



**UNIVERSIDADE
KIMPA VITA**

15 - 10 | 2026

Detecção de Ataques DDoS contra Controladores de Redes Definidas por Software (SDN) utilizando Machine Learning

Detection of DDoS Attacks against Software-Defined Network (SDN) Controllers using Machine Learning

Adão António Bagi | Amoussou Dorothée

Versão electrónica

URL: <https://ciencia.unikivi.ao/>

Data de publicação: 15-10-2026. Páginas: 12

Editor

Revista Científica Interdisciplinar da UNIKIVI

Referência electrónica

Bagi, A. A., & Dorothée, A. (2026). Detecção de Ataques DDoS contra Controladores de Redes Definidas por Software (SDN) utilizando Machine Learning. Revista da UNIKIVI. 02(01), 01-12.



DETECÇÃO DE ATAQUES DDoS CONTRA CONTROLADORES DE REDES DEFINIDAS POR SOFTWARE (SDN) UTILIZANDO MACHINE LEARNING

Detection of DDoS Attacks against Software-Defined Network (SDN) Controllers Using Machine Learning

Adão António Bagi

Universidade Kimpa Vita, Angola

adsubagi@gmail.com | <https://orcid.org/0009-0007-8353-4591>

Amoussou Dorothée

Faculdade de Economia da Universidade Kimpa Vita, Angola

Amousdorothe@yahoo.com.br | <https://orcid.org/0009-0007-8732-097X>

RESUMO

As Redes Definidas por Software (SDN), trouxeram uma nova forma de gerenciar redes de comunicação porque separa o plano de controlo do plano de dados, o que oferece mais flexibilidade e permite que as redes fossem gerenciadas de forma centralizada. No entanto, essa centralização fez com que o controlador da rede se tornasse um ponto vulnerável, especialmente quando se trata de ataques de negação de serviço (DDoS), que afecta a disponibilidade e o desempenho da comunicação. Sendo assim, é indispensável encontrar maneiras mais eficazes de detectar e mitigar esses ataques, pois que as técnicas comuns de detecção de intrusão têm limitações, quando se trata de identificar padrões complexos de tráfego malicioso. Portanto, este estudo teve como objectivo criar e avaliar um modelo de detecção de ataques DDoS ao controlador da rede SDN, utilizando algoritmos de Machine Learning. Para isso foi criado, um ambiente de simulação de uma rede SDN com hosts, switch e controlador Ryu virtual, emulados pelo Mininet, o que permitiu colectar o tráfego normal da rede através das comunicações entre os *hosts* baseada no protocolo TCP e UDP e o tráfego malicioso foi obtido, por meio da utilização da ferramenta *hping3* para reproduzir dados com padrão característico de um ataque DDoS e consequentemente a criação do *dataset* usado no treinamento dos modelos. O conjunto de testes feito no treinamento dos modelos de *Machine Learning* foi composto por 10 000 fluxos de rede, onde 60% da amostra representa o tráfego normal e 40% tráfego malicioso. Os resultados dos testes são satisfatórios nas métricas acurácias (98,63), precisão (98,66%) *recall* (97,92%), e (97,9%) e F1-score (98,29%) evidenciando a eficácia da abordagem proposta de tráfego malicioso, na mitigação de ataques distribuídos e na utilização de algoritmos de *Machine Learning* como precisos em identificar padrões associados a ataques DDoS.

Palavras-chave: SDN; Segurança de Redes; Controlador SDN; DDoS; Machine Learning.

