabalho listou os estudos identificados que propõem soluções para a predição de ataques DDoS. Com base nestes estudos, este trabalho propôs uma classificação seguindo os principais aspectos das soluções, como tempo, arquitetura, metodologia e tipos de dados utilizados pelos estudos. Além disso, este trabalho destacou importantes questões em aberto de pesquisa, como a aplicação de algoritmos de *deep learning* modernos, a redução da dependência de dados rotulados para treinamento de modelos, a instanciação de soluções para diferentes ambientes, como IoT e SDN e a proposta de novos estudos para realizar predições de longo prazo com uma baixa taxa de erro. Por fim, este trabalho enfatiza que projetar soluções distribuídas e cooperativas ainda podem representar avanços na predição e no combate aos ataques DDoS.

A segunda contribuição deste trabalho foi a abordagem ESPA em todas as suas vertentes. O principal objetivo da abordagem é aumentar o tempo que os administradores de rede têm para lidar com ataques DDoS identificando sinais da preparação dos ataques. Para isso, este trabalho hipotetizou e avaliou o uso da teoria dos sinais precoces de alerta para criar novas características que revelem a preparação dos ataques DDoS, proporcionando a predição dos ataques DDoS. Os resultados apresentados neste manuscrito apontam a capacidade da abordagem ESPA em antecipar o ataque DDoS e prover mais tempo para combatê-los. A abordagem ESPA foi projetada para se adequar a diferentes necessidades dos administradores de rede. Portanto, os administradores de rede podem configurá-la de diferentes modos para obter o melhor da proposta.

O Caso de uso 1 contribui com a primeira solução de predição de ataques DDoS distribuída e cooperativa da literatura. Os agentes coletam o tráfego de rede distribuidamente e enviam apenas dados dos atributos coletados para a central de inteligência. A cooperação acontece quando a central de inteligência aplica a engenharia de sinais sobre todos os dados coletados e utiliza todos os dados para realizar a predição dos ataques DDoS. Outra contribuição é a verificação de que técnicas de aprendizado de máquina do tipo *shallow* e *deep learning* podem automatizar a predição dos ataques DDoS usando o resultado da colaboração de seis indicadores estatísticos sobre o tráfego de rede (engenharia de sinais).

O Caso de uso 2 apresenta contribuições relacionadas a não utilização de dados rotulados para treinar os modelos, a automatização da configuração dos modelos, explicabilidade da predição e a independência de limiares. A independência de dados rotulados ocorre quando a central de inteligência é equipada com modelos baseados nos algoritmos LSTM *Autoencoder*

ou no *One-class* SVM. Esses algoritmos aprendem as propriedades normais do tráfego de rede. Assim, eles produzem saídas utilizadas para predizer os ataques DDoS quando o tráfego da rede possuir propriedades diferentes das normais. Utilizar *Autoencoders* também é uma contribuição deste trabalho. Pois uma das questões em aberto identificadas no Capítulo 3 foi a aplicação de algoritmos de *deep learning* modernos. A base de dados Scopus indexou 447 estudos que citam os *Autoencoders* entre os anos de 1989 e 2015. Após 2016, o total de estudos que utilizam *Autoencoders* supera os 26 mil. Portanto, este trabalho aplica uma técnica de *deep learning* moderna para predizer ataques DDoS.

Ainda no Caso de uso 2, os modelos baseados na rede LSTM *Autoencoder* foram definidos autonomamente por um *framework* AutoML. Assim, os administradores de rede não precisam definir parâmetros como a quantidade de camadas ocultas ou a quantidade de neurônios. A versão da abordagem ESPA equipada com a LSTM *Autoencoder* utiliza um limiar para automatizar a predição. Isso torna a solução customizável e facilita o entendimento das predições. Contudo, definir o limiar pode não ser trivial. Além disso, o tempo utilizado para selecionar os parâmetros e treinar a LSTM *Autoencoder* podem superar as expectativas dos administradores da rede. Para solucionar essas limitações, a abordagem ESPA foi equipada com o *One-class* SVM. Nessas duas versões, a abordagem ESPA predisse ataques DDoS, fornecendo mais tempo para os administradores de rede evitarem os prejuízos. Por fim, outra importante contribuição deste caso de uso foi verificar que a abordagem ESPA pode operar em redes locais e em redes onde a Internet conecta os *bots* e a vítima.

O Caso de uso 3 expande a contribuição do caso de uso anterior ao não utilizar dados rotulados para realizar a predição de ataques DDoS de dia zero, focando na explicabilidade dos resultados e configurando autonomamente os modelos. Os resultados indicam que a predição do ataque DDoS de dia zero é possível, pois a abordagem ESPA não utiliza conhecimentos prévios dos ataques ou das redes. A explicabilidade dos resultados decorre da representação em três dimensões dos agrupamentos que diferenciam o tráfego normal do tráfego onde os *bots* enviam pacotes. A figura em três dimensões ainda foi associada à teoria dos sinais precoces de alerta para justificar as possíveis mudanças futuras (Figuras 5.25 e 5.29 da Subseção 5.4.6). Este caso de uso também avaliou a seleção de características não supervisionada de 306 novas características geradas por seis indicadores estatísticos aplicados em 51 atributos do tráfego de rede. A seleção de características aliada ao AutoML não supervisionado proposto neste trabalho aumentaram a acurácia da predição de ataque DDoS. Por fim, este caso de uso estende as contribuições do caso de uso anterior ao mostrar que a abordagem ESPA pode predizer ataques DDoS em redes IoT, além de redes internas e Internet.

As contribuições do Caso de uso 4 também independem de dados rotulados. O PCA foi utilizado para realizar a seleção de características. A abordagem ESPA foi equipada com um *ensemble* de identificadores de *outliers*. Assim, esse trabalho está na vanguarda da aplicação de *ensembles* de identificadores de *outliers* para a predição de ataques DDoS. Outra importante contribuição dessa versão da abordagem ESPA é que ela independe de treinamento. Assim, a abordagem ESPA pode operar em diferentes cenários, sem necessitar de conhecimentos prévios e rótulos. Outra vantagem em aplicar o *ensemble* de identificadores de *outliers* é a possibilidade de tornar a abordagem ESPA menos suscetível a erros e, consequentemente, mais confiável. Por fim, ao aplicar o PCA a cada conjunto de 30 segundos, a abordagem ESPA torna-se adaptável frente a mudanças no tráfego de rede. Pois, a abordagem ESPA utiliza as melhores características para analisar o tráfego de rede e predizer os ataques DDoS. Mesmo que os sinais da preparação dos ataques DDoS se manifestem em diferentes características ao longo da preparação do ataque.

Com os resultados obtidos nos casos de uso, a abordagem ESPA enquadra-se na categoria predição de curto prazo para o aspecto temporal (Capítulo 3). Isso ocorre, pois as evidências

obtidas indicam que as predições ocorrem com menos de 24 horas de antecedência em relação aos ataques. A abordagem ESPA coopera no processamento dos dados coletados centralizadamente ou distribuidamente. Desde modo, a abordagem ESPA é cooperativa em relação ao aspecto arquitetônico no sentido de poder operar centralizadamente ou distribuidamente. Os casos de uso avaliaram vários algoritmos de aprendizado de máquina para equipar a central de inteligência. Deste modo, para o aspecto metodológico a classificação é aprendizado de máquina. Este trabalho utilizou atributos baseados no tráfego de rede para realizar a predição dos ataques DDoS. Assim, o tráfego de rede é a classificação para o aspecto de dados. Esse ramo, combinação dos aspectos curto prazo, cooperativo, aprendizado de máquina e tráfego de rede, ainda não estava presente na Figura 3.1, fato esse que ressalta o diferencial do trabalho proposto frente a literatura. Por fim, a Figura 6.1 atualiza a figura anterior (Figura 3.1)) adicionando o novo ramo.

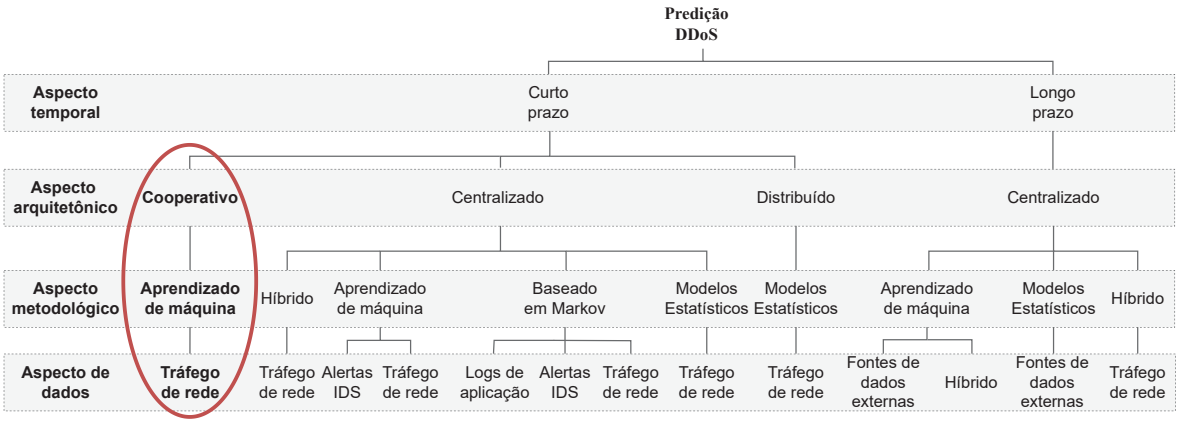


Figura 6.1: Atualização da Classificação das Soluções para a Predição de Ataque DDoS

Os resultados obtidos com a abordagem ESPA indicam a validade da hipótese avaliada neste trabalho. A hipótese definida neste trabalho é: ***É possível identificar os sinais da preparação dos ataques usando a teoria dos sinais precoces de alerta***. Ao manipular os atributos do tráfego de rede usando os sinais precoces de alerta, a abordagem conseguiu predizer ataques DDoS em redes locais, na Internet e em redes IoT. A abordagem foi avaliada usando diferentes tipos de aprendizado de máquina, atributos do tráfego de rede e indicadores estatísticos e bases de dados. Dentre os resultados apresentados, é oportuno ressaltar que as predições dos ataques ocorreram até 49 minutos antes do lançamento do ataque (Caso de uso 4, Seção 5.5), com uma acurácia máxima de 100% (Caso de uso 3, Seção 5.4). Esses resultados validam a hipótese de identificar sinais da preparação dos ataques DDoS usando a teoria dos sinais precoces de alerta.

Os resultados obtidos com a abordagem ESPA e a validação da hipótese possibilitam responder o problema de pesquisa definido neste trabalho. O problema tratado é: ***se existirem sinais da preparação dos ataques DDoS, como identificá-los?***. Os resultados indicam que aplicar a teoria dos sinais precoces de alertas sobre o tráfego de rede é uma forma de identificar sinais da preparação dos ataques DDoS. Isso ocorre, pois a teoria dos sinais precoces de alerta apresenta variações nos valores dos indicadores estatísticos calculados sobre o tráfego de rede (Subseção 4.2.4). Essas variações foram relacionadas com o tráfego originado pelos *bots* que, posteriormente, lançaram os ataques DDoS. Portanto, a teoria dos sinais precoces de alerta podem ser usadas para identificar sinais da preparação dos ataques, caso existam esses sinais.

6.2 TRABALHOS FUTUROS

Devido à extensão e a complexidade do tema abordado, a proposta apresentada proporciona trabalhos futuros com o intuito de evoluir esta e outras pesquisas sobre a predição dos ataques DDoS. Os trabalhos futuros definidos nesta seção são importantes para a evolução da abordagem ESPA bem como para a evolução do estado da arte relacionado a predição de ataques DDoS. Contudo, alguns trabalhos futuros demandam tempo para a plena condução das análises necessárias. Assim, esta subseção apresenta as oportunidades de evolução identificadas pelo autor. Elas serão priorizadas em futuras pesquisas.

Propor sinais precoces de alerta específicos para predizer ataques DDoS é um trabalho futuro que pode evoluir a literatura. O primeiro motivador para esse trabalho é a possibilidade de melhorar os resultados obtidos. Isso ocorre, pois os sinais precoces de alerta utilizados nos experimentos são genéricos. Portanto, novos sinais precoces de alerta que considerem as especificidades das redes de computadores e dos ataques DDoS podem produzir melhores resultados para a predição de ataques DDoS. Além de melhorar a qualidade da predição, é oportuno que os sinais precoces de alerta indiquem a probabilidade do ataque ocorrer. Isso facilitará o uso dos sistemas, visto que os administradores de rede podem tomar diferentes ações conforme as probabilidades variam. Outra possibilidade é que, utilizando sinais precoces de alerta específicos para ataque DDoS, seja possível estimar a magnitude e o momento do ataque. Novos sinais precoces de alerta podem viabilizar a produção de várias soluções com características específicas para predizer os ataques DDoS. Essas características auxiliarão os administradores de rede na difícil tarefa de evitar danos sofridos pelos ataques.

Outro importante trabalho futuro considera a possibilidade de projetar novas formas de colaboração entre os agentes (Seção 5.2). Uma opção de colaboração seria os agentes emitirem pré-alertas (mensagens) para os administradores e para os agentes com o intuito de melhorar os resultados da predição. Um modo para emitir os pré-alertas é definir limites de variação para os sinais precoces de alerta. Quando o limite for superado, o agente opcionalmente pode enviar uma mensagem para o administrador de rede informando que no seguimento de rede analisado foi identificado algum sinal que pode preceder um ataque. O agente que identificou o sinal pode avisar todos os outros agentes. O objetivo é fazer com que todos os agentes saibam da possibilidade da existência de um ataque. Os agentes podem mudar os limites tornando-os mais sensíveis a variações. Além disso, os agentes podem começar a analisar mais características para colaborar com a central de inteligência fornecendo mais sinais, possibilitando predições mais acuradas. Outra vantagem dessa colaboração é identificar quais segmentos da rede podem ter relação com o ataque.

A seleção de atributos coletados no tráfego de rede para a predição e a seleção dos melhores sinais precoces de alerta podem contribuir para a predição dos ataques DDoS. No Caso de uso 3, este trabalho utilizou o *FastICA* com o intuito de identificar os sinais para maximizar a acurácia das predições. No Caso de uso 4, a abordagem ESPA foi equipada com o PCA visando encontrar os principais componentes que representam a base de dados e maximizar a predição dos ataques DDoS. Para complementar esses resultados, a seleção de atributos para detecção de ataques DDoS pode ajudar a melhorar ainda mais os resultados obtidos. Como este é um tema tratado pela literatura, é possível começar as análises para a seleção de atributos coletados no tráfego de rede para a predição tendo como modelo estudos que realizaram a seleção de atributos para a detecção de ataques DDoS.

A seleção dos melhores atributos coletados no tráfego de rede está diretamente ligada com a seleção dos sinais precoces de alerta. Pois, o atributo do tráfego de rede pode ser discriminante e o sinal precoce de alerta não conseguir refletir isso em sinais capazes de

evidenciar a possibilidade da ocorrência de ataque DDoS. Deste modo, identificar os sinais precoces de alerta relevantes para a predição de ataques DDoS é importante. Pois o conjunto apropriado de sinais precoces de alerta evitam gasto de tempo processando atributos que não irão auxiliar na tarefa de predição, além de evitar que sinais inapropriados atrapalhem os modelos de predição e causem erros.

Um trabalho futuro complementar é a avaliação em outras bases de dados. As bases de dados utilizadas nos casos de uso rotulam os *bots* e o lançamento dos ataques DDoS. A definição do lançamento do ataque auxilia na avaliação da predição dos ataques DDoS, pois o foco das análises propostas é o tráfego de rede anterior ao lançamento do ataque para predizê-los. Por meio do rótulo dos *bots*, pode-se verificar se a abordagem ESPA identificou indícios da preparação do ataque antes do seu lançamento e relacioná-los com as ações dos *bots*. Portanto, a correta rotulagem dos dados permite verificar se o sistema proposto consegue identificar sinais da preparação dos ataques. Desta forma, para que a base de dados possa ser utilizada, elas devem apresentar alguma ação relacionada com a preparação dos ataques e rotulá-las. Exemplos da preparação dos ataques são a comunicação dos *bots* ou alguma ação anterior ao ataque realizada pelo atacante, como a infecção dos dispositivos ou algum teste de ataque. Poucas bases de dados suprem esses requisitos. Em geral, as bases de dados candidatas a participarem da avaliação apresentam apenas o tráfego malicioso após o lançamento do ataque ou não rotulam devidamente os dados. Isso limita as opções de avaliação. Assim, trabalhos futuros podem focar também na produção de bases de dados que possuam propriedades das redes atuais e apresentem alguma ação de preparação dos ataques.

Evitar erros de predição dos ataques DDoS é imprescindível. A ocorrência de predições equivocadas diminui a confiança sobre a abordagem. Portanto, pesquisar e propor maneiras para evitar os erros deve ser um trabalho futuro recorrente. Este trabalho hipotetiza que a união de diferentes soluções com aspectos diferentes pode evitar erros de predições. Por exemplo, usar diferentes fontes de dados pode ser benéfico para as soluções. Visto que, a preparação do ataque pode não ser verificável por uma fonte de dados, mas pode ser por outra. Caso seja possível verificar a preparação do ataque nas duas fontes de dados, a confiança na correta predição será elevada. Outra oportunidade para evitar erros é a utilização de algoritmos de aprendizado profundo ainda mais modernos que os *Autoencoders*. Assim, pesquisas futuras irão equipar a central de inteligência com modelos baseados nas redes neurais do tipo *Transformers*.

