UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

HENRIQUE WERNER DELAZERI

Explorando Aprendizado por Reforço para Detecção e Mitigação de Ataques DDoS em Redes de Comunicação

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação

Orientador: Prof. Dr. Luciano Paschoal Gaspary

Porto Alegre
2024

”Computers are useless.

They can only give you answers.”

— Pablo Picasso

AGRADECIMENTOS

Primeiramente, expresso minha gratidão à minha família, que me forneceu todo o suporte e incentivo necessários para concluir o curso. A minha mãe, Cristina, e meu pai, Airton, por todo o esforço para que nada me faltasse e todos os anos formativos de minha vida fossem os melhores possíveis, desde a creche até a presente graduação. A minha irmã, Giovana, por muitas vezes dizer o óbvio que era necessário. A minha namorada, Camille, por entender meus momentos de estresse e me incentivar a todos os momentos.

Agradeço também a meu orientador, Luciano Gaspary, pelo apoio e incentivo no desenvolvimento deste trabalho, me levando ao caminho certo com entusiasmo e compartilhando comigo o conhecimento e experiência adquirido na carreira acadêmica.

Agradeço também a UFRGS, ao Instituto de Informática e seus professores por todo o conhecimento compartilhado, desafios propostos e conquistas alcançadas.

Além disto, agradeço a meus amigos e colegas de curso que contribuíram, direta ou indiretamente, para minha formação como pessoa e profissional, sem eles o caminho teria sido muito mais difícil.

RESUMO

À medida que o número de sistemas conectados em redes de comunicação aumenta, também cresce a atenção de atores maliciosos a esses sistemas e, consequentemente, a preocupação com a proteção dos dados sensíveis dos usuários. Ataques DDoS são considerados um dos principais riscos à segurança, sobrecarregando a infraestrutura e impedindo o acesso legítimo aos sistemas. Estes ataques, muitas vezes realizados por meio de *BotNets*, têm se tornado mais frequentes e de maior escala, representando um desafio significativo para a detecção e mitigação eficazes. Esses ataques estão em constante evolução, exigindo sistemas de detecção adaptáveis e eficientes. Neste contexto, propõe-se um modelo de detecção de ataques distribuídos de negação de serviço baseado em aprendizado por reforço. Este modelo diminui a necessidade de grandes conjuntos de dados categorizados e etapas de treinamento, uma vez que os agentes de detecção podem ajustar seu comportamento de acordo com suas necessidades. Além disso, a capacidade de aprendizado contínuo dos agentes permite a detecção de novos tipos de ataques sem intervenção humana. A modelagem proposta foi analisada através da criação de cenários de ataque com base em dados sintéticos contendo diversos tipos de ataques DDoS. Os resultados sugerem que o modelo baseado em aprendizado por reforço consegue detectar e mitigar ataques DDoS eficientemente, adaptando-se a novos tipos de ataques sem a necessidade de retreinamento.

Palavras-chave: Redes de Computadores. DDoS. Classificação de Tráfego de Rede. Aprendizado por Reforço.

Exploring Reinforcement Learning for Detection and Mitigation of Attacks in Communication Networks

ABSTRACT

As the number of connected systems in communication networks increases, so does the attention of malicious actors to these systems and, consequently, the concern for protecting users' sensitive data. DDoS attacks are considered one of the main security risks, overloading infrastructure and preventing legitimate access to systems. These attacks, often carried out through BotNets, have become more frequent and larger in scale, posing a significant challenge for effective detection and mitigation. These attacks are constantly evolving, requiring adaptable and efficient detection systems. In this context, we propose a reinforcement learning-based distributed denial-of-service attack detection model. This model reduces the need for large labeled datasets and training steps, as detection agents can adjust their behavior according to their needs. Additionally, the agents' continuous learning capability allows for the detection of new attack types without human intervention. The proposed model was evaluated by creating attack scenarios based on synthetic data containing various types of DDoS attacks. The results suggest that the reinforcement learning-based model can efficiently detect and mitigate DDoS attacks, adapting to new attack types without the need for retraining.

Keywords: Computer Networks. DDoS. Network Traffic Classification. Reinforcement Learning.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

DoS	*Denial of Service*
DDoS	*Distributed Denial of Service*
HTTP	*Hypertext Transfer Protocol*
IP	*Internet Protocol*
LDAP	*Lightweight Directory Access Protocol*
MDP	*Markov Decision Process*
RL	*Reinforcement Learning*
TCP	*Transmission Control Protocol*
UDP	*User Datagram Protocol*

SUMÁRIO

1 INTRODUÇÃO

Com o crescimento das redes de comunicação, cada vez mais sistemas estão sendo conectados, vários deles processando dados sensíveis de seus usuários. Com a quantidade de sistemas conectados, maior o interesse de atores maliciosos em comprometer o funcionamento desses sistemas. Como reportado pela Akamai Technologies (2023), os dois setores mais atacados são serviços financeiros e de comércio, onde existem grandes incentivos financeiros, seja em recompensas para atacantes ou causando perdas para a vítima.

Ataques de negação de serviço (*Denial of Service*, DoS), mais especificamente ataques distribuídos (*Distributed Denial of Service*, DDoS), são considerados um dos principais riscos à segurança das redes de comunicação, tendo, aproximadamente, 7,9 milhões de ataques dessa natureza sido registrados na primeira metade de 2023, representando mais de 44 mil ataques por dia (NETSCOUT, 2023). Esses tem como principal objetivo impedir que usuários legítimos do sistema não consigam utilizá-lo, sobrecarregando a infraestrutura que provê o serviço. Ataques distribuídos normalmente fazem o uso de *BotNets*, conjuntos de dispositivos infectados com algum *malware* que permita um controle remoto, iniciando diversas requisições ao serviço alvo do ataque (Cloudflare, 2023).

Além da quantidade, o tamanho desses ataques tem também crescido (Menscher, 2020), e em 2023 o Google relatou ter impedido o maior ataque DDoS já registrado, tendo mais de 398 milhões de requisições por segundo (Kiner; April, 2023). Além disso, conforme reportado por NETSCOUT (2023), menos da metade dos ataques DDoS são mitigados. Vale ressaltar, ainda, que os ataques estão sempre evoluindo, fazendo o uso de novas técnicas e vetores de exploração para atingir suas vítimas, como o maior ataque registrado pelo Google, que fez uso de um método até então desconhecido.

Segundo a análise feita por Xia et al. (2015), o uso de redes definidas por software tem se tornado cada vez mais comum em grandes centros de dados, separando o plano de controle dos dispositivos de encaminhamento da rede do plano de dados. Com esta separação, os planos de dados de cada dispositivo tomam decisões de encaminhamento com base em políticas definidas pelo plano de controle. Uma das aplicações considerada na definição destas políticas é a classificação de tráfego e detecção de ataques, fazendo uso de algoritmos de aprendizado de máquina (Isyaku et al., 2023). Entretanto, diversos desses modelos de aprendizado, uma vez treinados e instalados na rede, não sofrem novas mudanças e adaptações sem a necessidade de uma nova etapa de treinamento e instalação

utilizando um novo conjunto de dados. Além do problema de rigidez dos modelos, grande parte desses algoritmos possui uma fase de treinamento supervisionada, ou seja, necessita de uma grande quantidade de dados rotulados. A obtenção desses dados é uma tarefa difícil devido ao volume de dados necessários e à mudança constante nos métodos de ataque, limitando a evolução dos agentes de detecção.

Neste trabalho, propõe-se um modelo de sistema de detecção de ataques distribuídos de negação de serviço, fazendo uso de um agente de detecção com algoritmos de aprendizado por reforço, visando a reduzir o impacto das dificuldades apresentadas. A necessidade de dados e fases de treinamento é reduzida, dada a capacidade dos algoritmos de adaptarem o seu comportamento conforme seu uso. Outra vantagem potencial dessa adaptabilidade do agente é sua capacidade de detectar novos tipos de ataque sem a necessidade de intervenção.

A modelagem proposta foi avaliada por meio da construção de cenários de ataque utilizando os dados disponibilizados por Sharafaldin et al. (2019). Os resultados sugerem que o uso de modelos baseados em aprendizado de máquina é uma abordagem viável para detecção e mitigação de ataques, especialmente devido à adaptabilidade dos agentes, levando os agentes a aprender a detectar ataques para os quais o agente não foi treinado.