ABSTRACT

Software-Defined Networking (SDN) has introduced a new paradigm for communication network management by decoupling the control plane from the data plane, providing greater flexibility, programmability, and centralized network control. However, this centralized architecture also introduces new security challenges, making the SDN controller a potential single point of failure, particularly under Distributed Denial-of-Service (DDoS) attacks, which can significantly impact network availability and performance. Therefore, developing efficient mechanisms for detecting and mitigating such attacks is essential, as traditional intrusion detection techniques often present limitations when identifying complex patterns of malicious network behavior. This study aims to develop and evaluate a Machine Learning-based approach for detecting DDoS attacks targeting SDN controllers. To achieve this objective, an experimental SDN environment was implemented using Mininet, comprising virtual hosts, an OpenFlow switch, and a Ryu-based SDN controller. Normal network traffic was generated through TCP-based communications among virtual hosts, while malicious traffic was simulated using the hping3 tool to reproduce traffic patterns associated with DDoS attacks. The collected network flows were processed and labeled to construct the dataset used for training and evaluating the Machine Learning models. The experimental dataset consisted of 10,000 network flows, including 60% of normal traffic samples and 40% of malicious traffic samples. Two classification algorithms, Random Forest and Decision Tree, were evaluated. The results demonstrated that the Random Forest model achieved superior performance, obtaining 98.7% precision, 97.9% recall, and an F1-score of 98.3%. These findings indicate that the proposed approach is effective in identifying DDoS-related traffic patterns and highlights the potential of Machine Learning algorithms as a reliable solution for improving the security and resilience of SDN environments against distributed attacks.

Keywords: *Software-Defined Networking; Network Security; SDN Controller; Distributed Denial-of-Service; Machine Learning.*

1 INTRODUÇÃO

O crescimento da Internet, da computação em nuvem e das infraestruturas digitais tem aumentado a complexidade das redes, exigindo soluções mais flexíveis, escaláveis e programáveis. Neste contexto, as Redes Definidas por Software (Software-Defined Networking — SDN) constituem um paradigma de gestão de redes baseado, na separação entre o plano de controlo e o plano de dados. Esta arquitectura permite centralizar as decisões de encaminhamento num controlador lógico, proporcionando maior visibilidade, simplificação da administração e implementação dinâmica de políticas de rede (Karthika & Arockiasamy, 2023).

Apesar destas vantagens, a centralização cria novos desafios de segurança, uma vez que o controlador assume uma função crítica na gestão e configuração dos dispositivos de rede. A sua sobrecarga ou comprometimento pode, por conseguinte, afectar o funcionamento de uma parte significativa da infra-estrutura (Ali et al., 2023). Entre as principais ameaças destacam-se, os ataques de negação de serviço distribuído (Distributed Denial of Service — DDoS), que recorrem a múltiplos dispositivos comprometidos para gerar tráfego ou solicitações simultâneas, com o propósito de esgotar os recursos do alvo (Mohsin & Hamad, 2022).

Em ambientes SDN, os ataques DDoS podem explorar especificamente a centralização do plano de controlo, gerando um elevado número de solicitações de novos fluxos e aumentando a carga sobre o controlador. Entre os mecanismos possíveis encontram-se a inundação de pacotes, o SYN flood e a saturação das tabelas de fluxos dos switches. Estas situações podem resultar em degradação do desempenho, aumento da latência e, em casos mais graves, indisponibilidade de serviços ou componentes da rede (Ali et al., 2023).

A detecção atempada destes ataques é, portanto, essencial para preservar a segurança e a disponibilidade das redes SDN. As abordagens tradicionais baseadas em assinaturas e regras

estáticas, presentes em ferramentas como *Snort* e *Suricata*, dependem principalmente de padrões previamente conhecidos e de regras configuradas manualmente. Embora sejam úteis na identificação de ameaças conhecidas, podem apresentar limitações perante ataques novos, variantes desconhecidas ou padrões de tráfego não previamente definidos. Esta limitação tem incentivado, a utilização de técnicas de aprendizagem automática (*Machine Learning* — ML) como complemento aos mecanismos convencionais (Mansoor et al., 2023).

A aprendizagem automática permite analisar características do tráfego de rede e reconhecer padrões associados a comportamentos potencialmente anómalos. Os algoritmos de classificação podem, com base nos dados utilizados no processo de treino, distinguir entre tráfego legítimo e tráfego potencialmente malicioso. Entre as técnicas estudadas neste domínio destacam-se o *Support Vector Machine* (SVM), *Decision Tree* e *Random Forest*, que apresentam resultados promissores na detecção de ataques DDoS em ambientes SDN (Altamemi et al., 2022).