Substituir a teoria dos sinais precoces de alerta por uma teoria alternativa também é um trabalho futuro. Isso é possível, pois a engenharia de sinais da abordagem ESPA foi projetada para aplicar outras teorias sobre o tráfego de rede. A possibilidade da abordagem aceitar outras teorias foi pouco explorada no Capítulo 4, pois o foco deste trabalho é avaliar a teoria dos sinais precoces de alerta. Além disso, não explorar essa possibilidade permitiu focar apenas na teoria avaliada e evitar possíveis erros. Contudo, a abordagem ESPA pode utilizar teorias alternativas ao invés da teoria dos sinais precoces de alerta. Essas teorias alternativas podem apresentar resultados competitivos frente aos apresentados neste trabalho. Um primeiro esforço nesse sentido foi realizado nos estudos de Albano et al. (2023) e Borges et al. (2024), onde a teoria dos padrões ordinais substituiu a teoria dos sinais precoces de alerta. Contudo, trabalhos futuros podem avaliar a abordagem ESPA aplicando novas teorias sobre o tráfego de rede.

Avaliar e melhorar a eficiência da abordagem proposta é outro trabalho futuro. A hipótese deste trabalho foca na possibilidade da teoria dos sinais auxiliarem na identificação de sinais da preparação dos ataques DDoS. Portanto, o trabalho apresentado foca em verificar a eficácia da proposta em predizer os ataques para verificar a validade da hipótese. Contudo, avaliar a eficiência também é importante para a evolução da pesquisa. Este trabalho mostrou preocupação com o tema quando apresentou a análise da complexidade da central de inteligência

e quando optou por utilizar mais modelos baseados no *One-class* SVM com SGD do que com base no *One-class* SVM padrão. Assim, o Caso de uso 4 utilizou mais modelos que possuem menor complexidade. Contudo, evoluções e avaliações focadas na eficiência da abordagem ESPA serão conduzidas para proporcionar novas evoluções com o intuito de criar abordagens ainda melhores. Por exemplo, a utilização de soluções relacionadas com o *Big Data* podem auxiliar na melhora da eficiência da proposta.

6.3 PRODUÇÃO BIBLIOGRÁFICA E ATIVIDADES COMPLEMENTARES

Esta seção apresenta a produção bibliográfica e as atividades complementares realizadas durante o período do curso. Os artigos e atividades apresentadas nesta seção contribuíram para a maturidade da proposta apresentada neste manuscrito. Os artigos publicados em periódicos são:

- Anderson Bergamini de Neira, Burak Kantarci e Michele Nogueira (2023). *Distributed denial of service attack prediction: Challenges, open issues and opportunities*. **Computer Networks**, 222:109553.
- Anderson Bergamini de Neira, Alex Medeiros e Michele Nogueira (2023). *An intelligent system for DDoS attack prediction based on early warning signals*. **IEEE Transactions on Network and Service Management (IEEE TNSM)**, 20(2):1254–1266.
- Alex Medeiros Araujo, Anderson Bergamini de Neira, Michele Nogueira (2022). *Autonomous machine learning for early bot detection in the internet of things*. **Digital Communications and Networks (DCN)** 9(6):1301–1309.

Os trabalhos publicados em anais de congressos internacionais são:

- Anderson Bergamini de Neira, Ligia Borges, Alex Medeiros e Michele Nogueira (2023). *Unsupervised feature engineering approach to predict DDoS attacks*. Em **IEEE Global Communications Conference (IEEE Globecom)**, Malaysia. IEEE.
- Davi Brito, Anderson Bergamini de Neira, Ligia Borges e Michele Nogueira (2023). *An Autonomous System for Predicting DDoS Attacks on Local Area Networks and the Internet.* **Em IEEE Latin-American Conference on Communications (LATINCOM)**, páginas 1–6.
- Gabriel Lucas F. M. e Silva, Anderson Bergamini de Neira e Michele Nogueira (2022). *A Deep Learning-based System for DDoS Attack Anticipation*. Em 2022 **IEEE Latin-American Conference on Communications (LATINCOM)**, páginas 1–6.
- Alex Medeiros Araujo, Anderson Bergamini de Neira e Michele Nogueira. *Lifelong Autonomous Botnet Detection*. **Em 2022 IEEE Global Communications Conference (IEEE Globecom)**, Rio de Janeiro, Brazil, 2022, pp. 3742-3747.
- Anderson Bergamini de Neira, Alex Medeiros Araújo e Michele Nogueira. *Early botnet detection for the Internet and the Internet of Things by autonomous machine learning*. **Em 2020 16th International Conference on Mobility, Sensing and Networking (MSN)**, páginas 516–523, Tokyo, Japan, 2020a. IEEE.

- Ligia Borges, Anderson Bergamini de Neira, Lucas Albano e Michele Nogueira. *Multifaceted DDoS Attack Prediction by Multivariate Time Series and Ordinal Patterns*, **2024 IEEE International Conference on Communications Workshops (ICC Workshops)**, Denver, CO, USA, 2024.

Os trabalhos publicados em anais de congressos nacionais são:

- Anderson Bergamini de Neira, Ligia Borges, Alex Medeiros e Michele Nogueira (2023). Engenharia de sinais precoces de alerta para a predição de ataques DDoS. **Em Anais do XXVIII Workshop de Gerência e Operação de Redes e Serviços (WGRS)**, páginas 139–152, Porto Alegre, RS, Brasil. SBC.

- Davi Brito, Anderson Bergamini de Neira, Ligia Borges, Alex Medeiros e Michele Nogueira (2023). Um sistema autônomo para a predição de ataques de DDoS em redes locais e Internet. **Em Anais do XXVIII Workshop de Gerência e Operação de Redes e Serviços (WGRS)**, páginas 29–42, Porto Alegre, RS, Brasil. SBC.

- Matheus Henrique dos Santos Lima, Anderson Bergamini de Neira, Ligia Borges e Michele Nogueira (2023). Predição não-supervisionada de ataques DDoS por sinais precoces e *One-class* SVM. Em **23ºSimpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)**, Juiz de Fora, MG.

- Lucas Albano, Ligia Borges, Anderson Bergamini de Neira e Michele Nogueira (2023). Predição de Ataques DDoS pela Correlação de Séries Temporais via Padrões Ordinais. Em **23ºSimpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)**, Juiz de Fora, MG.

- Caio Maciel, Anderson Bergamini de Neira, Ligia Borges e Michele Nogueira (2023). Detecção de Botnets em Dispositivos IoTs baseado em LSTM Autoencoder. Em **23ºSimpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)**, Juiz de Fora, MG.

- Gabriel Lucas F. M. e Silva, Anderson Bergamini de Neira e Michele Nogueira (2022). Aprendizado profundo para a predição de ataques de negação de serviço distribuído. **Em Anais do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)**, páginas 475–488, Porto Alegre, RS, Brasil. SBC.

- Anderson Bergamini de Neira, Alex Medeiros e Michele Nogueira. Identificação antecipada de botnets por aprendizagem de máquina. **Em Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)**, páginas 896–909, Porto Alegre, RS, Brasil, 2020. SBC.

- Lucas Albano, Ligia Borges, Anderson Bergamini de Neira e Michele Nogueira (2024). Seleção de Características na Predição de Ataques DDoS com Transformação em Padrões Ordinais. Em **Workshop de Gerência e Operação de Redes e Serviços (WGRS)**, Niterói, RJ.

- Leonardo Henrique De Melo Leite, Michele Nogueira, Anderson Bergamini de Neira, Douglas Henrique Siqueira Abreu e Herbert Ramos. Uma Visão Sistemática Da Resiliência Das Redes De Comunicação Operativa Utilizadas No Sistema Elétrico De Potência, Suas Vulnerabilidades E Métodos De Proteção E Mitigação Frente A Ataques Cibernéticos. **Em XXVII Seminário Nacional de Produção e Transmissão de Energia Elétrica (SNPTEE), Brasília, 2023**.

Os seguintes artigos foram aceitos para a publicação, mas ainda não foram publicados:

- Matheus Henrique dos Santos Lima, Anderson Bergamini de Neira, Ligia Borges e Michele Nogueira (2024). Automatização da Seleção de Modelos Não Supervisionados na Predição de Ataques DDoS. Em **24º Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais**, São José dos Campos, São Paulo.
- Anderson B. de Neira, Ligia F. Borges, Daniel M. Batista e Michele Nogueira. *Unsupervised AutoML and Dimensionality Reduction for DDoS Attack Prediction.* In: **IEEE Latin-American Conference on Communications (IEEE Latincom)**, Medellin, Colômbia.

O seguinte minicurso foi aceito para a publicação, mas ainda não foi publicado:

- Michele Nogueira, Ligia Borges, Anderson Bergamini de Neira, Lucas Albano e Kristtopher Coelho (2024). Ciência de Dados Aplicada à Cibersegurança: Teoria e Prática. Em **24º Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais**, São José dos Campos, São Paulo.

Dois artigos estão em preparação para a submissão:

- Anderson Bergamini de Neira, Ligia Borges e Michele Nogueira (2024). *A Novel Ensemble of Outlier Identification Techniques for DDoS Attack Prediction* (Título provisório). Em **IEEE *Access***.
- Ligia Borges, Anderson Bergamini de Neira, Lucas Albano e Michele Nogueira. *Unsupervised Online AutoML for DDoS Attack Prediction by Heterogeneous Data Correlation,* **2024 IEEE International Conference on Communications (ICC 25)**, Montreal, Canada, 2025.

As participações em projetos de pesquisas e comitês foram fundamentais para a obtenção dos resultados apresentados neste manuscrito. Elas são:

- 2019/2021 - RNP/GT-PERISCOPE: Uma Ferramenta para Predição de Ataques DDoS por Meio da Identificação Precoce de *Botnets* (Bolsista).
- 2021/2022 - RNP/GT-ARQUIMEDES: Uma Ferramenta para Se Esquivar de Vazamentos de Informação na IoT (Voluntário).
- 2022/2023 Huawei/Fundação para Inovações Tecnológicas (FITec) - Projeto Desenvolvimento de Estudos para aplicações relacionadas à Segurança Cibernética para o Setor Elétrico Brasileiro (Bolsista).
- 2023/2024 - MCTIC/FAPESP MENTORED: da modelagem à experimentação - predizendo e detectando ataques DDoS e zero-day (Bolsista TT-5).
- 2024/Atual - INEP/RNP: Prospecção e implantação de soluções de dados para educação (Bolsista).

Prêmios recebidos durante o curso:

- Destaque Núcleo de Redes Sem Fio e Redes Avançadas (NR2) do ano 2021.

- Melhor avaliador do Comitê de Artefatos do WGRS 2024.

O autor deste trabalho participou do registro de três patentes relacionadas a programas de computadores:

- MONTEVERDE, W. A. ; NEIRA, A. B. ; ARAÚJO, A. M. ; LIMA, M. N. ; TAVARES, A. C. J. ; COELHO, K. K. . SHERLOCK-X-DETECÇÃO: MÓDULO DE DETECÇÃO DE BOTNETS. 2020. Patente: Programa de Computador. Número do registro: BR512021002081-2, data de registro: 30/06/2020, título: "SHERLOCK-X-DETECÇÃO: MÓDULO DE DETECÇÃO DE BOTNETS", Instituição de registro: INPI - Instituto Nacional da Propriedade Industrial.
- MONTEVERDE, W. A. ; NEIRA, A. B. ; ARAÚJO, A. M. ; LIMA, M. N. . SHERLOCK-X-GESTÃOINCIDENTES: MÓDULO DE GESTÃO DE INCIDENTES. 2020. Patente: Programa de Computador. Número do registro: BR512021002080-4, data de registro: 30/06/2020, título: "SHERLOCK-X-GESTÃOINCIDENTES: MÓDULO DE GESTÃO DE INCIDENTES", Instituição de registro: INPI - Instituto Nacional da Propriedade Industrial.
- MONTEVERDE, W. A. ; NEIRA, A. B. ; ARAÚJO, A. M. ; LIMA, M. N. . SHERLOCK-X-INDICAÇÃORISCO: MÓDULO DE INDICAÇÃO DE RISCO EM TEMPO REAL. 2020. Patente: Programa de Computador. Número do registro: BR512021002077-4, data de registro: 30/06/2020, título: "SHERLOCK-X-INDICAÇÃORISCO: MÓDULO DE INDICAÇÃO DE RISCO EM TEMPO REAL", Instituição de registro: INPI - Instituto Nacional da Propriedade Industrial.

Por fim, o autor deste trabalho participou da coorientação na condução de Iniciação Científica e/ou na elaboração do Trabalho de Conclusão de Curso de alunos de graduação da Universidade Federal de Minas Gerais com a orientação da Professora Doutora Michele Nogueira Lima. Os alunos coorientados pelo autor deste trabalho são:

- Gabriel Lucas Freire Martins (Trabalho de Conclusão de Curso - Finalizado).
- Maíla Ferreira Silva (Trabalho de Conclusão de Curso - Finalizado).
- Luís Gustavo Costa da Silva (Iniciação Científica e Trabalho de Conclusão de Curso - Finalizado).
- Caio Alexandre Campos Maciel (Trabalho de Conclusão de Curso - Finalizado).
- Davi Brito (Iniciação Científica - Finalizado).
- Matheus Henrique dos Santos Lima (Iniciação Científica - Finalizado).
- Lucas Albano (Iniciação Científica - Em andamento).
- Caroline Campos Carvalho (Iniciação Científica - Em andamento).
- Giovanni Soares (Trabalho de Conclusão de Curso - Em andamento).

REFERÊNCIAS

Abaid, Z., Sarkar, D., Kaafar, M. A. e Jha, S. (2016). The early bird gets the botnet: A markov chain based early warning system for botnet attacks. Em *2016 IEEE 41st Conference on Local Computer Networks (LCN)*, páginas 61–68, Dubai, United Arab Emirates. IEEE.

Abdlhamed, M., Kifayat, K., Shi, Q. e Hurst, W. (2017). *Intrusion Prediction Systems*, páginas 155–174. Springer International Publishing, Cham.

Abu Rajab, M., Zarfoss, J., Monrose, F. e Terzis, A. (2006). A multifaceted approach to understanding the botnet phenomenon. Em *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, IMC '06, página 41–52, New York, NY, USA. ACM.

Adegboyega, A. (2015). An adaptive score model for effective bandwidth prediction and provisioning in the cloud network. Em *2015 IEEE Globecom Workshops (GC Wkshps)*, páginas 1–7, San Diego, CA, USA. IEEE.

Adobe (2021). Adobe digital economy index: COVID-19 report. Acessado em: 08/2021. https://blog.adobe.com/en/publish/2021/03/15/adobe-digital-economy-index-covid-19-report.html.

Alam, M. M., Arafat, M. Y. e Ahmed, F. (2015). Study on auto detecting defence mechanisms against application layer DDoS attacks in SIP server. *Journal of Networks*, 10(6):344–352.

Alarifi, S. e Wolthusen, S. D. (2013). Robust coordination of cloud-internal denial of service attacks. Em *2013 International Conference on Cloud and Green Computing*, páginas 135–142, Karlsruhe, Germany. IEEE.

Albano, L., Borges, L., Neira, A. e Nogueira, M. (2023). Predição de ataques DDoS pela correlação de séries temporais via padrões ordinais. Em *Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2023)*, Juiz de Fora, MG.

Aldweesh, A., Derhab, A. e Emam, A. Z. (2020). Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowledge-Based Systems*, 189:105124.

Alhanahnah, M., Lin, Q., Yan, Q., Zhang, N. e Chen, Z. (2018). Efficient signature generation for classifying cross-architecture IoT malware. Em *2018 IEEE Conference on Communications and Network Security (CNS)*, páginas 1–9, Beijing, China. IEEE.

Ali, M. Q. e Al-Shaer, E. (2013). Configuration-based IDS for advanced metering infrastructure. Em *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, página 451–462, New York, NY, USA. ACM.

Amarjyoti, S. (2017). Deep reinforcement learning for robotic manipulation - the state of the art. *CoRR*, abs/1701.08878.

Amer, M., Goldstein, M. e Abdennadher, S. (2013). Enhancing one-class support vector machines for unsupervised anomaly detection. Em *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, ODD '13, página 8–15, New York, NY, USA. Association for Computing Machinery.

Amini, P., Araghizadeh, M. A. e Azmi, R. (2015). A survey on botnet: Classification, detection and defense. Em *2015 International Electronics Symposium (IES)*, páginas 233–238, Surabaya, Indonesia. IEEE.

An, L. e Ahmed, S. E. (2008). Improving the performance of kurtosis estimator. *Computational Statistics & Data Analysis*, 52(5):2669–2681.

Anggraini, S., Wijaya, S. K. e Daryono (2021). Earthquake detection and location for earthquake early warning using deep learning. *Journal of Physics: Conference Series*, 1951(1):012056.

Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J. A., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K. e Zhou, Y. (2017). Understanding the Mirai botnet. Em *26th USENIX Security Symposium (USENIX Security 17)*, páginas 1093–1110, Vancouver, BC. USENIX Association.

Anuar, S., Ahmad, N. A., Sahibuddin, S., Ariffin, A., Saupi, A., Zamani, N. A., Jeffry, Y. e Efendy, F. (2018). Modeling malware prediction using artificial neural network. Em Fujita, H. e Herrera-Viedma, E., editores, *New Trends in Intelligent Software Methodologies, Tools and Techniques - Proceedings of the 17th International Conference SoMeT_18, Granada, Spain, 26-28 September 2018*, volume 303 de *Frontiers in Artificial Intelligence and Applications*, páginas 240–248, GRANADA, SPAIN. IOS Press.