O restante do trabalho está organizado como segue. O Capítulo 2 apresenta os fundamentos teóricos necessários para o total entendimento do trabalho. O Capítulo 3 propõe um modelo de sistema para detecção de ataques DDoS utilizando técnicas de aprendizado por reforço. O Capítulo 4 apresenta uma implementação do modelo de detecção, enquanto o Capítulo 5 apresenta os experimentos e resultados obtidos com essa implementação. Para finalizar, o Capítulo 6 apresenta uma análise sobre o trabalho e encaminhamentos para possíveis trabalhos futuros.

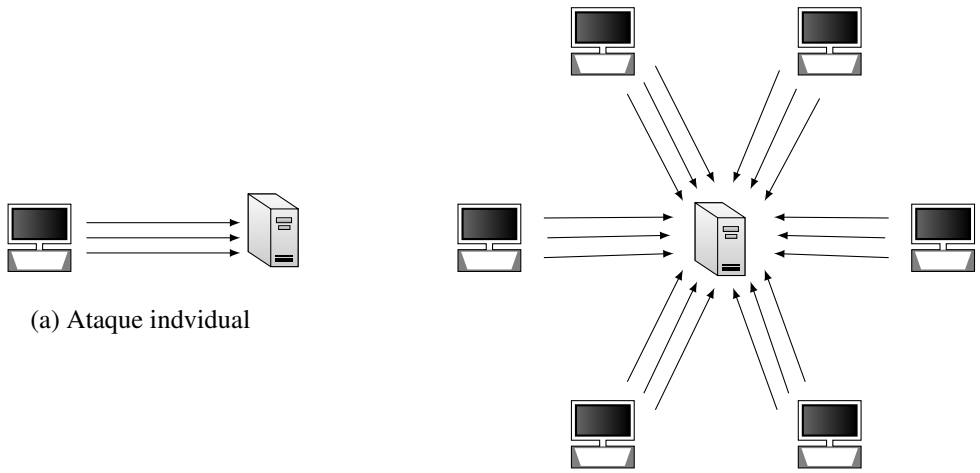
2 FUNDAMENTOS

Neste capítulo são apresentados os conceitos fundamentais para o entendimento do trabalho. Primeiramente são apresentados os conceitos de ataques de negação de serviço. Na sequência, são apresentados os fundamentos de aprendizado por reforço.

2.1 Ataques de Negação de Serviço

O objetivo de um ataque de negação de serviço é impedir que usuários legítimos acessem um dado serviço explorando vulnerabilidades de protocolos de comunicação ou exaurindo recursos de infraestrutura empregados pelo serviço (Zargar; Joshi; Tipper, 2013). Esses ataques podem partir de um ou mais dispositivos. Ataques individuais (*Denial of Service*, DoS) partem de um único dispositivo operado pelo ator malicioso, enquanto ataques distribuídos (*Distributed Denial of Service*, DDoS) partem de diversos dispositivos. Ambos os tipos de ataque são exemplificados na Figura 2.1.

Figura 2.1: Ataques de negação de serviço



(a) Ataque indvidual

(b) Ataque distribuído

Fonte: Elaborada pelo autor

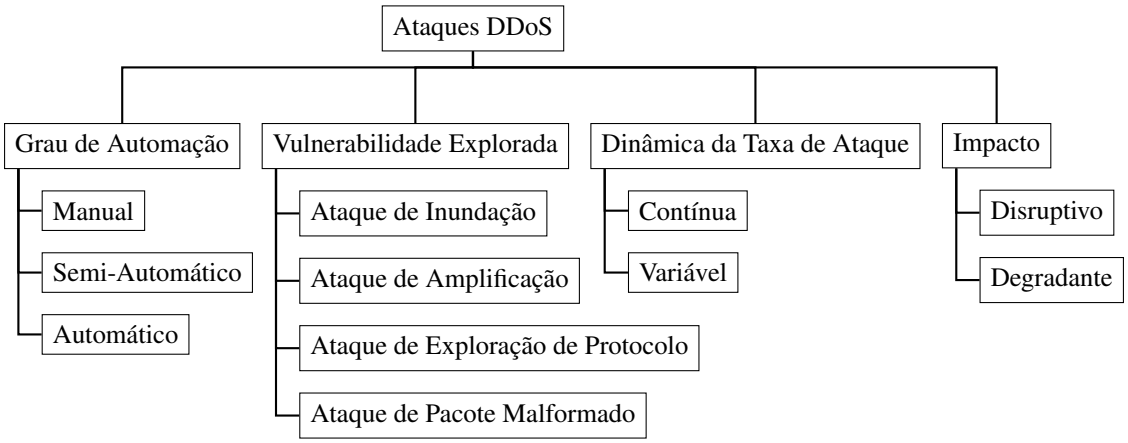
Na Figura 2.1, os computadores representam os dispositivos de origem do ataque, enquanto o servidor representa os recursos de infraestrutura do serviço atacado. Na Figura 2.1a a infraestrutura é atacada a partir de apenas uma origem, exemplificando um ataque DoS, enquanto na Figura 2.1b é exemplificado um ataque DDoS, onde diversas origens realizam o ataque em conjunto.

Ataques distribuídos tornaram-se maiores e mais frequentes, tendo um crescimento exponencial entre 2010 e 2020 (Menscher, 2020). Esses ataques têm sido fa-

cilitados pelo crescimento das chamadas *BotNets*, redes de dispositivos infectados com *malware*, permitindo que um atacante os utilize para exaurir os recursos de infraestrutura do serviço (Cloudflare, 2023). Devido a essa característica, detectar e mitigar esse tipo de ataque é uma tarefa mais complexa que a detecção de um ataque individual. Isto se dá, pois, em ataques distribuídos, o fluxo de dados do ataque é dividido entre as origens, cada uma responsável por uma pequena fração do fluxo total do ataque, dificultando a detecção de cada fluxo como maligno.

Segundo Douligeris e Mitrokotsa (2004), ataques de negação de serviço distribuídos podem ser classificados por meio de suas características. Essas características são apresentadas na Figura 2.2 e abordadas mais profundamente na sequência.

Figura 2.2: Classificação de ataques DDoS



Fonte: Adaptada de Douligeris e Mitrokotsa (2004)

Classificação por Grau de Automação

Dada a forma como os ataques são iniciados e controlados, eles podem ser divididos em três categorias, como segue.

- *Ataques Manuais:* o atacante busca, infecta e instala o código do ataque manualmente nos dispositivos que serão utilizados para o ataque, necessitando que comandos sejam enviados para os dispositivos infectados para iniciar o ataque.
- *Ataques Semi-Automáticos:* o atacante faz o uso de buscas automáticas para detectar dispositivos vulneráveis e infectá-los com o código do ataque, porém ainda necessitam que comandos de início do ataque sejam enviados.
- *Ataques Automáticos:* neste tipo de ataque, todas as etapas são automatizadas, tendo as informações do ataque (quando ocorrerá, duração e vítima) definidas de

maneira estática no *malware* utilizado.

Classificação por Vulnerabilidade Explorada

O ataque DDoS pode explorar uma ou mais vulnerabilidades existentes nas vítimas, sendo classificados conforme o vetor do ataque nas seguintes categorias, como segue.

- *Ataque de Inundação:* os dispositivos infectados enviam grandes volumes de tráfego à vítima, exaurindo os recursos responsáveis por prover o serviço.
- *Ataque de Amplificação:* o atacante envia requisições com o endereço IP de origem mascarado como endereços IP da vítima para serviços que não validam a identidade da requisição, logo esses serviços encaminham as respostas à vítima, gerando um tráfego não solicitado.
- *Ataque de Exploração de Protocolo:* esses ataques exploram uma funcionalidade ou erro de implementação de algum protocolo específico para consumir excessivamente recursos de infraestrutura.
- *Ataque de Pacote Malformado:* pacotes IP malformados são enviados à vítima visando a colapsar o serviço, impedindo que ele responda a solicitações legítimas.