Neste contexto, o presente artigo tem como objectivo investigar a aplicação de técnicas de aprendizagem automática na detecção de ataques DDoS dirigidos ao controlador de redes definidas por *Software*. Para esse efeito, é desenvolvida uma arquitectura experimental baseada em ML, na qual são avaliados diferentes algoritmos de classificação quanto à sua capacidade de identificar tráfego malicioso. O estudo pretende, assim, contribuir para a análise de mecanismos inteligentes de detecção aplicáveis ao reforço da segurança e da disponibilidade das infraestruturas SDN.

2 ENQUADRAMENTO TEÓRICO

2.1 Redes Definidas por Software (SDN)

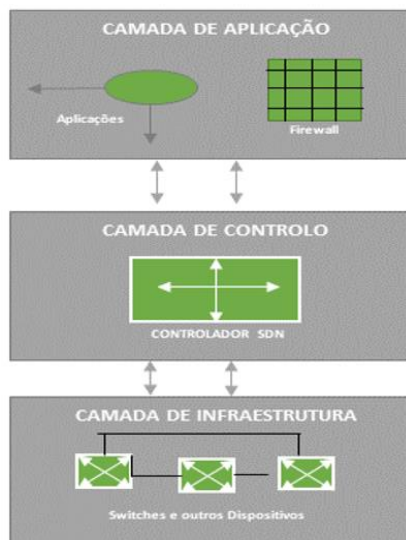
As Redes Definidas por *Software* (*Software-Defined Networking* — SDN) constituem um paradigma de arquitectura e gestão de redes baseado na separação entre o plano de controlo e o plano de dados. Enquanto este último assegura o encaminhamento dos pacotes, o plano de controlo concentra as decisões de gestão num controlador SDN, permitindo maior flexibilidade e programabilidade da rede (Karthika & Arockiasamy, 2023).

A arquitectura SDN organiza-se, geralmente, em três camadas: infra-estrutura, controlo e aplicação. A camada de infra-estrutura integra os *switches* e demais dispositivos de encaminhamento; a camada de controlo contém o controlador, responsável pela gestão da rede; e a camada de aplicação reúne os serviços e políticas de rede. A comunicação entre estas camadas permite configurar e adaptar dinamicamente, o funcionamento da infra-estrutura através de interfaces de programação (Ali et al., 2023).

Apesar das vantagens, a centralização do controlo torna o controlador um componente crítico, cuja sobrecarga ou comprometimento pode afectar o funcionamento da rede.

Figura 1

Arquitectura SDN



Essa arquitetura proporciona diversas vantagens, incluindo maior flexibilidade, automação, gestão centralizada e rápida adaptação a mudanças na rede. No entanto, a centralização do controle também cria um ponto único de falha, tornando o controlador um alvo crítico para ataques cibernéticos (Mohsin & Hamad, 2022).

2.2 Ataques DDoS em Redes SDN

2.2.1 Conceito e características dos ataques DDoS

Os ataques de negação de serviço distribuída (Distributed Denial of Service — DDoS) constituem uma ameaça relevante às infra-estruturas de rede. Caracterizam-se pela utilização coordenada de múltiplos dispositivos comprometidos para gerar elevados volumes de tráfego ou solicitações dirigidas a um determinado alvo, procurando esgotar os seus recursos e impedir o acesso de utilizadores legítimos (Altamemi et al., 2022).

2.2.2 Ataques DDoS no contexto das redes SDN

Nas Redes Definidas por Software (SDN), a arquitectura centralizada do plano de controlo introduz vulnerabilidades específicas. O controlador coordena decisões de encaminhamento e comunica com os dispositivos da rede, podendo ser sobrecarregado por um elevado número de solicitações de novos fluxos. Esta situação pode afectar o canal de controlo, os recursos computacionais do controlador e, conseqüentemente, o desempenho global da rede (Ali et al., 2023).