Apple (2021). Apple security bounty. Acessado em: 07/2021. https://developer.apple.com/security-bounty/.

Araújo, A. M., de Neira, A. B. e Nogueira, M. (2022). Lifelong autonomous botnet detection. Em *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, páginas 3742–3747, Brasil.

Armor (2018). Armor's 'black market' report highlights the big business of cybercrime. Acessado em: 07/2021. https://www.armor.com/resources/press-release/black-market-report-highlights-cybercrime/.

Arteaga, J. e Mejia, W. (2017). CLDAP reflection DDoS. Acessado em: 07/2021. https://www.akamai.com/us/en/resources/our-thinking/threat-advisories/connection-less-lightweight-directory-access-protocol-reflection-ddos-threat-advisory.jsp.

Asosheh, A. e Ramezani, N. (2008). A comprehensive taxonomy of DDoS attacks and defense mechanism applying in a smart classification. *WSEAS Transactions on Computers*, 7(4):281–290.

Avizienis, A., Laprie, J.-C., Randell, B. e Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33.

AWS-Shield (2020). Threat landscape report – Q1 2020. Acessado em: 08/2021. https://aws-shield-tlr.s3.amazonaws.com/2020-Q1_AWS_Shield_TLR.pdf.

Barnett, T., Jain, S., Andra, U. e Khurana, T. (2018). Cisco visual networking index (VNI), complete forecast update, 2017–2022.

Acessado em: 06/2021. https://www.cisco.com/c/dam/m/en_us/network-intelligence/service-provider/digital-transformation/knowledge-network-webinars/pdfs/1211_BUSINESS_SERVICES_CKN_PDF.pdf.

Barreno, M., Nelson, B., Joseph, A. D. e Tygar, J. D. (2010). The security of machine learning. *Machine Learning*, 81(2):121–148.

Batchu, R. K. e Seetha, H. (2021). A generalized machine learning model for DDoS attacks detection using hybrid feature selection and hyperparameter tuning. *Computer Networks*, 200:108498.

BBC (2020). New zealand stock exchange halted by cyber-attack. Acessado em: 07/2021. https://www.bbc.com/news/53918580.

Bedeian, A. G. e Mossholder, K. W. (2000). On the use of the coefficient of variation as a measure of diversity. *ORM*, 3(3):285–297.

Belenky, A. e Ansari, N. (2003). On IP traceback. *IEEE Communications Magazine*, 41(7):142–153.

Bendovschi, A. (2015). Cyber-attacks – trends, patterns and security countermeasures. *Procedia Economics and Finance*, 28:24–31. 7th International Conference On Financial Criminology 2015, 7th ICFC 2015, 13-14 April 2015,Wadham College, Oxford University, United Kingdom.

Benzekki, K., El Fergougui, A. e Elbelrhiti Elalaoui, A. (2016). Software-defined networking (SDN): a survey. *Security and Communication Networks*, 9(18):5803–5833.

Beslin Pajila, P. J. e Golden Julie, E. (2020). Detection of DDoS attack using SDN in IoT: A survey. Em *Intelligent Communication Technologies and Virtual Mobile Networks*, páginas 438–452, Cham. Springer International Publishing.

Bhatia, S., Behal, S. e Ahmed, I. (2018). *Distributed Denial of Service Attacks and Defense Mechanisms: Current Landscape and Future Directions*, páginas 55–97. Springer International Publishing, Cham.

Biggs, R., Carpenter, S. R. e Brock, W. A. (2009). Turning back from the brink: Detecting an impending regime shift in time to avert it. *Proceedings of the National Academy of Sciences*, 106(3):826–831.

Biju, J. M., Gopal, N. e Prakash, A. J. (2019). Cyber attacks and its different types. *International Research Journal of Engineering and Technology*, 6(3):4849–4852.

Bitit, R., Derhab, A., Guerroumi, M. e Khan, F. A. (2024). DDoS attack forecasting based on online multiple change points detection and time series analysis. *Multimedia Tools and Applications*, 83(18):53655–53685.

Boers, N. e Rypdal, M. (2021). Critical slowing down suggests that the western Greenland Ice Sheet is close to a tipping point. *PNAS*, 118(21).

Bondy, J. A., Murty, U. S. R. et al. (1976). *Graph theory with applications*, volume 290. The Macmillan Press Ltd, Great Britain.

Borges, L., de Neira, A. B., Albano, L. e Nogueira, M. (2024). Multifaceted DDoS attack prediction by multivariate time series and ordinal patterns (to appear). Em *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*, USA.

Box, G. E., Jenkins, G. M., Reinsel, G. C. e Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Brasil (2018). Lei geral de proteção de dados pessoais (LGPD). Acessado em: 06/2021. http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Brito, D., de Neira, A. B., Borges, L. F. e Nogueira, M. (2023a). An autonomous system for predicting DDoS attacks on local area networks and the internet. Em *2023 IEEE Latin-American Conference on Communications (LATINCOM 2023)*, página 6, Panama.

Brito, D., Neira, A., Borges, L., Araújo, A. e Nogueira, M. (2023b). Um sistema autônomo para a predição de ataques de DDoS em redes locais e Internet. Em *Anais do XXVIII Workshop de Gerência e Operação de Redes e Serviços*, páginas 29–42, Porto Alegre, RS, Brasil. SBC.

Brodić, D. e Amelio, A. (2020). *Types of CAPTCHA*, páginas 29–32. Springer International Publishing, Cham.

Brooks, R. R., Yu, L., Ozcelik, I., Oakley, J. e Tusing, N. (2022). Distributed denial of service (ddos): A history. *IEEE Annals of the History of Computing*, 44(2):44–54.

Burnham, K. P. e Anderson, D. R., editores (2004). *Model Selection and Multimodel Inference*. Springer New York.

Bury, T. M., Bauch, C. T. e Anand, M. (2020). Detecting and distinguishing tipping points using spectral early warning signals. *Journal of The Royal Society Interface*, 17(170):20200482.

Bury, T. M., Sujith, R. I., Pavithran, I., Scheffer, M., Lenton, T. M., Anand, M. e Bauch, C. T. (2021). Deep learning for early warning signals of tipping points. *Proceedings of the National Academy of Sciences*, 118(39).

CALEM, R. E. (1196). New York's Panix service is crippled by hacker attack. Acessado em: 02/2024. https://archive.nytimes.com/www.nytimes.com/library/cyber/week/0914panix.html.

Califórnia (2018). California consumer privacy act. Acessado em: 06/2021. https://oag.ca.gov/privacy/ccpa.

Caliński, T. e Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27.

Campbell, M. (2015). Summarising numerical variables. *African Journal of Midwifery and Women's Health*, 9(3):96–104.

Cao, X. H. e Obradovic, Z. (2015). A robust data scaling algorithm for gene expression classification. Em *2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE)*, páginas 1–4, Belgrade, Serbia. IEEE.

Carbonell, J. G., Michalski, R. S. e Mitchell, T. M. (1983). 1 - An overview of machine learning. Em Michalski, R. S., Carbonell, J. G. e Mitchell, T. M., editores, *Machine Learning*, páginas 3–23. Morgan Kaufmann, San Francisco (CA).

Carpenter, S. R. e Brock, W. A. (2006). Rising variance: a leading indicator of ecological transition. *Ecology Letters*, 9(3):311–318.

Castro, C. L. d. e Braga, A. P. (2011). Aprendizado supervisionado com conjuntos de dados desbalanceados. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, 22:441–466.

Chahal, J. K., Bhandari, A. e Behal, S. (2019). Distributed denial of service attacks: A threat or challenge. *New Review of Information Networking*, 24(1):31–103.

Chandrashekar, G. e Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28.

Chen, S., Wu, Z. e Christofides, P. D. (2021). Cyber-security of centralized, decentralized, and distributed control-detector architectures for nonlinear processes. *Chemical Engineering Research and Design*, 165:25–39.

Chen, Y., Zhao, X. e Jia, X. (2015). Spectral–spatial classification of hyperspectral data based on deep belief network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):2381–2392.

Choi, H. e Lee, H. (2012). Identifying botnets by capturing group activities in DNS traffic. *Computer Networks*, 56(1):20–33.

CISA (2019). Understanding denial-of-service attacks. Acessado em: 07/2021. https://us-cert.cisa.gov/ncas/tips/ST04-015.

Čisar, P. e Čisar, S. M. (2010). Skewness and kurtosis in function of selection of network traffic distribution. *Acta Polytechnica Hungarica*, 7(2):95–106.

Cisco (2014). A Cisco guide to defending against distributed denial of service attacks. Acessado em: 07/2021. https://tools.cisco.com/security/center/resources/guide_ddos_defense.html.

Cloudflare (2021a). What is a DDoS attack? Acessado em: 07/2021. https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack.

Cloudflare (2021b). What is a DNS amplification attack? Acessado em: 07/2021. https://www.cloudflare.com/learning/ddos/dns-amplification-ddos-attack/.

Cloudflare (2021c). What is a low and slow attack? Acessado em: 07/2021. https://www.cloudflare.com/learning/ddos/ddos-low-and-slow-attack/.

Cloudflare (2021d). What is a NTP amplification attack? Acessado em: 07/2021. https://www.cloudflare.com/learning/ddos/ntp-amplification-ddos-attack/.

Cloudflare (2021e). What is a ping (ICMP) flood attack? Acessado em: 07/2021. https://www.cloudflare.com/learning/ddos/ping-icmp-flood-ddos-attack.

Cloudflare (2021f). What is a SYN flood attack? Acessado em: 07/2021. https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/.

Cloudflare (2021g). What is a UDP flood attack? Acessado em: 07/2021. https://www.cloudflare.com/learning/ddos/udp-flood-ddos-attack/.

Cloudflare (2021h). What is a zero-day exploit? Acessado em: 08/2021. https://www.cloudflare.com/learning/security/threats/zero-day-exploit/.

Cloudflare (2021i). What is an HTTP flood DDoS attack? Acessado em: 07/2021. https://www.cloudflare.com/learning/ddos/http-flood-ddos-attack/.

Comer, D. E. (2015). *Computer networks and Internets*. Pearson Education, United States of America.

Corero (2021). The need for always-on in real-time DDoS security solutions. Acessado em: 10/2023. https://f.hubspotusercontent20.net/hubfs/6483172/2021/website-content/whitepapers/always-on-real-time-ddos-security-whitepaper.pdf.

Cormen, T. H., Leiserson, C. E., Rivest, R. L. e Stein, C. (2012). *Algoritmos: teoria e prática*, volume 3. Elsevier Editora Ltda.

CPJ (2021). Investigative outlet Repórter Brasil targeted with cyberattacks, threats, attempted break-in. Acessado em: 07/2021. https://cpj.org/2021/01/investigative-outlet-reporter-brasil-targeted-with-cyberattacks-threats-attempted-break-in/.

Cramér, H. (1946). *Mathematical methods of statistics*. ASIA PUBLISHING HOUSE.

Crevier, D. (1993). *AI: The Tumultuous History of the Search for Artificial Intelligence*. Basic Books, Inc., New York, NY, USA.

Crisci, C., Ghattas, B. e Perera, G. (2012). A review of supervised machine learning algorithms and their applications to ecological data. *Ecological Modelling*, 240:113–122.

Dake, D. K., Gadze, J. D. e Klogo, G. S. (2021). DDoS and flash event detection in higher bandwidth SDN-IoT using multiagent reinforcement learning. Em *ICCMA*, páginas 16–20.

Dakos, V., Carpenter, S. R., Brock, W. A., Ellison, A. M., Guttal, V., Ives, A. R., Kéfi, S., Livina, V., Seekell, D. A., van Nes, E. H. e Scheffer, M. (2012). Methods for detecting early warnings of critical transitions in time series illustrated using simulated ecological data. *PLOS ONE*, 7(7):1–20.

Dakos, V., Scheffer, M., van Nes, E. H., Brovkin, V., Petoukhov, V. e Held, H. (2008). Slowing down as an early warning signal for abrupt climate change. *Proceedings of the National Academy of Sciences*, 105(38):14308–14312.

Dalmazo, B. L., Marques, J. A., Costa, L. R., Bonfim, M. S., Carvalho, R. N., da Silva, A. S., Fernandes, S., Bordim, J. L., Alchieri, E., Schaeffer-Filho, A., Paschoal Gaspary, L. e Cordeiro, W. (2021). A systematic review on distributed denial of service attack defense mechanisms in programmable networks. *International Journal of Network Management*, n/a(n/a):e2163.

Davies, D. L. e Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.

Dayan, P., Sahani, M. e Deback, G. (1999). Unsupervised learning. *The MIT encyclopedia of the cognitive sciences*, páginas 857–859.

DDoS-Stress (2023). DDoS-as-aService. Acessado em: 09/2023. https://ddos.services.

de Neira, A. B., Medeiro, A. e Nogueira, M. (2020). Identificação antecipada de botnets por aprendizagem de máquina. Em *Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, páginas 896–909, Porto Alegre, RS, Brasil. SBC.

de Prado, M. L. e Lewis, M. J. (2019). Detection of false investment strategies using unsupervised learning methods. *Quantitative Finance*, 19(9):1555–1565.

de Souto, M. C. P., Prudencio, R. B. C., Soares, R. G. F., de Araujo, D. S. A., Costa, I. G., Ludermir, T. B. e Schliep, A. (2008). Ranking and selecting clustering algorithms using a meta-learning approach. Em *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, páginas 3729–3735, China.

DeCarlo, L. T. (1997). On the meaning and use of kurtosis. *Psychological methods*, 2(3):292.

Dei Rossi, G.-L., Iacono, M. e Marin, A. (2016). Evaluating the impact of EDoS attacks to cloud facilities. Em *Proceedings of the 9th EAI International Conference on Performance Evaluation Methodologies and Tools*, VALUETOOLS'15, página 188–195, Brussels, BEL. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Deshmukh, R. V. e Devadkar, K. K. (2015). Understanding DDoS attack & its effect in cloud environment. *Procedia Computer Science*, 49:202–210. Proceedings of 4th International Conference on Advances in Computing, Communication and Control (ICAC3'15).

Devi, D., Biswas, S. K. e Purkayastha, B. (2019). Learning in presence of class imbalance and class overlapping by using one-class SVM and undersampling technique. *Connection Science*, 31(2):105–142.

Dey, R. e Salem, F. M. (2017). Gate-variants of gated recurrent unit (GRU) neural networks. Em *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, páginas 1597–1600, Boston, MA, USA. IEEE.

Dhankhad, S., Mohammed, E. e Far, B. (2018). Supervised machine learning algorithms for credit card fraudulent transaction detection: A comparative study. Em *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, páginas 122–125.

Diks, C., Hommes, C. e Wang, J. (2018). Critical slowing down as an early warning signal for financial crises? *Empirical Economics*, 57(4):1201–1228.

Doane, D. P. e Seward, L. E. (2011). Measuring skewness: a forgotten statistic? *Journal of statistics education*, 19(2).

Domo, I. (2023). Data never sleeps 11.0. Acessado em: 02/2024. https://www.domo.com/blog/data-never-sleeps-hits-double-digits/.

dos Santos Lima, M. H., Neira, A., Borges, L. e Nogueira, M. (2023). Predição não-supervisionada de ataques DDoS por sinais precoces e One-Class SVM. Em *Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2023)*, Juiz de Fora, MG.

dos Santos Lima, M. H., Neira, A., Borges, L. e Nogueira, M. (2024). Automatização da seleção de modelos não supervisionados na predição de ataques DDoS (to appear). Em *Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2024)*, São José dos Campos.

Dosovitskiy, A. e Brox, T. (2016). Generating images with perceptual similarity metrics based on deep networks. Em Lee, D., Sugiyama, M., Luxburg, U., Guyon, I. e Garnett, R., editores, *Advances in Neural Information Processing Systems*, volume 29, Barcelona, Spain. Curran Associates, Inc.

Douligeris, C. e Mitrokotsa, A. (2004). DDoS attacks and defense mechanisms: Classification and state-of-the-art. *Computer Networks*, 44(5):643–666.

Došilović, F. K., Brčić, M. e Hlupić, N. (2018). Explainable artificial intelligence: A survey. Em *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, páginas 0210–0215, Opatija, Croatia. IEEE.

Drake, J. M. e Griffen, B. D. (2010). Early warning signals of extinction in deteriorating environments. *Nature*, 467(7314):456–459.

El-Sofany, H. F. (2020). A new cybersecurity approach for protecting cloud services against DDoS attacks. *International Journal of Intelligent Engineering and Systems*, 13(2):205–215.

Elamin, N. e Fukushige, M. (2018). Modeling and forecasting hourly electricity demand by SARIMAX with interactions. *Energy*, 165:257–268.

Erdogdu, E. (2007). Electricity demand analysis using cointegration and ARIMA modelling: A case study of Turkey. *Energy Policy*, 35(2):1129–1146.

Erickson, B. J., Korfiatis, P., Akkus, Z. e Kline, T. L. (2017). Machine learning for medical imaging. *RadioGraphics*, 37(2):505–515.

Ester, M., Kriegel, H.-P., Sander, J. e Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. Em *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, página 226–231, Portland, Oregon. AAAI Press.

Facebook (2021). Facebook bug bounty program info. Acessado em: 07/2021. https://www.facebook.com/whitehat.

Fadlullah, Z. M., Fouda, M. M., Kato, N., Shen, X. e Nozaki, Y. (2011). An early warning system against malicious activities for smart grid communications. *IEEE Network*, 25(5):50–55.