Classificação por Dinâmica da Taxa de Ataque

Um ataque DDoS pode ter comportamentos diferentes conforme o avanço do tempo de ataque. Um ataque contínuo gera a mesma quantidade de tráfego durante toda a sua duração, sem pausas ou reduções de carga. Ataques variáveis alteram suas características de tráfego, aumentando gradativamente ou incluindo flutuações, visando a dificultar a sua detecção.

Classificação por Impacto

Dadas as repercussões do ataque, ele pode ser classificado em disruptivo ou degradante. Um ataque disruptivo inviabiliza por completo o acesso ao serviço atacado, enquanto um ataque degradante consome apenas uma parte dos recursos, dificultando o acesso e aumentando o tempo de resposta do serviço.

2.2 Aprendizado por Reforço

Em um problema de Aprendizado por Reforço (*Reinforcement Learning*, RL), um agente autônomo é inserido em um ambiente com o qual ele pode interagir por meio de ações, resultando em um novo estado do ambiente e em uma recompensa. O processo de decisão de qual ação deve ser tomada a cada momento é modelado por um Processo de Decisão de Markov (*Markov Decision Process*, MDP), pois cada escolha depende exclusivamente do estado atual, desconsiderando a sequência de ações que levaram o agente a essa situação (Sutton; Barto, 2018).

Um MDP é caracterizado por uma tupla (S, A, T, R), onde S é o conjunto de possíveis estados do ambiente, A é o conjunto de ações que podem ser executadas em cada estado, T é o modelo de transição, indicando qual a probabilidade de, estando em um estado s e executando a ação a, o agente ir para o estado s', e R é a função que retorna a recompensa da ação a aplicada ao estado s.

Segundo Russel e Norvig (2021, p. 841), as abordagens para solução de um problema de RL podem ser divididas em duas grandes categorias: aprendizado baseado em modelos e aprendizado livre de modelos. Em abordagens baseadas em modelos, o agente observa o resultado de suas ações e utiliza os sinais de recompensa recebidos para construir um modelo do ambiente onde o agente se encontra, aprendendo uma função $U(s)$ representando o valor esperado para a soma das recompensas obtidas no futuro, considerando que o agente se encontra no estado atual s.

Nos casos em que a construção de modelo do ambiente não é desejável ou não facilita a resolução do problema, é necessário adotar uma abordagem livre de modelos. Essas abordagens podem ser divididas em duas categorias: busca por políticas e aprendizado de ação-utilidade. Em abordagens de busca por política, o agente aprende uma política $\pi(s)$ que mapeia diretamente o estado s para a ação a ser tomada. Já em abordagens de aprendizado de ação-utilidade, o agente aprende uma função de qualidade $Q(s, a)$ representando a soma esperada das recompensas a partir do estado s se a ação a for tomada.

A título de exemplo, considere um turista visitando uma cidade na qual ele nunca esteve, sem acesso a um mapa e com o objetivo de visitar um ponto turístico específico. Em uma abordagem baseada em modelo, o viajante anda pela cidade construindo um mapa (modelo) da cidade, anotando a ação tomada a cada intersecção e a recompensa recebida. O mapa construído não é um mapa completo da cidade, porém completo o

suficiente para resolver a tarefa de visitar o ponto turístico utilizando a melhor rota.

Considerando o mesmo exemplo, em uma abordagem livre de modelo, o turista anda pela cidade sabendo apenas em qual intersecção (estado) ele se encontra e anotando a ação tomada e a recompensa recebida, ignorando a rota utilizada para chegar até esse ponto, para que, caso ele se encontre novamente na mesma intersecção, ele possa levar os acontecimentos anteriores em consideração na tomada de decisão. Neste caso não é construído um mapa da cidade, porém a cada intersecção o turista conhece a melhor ação a ser tomada.

Algoritmo Q-Learning

Q-Learning (Watkins, 1989) é um algoritmo para a solução de problemas de aprendizado por reforço utilizando uma abordagem livre de modelo e de aprendizado de ação-utilidade. O algoritmo visa a aprender uma função $Q(s, a)$, que aproxima a função q_*, a função de valor de ação ótima para o problema (Sutton; Barto, 2018, p. 131).

Os valores aproximados $Q(s, a)$ são armazenados em uma matriz $|S| \times |A|$, denominada *Q-Table*, inicializada com valores arbitrários em cada posição, exceto para posições nas quais o estado s é terminal, onde o valor é definido como 0. Os valores $Q(s, a)$ são atualizados conforme a seguinte equação:

$$Q(s,a) = (1-\alpha)Q(s,a) + \alpha \left[R(s,a) + \gamma \max_{a' \in A} Q(s',a') \right] \tag{2.1}$$

Na Equação (2.1), α é o coeficiente de aprendizado, controlando o peso do novo valor observado perante o valor atualmente armazenado na tabela, e γ é o fator de desconto das recompensas futuras, determinando qual o peso das recompensas futuras nas próximas escolhas. O pseudocódigo do algoritmo Q-Learning é apresentado no Algoritmo 1.

Problema da Exploração de Possíveis Ações

Abordagens de aprendizado por reforço livres de modelo precisam lidar com o problema da exploração/refinamento. No Algoritmo 1 esse problema se apresenta na Linha 4, onde o agente deve escolher uma ação a ser tomada baseado no seu conhecimento adquirido nas escolhas anteriores. Se o agente sempre escolher a opção com maior recompensa esperada, refinando seu modelo, pode ocorrer de ele não explorar uma parte do espaço de transições do modelo, porém, se o agente sempre escolher uma ação aleatória,

Algoritmo 1: Pseudocódigo do algoritmo Q-Learning

Entrada: $\alpha \in [0, 1], \gamma \in [0, 1]$
Saída: Função Q que aproxima q_*

```
para cada episódio de treinamento faça
    s ← estado inicial
    enquanto s é não terminal faça
        Obter a conforme o critério de exploração
        Tomar a ação a, observar a recompensa r e o estado resultante s'
        Q(s, a) = (1 − α)Q(s, a) + α [r + γ max_{a'∈A} Q(s', a')]
        s ← s'
    fim
fim
```

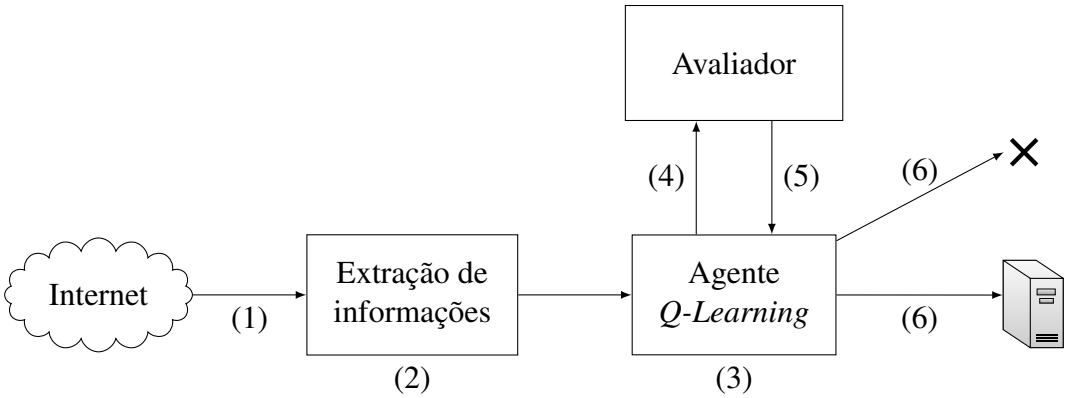
explorando novas partes do ambiente, ele necessita de uma maior quantidade de época de treinamento para aproximar a função q_*.

Watkins (1989) apresenta uma solução simples para esse problema, utilizando um fator $\epsilon \in [0, 1]$ que determina a probabilidade de o agente executar uma ação aleatória dentre as possíveis. No início do treinamento, esse valor é próximo de 1, fazendo com que o agente decida em favor da exploração na maioria das vezes. Conforme o treinamento avança e o agente conhece mais sobre o ambiente, esse valor vai sendo reduzido, favorecendo cada vez mais o refinamento do modelo conhecido.