2.2.3 Principais tipos de ataques DDoS em SDN

Os ataques DDoS podem explorar diferentes mecanismos da arquitectura SDN. Entre os mais relevantes encontram-se:

- a) SYN Flood. Explora o estabelecimento de ligações TCP através do envio massivo de segmentos SYN sem conclusão do *three-way handshake*. Em SDN, a chegada de fluxos sem regras correspondentes pode gerar numerosas mensagens *packet-in* para o controlador, aumentando a sua carga de processamento (Kreutz et al., 2015; Scott-Hayward et al., 2016).
- b) Saturação da tabela de fluxo. Os ataques *Flow Table Overflow* procuram preencher a capacidade limitada das tabelas de fluxo dos *switches*, através da criação de múltiplos fluxos distintos. A ausência de correspondência nas tabelas pode originar novas solicitações ao controlador, contribuindo para a degradação do desempenho da rede (Shin & Gu, 2013; Kreutz et al., 2015).
- c) Inundação de pacotes. Consiste no envio massivo de pacotes para um ou vários destinos, com o objectivo de consumir recursos de processamento ou transmissão. Em SDN, pacotes sem correspondência nas tabelas de fluxo podem aumentar o número de mensagens dirigidas ao controlador, sobrecarregando o canal de controlo (Mirkovic & Reiher, 2004; Kreutz et al., 2015).
- d) Amplificação. Explora serviços ou protocolos capazes de produzir respostas significativamente maiores do que as solicitações recebidas. DNS, NTP e SSDP constituem exemplos de protocolos susceptíveis de exploração. O tráfego amplificado pode provocar congestionamento dos enlaces e sobrecarga dos dispositivos de rede (Rossow, 2014; Douligeris & Mitrokotsa, 2004).

2.2.4 Impactos e implicações para a segurança

Independentemente do mecanismo utilizado, os ataques DDoS podem provocar aumento da latência, perda de pacotes, degradação do desempenho, saturação dos recursos e indisponibilidade de serviços (Mansoor et al., 2023). Em ambientes SDN, estes efeitos podem ser agravados quando o ataque afecta simultaneamente os dispositivos de encaminhamento e o controlador.

A diversidade dos mecanismos de ataque evidencia a necessidade de soluções capazes de analisar diferentes características do tráfego e identificar padrões anómalos. Neste sentido, técnicas de aprendizagem automática (*Machine Learning* - ML) constituem uma abordagem

promissora para complementar os mecanismos tradicionais de detecção, permitindo explorar características do tráfego e apoiar a identificação de comportamentos potencialmente maliciosos.

3 METODOLOGIA

A metodologia adoptada neste estudo foi estruturada em cinco etapas principais: construção do ambiente experimental, geração e recolha do tráfego de rede, extracção das características, treino dos modelos de *Machine Learning* e avaliação do seu desempenho. Este procedimento permitiu desenvolver, de forma sistemática, o cenário experimental e analisar a capacidade dos modelos na detecção de tráfego associado a ataques DDoS em redes SDN.

3.1 Ambiente Experimental

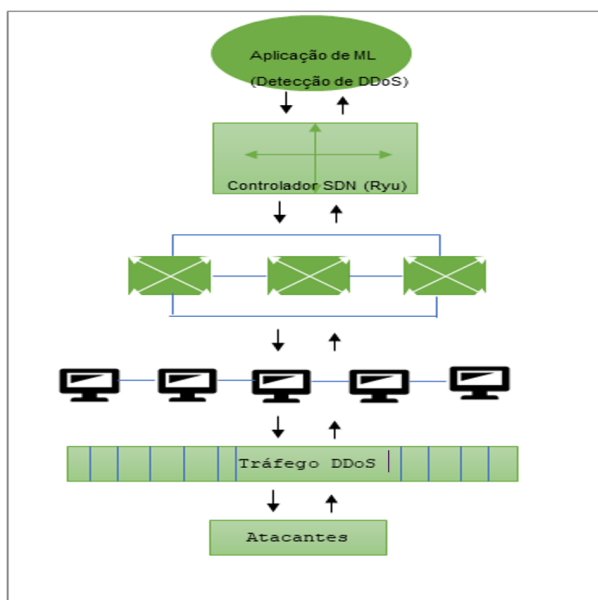
O ambiente experimental foi implementado utilizando o emulador de redes *Mininet*, versão 2.3.0, para a criação e simulação da rede definida no estudo. O plano de controlo foi gerido pelo controlador Ryu, versão 4.34, responsável pela administração e programação dos dispositivos de encaminhamento da rede SDN. A topologia experimental foi constituída por um controlador SDN, três *switches OpenFlow* e cinco *hosts*, permitindo reproduzir um cenário controlado para a análise do tráfego de rede e a avaliação do mecanismo de detecção de ataques DDoS.