Fan, Z., Tan, Z., Tan, C. e Li, X. (2018). An improved integrated prediction method of cyber security situation based on spatial-time analysis. *Journal of Internet Technology*, 19:1789–1800.

Feng, Y., Akiyama, H., Lu, L. e Sakurai, K. (2018). Feature selection for machine learning-based early detection of distributed cyber attacks. Em *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, páginas 173–180, Athens, Greece. IEEE.

Feng, Y., Li, J. e Nguyen, T. (2020). Application-layer DDoS defense with reinforcement learning. Em *2020 IEEE ACM 28th International Symposium on Quality of Service (IWQoS)*, páginas 1–10.

Fenil, E. e Mohan Kumar, P. (2020). Survey on DDoS defense mechanisms. *Concurrency and Computation: Practice and Experience*, 32(4):e5114.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J. T., Blum, M. e Hutter, F. (2015). Efficient and robust automated machine learning. Em *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, página 2755–2763, Cambridge, MA, USA. MIT Press.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J. T., Blum, M. e Hutter, F. (2019). Auto-sklearn: efficient and robust automated machine learning. Em *Automated Machine Learning*, página 21. Springer.

Fortnow, L. (2009). The status of the P versus NP problem. *Communications of the ACM*, 52(9):78–86.

Foulds, L. R. (2012). *Graph theory applications*. Springer Science & Business Media, New York, NY, USA.

Freund, Y. e Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378. Nonlinear Methods and Data Mining.

Frost e Sullivan (2022). The new benchmark: Why fast DDoS detection is no longer good enough. Acessado em: 10/2023. https://www.link11.com/wp-content/uploads/2022/08/Frost_Sullivan_DDoS_Protection_Benchmark_Report_01-2022.pdf.

Gao, Y., Liu, Y., Jin, Y., Chen, J. e Wu, H. (2018). A novel semi-supervised learning approach for network intrusion detection on cloud-based robotic system. *IEEE Access*, 6:50927–50938.

Garcia, S., Grill, M., Stiborek, J. e Zunino, A. (2014). An empirical comparison of botnet detection methods. *Computers & Security*, 45:100–123.

Garcia, S., Parmisano, A. e Erquiaga, M. J. (2020). IoT-23: A labeled dataset with malicious and benign IoT network traffic Acessado em: 04/2023.

Geller, R. J. (1997). Earthquake prediction: a critical review. *Geophysical Journal International*, 131(3):425–450.

Ghahramani, Z. (2004). *Unsupervised Learning*, páginas 72–112. Springer Berlin Heidelberg, Berlin, Heidelberg.

Gligor, V. D. (1984). A note on denial-of-service in operating systems. *IEEE Transactions on Software Engineering*, SE-10(3):320–324.

Google (2021). Google vulnerability reward program (VRP) rules. Acessado em: 07/2021. https://www.google.com/about/appsecurity/reward-program/.

Gruschka, N. e Luttenberger, N. (2006). Protecting web services from DoS attacks by SOAP message validation. Em Fischer-Hübner, S., Rannenberg, K., Yngström, L. e Lindskog, S., editores, *Security and Privacy in Dynamic Environments*, páginas 171–182, Boston, MA. Springer US.

Grzymala-Busse, J. W. (1992). *LERS-A System for Learning from Examples Based on Rough Sets*, páginas 3–18. Springer Netherlands, Dordrecht.

Gupta, B. B. e Badve, O. P. (2017). Taxonomy of DoS and DDoS attacks and desirable defense mechanism in a cloud computing environment. *Neural Computing and Applications*, 28(12):3655–3682.

Gupta, B. B. e Dahiya, A. (2021). *Distributed Denial of Service (DDoS) Attacks: Classification, Attacks, Challenges, and Countermeasures*. CRC Press, Florida, United States of America.

Gupta, B. B., Joshi, R. C. e Misra, M. (2010). Distributed denial of service prevention techniques. *International Journal of Computer and Electrical Engineering*, 2(2):268–276.

Gutnikov, A., Kupreev, O. e Badovskaya, E. (2021). DDoS attacks in Q1 2021. Acessado em: 07/2021. https://securelist.com/ddos-attacks-in-q1-2021/102166.

Guttal, V. e Jayaprakash, C. (2008). Changing skewness: an early warning signal of regime shifts in ecosystems. *Ecology Letters*, 11(5):450–460.

Halkidi, M., Batistakis, Y. e Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2):107–145.

Halkidi, M. e Vazirgiannis, M. (2001). Clustering validity assessment: finding the optimal partitioning of a data set. Em *Proceedings 2001 IEEE International Conference on Data Mining*, páginas 187–194.

Halkidi, M. e Vazirgiannis, M. (2008). A density-based cluster validity approach using multi-representatives. *Pattern Recognition Letters*, 29(6):773–786.

Han, J., Pei, J. e Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Hanbanchong, A. e Piromsopa, K. (2012). SARIMA based network bandwidth anomaly detection. Em *2012 Ninth International Conference on Computer Science and Software Engineering (JCSSE)*, páginas 104–108.

Hanselmann, M., Strauss, T., Dormann, K. e Ulmer, H. (2020). CANet: an unsupervised intrusion detection system for high dimensional CAN bus data. *IEEE Access*, 8:58194–58205.

Hastie, T., Tibshirani, R. e Friedman, J. (2009). *Unsupervised Learning*, páginas 485–585. Springer New York, New York, NY.

He, X., Zhao, K. e Chu, X. (2021). AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622.

He, Z., Zhang, T. e Lee, R. B. (2017). Machine learning based DDoS attack detection from source side in cloud. Em *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, páginas 114–120, New York, NY, USA. IEEE.

Hochreiter, S. e Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Holgado, P., Villagrá, V. A. e Vázquez, L. (2020). Real-time multistep attack prediction based on hidden markov models. *IEEE Transactions on Dependable and Secure Computing*, 17(1):134–147.

Holland, J., Schmitt, P., Feamster, N. e Mittal, P. (2021). New directions in automated traffic analysis. Em *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, página 3366–3383, USA. ACM.

Horsanali, E., Yigit, Y., Secinti, G., Karameseoglu, A. e Canberk, B. (2021). Network-aware AutoML framework for software-defined sensor networks. Em *International Conference on Distributed Computing in Sensor Systems (DCOSS)*, páginas 451–457. IEEE.

Houssein, E. H., Hosney, M. E., Oliva, D., Mohamed, W. M. e Hassaballah, M. (2020). A novel hybrid harris hawks optimization and support vector machines for drug design and discovery. *Computers & Chemical Engineering*, 133:106656.

Huang, D. Y., Dharmdasani, H., Meiklejohn, S., Dave, V., Grier, C., McCoy, D., Savage, S., Weaver, N., Snoeren, A. C. e Levchenko, K. (2014). Botcoin: Monetizing stolen cycles. Em *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014*, página 16, San Diego, California, USA. The Internet Society.

Huang, L., Joseph, A. D., Nelson, B., Rubinstein, B. I. e Tygar, J. D. (2011). Adversarial machine learning. Em *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, AISec '11, página 43–58, New York, NY, USA. ACM.

Hummel, R., Dobbins, R., Bjarnasen, S., Conrad, C., Lara, R., Arenberg, C., Kristoff, J., Patel, K., Resing, M. e Modi, H. (2023). Netscout DDoS threat intelligence report / findings from 2nd half 2022. Acessado em: 09/2023. https://www.netscout.com/threatreport/wp-content/uploads/2023/04/Threat-Report-2H2022.pdf.

Hummel, R., Hildebrand, C., Modi, H., Conrad, C., Roland Dobbins, S. B., Belanger, J., Sockrider, G., Alcoy, P. e Bienkowski, T. (2021). Netscout threat intelligence report. Acessado em: 08/2021. https://www.netscout.com/sites/default/files/2021-04/ThreatReport_2H2020_FINAL_0.pdf.

Husák, M., Komárková, J., Bou-Harb, E. e Čeleda, P. (2019). Survey of attack projection, prediction, and forecasting in cyber security. *IEEE Communications Surveys Tutorials*, 21(1):640–660.

Hyvärinen, A. e Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural Networks*, 13(4–5):411–430.

Iacomin, R. (2015). Stock market prediction. Em *2015 19th International Conference on System Theory, Control and Computing (ICSTCC)*, páginas 200–205, Cheile Gradistei, Romania. IEEE.

Ibitoye, O., Abou-Khamis, R., Matrawy, A. e Shafiq, M. O. (2020). The threat of adversarial attacks on machine learning in network security – a survey.

il Jang, D., Kim, M., chul Jung, H. e Noh, B.-N. (2009). Analysis of HTTP2P botnet: case study waledac. Em *2009 IEEE 9th Malaysia International Conference on Communications (MICC)*, páginas 409–412, Kuala Lumpur, Malaysia. IEEE.

Ilascu, I. (2022). Extortion DDoS attacks grow stronger and more common. Acessado em: 01/2022. https://www.bleepingcomputer.com/news/security/extortion-ddos-attacks-grow-stronger-and-more-common/.

Imran, Jamil, F. e Kim, D. (2021). An ensemble of prediction and learning mechanism for improving accuracy of anomaly detection in network intrusion environments. *Sustainability*, 13(18):10057.

Ioulianou, P., Vasilakis, V. e Shahandashti, S. F. (2022). ML-based detection of blackhole and rank attacks in RPL networks. Em *International Symposium on Communications Systems, Networks and Digital Signal Processing: Proceedings*, Portugal. IEEE.

Isingizwe, D. F., Wang, M., Liu, W., Wang, D., Wu, T. e Li, J. (2021). Analyzing learning-based encrypted malware traffic classification with AutoML. Em *2021 IEEE 21st International Conference on Communication Technology (ICCT)*, páginas 313–322, Tianjin, China.

ITIC (2020). ITIC 2020 global server hardware, server os reliability report. Acessado em: 08/2021. https://www.ibm.com/downloads/cas/DV0XZV6R.

Jaber, A. N., Zolkipli, M. F., Majid, M. A. e Anwar, S. (2017). Methods for preventing distributed denial of service attacks in cloud computing. *Advanced Science Letters*, 23(6):5282–5285.

James, G., Witten, D., Hastie, T., Tibshirani, R. e Taylor, J. (2023). *Unsupervised Learning*, páginas 503–556. Springer International Publishing, Cham.

Janos, M. (2020). Deep learning – conceitos e aplicações. Acessado em: 12/2021. https://www.3dimensoes.com.br/post/deep-learning-conceitos-e-aplica%C3%A7%C3%B5es.

Jin, H., Song, Q. e Hu, X. (2019). Auto-keras: An efficient neural architecture search system. Em *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, página 1946–1956, New York, NY, USA. Association for Computing Machinery.

Jing, H. e Wang, J. (2020). DDoS detection based on graph structure features and non-negative matrix factorization. *Concurrency and Computation: Practice and Experience*, n/a(n/a):13.

Joanes, D. N. e Gill, C. A. (1998). Comparing measures of sample skewness and kurtosis. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(1):183–189.

Jog, M., Natu, M. e Shelke, S. (2015). Distributed and predictive-preventive defense against DDoS attacks. Em *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, ICDCN '15, New York, NY, USA. ACM.

Jyoti, N. e Behal, S. (2021). A meta-evaluation of machine learning techniques for detection of DDoS attacks. Em *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, páginas 522–526, New Delhi, India. IEEE.

Kaluarachchi, T., Reis, A. e Nanayakkara, S. (2021). A review of recent deep learning approaches in human-centered machine learning. *Sensors*, 21(7).

Kar, A. K. (2016). Bio inspired computing – a review of algorithms and scope of applications. *Expert Systems with Applications*, 59:20–32.

Kardian, A. R., Sudiro, S. A. e Madenda, S. (2018). Efficient implementation of mean, variance and skewness statistic formula for image processing using FPGA device. *Bulletin of Electrical Engineering and Informatics*, 7(3):386–392.

Karim, A., Salleh, R. B., Shiraz, M., Shah, S. A. A., Awan, I. e Anuar, N. B. (2014). Botnet detection techniques: review, future trends, and issues. *Journal of Zhejiang University SCIENCE C*, 15(11):943–983.

Kaur, H., Pannu, H. S. e Malhi, A. K. (2019). A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Computing Surveys*, 52(4).

Kavlakoglu, E. (2020). AI vs. machine learning vs. deep learning vs. neural networks: What's the difference?. Acessado em: 08/2021. https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks.

Kawaguchi, K. (2016). Deep learning without poor local minima. *Advances in Neural Information Processing Systems*, 29:586–594.

Kecman, V. (2005). *Support Vector Machines – An Introduction*, páginas 1–47. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kedziora, D. J., Musial, K. e Gabrys, B. (2020). AutonoML: Towards an integrated framework for autonomous machine learning. *CoRR*, abs/2012.12600:1–77.

Kemeny, J. G. e Snell, J. L. (1976). *Finite Markov Chains*. Springer-Verlag, New York, NY, USA.

Keshariya, A. e Foukia, N. (2010). DDoS defense mechanisms: A new taxonomy. Em Garcia-Alfaro, J., Navarro-Arribas, G., Cuppens-Boulahia, N. e Roudier, Y., editores, *Data Privacy Management and Autonomous Spontaneous Security*, páginas 222–236, Berlin, Heidelberg. Springer Berlin Heidelberg.

Kianpour, M. e Wen, S.-F. (2020). Timing attacks on machine learning: State of the art. Em *Intelligent Systems and Applications*, páginas 111–125, Cham. Springer International Publishing.

Kiner, E. e April, T. (2023). Google mitigated the largest DDoS attack to date, peaking above 398 million rps Acessado em: 10/2023. https://cloud.google.com/blog/products/identity-security/google-cloud-mitigated-largest-ddos-attack-peaking-above-398-million-rps.

Kivalov, S. e Strelkovskaya, I. (2022). Detection and prediction of DDoS cyber attacks using spline functions. Em *17th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering*, páginas 710–713, Ukraine.

Kleinbaum, D. G. e Klein, M. (2010). *Introduction to Logistic Regression*, páginas 1–39. Springer New York, New York, NY.

Klement, F., Pöhls, H. C. e Spielvogel, K. (2020). Towards privacy-preserving local monitoring and evaluation of network traffic from IoT devices and corresponding mobile phone applications. Em *2020 Global Internet of Things Summit (GIoTS)*, páginas 1–6, Dublin, Ireland. IEEE.

Kohonen, T. (1997). *The Basic SOM*, páginas 85–144. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kour, H. e Gondhi, N. (2020). Machine learning techniques: A survey. Em *Innovative Data Communication Technologies and Application*, páginas 266–275, Cham. Springer International Publishing.

Kshirsagar, D. e Kumar, S. (2021). A feature reduction based reflected and exploited DDoS attacks detection system. *Journal of Ambient Intelligence and Humanized Computing*, 13(1):393–405.

Kumar, M., Khandelwal, S., Wei, W., Lakshmanan, R., Jain, K., Bansal, S. K., Grover, M., Raam, G. e Sahay, P. (2022). Reasons why every business is a target of DDoS attacks. Acessado em: 02/2022. https://thehackernews.com/2022/01/reasons-why-every-business-is-target-of.html.

Kumbhare, T. A. e Chobe, S. V. (2014). An overview of association rule mining algorithms. *International Journal of Computer Science and Information Technologies*, 5(1):927–930.

Kurose, J. F. e Ross, K. W. (2014). *Redes de Computadores ea Internet uma abordagem top-down*. Pearson Education do Brasil.

Kwon, D., Kim, H., An, D. e Ju, H. (2017). DDoS attack volume forecasting using a statistical approach. Em *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, páginas 1083–1086, Lisbon, Portugal. IEEE.

Laboratory, L. (2000). 2000 DARPA intrusion detection scenario specific datasets. Acessado em: 06/2021. https://www.ll.mit.edu/r-d/datasets/2000-darpa-intrusion-detection-scenario-specific-datasets.

Laing, L., Salema, N., Jeffries, M., Shamsuddin, A., Sheikh, A., Waring, J., Avery, T. e Keers, R. (2021). Understanding the implementation and medium-longer term sustainability of the primary care prescribing safety intervention, pincer: preliminary results from a longitudinal process evaluation. *International Journal of Pharmacy Practice*, 29(Supplement_1):i8–i9.

Lam, J. e Abbas, R. (2020). Machine learning based anomaly detection for 5G networks.

Lee, K., Kim, J., Kwon, K. H., Han, Y. e Kim, S. (2008). DDoS attack detection method using cluster analysis. *Expert Systems with Applications*, 34(3):1659–1665.

Lerner, A. (2014). The cost of downtime. Acessado em: 08/2021. https://blogs.gartner.com/andrew-lerner/2014/07/16/the-cost-of-downtime.

Leros, A. P. e Andreatos, A. S. (2019). *Network Traffic Analytics for Internet Service Providers—Application in Early Prediction of DDoS Attacks*, páginas 233–267. Springer International Publishing, Cham.

Levine, S., Kumar, A., Tucker, G. e Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems.

Li, D., Zhao, J., Liu, H. e Hao, D. (2014). The application of FastICA combined with related function in blind signal separation. *Mathematical Problems in Engineering*, 2014:1–9.

Li, J., Liu, M., Xue, Z., Fan, X. e He, X. (2020). RTVD: A real-time volumetric detection scheme for DDoS in the Internet of things. *IEEE Access*, 8:36191–36201.