3 SISTEMA PARA DETECÇÃO DE ATAQUES DDOS POR MEIO DE APRENDIZADO POR REFORÇO

Este capítulo tem por objetivo apresentar o modelo de sistema proposto para detectar os ataques DDoS. A Figura 3.1 apresenta uma visão geral do sistema. Após a chegada de um novo pacote (1), as informações relevantes são extraídas (2) e encaminhadas para o agente de aprendizado por reforço, que toma a sua decisão (3) quanto ao pacote ser malicioso ou legítimo. Essa decisão é encaminhada para um avaliador (4), que devolve ao agente a recompensa pela sua decisão (5). Finalizando esse processo, o agente encaminha o pacote ao destino ou para descarte (6). Cada componente do sistema será explorado com mais detalhes nas seções a seguir.

Figura 3.1: Modelo do sistema proposto



Fonte: Elaborada pelo autor

3.1 Extração de Informações

Um pacote de rede carrega, além dos dados enviados ao destino, diversas informações utilizadas para que o roteamento desse pacote da origem ao seu destino seja concluído. Com essas informações, é possível realizar a medição e a agregação/consolidação de novas métricas relevantes para o tipo de ataque a ser detectado.

Como apresentado por Sharafaldin et al. (2019) nos exemplos de ataques utilizados, diferentes métricas são relevantes para a detecção de diferentes tipos de ataque. Um exemplo dessa afirmação é o valor do campo `ACK` de pacotes de dados, métrica com maior relevância para a detecção de ataques de inundação de pacotes `SYN` e irrelevante para a detecção de ataques de inundação UDP. Tal ocorre, pois pacotes `SYN` e o controle de transmissões e retransmissões usando os valores de `ACK` fazem parte do processo de

gerenciamento de uma conexão TCP (Postel, 1981) e não existem em envios via UDP (Postel, 1980).

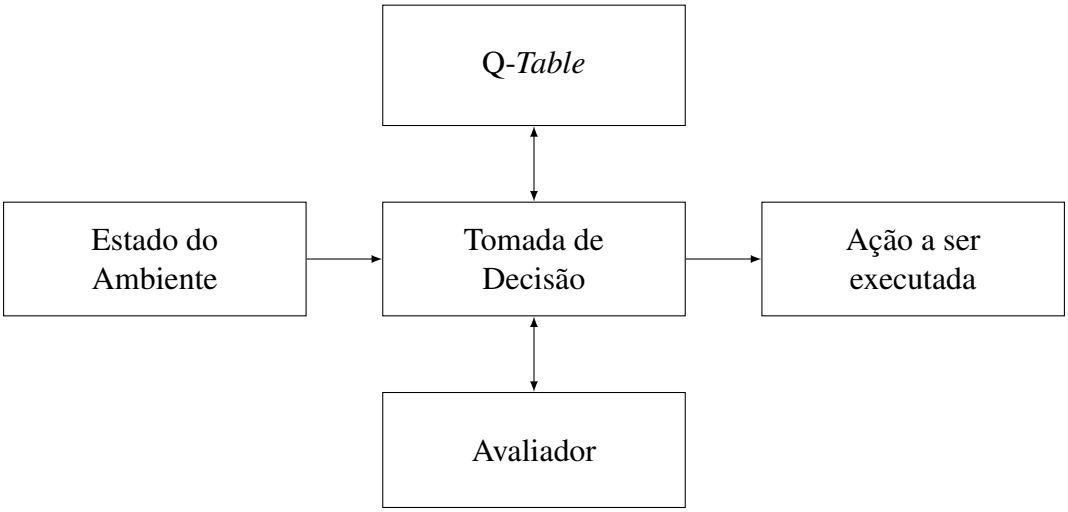
Parte das informações relevantes para a detecção de um ataque são computadas conforme novas conexões são abertas, necessitando um armazenamento e computação desses dados. Uma métrica relevante para a detecção de grande parte dos ataques DDoS é o intervalo entre chegada de pacotes vindos de uma mesma origem, onde um intervalo muito pequeno pode indicar um ataque.

No modelo proposto, esta etapa do processo recebe os dados do pacote que entra na rede, computa as métricas relevantes e encaminha os dados ao agente decisor. Esse componente é adaptável ao tipo de ataque a ser detectado. Existe a possibilidade de uma implementação do componente ser substituída por outra focada na detecção de outro ataque, ou, ainda, em uma implementação mais generalizada visando à detecção de mais de um tipo de ataque.

3.2 Agente

O agente presente no modelo proposto faz uso do mecanismo de Q-*Learning* para aprender a realizar a detecção dos ataques. A Figura 3.2 apresenta uma visualização detalhada sobre o funcionamento do agente.

Figura 3.2: Modelo de agente proposto



Fonte: Elaboarda pelo autor

Os dados extraídos do pacote trafegando pela rede são mapeados para um estado do ambiente, o qual o agente está analisando. Ao receber esses dados, a rotina de tomada de decisão do agente consulta a Q-*Table*, utilizando os valores do estado, para determinar

qual a melhor ação a ser tomada. Tendo essa ação, o agente se comunica com o Avaliador para determinar se essa escolha foi acertada e, recebendo a recompensa, atualiza a Q-*Table* seguindo o apresentado na Equação (2.1). O agente então, baseado em sua decisão, encaminha o pacote ao destino ou para descarte.

O algoritmo Q-*Learning* possui alguns parâmetros que podem ser ajustados para alterar o comportamento do agente na tomada de decisões. Para o modelo proposto, os parâmetros foram ajustados da seguinte maneira: o coeficiente de aprendizado α foi mantido entre 10% e 30%, e o fator de desconto das recompensas futuras γ foi definido como 0. A escolha do coeficiente de aprendizado nessa faixa de valores se dá para evitar grandes mudanças de comportamento quando o agente recebe uma recompensa diferente da esperada. Já a escolha do fator de desconto das recompensas foi definido da maneira mencionada porque o agente é utilizado para uma tarefa de classificação, importando apenas se a escolha atual foi correta ou não, independente do estado futuro do ambiente.

Além disso, para a política de escolha de ações, foram utilizados valores ϵ no intervalo $[0.1, 0.7]$, com um decaimento linear conforme as épocas de treinamento avançam. Tal permite ao agente uma abordagem mais exploratória no início do treinamento e a realização de escolhas mais informadas conforme o avanço do agente.

3.3 Avaliador

O avaliador presente no modelo proposto recebe do agente os dados do pacote e a decisão tomada, sendo responsável por avaliar essa decisão. Retorna ao agente uma recompensa pela decisão tomada, sendo ela positiva em caso de acerto ou negativa, caso contrário.

O avaliador pode ser implementado de diversas maneiras, podendo, em fases iniciais, ser substituído por um componente que conheça a decisão correta sobre cada pacote na entrada, sendo utilizado para treinar uma política para o agente. Após o treinamento inicial, quando instalado em uma rede de comunicação, esse componente pode fazer uso de um sistema de classificação de tráfego mais robusto que analisa uma quantidade maior de dados de cada pacote com maior profundidade. Este sistema pode ser utilizado para avaliar uma parcela dos pacotes que trafegam pela rede visando reduzir a penalidade de desempenho causada pela análise mais complexa realizada por este sistema.

4 IMPLEMENTAÇÃO

Neste capítulo são apresentados os métodos utilizados para implementar cada parte do modelo apresentado no Capítulo 3. Primeiramente é apresentado o conjunto de dados utilizado. Na sequência é descrita a implementação do processo de extração de informações e do avaliador, implementados em conjunto para a validação do modelo. Por fim, é apresentada a implementação do agente utilizado para classificação de tráfego.

4.1 Escolha dos Dados

Para o desenvolvimento do trabalho, foi utilizado o *dataset* `CIC-DDoS2019`, disponibilizado pelo Instituto Canadense de Cibersegurança (Sharafaldin et al., 2019). Os dados contidos nesse conjunto foram gerados visando a representar de maneira realista ataques DDoS comuns, também incluindo tráfego benigno entre os pacotes de rede pertencentes ao ataque.

Os dados disponibilizados são divididos em dois conjuntos, gerados em dois dias diferentes, com ataques simulados em cada dia. Esses conjuntos são utilizados, separadamente, para treinamento e validação dos agentes. Em cada dia, foram simulados diferentes tipos de ataques DDoS de inundação, amplificação e explorações de protocolos.