A infra-estrutura computacional utilizada para a execução das simulações apresentava 8 GB de memória RAM, processador Intel Core i5-6300U, a 2,40 GHz, placa gráfica Intel HD Graphics 520 com 128 MB e sistema operativo Windows 10.

Conforme ilustrado na figura esquemática do ambiente experimental, a arquitectura encontra-se organizada em três planos. No plano de infraestrutura, estão representados os switches OpenFlow, responsáveis pelo encaminhamento dos pacotes. No plano de controlo, encontra-se o controlador Ryu, responsável pela gestão e programação da rede. Por sua vez, no plano de aplicação, está integrado o módulo de detecção baseado em *Machine Learning*, responsável por analisar o tráfego durante os ataques DDoS e identificar padrões potencialmente anómalos.

Figura 2

Esquema estrutural do ambiente experimental



Fonte: Autor

O tráfego normal foi gerado através da comunicação entre os *hosts* utilizando protocolos TCP e UDP, enquanto o tráfego malicioso foi produzido utilizando *hping3* para reproduzir padrões característicos de ataques DDoS. Durante os experimentos foram coletadas características dos fluxos, incluindo número de pacotes, taxa de fluxo, duração das conexão e quantidade de solicitações encaminhadas ao controlador. Após a captura e separação dos dados das comunicações, os fluxos foram rotulados como tráfego normal e DDoS para compor o *dataset*,

que dividiu este fluxo em 70% para treinamento e 30% para teste, sendo todo o conjunto de avaliação é de 10 000 fluxos, dos quais 6 000 representam tráfego normal e 4 000 representam tráfego malicioso. Os modelos foram avaliados utilizando métricas como acurácia, precisão, *recall* e F1-score.

3.2 Colecta e extracção de Características de Dados

Durante as simulações, foram recolhidas estatísticas do tráfego de rede, utilizadas para constituir um conjunto de dados composto por amostras de tráfego legítimo e malicioso. A captura e a extracção das características foram realizadas com recurso ao *Wireshark*, ferramenta que permite efectuar a análise detalhada do tráfego de rede e identificar atributos relevantes para o processo de classificação. Após a captura, os dados foram tratados, seleccionados e organizados em formato CSV, constituindo um conjunto de 10 000 fluxos de rede classificados, posteriormente utilizado no treino e na avaliação dos modelos de aprendizagem automática. Entre as principais características extraídas e analisadas encontram-se a taxa de chegada dos pacotes, o número de fluxos por segundo, o tamanho médio dos pacotes e o número de requisições efectuadas ao controlador. A selecção destas características teve como finalidade representar, padrões distintos do comportamento do tráfego e fornecer aos modelos informações relevantes para a diferenciação entre tráfego legítimo e malicioso.