Li, S., Song, W., Fang, L., Chen, Y., Ghamisi, P. e Benediktsson, J. A. (2019a). Deep learning for hyperspectral image classification: An overview. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):6690–6709.

Li, X., Fan, Z., Xiao, Y., Xu, Q. e Zhu, W. (2019b). Improved automated graph and FCM based DDoS attack detection mechanism in software defined networks. *Journal of Internet Technology*, 20(7).

Lindemann, B., Müller, T., Vietz, H., Jazdi, N. e Weyrich, M. (2021). A survey on long short-term memory networks for time series prediction. *Procedia CIRP*, 99:650–655. 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020.

Liu, H. e Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences*, 9(20).

Liu, J. (2013). *RBF Neural Network Design and Simulation*, páginas 19–53. Springer Berlin Heidelberg, Berlin, Heidelberg.

Liu, J., Nogueira, M., Fernandes, J. e Kantarci, B. (2022). Adversarial machine learning: A multilayer review of the state-of-the-art and challenges for wireless and mobile systems. *IEEE Communications Surveys & Tutorials*, 24(1):123–159.

Liu, Y., Li, Z., Xiong, H., Gao, X. e Wu, J. (2010). Understanding of internal clustering validation measures. Em *2010 IEEE International Conference on Data Mining*, páginas 911–916.

Liu, Y., Zhang, J., Sarabi, A., Liu, M., Karir, M. e Bailey, M. (2015). Predicting cyber security incidents using feature-based characterization of network-level malicious activities. Em *Proceedings of the 2015 ACM International Workshop on International Workshop on Security and Privacy Analytics*, IWSPA '15, página 3–9, New York, NY, USA. ACM.

Liuliakov, A. e Hammer, B. (2021). AutoML technologies for the identification of sparse models. Em Yin, H., Camacho, D., Tino, P., Allmendinger, R., Tallón-Ballesteros, A. J., Tang, K., Cho, S.-B., Novais, P. e Nascimento, S., editores, *International Conference on Intelligent Data Engineering and Automated Learning*, páginas 65–75, Cham. Springer.

Livina, V. N., Lewis, A. P. e Wickham, M. (2020). Tipping point analysis of electrical resistance data with early warning signals of failure for predictive maintenance. *Journal of Electronic Testing*, 36(5):569–576.

Luong, T.-K., Tran, T.-D. e Le, G.-T. (2020). DDoS attack detection and defense in SDN based on machine learning. Em *2020 7th NAFOSTED Conference on Information and Computer Science (NICS)*, páginas 31–35, Ho Chi Minh City, Vietnam. IEEE.

Luzi, L., Puglia, R., Russo, E., D'Amico, M., Felicetta, C., Pacor, F., Lanzano, G., Çeken, U., Clinton, J., Costa, G., Duni, L., Farzanegan, E., Gueguen, P., Ionescu, C., Kalogeras, I., Özener, H., Pesaresi, D., Sleeman, R., Strollo, A. e Zare, M. (2016). The Engineering Strong-Motion Database: A Platform to Access Pan-European Accelerometric Data. *Seismological Research Letters*, 87(4):987–997.

Ma, R., Wu, F., Zhang, M., Lu, Z., Wan, J. e Xie, C. (2019). RBER-Aware lifetime prediction scheme for 3D-TLC NAND flash memory. *IEEE Access*, 7:44696–44708.

Machaka, P., Ajayi, O., Kahenga, F., Bagula, A. e Kyamakya, K. (2022). Modelling DDoS attacks in IoT networks using machine learning. Em *International Conference on Emerging Technologies for Developing Countries*, páginas 161–175. Springer.

Mahfouz, A., Abuhussein, A., Venugopal, D. e Shiva, S. (2020). Ensemble classifiers for network intrusion detection using a novel network attack dataset. *Future Internet*, 12(11):180.

Mahjabin, T., Xiao, Y., Sun, G. e Jiang, W. (2017). A survey of distributed denial-of-service attack, prevention, and mitigation techniques. *International Journal of Distributed Sensor Networks*, 13(12).

Mahmoud, N., Essam, Y., Elshawi, R. e Sakr, S. (2019). DLBench: An experimental evaluation of deep learning frameworks. Em *2019 IEEE International Congress on Big Data (BigDataCongress)*, páginas 149–156, Milan, Italy. IEEE.

Malhotra, P., Vig, L., Shroff, G. e Agarwal, P. (2015). Long short term memory networks for anomaly detection in time series. Em *European Symposium on Artificial Neural Networks*.

Mane, Y. D. (2017). Detect and deactivate P2P Zeus bot. Em *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, páginas 1–7, Delhi, India. IEEE.

Manigandan, P., Alam, M. S., Alharthi, M., Khan, U., Alagirisamy, K., Pachiyappan, D. e Rehman, A. (2021). Forecasting natural gas production and consumption in United States-Evidence from SARIMA and SARIMAX models. *Energies*, 14(19).

Marrow, A. e Stolyarov, G. (2021). Russia's Yandex says it repelled biggest DDoS attack in history. Acessado em: 10/2021. https://www.reuters.com/technology/russias-yandex-says-it-repelled-biggest-ddos-attack-history-2021-09-09/.

Martinez, A. e Kak, A. (2001). PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233.

Marzano, A., Alexander, D., Fonseca, O., Fazzion, E., Hoepers, C., Steding-Jessen, K., Chaves, M. H. P. C., Cunha, A., Guedes, D. e Meira, W. (2018). The evolution of Bashlite and Mirai IoT botnets. Em *2018 IEEE Symposium on Computers and Communications (ISCC)*, páginas 00813–00818, Natal, Brasil. IEEE.

Matsumoto, G. e Kunisawa, T. (1978). Critical slowing-down near the transition region from the resting to time-ordered states in squid giant axons. *Journal of the Physical Society of Japan*, 44(3):1047–1048.

Maćkiewicz, A. e Ratajczak, W. (1993). Principal components analysis (PCA). *Computers & Geosciences*, 19(3):303–342.

Meerschaert, M. M. (2013). *Mathematical modeling*. Academic Press/Elsevier, Waltham, MA, USA.

Meisel, C., Klaus, A., Kuehn, C. e Plenz, D. (2015). Critical slowing down governs the transition to neuron spiking. *PLOS Computational Biology*, 11(2):e1004097.

Menscher, D. (2020). Exponential growth in DDoS attack volumes. Acessado em: 07/2021. https://cloud.google.com/blog/products/identity-security/identifying-and-protecting-against-the-largest-ddos-attacks.

Miao, J. e Niu, L. (2016). A survey on feature selection. *Procedia Computer Science*, 91:919–926.

Mirkovic, J. e Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2):39–53.

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill, New York.

Monge, M., Vidal, J. e Villalba, L. (2017). Entropy-based economic denial of sustainability detection. *Entropy*, 19(12):649.

Monroe, D. (2021). Trouble at the source. *Communications of the ACM*, 64(12):17–19.

Moudoud, H., Khoukhi, L. e Cherkaoui, S. (2021). Prediction and detection of FDIA and DDoS attacks in 5G enabled IoT. *IEEE Network*, 35(2):194–201.

Mrabet, H., Belguith, S., Alhomoud, A. e Jemai, A. (2020). A survey of IoT security based on a layered architecture of sensing and data analysis. *Sensors*, 20(13):1–19.

Muhammad, A., Asad, M. e Javed, A. R. (2020). Robust early stage botnet detection using machine learning. Em *2020 International Conference on Cyber Warfare and Security (ICCWS)*, páginas 1–6, Islamabad, Pakistan. IEEE.

Muhammad, I. e Yan, Z. (2015). Supervised machine learning approaches: A survey. *ICTACT Journal on Soft Computing*, 5(3).

Muller, K.-R., Mika, S., Ratsch, G., Tsuda, K. e Scholkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201.

Musik, P. e Jaroensutasinee, K. (2007). Large-scale simulation using parallel computing toolkit and server message block. *WSEAS Transactions on Mathematics*, 6(2):369–372.

Nafea, M. S. e Ismail, Z. H. (2022). Supervised machine learning and deep learning techniques for epileptic seizure recognition using EEG signals—a systematic literature review. *Bioengineering*, 9(12).

Nakagawa, E. Y., Scannavino, K. R. F., Fabbri, S. C. P. F. e Ferrari, F. C. (2017). *Revisão sistemática da literatura em engenharia de software: teoria e prática*. Elsevier Brasil.

Nattkemper, T. W. e Wismüller, A. (2005). Tumor feature visualization with unsupervised learning. *Medical Image Analysis*, 9(4):344–351. Functional Imaging and Modeling of the Heart - FIMH03.

Neira, A., Borges, L., Araújo, A. e Nogueira, M. (2023a). Engenharia de sinais precoces de alerta para a predição de ataques DDoS. Em *Anais do XXVIII Workshop de Gerência e Operação de Redes e Serviços*, páginas 139–152, Porto Alegre, RS, Brasil. SBC.

Neira, A., Borges, L., Araújo, A. e Nogueira, M. (2023b). Unsupervised feature engineering approach to predict DDoS attacks. Em *IEEE Global Communications Conference*, Malaysia. IEEE.

Neira, A. B. d., Araujo, A. M. d. e Nogueira, M. (2023c). An intelligent system for DDoS attack prediction based on early warning signals. *IEEE Transactions on Network and Service Management*, 20(2):1254–1266.

Neira, A. B. d., Kantarci, B. e Nogueira, M. (2023d). Distributed denial of service attack prediction: Challenges, open issues and opportunities. *Computer Networks*, 222:109553.

Ngo, F. T., Agarwal, A., Govindu, R. e MacDonald, C. (2020). *Malicious Software Threats*, páginas 793–813. Springer International Publishing, Cham.

Nguyen, H., Tran, K., Thomassey, S. e Hamad, M. (2021). Forecasting and anomaly detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management. *International Journal of Information Management*, 57:102282.

Nilashi, M., Ahmadi, H., Manaf, A. A., Rashid, T. A., Samad, S., Shahmoradi, L., Aljojo, N. e Akbari, E. (2020). Coronary heart disease diagnosis through self-organizing map and fuzzy support vector machine with incremental updates. *International Journal of Fuzzy Systems*, 22(4).

Nogueira, M., Santos, A. A. e Moura, J. M. F. (2016). Non-parametric early warning signals from volumetric DDoS attacks. *CoRR*, abs/1609.09560.

Nooribakhsh, M. e Mollamotalebi, M. (2020). A review on statistical approaches for anomaly detection in DDoS attacks. *Information Security Journal: A Global Perspective*, 29(3):118–133.

Oikonomou, G., Mirkovic, J., Reiher, P. e Robinson, M. (2006). A framework for a collaborative DDoS defense. Em *2006 22nd Annual Computer Security Applications Conference (ACSAC'06)*, páginas 33–42, Miami Beach, FL, USA. IEEE.

Oja, H. (1981). On location, scale, skewness and kurtosis of univariate distributions. *Scandinavian Journal of Statistics*, 8(3):154–168.

Olabelurin, A., Veluru, S., Healing, A. e Rajarajan, M. (2015). Entropy clustering approach for improving forecasting in DDoS attacks. Em *2015 IEEE 12th International Conference on Networking, Sensing and Control*, páginas 315–320, Taipei, Taiwan. IEEE.

Omar, S., Ngadi, A. e Jebur, H. H. (2013). Machine learning techniques for anomaly detection: an overview. *International Journal of Computer Applications*, 79(2).

ONF (2021). Software-defined networking (SDN) definition: Acessado em: 07/2021. https://opennetworking.org/sdn-definition/.

O'Reilly (2021). Chapter 1. introduction to tensorflow: Acessado em: 12/2021. https://www.oreilly.com/library/view/ai-and-machine/9781492078180/ch01.html#introduction_to_tensorflow.

Padmanabhan, J. e Premkumar, M. J. J. (2015). Machine learning in automatic speech recognition: A survey. *IETE Technical Review*, 32(4):240–251.

Pak, M. e Kim, S. (2017). A review of deep learning in image recognition. Em *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, páginas 1–3.

Pan, M., Liu, Y., Cao, J., Li, Y., Li, C. e Chen, C.-H. (2020). Visual recognition based on deep learning for navigation mark classification. *IEEE Access*, 8:32767–32775.

Pang, G., Shen, C., Cao, L. e Hengel, A. V. D. (2021). Deep learning for anomaly detection: A review. *ACM Computing Surveys*, 54(2).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. e Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pelloso, M., Vergutz, A., Santos, A. e Nogueira, M. (2018). A self-adaptable system for DDoS attack prediction based on the metastability theory. Em *2018 IEEE Global Communications Conference (GLOBECOM)*, páginas 1–6, Abu Dhabi, United Arab Emirates. IEEE.

Peng, T., Leckie, C. e Ramamohanarao, K. (2007). Survey of network-based defense mechanisms countering the DoS and DDoS problems. *ACM Computing Surveys*, 39(1):3–es.

Pipis, G. (2020). Skewness and kurtosis in statistics. Acessado em: 06/2021. https://www.r-bloggers.com/2020/11/skewness-and-kurtosis-in-statistics/.

Pise, N. N. e Kulkarni, P. (2008). A survey of semi-supervised learning methods. Em *2008 International Conference on Computational Intelligence and Security*, volume 2, páginas 30–34.

Poulakis, G. (2020). Unsupervised AutoML: a study on automated machine learning in the context of clustering. Dissertação de Mestrado, Πανεπιστήμιο Πειραιώς.

Prathyusha, D. J., Naseera, S., Anusha, D. J. e Alisha, K. (2019). A review of biologically inspired algorithms in a cloud environment to combat DDoS attacks. Em *Smart Intelligent Computing and Applications*, páginas 59–68. Springer.

Proverbio, D., Kemp, F., Magni, S. e Gonçalves, J. (2022). Performance of early warning signals for disease re-emergence: A case study on COVID-19 data. *PLOS Computational Biology*, 18(3):e1009958.

Puterman, M. L. (1990). Chapter 8 Markov decision processes. Em *Stochastic Models*, volume 2 de *Handbooks in Operations Research and Management Science*, páginas 331–434. Elsevier, North-Holland.

Radain, D., Almalki, S., Alsaadi, H. e Salama, S. (2021). A review on defense mechanisms against distributed denial of service (DDoS) attacks on cloud computing. Em *2021 International Conference of Women in Data Science at Taif University (WiDSTaif)*, páginas 1–6, Taif, Saudi Arabia. IEEE.

Radware (2021). Teardrop attack. Acessado em: 07/2021. https://www.radware.com/security/ddos-knowledge-center/ddospedia/teardrop-attack.

Rahal, B. M., Santos, A. e Nogueira, M. (2020). A distributed architecture for DDoS prediction and bot detection. *IEEE Access*, 8:159756–159772.

Raschka, S. (2020). Chapter 1: Introduction to machine learning and deep learning. Acessado em: 12/2021. https://sebastianraschka.com/blog/2020/intro-to-dl-ch01.html.

Rauf, U. (2018). A taxonomy of bio-inspired cyber security approaches: Existing techniques and future directions. *Arabian Journal for Science and Engineering*, 43(12):6693–6708.

Ray, S. (2019). A quick review of machine learning algorithms. Em *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, páginas 35–39, Faridabad, India. IEEE.

Ren, P., Xiao, Y., Chang, X., Huang, P.-y., Li, Z., Chen, X. e Wang, X. (2021). A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Computing Surveys*, 54(4).

Riazul Islam, S., Uddin, M. N. e Kwak, K. S. (2016). The IoT: Exciting possibilities for bettering lives: Special application scenarios. *IEEE Consumer Electronics Magazine*, 5(2):49–57.

Ribin, J. S. e Kumar, N. (2019). Precursory study on varieties of DDoS attacks and its implications in cloud systems. Em *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, páginas 1003–1008, Tirunelveli, India. IEEE.

Riley, W. e Greenhall, C. (2004). Power law noise identification using the lag 1 autocorrelation. Em *2004 18th European Frequency and Time Forum (EFTF 2004)*, páginas 576–580, Inglaterra.

Rodrigues, B., Bocek, T., Lareida, A., Hausheer, D., Rafati, S. e Stiller, B. (2017). A Blockchain-Based architecture for collaborative DDoS mitigation with smart contracts. Em *Security of Networks and Services in an All-Connected World*, páginas 16–29, Cham. Springer International Publishing.

Rokach, L. (2005). *Ensemble Methods for Classifiers*, páginas 957–980. Springer US, Boston, MA.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

RU (2020). £1.2 billion for the world's most powerful weather and climate supercomputer. Acessado em: 08/2021. https://www.gov.uk/government/news/12-billion-for-the-worlds-most-powerful-weather-and-climate-supercomputer.

Ruijun, Y., Lijun, H., Xueqi, H. e Bin, Z. (2021). Research on industrial control network security based on automatic machine learning. Em *2021 6th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, volume 6, páginas 149–153, Japan.

Ruiz-Vanoye, J. A., Díaz-Parra, O., Trejo-Macotela, F. R. e Ramos-Fernández, J. C. (2020). Editorial: P versus NP problem from formal languages theory view. *International Journal of Combinatorial Optimization Problems and Informatics*, 12(1):1–8.

Russell, S. e Norvig, P. (2021). *Artificial Intelligence, A Modern Approach*. Pearson.

Sachdeva, M., Singh, G., Kumar, K. e Singh, K. (2009). A comprehensive survey of distributed defense techniques against DDoS attacks. *International Journal of Computer Science and Network Security*, 9(12):7–15.

Sahoo, K. S., Panda, S. K., Sahoo, S., Sahoo, B. e Dash, R. (2019). Toward secure software-defined networks against distributed denial of service attack. *The Journal of Supercomputing*, 75(8):4829–4874.