O conjunto de dados contém mais de 70 milhões de pacotes capturados. Cada pacote possui um conjunto de informações que o caracterizam, formando uma tupla com 87 itens de dados e um rótulo identificando se o pacote é benigno ou maligno e, em caso de tráfego maligno, a qual tipo de ataque o pacote pertence. Os dados de cada pacote apresentam, entre outros:

- dados de identificação do fluxo (identificador único, data e hora da captura);
- dados de identificação de origem e destino (IPs e portas);
- protocolo de camada de aplicação utilizado (HTTP, DNS, LDAP, etc.);
- métricas do pacote (tamanho do pacote, intervalo entre chegadas, valores de campos do cabeçalho do pacote, entre outros); e
- estatísticas calculadas sobre os dados dos pacotes (médias, desvios padrão, mínimos e máximos).

Devido à grande quantidade de informações disponibilizadas para cada pacote, a primeira etapa do trabalho consistiu em uma análise das informações visando a determi-

nar a relevância de cada item, tendo como objetivo reduzir a quantidade de dados a ser processada a cada iteração. Sharafaldin et al. (2019) disponibilizam, em conjunto com o *dataset*, quais métricas são mais relevantes para cada tipo de ataque e as métricas relevantes para a detecção de tráfego benigno. Após examinar os dados fornecidos, se optou por empregar essas métricas, resultando na criação de uma versão reduzida do conjunto de dados que inclui somente as informações pertinentes e suas correspondentes categorizações. No restante do trabalho, quando citado o conjunto de dados, está-se considerando essa versão reduzida.

Um exemplo de instância presente no conjunto de dados, para o ataque `SYN`, contém os dados `ACK Flag Count`, `Init_Win_bytes_forward`, `min_seg_size_forward`, `Fwd IAT Total` e `Flow Duration`, organizados em uma tupla, enquanto para o ataque `DNS` são considerados `Max Packet Length`, `Fwd Packet Length Max`, `Fwd Packet Length Min`, `Average Packet Size` e `Min Packet Length`. A lista completa de dados relevantes para cada ataque pode ser encontrada em Sharafaldin et al. (2019), enquanto a explicação de cada métrica pode ser encontrada em Canadian Institute for Cybersecurity (2020).

Realizando uma análise exploratória no *dataset*, foi detectado um desbalanceamento entre as diferentes classes. As proporções de classes sobre o total de pacotes para treinamento e teste são apresentadas nas Tabelas 4.1 e 4.2, respectivamente. Apesar dessa diferença, optou-se por não realizar um balanceamento entre as classes, visto que uma grande diferença de quantidade de tráfego é um comportamento esperado de ataques DDoS.

Tabela 4.1: Distribuição de classes no conjunto de treinamento

Rótulo	**Quantidade de pacotes**	**Proporção**
BENIGN	56.965	0,279727%
LDAP	1.915.122	9,404207%
MSSQL	5.787.453	28,419288%
NetBIOS	3.657.497	17,960139%
Portmap	186.960	0,918067%
SYN	4.891.500	24,019711%
UDP	3.867.155	18,989665%
UDPLag	1.873	0,009197%

Fonte: Elaborada pelo autor

Durante a análise exploratória, também foi detectada a discrepância entre os tipos de ataques realizados nos conjuntos de treinamento e teste. Essa diferença não foi tratada

Tabela 4.2: Distribuição de classes no conjunto de teste

Rótulo	**Quantidade de pacotes**	**Proporção**
BENIGN	56.863	0,113583%
DrDoS_DNS	5.071.011	10,129236%
DrDoS_LDAP	2.179.930	4,354364%
DrDoS_MSSQL	4.522.492	9,033581%
DrDoS_NTP	1.202.642	2,402252%
DrDoS_NetBIOS	4.093.279	8,176238%
DrDoS_SNMP	5.159.870	10,306730%
DrDoS_SSDP	2.610.611	5,214640%
DrDoS_UDP	3.134.645	6,261387%
SYN	1.582.289	3,160589%
TFTP	20.082.580	40,114526%
UDP-lag	366.461	0,731998%
WebDDoS	439	0,000877%

Fonte: Elaborada pelo autor

visando a analisar a capacidade do agente de se adaptar a novos tipos de ataques com os quais ele não teve contato.

4.2 Extração de Informações e Avaliador

Durante a implementação do modelo, optou-se por implementar conjuntamente a etapa de extração de informações e a etapa de avaliação dos pacotes. Essa escolha se deu devido ao uso do *dataset* como fonte única de informações, sem a implementação de um sistema separado para avaliação dos pacotes. Essa implementação garante uma avaliação precisa e sem incertezas para cada pacote, possibilitando que a recompensa dada ao agente identifique com precisão se as escolhas de ações foram adequadas.

A implementação das referidas etapas foi feita através da construção de um ambiente compatível com a biblioteca `Gymnasium` (Towers et al., 2023). Esse ambiente simula a chegada de um novo pacote e a avaliação da decisão tomada pelo agente a cada item do *dataset*.

Para realizar essa implementação, foi necessário converter os dados recebidos de entrada para uma tupla de itens representando o estado atual da simulação. A implementação feita extrai as informações relevantes para cada ataque do conjunto de dados e os converte em uma tupla.

O ambiente foi implementado de maneira a considerar todo estado subsequente à

aplicação da ação escolhida como um estado final. Isso se deve ao fato de que o agente é responsável pela classificação dos dados de entrada, não se importando com os resultados de suas ações no ambiente.

Para realizar a avaliação da escolha de ação do agente, o ambiente encaminha os dados para o agente e espera a decisão da ação a ser tomada. Com a resposta do agente contendo a ação escolhida, o ambiente compara essa escolha com a verdadeira classificação do pacote de dados, extraída em conjunto com as informações de estado, e retorna ao agente uma recompensa positiva, em caso de acerto, ou uma recompensa negativa, em caso de erro.

4.3 Agente Q-*Learning*

O agente Q-*Learning* foi implementado utilizando a linguagem Python 3 (Rossum; Drake, 2009), seguindo a definição do Algoritmo 1. Essa equação necessita de dois parâmetros: coeficiente de aprendizado (α) e coeficiente de desconto (γ).

Para o coeficiente de aprendizado, foi definido o valor de 10% visando a evitar mudanças bruscas no comportamento do agente em caso de escolhas incorretas e recompensas inesperadas. Já para o coeficiente de desconto, foi escolhido o valor 0, visto que o agente funciona como um modelo de classificação de pacotes, sendo irrelevantes as recompensas futuras (apenas se a decisão atual foi correta ou não).

O problema da exploração/refinamento foi abordado utilizando a estratégia ϵ-*greedy*, ou seja, durante a escolha de uma ação, o agente tem uma probabilidade ϵ de escolher uma ação aleatória ao invés de escolher a melhor ação possível. O valor desse parâmetro foi considerado em um intervalo $[0.1, 0.7]$, iniciando o treinamento do agente com o valor 0,7 e, conforme o avanço do treinamento, o valor segue um padrão linear de decaimento até o valor final de 0,1. Tal é necessário para que o agente possa explorar mais o ambiente no início do treinamento e reforçar o seu conhecimento no final.

Após a etapa de treinamento do agente, para classificar o tráfego de teste, foram implementadas duas funções de escolha de ações, uma considerando o conhecimento do agente como estático, sem alterações na Q-*Table* após o treinamento. A segunda implementação atualiza a Q-*Table*, porém com um coeficiente de aprendizado mais baixo que o da etapa de treinamento. A comparação entre essas implementações será apresentada no Capítulo 5.

5 AVALIAÇÃO EXPERIMENTAL

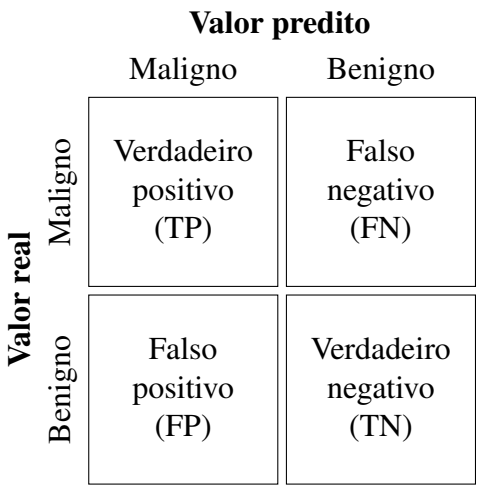
Neste capítulo são apresentadas as métricas utilizados para avaliar e contrastar os agentes. Em seguida, são detalhados e debatidos os resultados obtidos em cada cenário de teste.