Tabela 1

Características dos pacotes para a classificação

Características	Tráfego Normal	DDoS
Número médio de pacotes por fluxo	45	580
Taxa média de fluxo (pacotes/s)	18	1 150
Duração média da conexão (s)	12,8	0,9
Solicitações médias ao controlador por fluxo	3	52
Tamanho médio do fluxo (KB)	38,5	412,7
Intervalo médio entre pacotes (ms)	72	2

Fonte: Autor

A tabela 1 apresenta os valores médios das principais características extraídas dos fluxos de rede utilizados no *dataset* que posteriormente serviram como variáveis de entrada dos algoritmos de *Random Forest* e *Decision Tree*. Observa-se que o tráfego associado aos ataques DDoS apresenta, um número significativamente maior do que pacotes por fluxo, maior taxa de transmissão e maior quantidade de solicitações, enquanto o intervalo entre pacotes e a duração das conexões são substancialmente menores quando, comparados ao tráfego normal.

Tabela 2

Composição do conjunto de dados

Classe	Número de fluxos	Percentual
Tráfego normal	6 000	60%
DDoS	4 000	40%
Total	10 000	100%

Fonte: Autor

O conjunto de teste foi constituído por 10 000 fluxos de rede, dos quais 6 000 (60%) correspondem ao tráfego normal e 4 000 (40%) ao tráfego malicioso associado a ataques DDoS. Essa distribuição permitiu avaliar a capacidade dos classificadores em distinguir padrões legítimos e maliciosos.

Tabela 3

Divisão do Dataset

Conjunto	Percentual	Finalidade
Treinamento	70%	Treinamento dos modelos
Teste	30%	Avaliação dos modelos

Fonte: Autor

Conforme apresentado na tabela 3, o conjunto de dados foi dividido de forma aleatória em 70% (7 000 fluxos) para treinamento e 30% (3 000 fluxos) para teste preservando a proporção entre as classes. A utilização de conjuntos distintos para treinamento e avaliação permitiu analisar sua capacidade de generalização sobre dados não utilizados durante o processo de aprendizagem.

3.3 Algoritmos de *Machine Learning*

Foram seleccionados os algoritmos *Random Forest* e *Decision Tree*, por se adequarem à classificação de tráfego de rede e aos objectivos do estudo. O *Random Forest* foi escolhido pela capacidade de combinar múltiplas árvores de decisão, reduzir o risco de *overfitting* e processar conjuntos de dados com diversos atributos de fluxo de rede (Breiman, 2001). O *Decision Tree* permite estruturar regras de decisão, para classificar os fluxos de forma interpretável (Quinlan, 1986).

Ambos os modelos foram treinados com 7 000 fluxos do conjunto de treino. No *Random Forest*, foram construídas várias árvores a partir de subconjuntos aleatórios dos dados e dos atributos. Em cada nó, a divisão foi seleccionada segundo a maior redução da impureza, utilizando o índice de Gini. A classificação final resultou da votação maioritária entre as árvores, contribuindo para reduzir a variância e melhorar a generalização.

No *Decision Tree*, foi construída uma única árvore através de divisões sucessivas dos dados, seleccionando, em cada nó, o atributo que proporcionava maior redução da impureza. O processo terminou de acordo com os hiperparâmetros definidos, permitindo classificar os fluxos como tráfego normal ou associado a ataques DDoS.

Tabela 4

Hiperparâmetros definidos

nº Ordem	Parâmetro
1	Critério de divisão
2	Númer o de árvores
3	Profundidade máxima
4	Mínimo de amostras para divisão
5	Mínimo de amostras por folha
6	Seleccção de atributos
7	Bootstrap
8	Estado aleatório

Fonte: Autor

Os hiperparâmetros dos modelos foram ajustados utilizando técnicas de optimização fornecidas pela biblioteca Scikit-learn em que, no algoritmo *Decision Tree* foi empregue o *GridSearchCV*, enquanto para o *Random Forest* utilizou-se o *RandomizedSearchCV*, ambos com validação cruzada de cinco partições, seleccionando a configuração que apresentou o melhor desempenho em termos de F1-score.