Sahu, S. K. e Khare, R. (2020). DDoS attacks & mitigation techniques in cloud computing environments. *Gedrag & Organisatie Review*, 33(2):2426–2435.

Salam, R. A., Das, J. K., Lassi, Z. S. e Bhutta, Z. A. (2016). Adolescent health and well-being: Background and methodology for review of potential interventions. *Journal of Adolescent Health*, 59(4, Supplement):S4–S10. Interventions to Address Adolescent Health and Well-Being: Current State of the Evidence.

Salas, J. D., Delleur, J. W., Yevjevich, V. e Lane, W. L. (1980). *Applied modeling of hydrologic time series*. Water Resources Publication.

Salemi, H., Rostami, H., Talatian-Azad, S. e Khosravi, M. R. (2021). Leaesn: Predicting DDoS attack in healthcare systems based on lyapunov exponent analysis and echo state neural networks. *Multimedia Tools and Applications*, -(-):1–22.

Salim, M. M., Rathore, S. e Park, J. H. (2020). Distributed denial of service attacks and its defenses in IoT: a survey. *The Journal of Supercomputing*, 76(7):5320–5363.

Samta, R. e Sood, M. (2020). Analysis and mitigation of DDoS flooding attacks in software defined networks. Em *International Conference on Innovative Computing and Communications*, páginas 337–355, Singapore. Springer Singapore.

Santanna, J. J., van Rijswijk-Deij, R., Hofstede, R., Sperotto, A., Wierbosch, M., Granville, L. Z. e Pras, A. (2015). Booters — an analysis of DDoS-as-a-service attacks. Em *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, páginas 243–251, Ottawa, ON, Canada. IEEE.

Santos, A. A., Nogueira, M. e Moura, J. M. F. (2017). A stochastic adaptive model to explore mobile botnet dynamics. *IEEE Communications Letters*, 21(4):753–756.

Sapienza, A., Ernala, S. K., Bessi, A., Lerman, K. e Ferrara, E. (2018). Discover: Mining online chatter for emerging cyber threats. Em *Companion Proceedings of the The Web Conference 2018*, WWW '18, página 983–990, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Sarker, I. H. (2021). Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2(6):1–20.

Sasaki, T., Gañán, C. H., Yoshioka, K., Eeten, M. V. e Matsumoto, T. (2020). Pay the piper: DDoS mitigation technique to deter financially-motivated attackers. *IEICE Transactions on Communications*, E103.B(4):389–404.

Savchenko, V., Ilin, O., Hnidenko, N., Tkachenko, O., Laptiev, O. e Lehominova, S. (2020). Detection of slow DDoS attacks based on user's behavior forecasting. *International Journal of Emerging Trends in Engineering Research*, 8(5).

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M. e Monfardini, G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Schapire, R. E. (2013). *Explaining AdaBoost*, páginas 37–52. Springer Berlin Heidelberg, Berlin, Heidelberg.

Scheffer, M. (2009). *Critical Transitions in Nature and Society*. Princeton University Press.

Schneider, B. (1999). Scalable test solutions: a means of improving ASIC performance and time-to-market in mixed-signal engineering test. Em *Fifth Workshop on Electronics for the LHC Experiments*, páginas 1–10, EUA. CERN.

Scikit-learn (2021). Gaussian processes. Acessado em: 08/2021. https://scikit-learn.org/stable/modules/gaussian_process.html.

SecurityWeek (2023). A cyberattack has disrupted hospitals and health care in five states. Acessado em: 09/2022. https://www.securityweek.com/a-cyberattack-has-disrupted-hospitals-and-health-care-in-five-states/.

Sefara, T. J. (2019). The effects of normalisation methods on speech emotion recognition. Em *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, páginas 1–8, Vanderbijlpark, South Africa. IEEE.

Sewak, M. (2019). *Deep reinforcement learning*. Springer.

Shanmuganathan, S. (2016). *Artificial Neural Network Modelling: An Introduction*, páginas 1–14. Springer International Publishing, Cham.

Shao, M., Wang, X., Bu, Z., Chen, X. e Wang, Y. (2020). Prediction of energy consumption in hotel buildings via support vector machines. *Sustainable Cities and Society*, 57:102128.

Sharafaldin, I., Lashkari, A. H. e Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. Em *International Conference on Information Systems Security and Privacy (ICISSP 2018)*. SCITEPRESS.

Sharafaldin, I., Lashkari, A. H., Hakak, S. e Ghorbani, A. A. (2019). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. Em *2019 International Carnahan Conference on Security Technology (ICCST)*, páginas 1–8.

Shin, S., Lee, S., Kim, H. e Kim, S. (2013). Advanced probabilistic approach for network intrusion forecasting and detection. *Expert Systems with Applications*, 40(1):315–322.

Shinde, P. P. e Shah, S. (2018). A review of machine learning and deep learning applications. Em *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, páginas 1–6, Pune, India. IEEE.

Shumway, R. H. e Stoffer, D. S. (2017). *Characteristics of Time Series*, páginas 1–44. Springer International Publishing, Cham.

Silva, G. L. F. M., de Neira, A. B. e Nogueira, M. (2022a). A deep learning-based system for DDoS attack anticipation. Em *2022 IEEE Latin-American Conference on Communications (LATINCOM)*, páginas 1–6.

Silva, G. M., Neira, A. e Nogueira, M. (2022b). Aprendizado profundo para a predição de ataques de negação de serviço distribuído. Em *Anais do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, páginas 475–488, Porto Alegre, RS, Brasil. SBC.

Silva, S. S., Silva, R. M., Pinto, R. C. e Salles, R. M. (2013). Botnets: A survey. *Computer Networks*, 57(2):378–403. Botnet Activity: Analysis, Detection and Shutdown.

Singh, A., Amutha, J., Nagar, J., Sharma, S. e Lee, C.-C. (2022). AutoML-ID: automated machine learning model for intrusion detection using wireless sensor network. *Scientific Reports*, 12(1).

Singh, A., Thakur, N. e Sharma, A. (2016). A review of supervised machine learning algorithms. Em *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, páginas 1310–1315.

Singh, P. (2019). *Supervised Machine Learning*, páginas 117–159. Apress, Berkeley, CA.

Singh, P., Manickam, S. e Rehman, S. U. (2014). A survey of mitigation techniques against economic denial of sustainability (EDoS) attack on cloud computing architecture. Em *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization*, páginas 1–4, Noida, India. IEEE.

Somani, G., Gaur, M. S., Sanghi, D., Conti, M. e Buyya, R. (2017). DDoS attacks in cloud computing: Issues, taxonomy, and future directions. *Computer Communications*, 107:30–48.

Sonar, K. e Upadhyay, H. (2014). A survey: DDoS attack on Internet of things. *International Journal of Engineering Research and Development*, 10(11):58–63.

Song, Y., Zheng, S., Li, L., Zhang, X., Zhang, X., Huang, Z., Chen, J., Wang, R., Zhao, H., Zha, Y., Shen, J., Chong, Y. e Yang, Y. (2021). Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, páginas 1–1.

Spaan, M. T. J. (2012). *Partially Observable Markov Decision Processes*, páginas 387–414. Springer Berlin Heidelberg, Berlin, Heidelberg.

Specht, S. M. e Lee, R. B. (2004). Distributed denial of service: Taxonomies of attacks, tools, and countermeasures. Em *Proceedings of the ISCA 17th International Conference on Parallel and Distributed Computing Systems, September 15-17*, páginas 543–550, California, USA. ISCA.

Squartini, T., van Lelyveld, I. e Garlaschelli, D. (2013). Early-warning signals of topological collapse in interbank networks. *Scientific Reports*, 3(1).

Srinivasan, K., Mubarakali, A., Alqahtani, A. S. e Dinesh Kumar, A. (2020). A survey on the impact of DDoS attacks in cloud computing: Prevention, detection and mitigation techniques. Em *Intelligent Communication Technologies and Virtual Mobile Networks*, páginas 252–270, Tirunelveli, India. Springer International Publishing.

Stamp, M. (2017). *Introduction to Machine Learning with Applications in Information Security*. CRC Press, United States of America.

Staudemeyer, R. C. e Morris, E. R. (2019). Understanding LSTM – a tutorial into long short-term memory recurrent neural networks.

Stresser-App (2021). DDoS-as-aService. Acessado em: 07/2021. https://stresser.app/.

Strogatz, S. H. (2018). *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. CRC press.

SU, Y., MENG, X., MENG, Q. e HAN, X. (2018). DDoS attack detection algorithm based on hybrid traffic prediction model. Em *2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, páginas 1–5, Qingdao, China. IEEE.

Sunindyo, W. D., Moser, T., Winkler, D. e Biffl, S. (2010). A process model discovery approach for enabling model interoperability in signal engineering. Em *Proceedings of the First International Workshop on Model-Driven Interoperability*, MDI '10, página 15–21, New York, NY, USA. Association for Computing Machinery.

Sutton, R. S. e Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Suzuki, K. (2017). Overview of deep learning in medical imaging. *Radiological Physics and Technology*, 10(3):257–273.

Sáez, J. A. e Corchado, E. (2019). A meta-learning recommendation system for characterizing unsupervised problems: On using quality indices to describe data conformations. *IEEE Access*, 7(-):63247–63263.

Takimoto, G. (2009). Early warning signals of demographic regime shifts in invading populations. *Population Ecology*, 51(3):419–426.

Tasa, F. A., Istiqomah, Murti, M. A. e Alinursafa, I. (2022). Classification of earthquake vibrations using the ANN (Artificial Neural Network) algorithm. Em *Artificial Intelligence, and Communications Technology (IAICT)*, páginas 102–107.

Taud, H. e Mas, J. (2018). *Multilayer Perceptron (MLP)*, páginas 451–455. Springer International Publishing, Cham.

Tayfour, O. E. e Marsono, M. N. (2020). Collaborative detection and mitigation of distributed denial-of-service attacks on software-defined network. *Mobile Networks and Applications*, 25(4):1338–1347.

Thatipalli, A., Aravamudu, P., Kartheek, K. e Dennisan, A. (2022). Exploring and comparing various machine and deep learning technique algorithms to detect domain generation algorithms of malicious variants. *Computer Science and Information Technologies*, 3(2).

Tian, Q., Han, D. e Du, Z. (2019). DDoS attack detection based on global unbiased search strategy bee colony algorithm and artificial neural network. *International Journal of Embedded Systems*, 11(5):584–593.

Tingley, B. (2021). The pentagon is experimenting with using artificial intelligence to "see days in advance". Acessado em: 08/2021. https://www.thedrive.com/the-war-zone/41771/the-pentagon-is-experimenting-with-using-artificial-intelligence-to-see-days-in-advance.

Toh, A., Vij, A. e Pasha, S. (2022). Azure DDoS protection—2021 Q3 and Q4 DDoS attack trends. Acessado em: 01/2022. https://azure.microsoft.com/en-us/blog/azure-ddos-protection-2021-q3-and-q4-ddos-attack-trends/.

Tomašev, N. e Radovanović, M. (2016). *Clustering Evaluation in High-Dimensional Data*, páginas 71–107. Springer International Publishing, Cham.

Torrecilla, J. S., García, J., García, S. e Rodríguez, F. (2011). Application of lag-k autocorrelation coefficient and the TGA signals approach to detecting and quantifying adulterations of extra virgin olive oil with inferior edible oils. *Analytica Chimica Acta*, 688(2):140–145.

Tse, A. e Carley, K. M. (2017). Event-based model simulating the change in DDoS attack trends after P/DIME events. Em *Social, Cultural, and Behavioral Modeling*, páginas 120–126, Cham. Springer International Publishing.

UE (2016). General data protection regulation: Acessado em: 06/2021. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679.

Uhlenbeck, G. E. e Ornstein, L. S. (1930). On the theory of the brownian motion. *Physical review*, 36:823–841.

ur Rasool, R., Wang, H., Ashraf, U., Ahmed, K., Anwar, Z. e Rafique, W. (2020). A survey of link flooding attacks in software defined network ecosystems. *Journal of Network and Computer Applications*, 172:102803.

Vaseghi, S. V. (1996). *Power Spectrum Estimation*, páginas 214–241. Vieweg+Teubner Verlag, Wiesbaden.

Venkatesh, B. e Anuradha, J. (2019). A review of feature selection and its methods. *Cybernetics and Information Technologies*, 19(1):3–26.

Vezhnevets, A. e Vezhnevets, V. (2005). Modest adaboost-teaching adaboost to generalize better. Em *Computer Graphics and Applications (GraphiCon'2005)*, páginas 987–997, Novosibirsk Akademgorodok, Rússia.

Vijayan, J. (2020). DDoS attacks spiked, became more complex in 2020. Acessado em: 07/2021. https://www.darkreading.com/attacks-breaches/ddos-attacks-spiked-became-more-complex-in-2020/d/d-id/1339814.

Visalatchi, L., Yazhini, P. e Scholar, M. P. (2020). The survey DDoS attack prevention and defense technique. *International Journal of Innovative Science and Research Technology*, 5(2).

Vlajic, N. e Slopek, A. (2014). Web bugs in the cloud: Feasibility study of a new form of EDoS attack. Em *2014 IEEE Globecom Workshops (GC Wkshps)*, páginas 64–69, Austin, TX, USA. IEEE.

Vormayr, G., Zseby, T. e Fabini, J. (2017). Botnet communication patterns. *IEEE Communications Surveys Tutorials*, 19(4):2768–2796.

Wang, A., Mohaisen, A. e Chen, S. (2017). An adversary-centric behavior modeling of DDoS attacks. Em *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, páginas 1126–1136, Atlanta, GA, USA. IEEE.

Wang, L., Han, M., Li, X., Zhang, N. e Cheng, H. (2021). Review of classification methods on unbalanced data sets. *IEEE Access*, 9:64606–64628.

Wang, P., Sparks, S. e Zou, C. C. (2010). An advanced hybrid peer-to-peer botnet. *IEEE Transactions on Dependable and Secure Computing*, 7(2):113–127.

Wang, Z., Hong, T. e Piette, M. A. (2020). Building thermal load prediction through shallow machine learning and deep learning. *Applied Energy*, 263:114683.

Wang, Z. e Zhang, Y. (2017). DDoS event forecasting using twitter data. Em *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, página 4151–4157, Melbourne, Australia. AAAI Press.

Webster, P. J. (2013). Improve weather forecasts for the developing world. *Nature*, 493(7430):17–19.

Wei, Y., Jang-Jaccard, J., Sabrina, F., Singh, A., Xu, W. e Camtepe, S. (2021). AE-MLP: a hybrid deep learning approach for DDoS detection and classification. *IEEE Access*, 9.

Welch, P. (1967). The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73.

Westfall, P. H. (2014). Kurtosis as peakedness, 1905–2014. r.i.p. *The American Statistician*, 68(3):191–195.

Wijman, T. (2023). Newzoo's global games market report 2023 Acessado em: 09/2023. https://newzoo.com/resources/trend-reports/newzoo-global-games-market-report-2023-free-version.

Williams, C. (2016). Today the web was broken by countless hacked devices – your 60-second summary: Acessado em: 06/2021. https://www.theregister.com/2016/10/21/dyn_dns_ddos_explained.

Winterford, B. (2010). Virgin to chase $20m in outage losses. Acessado em: 08/2021. https://www.itnews.com.au/news/virgin-to-chase-20m-in-outage-losses-234854.

Wissel, C. (1984). A universal law of the characteristic return time near thresholds. *Oecologia*, 65(1):101–107.

Wlosinski, L. G. (2019). Cybersecurity takedowns. *Information Systems Audit and Control Association (ISACA) Journal*, 6.

Wold, S., Esbensen, K. e Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37–52. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

Wood, A. e Stankovic, J. (2002). Denial of service in sensor networks. *Computer*, 35(10):54–62.

Wu, J. (2012). *Cluster Analysis and K-means Clustering: An Introduction*, páginas 1–16. Springer Berlin Heidelberg, Berlin, Heidelberg.

Wu, M. e Chen, L. (2015). Image recognition based on deep learning. Em *2015 Chinese Automation Congress (CAC)*, páginas 542–546.

Xie, Q., Guo, T., Chen, Y., Xiao, Y., Wang, X. e Zhao, B. Y. (2020). *Deep Graph Convolutional Networks for Incident-Driven Traffic Speed Prediction*, página 1665–1674. Association for Computing Machinery, New York, NY, USA.

Xing, J., Wu, W. e Chen, A. (2021). Ripple: A programmable, decentralized link-flooding defense against adaptive adversaries. Em *30th USENIX Security Symposium (USENIX Security 21)*, página 16, USA.

Xiong, X., Ozbay, K., Jin, L. e Feng, C. (2019). Dynamic prediction of origin-destination flows using fusion line graph convolutional networks. *CoRR*, abs/1905.00406.

Xu, H., Yu, W., Griffith, D. e Golmie, N. (2018). A survey on Industrial Internet of Things: A cyber-physical systems perspective. *IEEE Access*, 6:78238–78259.

Xu, J., Zhu, S., Guo, H. e Wu, S. (2021a). Automated labeling for robotic autonomous navigation through multi-sensory semi-supervised learning on big data. *IEEE Transactions on Big Data*, 7(1):93–101.

Xu, X. Y., Huang, Y. Z., Luo, Y. C., Ouyang, G. e Zhuang, H. (2021b). Research on traffic classification of auto machine-learning base on meta-QNN. *Journal of Physics: Conference Series*, 1757(1):012006.