5.1 Métricas de Avaliação

Devido ao uso do agente para classificação de tráfego em pacotes malignos e benignos, as métricas escolhidas utilizam como base a matriz de confusão apresentada na Figura 5.1. A matriz é dividida em quatro grupos nos quais as classificações realizadas pelo agente podem ser agrupadas.

- *Verdadeiro Positivo:* instância maligna classificada pelo agente como maligna
- *Falso Positivo:* instância benigna classificada pelo agente como maligna
- *Verdadeiro Negativo:* instância benigna classificada pelo agente como benigna
- *Falso Negativo:* instância maligna classificada pelo agente como benigna

Figura 5.1: Matriz de confusão



Fonte: Elaborada pelo autor

A métrica utilizada para indicar o desempenho geral do modelo, denominada acurácia, mede a taxa de escolhas corretas feitas pelo agente sobre o total de escolhas feitas. A Equação (5.1) apresenta a fórmula utilizada para calcular essa métrica.

$$\mathrm{Ac} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN}} \tag{5.1}$$

Devido ao desbalanceamento entre classes no *dataset*, apenas a acurácia pode não indicar um modelo com um bom desempenho. Um modelo que indique todos os dados como malignos pode ter uma acurácia alta, porém está considerando todo o tráfego benigno como ataque. Para avaliar o modelo com mais precisão, foram utilizadas, também, as métricas de precisão, revocação e F1-Score.

A precisão do modelo indica a taxa de acertos que o modelo teve quando previu algo como positivo, ou seja, dentre as instâncias previstas como positivas, quantas delas realmente eram positivas. A Equação (5.2) indica o cálculo utilizado para obter essa métrica.

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{5.2}$$

A métrica de revocação indica a taxa de instâncias previstas como positivas corretamente pelo modelo dentre todas as instâncias positivas existentes no conjunto de dados utilizado. O cálculo dessa métrica é apresentado na Equação (5.3).

$$\mathrm{Rc} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5.3}$$

A métrica de precisão é utilizada em contextos onde a existência de falsos positivos é considerada mais prejudicial que a existência de falsos negativos; no contexto deste trabalho, indicando a existência de tráfego benigno sendo considerado maligno. Já a métrica de revocação é utilizada quando a existência de falsos negativos é mais prejudicial que a de falsos positivos; no contexto deste trabalho, indicando a existência de tráfego maligno sendo considerado tráfego normal da rede.

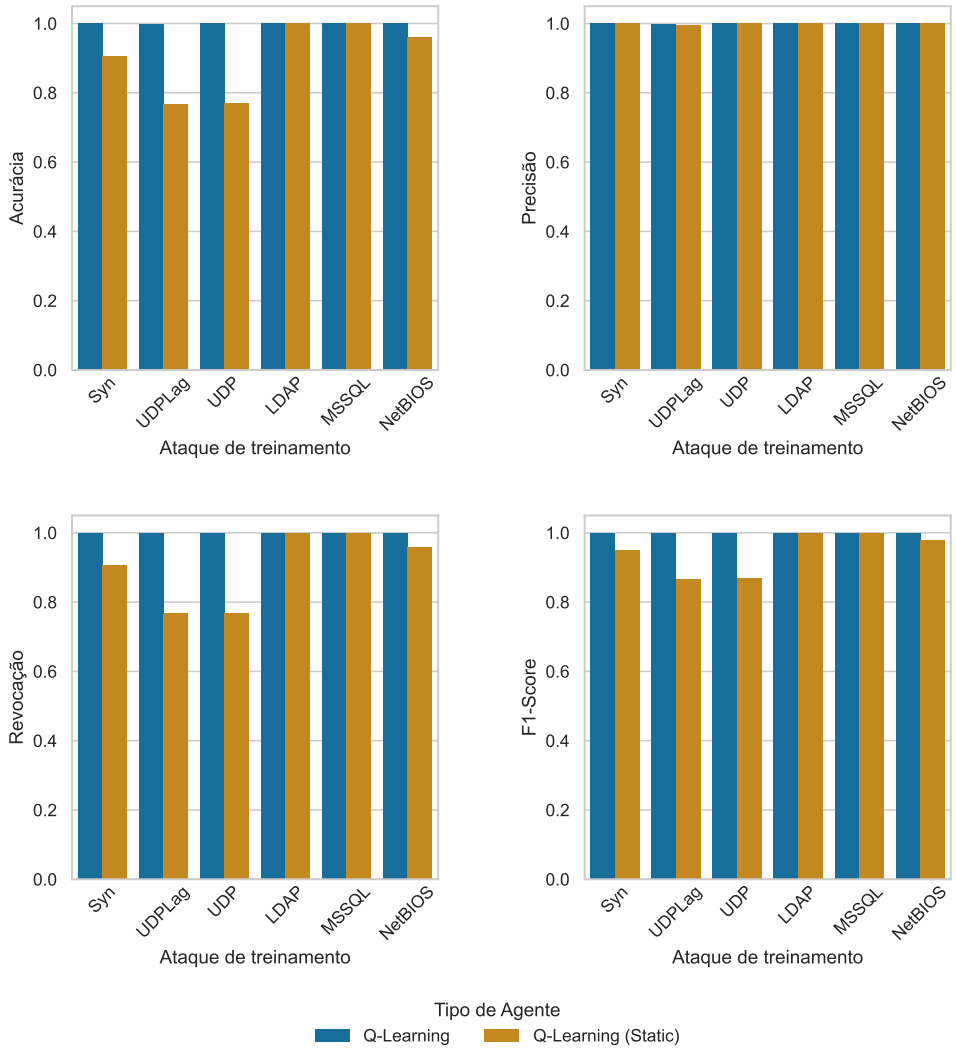
É necessário um equilíbrio entre as duas métricas, uma vez que não é desejável bloquear o tráfego benigno nem ignorar o tráfego maligno na rede. Para isto, foi proposto o F1-Score, uma média harmônica entre precisão e revocação. Quando essa métrica está baixa, tal indica que precisão, revocação ou ambas estão baixas. A Equação (5.4) apresenta o cálculo utilizado para obter o F1-Score.

$$\mathrm{F1} = 2 \times \frac{\mathrm{Pr} \times \mathrm{Rc}}{\mathrm{Pr} + \mathrm{Rc}} \tag{5.4}$$

5.2 Detecção de um Ataque

O primeiro cenário de testes foi a construção de um sistema de detecção especializado em apenas um tipo de ataque. Para esse cenário, o componente de extração de informações obtém as tuplas com as informações relevantes para detecção de cada tipo de ataque, conforme indicado por Sharafaldin et al. (2019). O agente foi treinado e avaliado apenas para ataques que existem tanto no conjunto de treinamento quanto no teste. Treinou-se um agente de cada tipo para enfrentar cada ataque, e os resultados de sua eficácia estão detalhados na Figura 5.2.

Figura 5.2: Resultados obtidos no cenário de detecção do mesmo ataque de treinamento



Fonte: Elaborada pelo autor

Como é possível perceber na figura, o agente que continua atualizando o seu modelo de decisão durante a fase de avaliação se sobressai ao agente que considera o conhecimento adquirido na fase de treinamento como estático. Esse comportamento se apresenta

bastante na acurácia e revocação. A métrica F1-Score é afetada pela revocação e tem seu valor reduzido.

A partir desses resultados, é possível concluir que o agente com modelo estático apresenta um desempenho bastante inferior ao agente que continua aprendendo, tendo uma acurácia média de 89,9% enquanto o agente dinâmico possui acurácia média de 99,9%.

5.3 Detecção de Ataque Diferente

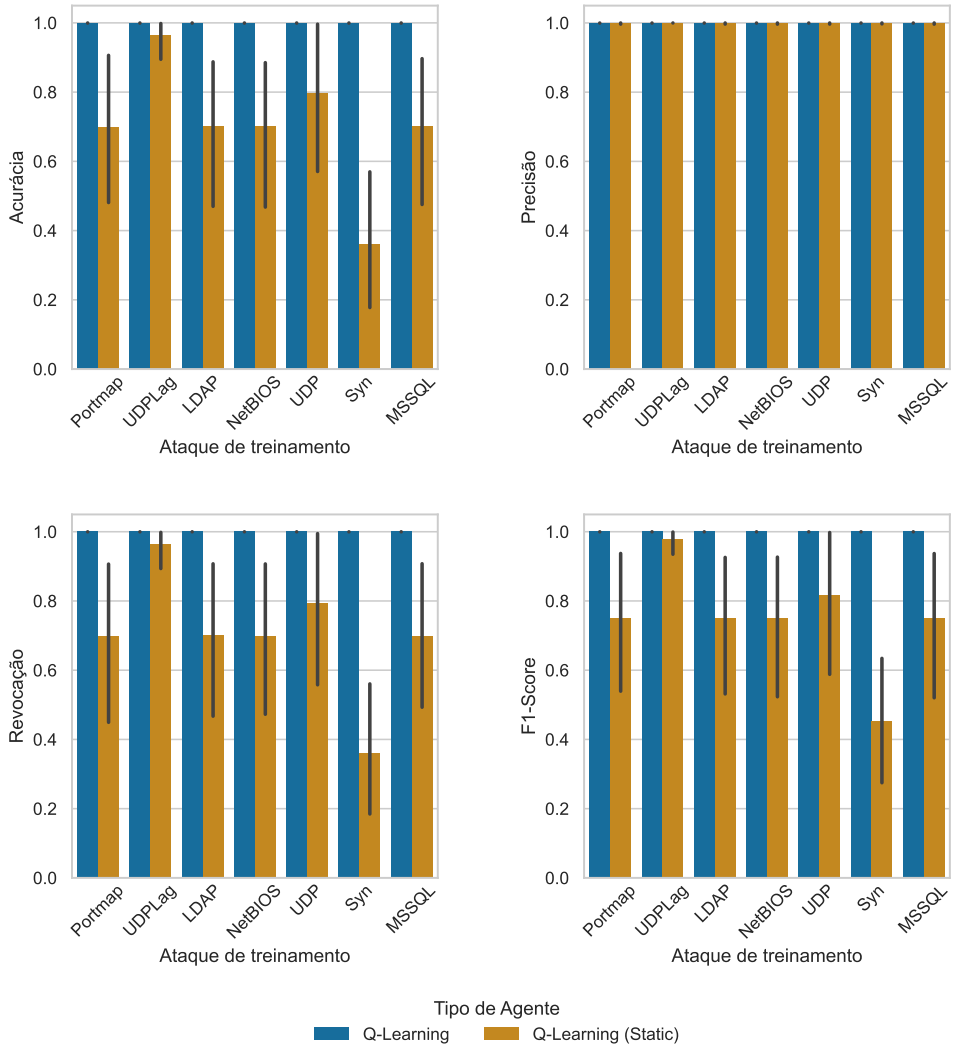
O segundo cenário de teste construído foi o treinamento de um agente utilizando os dados de cada um dos ataques presentes no conjunto de treinamento e a avaliação do seu desempenho nos dados dos ataques presentes no conjunto de teste. A fim de assegurar uma comparação justa entre os agentes em relação à detecção de ataques diferentes dos utilizados no treinamento, o módulo de extração de informações busca os dados relevantes para detecção de tráfego benigno reportadas por Sharafaldin et al. (2019). A Figura 5.3 apresenta os resultados dessas avaliações.

Neste experimento, as métricas foram computadas para cada combinação de ataque de treinamento e ataque de teste. As barras dos gráficos indicam as médias de cada métrica, e as linhas pretas indicam o intervalo que comporta 95% dos valores obtidos.

Analisando os resultados obtidos, é possível perceber que o agente dinâmico atinge um desempenho bastante consistente, com resultados próximos de 1 em todas as métricas analisadas, evidenciando a habilidade do agente em aprender a detectar ataques anteriormente desconhecidos por ele e classificar o tráfego da rede de maneira correta. Em contraste a esses resultados, o agente estático não atinge um desempenho satisfatório na maioria dos ataques, tendo sua fraqueza mais demonstrada quando o agente é treinado com o ataque `SYN`, onde acurácia, revocação e F1-Score estão abaixo de 0,5, indicando que grande parte do tráfego na rede está sendo classificado de maneira incorreta.

É possível perceber, ainda, uma grande variação entre resultados obtidos pelo agente estático, demonstrada pelo tamanho dos intervalos de valores apresentados, em comparação com os intervalos apresentados pelo agente dinâmico. As Tabelas 5.1 e 5.2 ilustram este comportamento, apresentando os resultados obtidos com agentes treinados para detectar o ataque `LDAP`. Os resultados obtidos pelo agente dinâmico apresentam um intervalo de valores bastante pequeno, com a maior variação sendo na acurácia, com a diferença de aproximadamente 0,15% entre o melhor e o pior resultado. Em contrapartida,

Figura 5.3: Resultados obtidos no cenário de detecção de ataque diferente do ataque de treinamento



Fonte: Elaborada pelo autor

dentre os resultados apresentados pelo agente estático, alguns ataques são detectados de maneira satisfatória, apresentando acurácia acima de 95%, enquanto para outros, a acurácia é próxima de 50%, aproximando uma escolha aleatória, tendo ainda resultados abaixo de 10%, onde quase nenhum tráfego foi classificado de maneira correta. Fenômeno semelhante é observado nas medições associadas à revocação, onde, para testes com agentes treinados com o ataque SYN, é próxima de 0, indicando que quase nenhum tráfego maligno foi detectado corretamente.

O comportamento dos agentes reforça a necessidade de adaptabilidade do modelo de detecção para novos ataques, visto que eles fazem uso de técnicas diferentes, explorando vulnerabilidades diferentes dos sistemas alvo.

Tabela 5.1: Métricas do agente dinâmico treinado para LDAP

Ataque Teste	Acurácia	Precisão	Revocação	F1-Score
DNS	0.999794	0.999828	0.999966	0.999897
MSSQL	0.999839	0.999865	0.999974	0.999920
NetBIOS	0.999828	0.999862	0.999966	0.999914
NTP	0.998362	0.998998	0.999344	0.999171
SNMP	0.999825	0.999859	0.999966	0.999913
SSDP	0.999863	0.999887	0.999976	0.999932
UDP	0.999727	0.999770	0.999957	0.999863
SYN	0.999885	0.999900	0.999985	0.999942
TFTP	0.999716	0.999782	0.999934	0.999858
UDPLag	0.998780	0.999194	0.999575	0.999384

Fonte: Elaborada pelo autor

Tabela 5.2: Métricas do agente estático treinado para LDAP

Ataque Teste	Acurácia	Precisão	Revocação	F1-Score
DNS	0.998300	0.999958	0.998341	0.999149
MSSQL	0.999867	0.999973	0.999894	0.999934
NetBIOS	0.999562	0.999989	0.999572	0.999781
NTP	0.626703	0.999553	0.622522	0.767220
SNMP	0.999744	0.999967	0.999778	0.999872
SSDP	0.509012	0.999954	0.508891	0.674513
UDP	0.497504	0.999933	0.497192	0.664151
SYN	0.000717	0.988173	0.000475	0.000950
TFTP	0.992125	0.999966	0.992148	0.996042
UDPLag	0.084856	0.998347	0.075740	0.140799

Fonte: Elaborada pelo autor

5.4 Detecção de Múltiplos Ataques

O último cenário analisado foi a construção de um conjunto de dados de treinamento com todos os dados dos ataques de treinamento e um conjunto de dados único para todos os dados de teste. Nesse cenário também foram utilizadas as informações relevantes para a detecção de tráfego benigno. A Tabela 5.3 apresenta os resultados obtidos.

Nesse cenário, ambos os agentes apresentaram resultados bastante satisfatórios. O agente estático teve resultados levemente inferiores ao dinâmico, porém ainda acima de 0,99. Outra característica detectada é que os agentes treinados em uma variedade de ataques foram mais capazes de detectar ataques nunca vistos antes, mesmo sem aprender nenhuma informação nova, no caso do agente estático.