Yadav, J. e Thakur, J. (2020). Botnet: Evolution life cycle architecture and detection techniques. *Mukt Shabd Journal*, 9(6):4265–4281.

Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T. e Yu, H. (2019a). Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13(3):1–207.

Yang, Y., Nan, F., Yang, P., Meng, Q., Xie, Y., Zhang, D. e Muhammad, K. (2019b). GAN-Based semi-supervised learning approach for clinical decision support in Health-IoT platform. *IEEE Access*, 7:8048–8057.

Yassein, M. B., Hmeidi, I., Al-Harbi, M., Mrayan, L., Mardini, W. e Khamayseh, Y. (2019). IoT-Based healthcare systems: A survey. Em *Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems*, DATA '19, New York, NY, USA. ACM.

Yates, R. F. (1920). Succeeding in signal engineering. *Scientific American*, 123(25):610–610.

Yin, K. e Nianqing, T. (2009). Study on the risk detection about network security based on grey theory. Em *2009 International Forum on Information Technology and Applications*, volume 1, páginas 411–413, Chengdu, China. IEEE.

Yoachimik, O. (2021). Cloudflare thwarts 17.2M rps DDoS attack — the largest ever reported. Acessado em: 08/2021. https://blog.cloudflare.com/cloudflare-thwarts-17-2m-rps-ddos-attack-the-largest-ever-reported/.

Yoachimik, O., Desgats, J. e Forster, A. (2023). Cloudflare mitigates record-breaking 71 million request-per-second DDoS attack. Acesso em: 05/2023. https://blog.cloudflare.com/cloudflare-mitigates-record-breaking-71-million-request-per-second-ddos-attack/.

Yoachimik, O. e Pacheco, J. (2023). DDoS threat report for 2023 Q2. Acessado em: 09/2022. https://blog.cloudflare.com/ddos-threat-report-2023-q2.

Yu, X. e Guo, H. (2019). A survey on IIoT security. Em *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, páginas 1–5, Singapore. IEEE.

Yusof, A. R., Udzir, N. I. e Selamat, A. (2019). Systematic literature review and taxonomy for DDoS attack detection and prediction. *International Journal of Digital Enterprise Technology*, 1(3):292–315.

Zahler, R. S. e Sussmann, H. J. (1977). Claims and accomplishments of applied catastrophe theory. *Nature*, 269(5631):759–763.

Zargar, S. T., Joshi, J. e Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Communications Surveys Tutorials*, 15(4):2046–2069.

Zayo (2023). The state of DDoS attacks. Acessado em: 10/2023. https://go.zayo.com/zayo-ddos-protection-ebook.

Zeidanloo, H. R. e Manaf, A. B. A. (2010). Botnet detection by monitoring similar communication patterns. *International Journal of Computer Science and Information Security (IJCSIS)*, 7(3):10.

Zhang, W., Yang, G., Lin, Y., Ji, C. e Gupta, M. M. (2018). On definition of deep learning. Em *2018 World Automation Congress (WAC)*, páginas 1–5.

Zhijun, W., Wenjing, L., Liang, L. e Meng, Y. (2020). Low-rate DoS attacks, detection, defense, and challenges: A survey. *IEEE Access*, 8:43920–43943.

Zhong, L., Cheng, L., Xu, H., Wu, Y., Chen, Y. e Li, M. (2017). Segmentation of individual trees from TLS and MLS data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(2):774–787.

Zhou, C. V., Leckie, C. e Karunasekera, S. (2010). A survey of coordinated attacks and collaborative intrusion detection. *Computers & Security*, 29(1):124–140.

Zhou, X. e Belkin, M. (2014). Chapter 22 - semi-supervised learning. Em *Academic Press Library in Signal Processing: Volume 1*, volume 1 de *Academic Press Library in Signal Processing*, páginas 1239–1269. Elsevier.

Zhu, X. e Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–130.

Zi, L., Yearwood, J. e Wu, X.-W. (2010). Adaptive clustering with feature ranking for DDoS attacks detection. Em *International Conference on Network and System Security*, páginas 281–286.

Zivot, E. e Wang, J. (2003). *Rolling Analysis of Time Series*, páginas 299–346. Springer New York, New York, NY.

Zou, Y. e Quan, L. (2017). A new service-oriented grid-based method for AIoT application and implementation. *Modern Physics Letters B*, 31(19-21):1740064.

Şenol, A. (2022). Viasckde index: A novel internal cluster validity index for arbitrary-shaped clusters based on the kernel density estimation. *Computational Intelligence and Neuroscience*, 2022:1–20.

APÊNDICE A – METODOLOGIA DA REVISÃO BIBLIOGRÁFICA

Este apêndice detalha o planejamento e a execução da revisão bibliográfica apresentada no Capítulo 3. Este trabalho apresentou uma extensa e rigorosa revisão da literatura sobre a predição de ataques DDoS. Para atingir este objetivo, este trabalho definiu e avaliou uma metodologia de busca baseada no trabalho de Nakagawa et al. (2017). O planejamento consiste em cinco etapas: definição do grupo de controle; criação e validação da *string* de busca; definição das fontes da pesquisa; definição dos critérios de inclusão e exclusão; e a avaliação do planejamento. A execução da revisão bibliográfica foi realizada em cinco etapas, busca de estudos, remoção de estudos duplicados, seleção inicial, seleção avançada e seleção final. Este apêndice está dividido em seis seções. A Seção A.1 apresenta o grupo de controle. A Seção A.2 detalha a *string* de busca usada para encontrar os estudos. A Seção A.3 especifica as fontes de pesquisa usadas na revisão bibliográfica. A Seção A.4 apresenta os critérios de seleção usados para selecionar ou rejeitar os estudos. Por fim, as Seções A.5 e A.6 apresentam a avaliação e a execução do planejamento respectivamente.

A.1 GRUPO DE CONTROLE DA REVISÃO BIBLIOGRÁFICA

O grupo de controle representa estudos previamente identificados pelo autor. Este conjunto de estudos trata da predição de ataques DDoS, e a versão final deste trabalho deve incluir esses estudos. Além disso, a *string* de busca é composta de termos presentes nesses estudos. A validação do planejamento também utiliza os estudos do grupo de controle. A Tabela A.1 descreve o conjunto de estudos selecionados para construir o grupo de controle. Este conjunto é composto por quatro estudos. O mais antigo foi publicado em 2015, e o mais novo no ano de 2018. O estudo de Pelloso et al. (2018) presente no grupo de controle foi publicado por membros do CCSC (do inglês, *Computational and network seCurity SCience*).

Tabela A.1: Conjunto de Estudos que Constituem o Grupo de Controle

Título	Ano	Referência
Entropy clustering approach for improving forecasting in DDoS attacks	2015	Olabelurin et al. (2015)
The early bird gets the Botnet: a markov chain based early warning system for Botnet attacks	2016	Abaid et al. (2016)
DDoS attack volume forecasting using a statistical approach	2017	Kwon et al. (2017)
A self-adaptable system for DDoS attack prediction based on the metastability theory	2018	Pelloso et al. (2018)

A.2 *STRING* DE BUSCA DA REVISÃO BIBLIOGRÁFICA

Após definir o objetivo e o grupo de controle, é necessário construir a *string* de busca. A *string* de busca é vital para o sucesso desta revisão. Ela define os termos utilizados para identificar as soluções existentes. Deste modo, a busca pode atingir todos os estudos que constituem a literatura relacionada à predição de ataques DDoS. A Tabela A.2 apresenta todas as palavras-chave utilizadas nesta revisão e seus sinônimos. O primeiro termo da *string* de busca é *distributed denial of service*, que tem o termo DDoS como sinônimo. O objetivo deste primeiro termo é delimitar a área de atuação deste trabalho, visto que o objetivo desta revisão considera apenas as soluções relacionadas ao ataque DDoS. O segundo parâmetro da *string* de busca é o termo *predict* com os sinônimos *early*, *forecast*, *forecasts*, *predicting*, *prediction*. O termo *predict*

também delimita o escopo deste trabalho devido ao objetivo, assim para serem considerados neste trabalho os estudos devem realizar a predição do ataque DDoS.

Neste trabalho, a predição de ataques acontece quando as soluções identificam sinais de possíveis ataques e produzem evidências do ataque. Ou seja, eles criam alertas relacionados a um ataque DDoS antes que o atacante inicie o ataque. Os termos *predicting* and *prediction* são variações do termo *predict*. Assim, essas variações possibilitam a *string* de busca ser executada em qualquer fonte de pesquisa. Os termos *forecast*, *forecasts* fornecem um novo horizonte de possibilidades para encontrar soluções que possam projetar o futuro e, assim, inferir a possibilidade de um ataque DDoS ocorrer antes do atacante iniciar o ataque. O termo *early* está em grupo sinônimo devido à possibilidade deste termo representar estudos que lidam com a predição de ataques DDoS. Além disso, o termo *early* está presente no estudo de Abaid et al. (2016) presente no grupo de controle. Por fim, o termo *solution* é a última palavra-chave definida. Seu objetivo é representar qualquer estudo disponível na literatura relacionado à predição de ataques DDoS. Por esse motivo, o termo solução possui 17 sinônimos.

Tabela A.2: Palavras-chave e Sinônimos da *String* de Busca

Palavras-chave	Sinônimos
distributed denial of service	DDoS
predict	*early*, *forecast*, *forecasts*, *predicting*, *prediction*
solution	*approach*, *approaches*, *framework*, *frameworks*, *method*, *methodologies*, *methodology*, *methods*, *procedure*, *procedures*, *solutions*, *system*, *systems*, *technologies*, *technology*, *tool*, *tools*

A Figura A.1 mostra a *string* de busca genérica usada neste trabalho. A *string* de busca é genérica, pois, em algumas fontes de pesquisa, é necessário especializá-la, mas sempre seguindo o padrão especificado na Figura A.1. As chaves "AND" e "OR" são os operadores booleanos que realizam a combinação de palavras-chave e sinônimos para formar a *string* de busca. O operador "OR" é usado para juntar os sinônimos correspondentes. Por exemplo, "*distributed denial of service*" OR "DDoS". O operador "AND" é usado para juntar toda a *string* de busca.

("distributed denial of service" OR "DDoS") AND
("predict" OR "early" OR "forecast" OR "forecasts" OR "predicting" OR "prediction") AND
("solution" OR "approach" OR "approaches" OR "framework" OR "frameworks" OR "method" OR "methodologies"OR "methodology" OR "methods" OR "procedure" OR "procedures" OR "solutions" OR "system" OR "systems" OR "technologies" OR "technology" OR "tool" OR "tools")

Figura A.1: *String* de Busca Genérica da Revisão Bibliográfica

A.3 FONTES DE PESQUISA DA REVISÃO BIBLIOGRÁFICA

A escolha das fontes de pesquisa também impacta o resultado deste trabalho. Este trabalho usa cinco fontes de pesquisa diferentes: ACM Digital Library, IEEE Xplore, ScienceDirect - Elsevier, Scopus e SpringerLink. IEEE Xplore, SpringerLink e ScienceDirect são fontes de pesquisa que indexam apenas os estudos publicados pelas plataformas. O Scopus é uma fonte de pesquisa que opera apenas como mecanismo de busca, indexando estudos publicados em outras plataformas. A ACM Digital Library opera de forma híbrida, publicando estudos e indexando

estudos publicados em outras plataformas. O objetivo da escolha dessas fontes de pesquisa é buscar identificar a maior quantidade de estudos relevantes para a composição deste trabalho. Para isso, este trabalho aplica a *string* de busca nos metadados dos estudos. O título, o resumo e as palavras-chave são os metadados considerados neste trabalho. Assim, as fontes de pesquisa encontraram apenas os estudos que satisfaçam a *string* de busca nesses campos. O objetivo desta decisão é aprimorar as pesquisas e identificar os estudos que tratam do tema. Deste modo, a busca não identifica estudos que apenas mencionam os termos ao longo do texto e fazem parte de outros domínios. Essa ação proporciona objetividade ao identificar apenas estudos relevantes. No entanto, a ACM Digital Library e o SpringerLink não possuem esse mecanismo. Assim, exclusivamente para a ACM Digital Library e SpringerLink, este trabalho utilizou todos os campos disponíveis para realizar a pesquisa.

A.4 CRITÉRIO DE SELEÇÃO DA REVISÃO BIBLIOGRÁFICA

Os critérios de seleção determinam a aceitação ou rejeição dos resultados encontrados nas buscas. Existem três critérios de inclusão e dez critérios de exclusão. A Tabela A.3 apresenta todos os critérios. Os dois primeiros critérios de inclusão definem o tipo de estudo e o período de atividade contemplado neste trabalho. Assim, para ser incluído neste trabalho, o resultado da pesquisa deve ser um estudo que propõe uma solução para predição de ataques DDoS e ter sido publicado até 22/08/2021. O terceiro critério define a inclusão de estudos que proponham soluções para a detecção precoce de ataques DDoS. Assim, esses estudos foram aceitos e analisados após a conclusão das buscas. Essa condição é necessária, pois alguns estudos podem considerar o termo detecção precoce de ataque DDoS como sinônimo de predição de ataque DDoS. Nestes casos, é necessário verificar quando as soluções produzem as evidências do ataque. Se a criação da evidência do ataque ocorrer antes do lançamento do ataque, este trabalho considera a solução como uma solução para a predição de ataques DDoS. Se a solução chamada de detecção antecipada produzir evidências do ataque depois que o atacante lançar o ataque, esses estudos serão descartados.

Tabela A.3: Critérios de Inclusão e Exclusão

Critério	Tipo	#
O estudo apresenta uma solução para a predição ou previsão de ataques DDoS	Inclusão	1º
O estudo foi publicado até 22/08/2021	Inclusão	2º
O estudo apresenta uma solução para detectar ataques DDoS antecipadamente	Inclusão	3º
Anais de conferências, apresentações e similares	Exclusão	4º
Revisões sistemáticas, *surveys* e similares	Exclusão	5º
Estudos não escritos em inglês	Exclusão	6º
Não avaliar a proposta	Exclusão	7º
Estudos sem resumo	Exclusão	8º
Estudos apenas com resumo disponível	Exclusão	9º
Estudos duplicados	Exclusão	10º
Versões mais antigas de estudos mais recentes	Exclusão	11º
Não ter acesso ao texto completo	Exclusão	12º
Estudo sobre outro tema	Exclusão	13º

Os critérios de exclusão complementam os critérios de inclusão e adicionam algumas restrições específicas. Os resultados das buscas são retirados da análise se forem slides de uma apresentação, revisões da literatura, estudos escritos em outro idioma que não o inglês, estudos

que não apresentam resultados para a proposta, estudos sem resumo ou apenas com resumo disponível. Algumas fontes de pesquisa podem indexar o mesmo estudo, causando duplicidade de resultados. Portanto, apenas uma das ocorrências foi considerada. Se dois ou mais resultados corresponderem a versões mais antigas de estudos mais recentes, este trabalho descartou as versões mais antigas. Este trabalho removeu os estudos caso o texto completo não estivesse disponível para acesso. Por fim, o último critério de exclusão prevê a remoção de resultados que abordam a identificação/detecção do ataque DDoS, a predição de um ataque diferente do ataque DDoS ou que trate de qualquer problema que não seja a predição do ataque DDoS.

A.5 AVALIAÇÃO DO PLANEJAMENTO DA REVISÃO BIBLIOGRÁFICA

A avaliação do planejamento analisa se as decisões tomadas até o momento são compatíveis com o objetivo definido. A avaliação do planejamento consiste em aplicar a *string* de pesquisa na fonte de pesquisa IEEE Xplore e verificar se o resultado da pesquisa abrange todo o grupo de controle. Esta ação foi projetada desta forma pois todos os estudos do grupo de controle estão disponíveis no IEEE Xplore. Como resultado, todos os quatro estudos do grupo controle foram encontrados nos resultados desta pesquisa preliminar, validando assim o planejamento apresentado.

A.6 EXECUÇÃO DA REVISÃO BIBLIOGRÁFICA

A Figura A.2 apresenta a execução e os resultados das cinco etapas utilizadas para a execução da revisão bibliográfica. A primeira etapa consiste em aplicar a *string* de busca em todas as fontes de pesquisa definidas no planejamento. A Tabela A.4 detalha o número de estudos encontrados em cada fonte de pesquisa. A pesquisa na ACM Digital Library resultou em 1266 estudos, sendo a fonte de pesquisa que retornou mais resultados. A busca realizada na Scopus resultou em 681 estudos, sendo a segunda fonte de pesquisa com mais resultados. O fato da busca realizada na ACM Digital Library possuir quase o dobro da Scopus pode ser explicada por dois fatores. Primeiramente a ACM Digital Library é uma fonte de pesquisa híbrida, deste modo ela indexa trabalhos de outras fontes de pesquisas. Além disso, como especificado anteriormente, a ACM Digital Library não possui a opção para realizar buscas nos metadados, deste modo, a *string* de busca foi aplicada em todo o texto, apresentando assim mais resultados.