Tabela 5.3: Resultados da detecção de múltiplos ataques

Métrica	**Q-Learning**	**Q-Learning (Static)**
Acurácia	0.999754	0.990524
Precisão	0.999811	0.999959
Revocação	0.999942	0.990554
F1-Score	0.999877	0.995234

Fonte: Elaborada pelo autor

5.5 Avaliação dos Resultados

Conforme analisado no decorrer do capítulo, o agente dinâmico apresentou um comportamento mais satisfatório que o agente estático. Isto se demonstra nos resultados apresentados nas Seções 5.3 e 5.4 e, conforme já comentado, para um funcionamento satisfatório é necessária a capacidade de adaptabilidade que o aprendizado do algoritmo Q-*Learning* provê.

Outro comportamento detectado nos agentes é demonstrado pela métrica de precisão. Em todos os testes, a precisão computada é próxima de 1, não auxiliando na análise comparativa entre os agentes. Além disso, a métrica indica que, independente do agente e do ataque utilizados para treinamento, o agente consegue detectar grande parte do tráfego maligno de maneira correta. Entretanto, devido ao grande desbalanceamento de dados, essa métrica não pode ser considerada com grande peso na avaliação.

Embora os resultados obtidos com a aplicação da técnica de Q-Learning tabular sejam promissores, sua aplicação em problemas com um grande número de variáveis de estado, além de uma grande quantidade de valores possíveis para cada variável, pode ser desafiadora, levando a uma dificuldade de generalização de aprendizado. Essa dificuldade de generalização levanta dúvidas sobre os resultados obtidos pelos agentes na detecção de tráfego nunca visto antes. Como trabalho futuro, faz-se necessária uma verificação mais profunda para determinar se a forma como os agentes são inicializados os induz a um comportamento que distorce a realidade do aprendizado.

6 CONCLUSÃO

Neste trabalho foi realizado o projeto, o desenvolvimento e a avaliação de um modelo de sistema para detecção de ataques DDoS em redes de comunicação. Para atingir esse objetivo, o trabalho foi dividido em várias etapas: análise das dificuldades existentes; busca e análise exploratória de dados para treinamento e avaliação; proposta de um modelo de sistema para classificação de tráfego; implementação deste modelo; análise e comparação das implementações.

Os resultados obtidos sugerem que a arquitetura proposta é viável para detecção e mitigação de ataques sofridos por um serviço de comunicação. O modelo proposto é bastante simples e pode ser implementado de maneira ”modularizável”, permitindo que partes do sistema possam ser substituídas sem a necessidade de grandes mudanças em outros componentes. Apesar dessa simplicidade, o modelo proposto atinge resultados satisfatórios, detectando e classificando o tráfego maligno na rede com bastante exatidão.

Durante a análise dos resultados das avaliações experimentais realizadas, pode-se validar a hipótese de que a adaptabilidade de agentes autônomos utilizando algoritmos de aprendizado por reforço seria benéfica para esse cenário de detecção. Esta característica se mostrou bastante necessária quando um agente é treinado para detecção de um ataque, porém o ataque sofrido não consta no conjunto de treinamento, algo bastante comum no cenário de constante evolução de ataques DDoS.

Embora os resultados obtidos com a aplicação da técnica de Q-Learning tabular sejam promissores, sua aplicação em problemas com um grande número de variáveis de estado pode ser desafiadora. A necessidade de armazenar a função de valor em uma tabela torna a técnica inviável para espaços de estados extensos, devido ao consumo excessivo de memória e à dificuldade de aprendizado eficiente. Essa limitação se torna ainda mais crítica quando as variáveis em questão possuem grandes intervalos possíveis, o que leva a um crescimento exponencial do tamanho da tabela de valores. Ambos os problemas sugerem ser necessária uma maior análise da viabilidade computacional de implementação da solução.

Além da análise de viabilidade computacional, é necessária uma avaliação do impacto da solução na vazão e latência da rede de comunicação em que se planeja implementar o modelo. Essa análise se faz necessária para determinar se o custo de uso da solução é aceitável, considerando-se os benefícios providos por ela. Além disso, é necessário examinar a maneira como a solução pode ser implementada em uma rede de comunicação e

possíveis falhas na própria solução, uma vez que ataques de DDoS visam esgotar recursos da infraestrutura do serviço e, uma vez instalada, a solução proposta torna-se parte dessa infraestrutura.

Possíveis evoluções deste trabalho, além das análises já citadas, incluem: implementação de interfaces de comunicação fixas entre os componentes do sistema, permitindo uma maior modularização; avaliação da sensibilidade do modelo implementado a variações no coeficiente de aprendizado e intervalo de valores utilizados na estratégia ϵ-*greedy*; implementações de outros algoritmos de aprendizado por reforço; integração do modelo com sistemas de detecção de ataques já existentes como avaliadores de decisões; emprego de um conjunto de agentes, com diferentes modelos e algoritmos, para detectar ataques; e expansão do modelo de sistema para permitir a escolha entre mais ações que apenas encaminhamento e descarte de pacotes.

REFERÊNCIAS

Akamai Technologies. **A Year in Review — A Look at 2023's Cyber Trends and What's to Come**. [S.l.], 2023. Disponível em: <https://www.akamai.com/our-thinking/the-state-of-the-internet>.

Canadian Institute for Cybersecurity. **CICFlowMeter**. 2020. Disponível em: <https://github.com/CanadianInstituteForCybersecurity/CICFlowMeter>.

Cloudflare. **What is a distributed denial-of-service (DDoS) attack?** 2023. Disponível em: <https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/>.

DOULIGERIS, C.; MITROKOTSA, A. DDoS attacks and defense mechanisms: classification and state-of-the-art. **Computer Networks**, Elsevier, v. 44, n. 5, p. 643–666, 4 2004. ISSN 1389-1286.

ISYAKU, B. et al. Performance Comparison of Machine Learning Classifiers for DDOS Detection and Mitigation on Software Defined Networks. **2023 IEEE International Conference on Automatic Control and Intelligent Systems, I2CACIS 2023 - Proceedings**, Institute of Electrical and Electronics Engineers Inc., p. 69–74, 2023.

KINER, E.; APRIL, T. **Google Cloud mitigated largest DDoS attack, peaking above 398 million rps**. 2023. Disponível em: <https://cloud.google.com/blog/products/identity-security/google-cloud-mitigated-largest-ddos-attack-peaking-above-398-million-rps>.

MENSCHER, D. **Exponential growth in DDoS attack volumes**. 2020. Disponível em: <https://cloud.google.com/blog/products/identity-security/identifying-and-protecting-against-the-largest-ddos-attacks>.

NETSCOUT. **DDoS Threat Intelligence Report**. 2023. Disponível em: <https://www.netscout.com/threatreport>.

POSTEL, J. **User Datagram Protocol**. [S.l.], 1980. Disponível em: <https://www.rfc-editor.org/info/rfc768>.

POSTEL, J. **Transmission Control Protocol**. [S.l.], 1981. Disponível em: <https://www.rfc-editor.org/info/rfc793>.

ROSSUM, G. V.; DRAKE, F. L. **Python 3 Reference Manual**. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697.

RUSSEL, S. J.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 4. ed. [S.l.]: Pearson, 2021. ISBN 9781292401133.

SHARAFALDIN, I. et al. Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. In: **2019 International Carnahan Conference on Security Technology (ICCST)**. [S.l.]: IEEE, 2019. p. 1–8. ISBN 978-1-7281-1576-4.

SUTTON, R. S.; BARTO, A. G. **Reinforcement Learning: An Introduction**. 2nd edition. ed. Cambridge: A Bradford Book, 2018. ISBN 978-0262039249.

TOWERS, M. et al. **Gymnasium**. Zenodo, 2023. Disponível em: <https://doi.org/10.5281/zenodo.8269265>.

WATKINS, C. J. C. H. **Learning from Delayed Rewards**. Tese (Doutorado) — King's College, Oxford, 1989.

XIA, W. et al. A Survey on Software-Defined Networking. **IEEE Communications Surveys and Tutorials**, Institute of Electrical and Electronics Engineers Inc., v. 17, n. 1, p. 27–51, 1 2015. ISSN 1553877X.

ZARGAR, S. T.; JOSHI, J.; TIPPER, D. A survey of defense mechanisms against distributed denial of service (DDOS) flooding attacks. **IEEE Communications Surveys and Tutorials**, v. 15, n. 4, p. 2046–2069, 2013. ISSN 1553877X.