Na Etapa 2, a ferramenta Parsifal[1] removeu 361 estudos duplicados. A duplicação dos estudos é possível devido ao Scopus e a ACM Digital Library encontrarem estudos em outras fontes de pesquisas, duplicando assim os resultados. A Etapa 3 compreende a leitura dos resumos de todos os 2121 estudos. Ao ler os resumos do estudo, este trabalho avaliou se cada estudo satisfaz os critérios de seleção. Caso este trabalho identifique que o estudo satisfaz quaisquer critérios de exclusão, este trabalho exclui o estudo. Quando este trabalho identifica que os estudos satisfazem os critérios de inclusão, os estudos seguem para a Etapa 4. Caso exista dúvida sobre a aceitação ou exclusão do estudo apenas com a leitura do resumo, o estudo é aceito e segue para a quarta etapa. Essa ação é adequada, pois na quarta etapa este trabalho conduz a leitura da introdução e a conclusão de todos os estudos aceitos. Durante a quarta etapa, este trabalho aplicou novamente os critérios de seleção mencionados anteriormente. Assim, ao final desta etapa, este trabalho retirou 264 estudos e iniciou a quinta etapa com 62 estudos. Na Etapa 5, este trabalho realizou a leitura completa dos estudos, resultando na seleção de 25 estudos. Por fim, em maio de 2022 foi conduzida uma atualização na revisão da literatura. Essa atualização seguiu o protocolo apresentado anteriormente, além da adição do motor de busca Google Scholar. Nesta

[1] https://parsif.al/

atualização foram revisados cerca de 200 estudos, dos quais apenas dois foram considerados aptos a serem incluídos nesta revisão. Assim, a revisão da literatura apresentada no Capítulo 3 conta com 27 estudos publicados até maio de 2022.

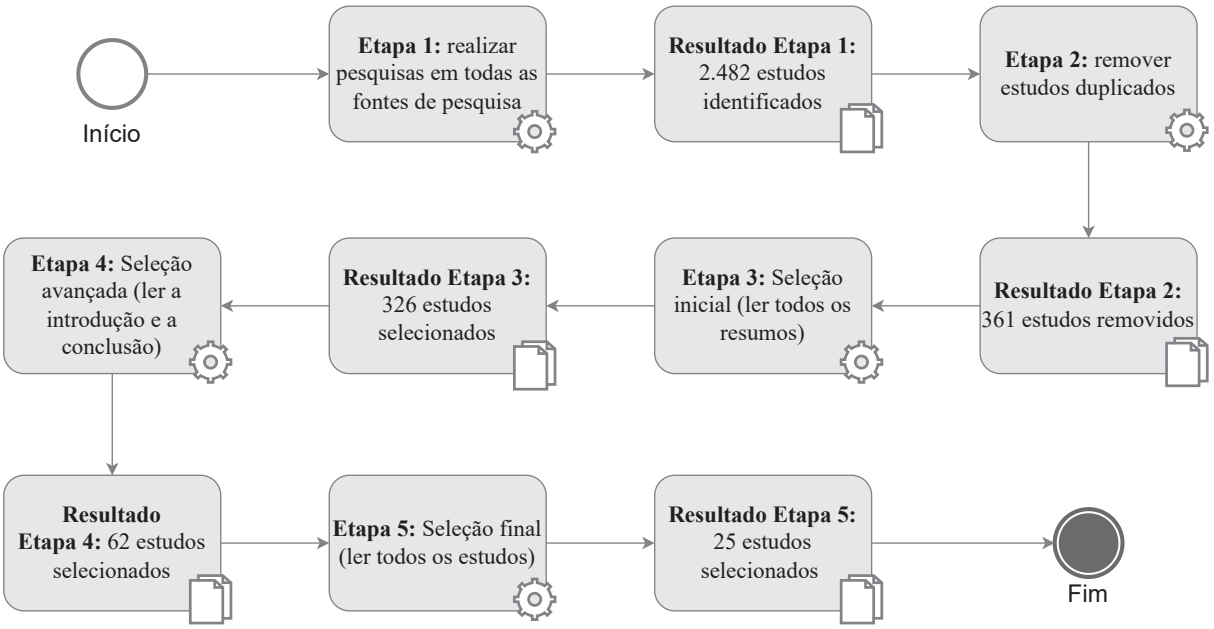


Figura A.2: Execução da Revisão Bibliográfica

Tabela A.4: Total de Estudos Identificados por Fonte de Pesquisa

Fonte de pesquisa	Total de estudos
ACM Digital Library	1266
IEEE Xplore	241
ScienceDirect - Elsevier	29
Scopus	681
SpringerLink	265
Total	**2482**

APÊNDICE B – TRABALHOS RELACIONADOS COM OS SINAIS PRECOCES DE ALERTA

A literatura apresenta o uso de sinais precoces de alerta para prever ocorrências futuras em diferentes áreas. Para complementar a revisão bibliográfica (Capítulo 3), este apêndice apresenta trabalhos relacionados com o uso dos sinais precoces de alerta. A Seção B.1 apresenta o uso de sinais precoces de alerta usados para predizer mudanças climáticas. A Seção B.2 apresenta como os terremotos podem ser antecipados usando os sinais precoces de alerta. A Seção B.3 expõe trabalhos onde a variação populacional pode ser prevista por meio da teoria dos sinais precoces de alerta. A Seção B.4 apresenta o uso de vários sinais precoces em diferentes áreas. Por fim, a Seção B.5 discute o uso de sinais precoces de alerta presentes na literatura.

B.1 SINAIS PRECOCES DE ALERTA EM MUDANÇAS CLIMÁTICAS

Dakos et al. (2008) investigaram a possibilidade de os sinais precoces de alerta anteciparem transições críticas em mudanças climáticas abruptas que aconteceram no passado. Para realizar essa pesquisa, os autores utilizaram registros geológicos de oito eventos com mudanças climáticas abruptas e a da autocorrelação de *lag-1* (AC-1) como sinal precoce de alerta. Os autores verificaram que antes das mudanças climáticas abruptas, em todos os oito eventos, existe um aumento na AC-1. Esse aumento ocorreu devido à desaceleração dos sistemas observados conforme um ponto crítico estava se aproximando. Isso ocorre, pois, a desaceleração é refletida na diminuição das taxas de mudança do sistema, apresentando um aumento na autocorrelação da série temporal. Essa pesquisa é importante, pois, mesmo atualmente, predizer eventos futuros não é uma tarefa trivial. Portanto, identificar que em todos esses eventos ocorreu um aumento da AC-1 antes da mudança real é um ponto de partida para a literatura poder prever novos acontecimentos semelhantes.

Boers e Rypdal (2021) avaliaram a aplicabilidade dos sinais precoces de alerta (variância e a AC-1) para a predição do derretimento da camada de gelo da Groenlândia. Os autores avaliaram os sinais precoces sobre o derretimento do gelo no centro-oeste da Groenlândia e sobre as flutuações da altura do manto de gelo. Em ambos os casos, os autores identificaram aumentos nos dois sinais precoces de alerta utilizados, sugerindo que, em breve, o derretimento de gelo da Groenlândia pode sofrer uma transição crítica. Ou seja, uma mudança permanente de comportamento poderá ocorrer. Essa conclusão sugere que medidas para evitar a progressão do derretimento do gelo sejam adotadas. Os resultados apresentados por Boers e Rypdal (2021) corroboram parte da literatura sobre o tema. Contudo, o tema é sensível e deve continuar recebendo a atenção dos pesquisadores.

B.2 SINAIS PRECOCES DE ALERTA PARA PREDIZER TERREMOTOS

Sinais precoces de alerta também podem ser utilizados para predizer terremotos. Um terremoto possui ao menos dois tipos de ondas, as primárias (ondas P) e as secundárias (ondas S) (Anggraini et al., 2021). As ondas P ocorrem antes de um terremoto e não são destrutivas. As ondas S são destrutivas e ocorrem após as ondas P. Em geral, o intervalo de tempo entre essas ondas é de 60 e 90 segundos. A proposta de Anggraini et al. (2021) é identificar as ondas P usando o *skewness*, o *kurtosis* e algoritmos de *deep learning*. Para isso, os autores utilizam séries temporais que representam a medição do movimento do solo que foram realizadas por três

estações diferentes. Esses dados são utilizados por uma DNN para detectar a ocorrência de uma onda P, e consequentemente predizer um terremoto. Os autores treinaram a DNN usando 1455 medições do movimento do solo relacionados com terremotos e testaram a DNN usando 624 medições do movimento do solo. A solução obteve 93% de acurácia para predizer os terremotos. Contudo, os autores verificaram que a solução proposta não conseguiu predizer os terremotos originados longe dos sensores. Pois a acurácia da proposta neste caso foi de 42%.

Tasa et al. (2022) usaram uma ANN para categorizar as vibrações relacionadas ou não a terremotos. Para isso, os autores propõem o treinamento da ANN usando as seguintes características: média, mediana, máximo, mínimo, *skewness* e *kurtosis*. Essas características são extraídas de dados que representam o movimento do solo, incluindo terremotos. Os dados correspondem a 54 terremotos e estão disponíveis publicamente (Luzi et al., 2016). O melhor resultado obtido pelos autores foi utilizar a ANN com três camadas ocultas, sigmoide como função de ativação e usando 90% dos dados para o treinamento e 10% para o teste. Neste caso, a acurácia no teste foi de 99,12%. Para verificar a relevância dos resultados, os autores reproduziram o teste sem a geração de novas características, ou seja, sem utilizar o *skewness* e o *kurtosis*. Neste caso, a acurácia ficou em torno de 70% bem abaixo do resultado anterior. Contudo, a acurácia de 99,12% está fortemente relacionada com a configuração da ANN e com os dados de treinamento. Variando a arquitetura da ANN para duas camadas ocultas, mudando a função de ativação, a acurácia da proposta não atingiu 98%.

B.3 SINAIS PRECOCES DE ALERTA EM VARIAÇÕES POPULACIONAIS

Takimoto (2009) utilizou os sinais precoces de alerta *skewness*, o desvio padrão e a taxa de retorno para predizer variações da população de espécies invasoras. Takimoto (2009) aplica os sinais precoces de alerta sobre a variação da população invasora para predizer uma explosão populacional repentina. É importante prever mudanças populacionais, pois os invasores podem superar as espécies nativas, encaminhando-as para a extinção, ou seja, uma transição crítica. Os resultados indicam que a taxa de retorno foi o sinal precoce de alerta que melhor predisse a explosão populacional. Porém, sempre é importante ter cautela ao avaliar predições baseadas em apenas um indicador estatístico. Portanto, é conveniente utilizar outras características dos sistemas para evitar que interpretações equivocadas causem atitudes equivocadas.

Drake e Griffen (2010) avaliaram empiricamente a capacidade dos sinais precoces de alerta em antecipar variações populacionais. Para isso, os autores utilizaram o CV, a autocorrelação, o *skewness* e a correlação espacial. Eles aplicaram esses indicadores estatísticos sobre os dados de duas populações da mesma espécie de zooplâncton. A primeira população de zooplâncton compõe o grupo de controle, nesse caso, nenhuma variação nas condições ambientais foi imposta a essa população. A segunda população é referente ao grupo de tratamento, onde efetivamente os autores aplicaram mudanças ambientais com o intuito de avaliar a variação populacional e, consequentemente, a extinção populacional. Nos primeiros dias do experimento, os autores mantiveram as mesmas condições para as duas populações. Após o dia 154, o grupo de tratamento começou a enfrentar um lento declínio na disponibilidade de alimentos. Os autores mantiveram os níveis iniciais de alimentos para o grupo de controle. A limitação da disponibilidade de alimentos causou a diminuição da população do grupo de tratamento, culminando na extinção desse grupo. O mesmo não aconteceu com o grupo de controle, que continuou existindo por vários dias após a extinção do grupo de tratamento. Os autores verificaram que os quatro indicadores estatísticos (o CV, a autocorrelação, o *skewness* e a correlação espacial) apresentaram aumento substancial em comparação ao grupo de controle. O coeficiente de variação (CV) e o *skewness* foram os indicadores estatísticos que mais apresentaram variação.

Os autores verificaram aumentos significativos nesses indicadores estatísticos 110 dias antes da transição crítica.

Bury et al. (2020) evoluem os sinais precoces de alerta ao propor meios para identificar pistas sobre o tipo da transição crítica vindoura, e não apenas que uma transição crítica está se aproximando. Os autores utilizaram a teoria de Ornstein–Uhlenbeck (Uhlenbeck e Ornstein, 1930) para identificar características únicas antes de cada tipo de transição crítica. Esses resultados geraram sinais precoces de alerta espectrais e fornecem menor influência frente a ruídos nos dados, além das pistas sobre a transição crítica que está se aproximando. Os sinais precoces de alerta espectral baseiam-se no fenômeno de avermelhamento espectral. Assim, à medida que uma transição crítica se aproxima, o espectro de potência muda em direção a frequências menores. Os autores utilizam o S_{max}, S_{null}, S_{fold} e S_{hopf}. O S_{max} indica a proximidade da transição crítica. O S_{fold} e o S_{hopf} indicam qual é o tipo da transição crítica que está por vir, e o S_{null} indica que nenhuma transição do tipo *fold* ou *hopf* está se aproximando. Os autores aplicaram esses sinais precoces de alerta sobre dados com variações populacionais. Esses dados possuem duas transições críticas, uma *fold* e uma *hopf*. Em ambos os casos, o S_{max} foi um sinal precoce de alerta mais confiável que a variância. Além disso, os sinais precoces de alerta identificaram corretamente o tipo de transição crítica em ambos os casos. Apesar dos resultados relevantes, os autores citam que existem casos como nas transições críticas do tipo *hopf*, o avermelhamento espectral não precede a transição crítica. Nesses casos, os sinais precoces de alerta espectrais não fornecem sinais da ocorrência de transições críticas futuras.

B.4 SINAIS PRECOCES DE ALERTA APLICADOS EM OUTRAS ÁREAS

Carpenter e Brock (2006) avaliaram desvio padrão da concentração de fósforo na água para antecipar mudanças de regimes em lagos. Antecipar mudanças de regimes é importante para controlar a qualidade da água e evitar a mortandade de peixes que podem ocasionar perdas econômicas. Por meio de simulações, os autores verificaram que a mudança de regime pode ser prevista anos antes da real mudança. Essa predição ocorreu pois os autores identificaram um aumento no desvio padrão de séries temporais de fósforo na água do lago. Assim, mesmo nos casos em que o modelo matemático que rege o sistema não é conhecido, variações no desvio padrão podem ser um indício importante de mudanças no sistema.

Bury et al. (2021) usam a teoria dos sinais precoces de alerta de modo diferente da literatura convencional. Os autores não utilizam indicadores estatísticos como *skewness*, *kurtosis*, CV ou AC-1 para antecipar as transições críticas. Ao invés disso, os autores simularam séries temporais que possuem transições críticas *fold*, *hopf* e *transcritical*, além de séries temporais que não possuem transições críticas. Com os resultados dessas simulações, os autores treinaram uma rede neural do tipo CNN-LSTM para assim identificar possíveis ocorrências de transições críticas. A saída de rede neural é a probabilidade de, no futuro, a série temporal apresentar uma transição crítica (*fold*, *hopf* e *transcritical*) ou de não apresentar uma transição crítica. Os autores compararam a proposta com a variação e a AC-1. Os resultados mostram que, para os cenários avaliados, a proposta dos autores apresentou AUC superior ou equivalente aos sinais precoces de alerta. Contudo, a proposta pode falhar em qualquer cenário que fuja do treinamento simulado pelos autores. Os sinais precoces de alerta (variação e AC-1) podem se adaptar melhor aos diferentes cenários por serem genéricos.

Proverbio et al. (2022) avaliaram os sinais precoces de alerta para predizer novas ondas da pandemia da COVID-19 em países como Cingapura, Coreia do Sul, Japão e países que integram a União Europeia. Os autores realizaram a predição por meio do cálculo da variância, AC-1, o CV e o *skewness* sobre o total de casos ativos. Os resultados indicam que os sinais

precoces de alerta são úteis para o monitoramento ativo da dinâmica epidêmica. A variância e o AC-1 foram os sinais precoces de alerta que apresentaram melhores resultados dentre todos os avaliados. Contudo, os autores advertem ser necessário cautela na interpretação dos sinais precoces de alerta, pois ruídos podem obscurecer os resultados.

B.5 DISCUSSÃO SOBRE OS TRABALHOS RELACIONADOS COM OS SINAIS PRECOCES DE ALERTA

Compreender a dinâmica populacional e aprimorar a capacidade de previsão de eventos futuros é essencial para a humanidade (Drake e Griffen, 2010). A literatura dos sinais precoces de alerta é uma linha de pesquisa capaz de fornecer evidências sobre a dinâmica dos sistemas observáveis, podendo antecipar eventos futuros. Contudo, prever corretamente eventos futuros ainda é um desafio complexo (Bury et al., 2020; Proverbio et al., 2022). A literatura apresenta a utilização de sinais precoces de alerta em aplicações de diferentes áreas. Este apêndice apresenta o uso dos sinais precoces de alerta para antecipar a extinção de espécies (Takimoto, 2009) e o derretimento de gelo (Boers e Rypdal, 2021), para predizer a ocorrência de terremotos (Tasa et al., 2022) e a evolução da pandemia de COVID-19 (Proverbio et al., 2022). Além desses estudos, a literatura avalia a utilização dos sinais precoces de alerta no mercado financeiro (Squartini et al., 2013; Diks et al., 2018), na neurociência (Meisel et al., 2015) e na análise de falhas de componentes eletrônicos (Livina et al., 2020). A literatura dos sinais precoces de alerta possui estudos recentes como os trabalhos de Proverbio et al. (2022) e Tasa et al. (2022), porém o tema é estudado desde a década de 1970 (Zahler e Sussmann, 1977; Matsumoto e Kunisawa, 1978; Wissel, 1984). Por fim, estudos sobre o tema foram publicados em veículos de relevância internacional como o *Springer Nature* (Drake e Griffen, 2010; Squartini et al., 2013).