Tabela 5

Tempo estimado de treinamento dos modelos

Algoritmo	Tempo de treinamento	Tempo médio de inferência por fluxo
Random Forest	0,86 s	0,52 ms
Decision Tree	0,18 s	0,09 ms

Fonte: Autor

Observa-se que a *Decision Tree* apresentou menor tempo de treinamento e menor tempo médio de inferência, por fluxo devido a sua estrutura composta por uma única árvore de decisão. Em contrapartida, o *Random Forest* necessitou de maior tempo de treinamento e inferência em virtude da construção e avaliação de muitas árvores.

4 RESULTADOS

Tabela 6

Matriz de confusão de conjunto de teste

Elemento	Descrição
True Positive (TP)	Ataque correctamente identificado
True Negative (TN)	Tráfego normal correctamente identificado
False Positive (FP)	Tráfego normal classificado incorrectamente como ataque
False Negative (FN)	Ataque não detectado

Fonte: Sommer & Paxson, 2010

A matriz de confusão, para o conjunto de teste de fluxos foi de 3 000 fluxos de rede que representa 30% do fluxo total do *dataset* experimental, que é 10 000 fluxos.

Tabela 7

Matriz do conjunto de teste

Classe	Quantidade no teste
DDoS	1 200 fluxos
Tráfego Normal	1 800 fluxos
Total	3 000 fluxos

Fonte: Autor

Após o treinamento, o modelo *Forest* classificou cada um dos 3 000 fluxos gerando a seguinte previsão do modelo nas respectivas classes:

Tabela 8

Classificação realizada pelo Random Forest

Classe	Previsto DDoS	Previsto Normal	Total
DDoS	1 175	25	1 200
Tráfego Normal	16	1 784	1 800
Total	1 191	1 809	3 000

Fonte: Autor

A análise dos resultados da tabela de classificação indica, que as principais características responsáveis pela diferenciação entre tráfego normal e tráfego DDoS foram: número de pacotes por fluxo, taxa de fluxo, número de solicitações ao controlador, duração da conexão. A partir destes valores foram obtidos, os indicadores das variáveis da matriz de confusão, como apresenta a tabela.

Tabela 9

Variáveis da matriz da confusão

Elemento	Random Forest	Decision Tree
TP	1 175	1 135
TN	1 784	1 743
FP	16	57
FN	25	65

Fonte: Autor

Tabela 10

Matriz de confusão do algoritmo de Random Forest

Métrica	Valor
Acurácia	98,63%
Precision	98,66%
Recall	97,92%
F1-score	98,29%

Taxa de falso positivo	0,89%
Taxa de falso Negativo	2,08%

Fonte: Autor

Tabela 11

Matriz de confusão do algoritmo de decisão Tree

Algoritmo	Accuracy	Precision	Recall	F1-score	FPR	FNR
Decision Tree	95,93%	95,20%	94,60%	94,90%	3,17%	5,42%

Fonte: Autor

4.1 Cálculo da métrica Accuracy dos algoritmos de Random e Decision

$$\begin{aligned} \text{Accuracy} &= \frac{TP+TN}{TP+TN+FN+FP} ; & \text{Accuracy} &= \frac{TP+TN}{TP+TN+FN+FP} ; \\ \text{Accuracy} &= \frac{1175+1784}{1175+1784+16+25} ; & \text{Accuracy} &= \frac{1135+1743}{1135+1743+57+65} ; \\ \text{Accuracy} &= 98,63\% ; & \text{Accuracy} &= 95,93\% ; \end{aligned}$$

4.2 Cálculo da métrica Precision dos algoritmos de Random e Decision

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP+FP} ; & \text{Precision} &= \frac{TP}{TP+FP} ; \\ \text{Precision} &= \frac{1175}{1175+16} ; & \text{Precision} &= \frac{1135}{1135+57} ; \\ \text{Precision} &= 98,66\% ; & \text{Precision} &= 95,22\% ; \end{aligned}$$

4.3 Cálculo da métrica Recall dos algoritmos de Random e Decision

$$\begin{aligned} \text{Recall} &= \frac{TP}{TP+FN} ; & \text{Recall} &= \frac{TP}{TP+FN} ; \\ \text{Recall} &= \frac{1175}{1175+25} ; & \text{Recall} &= \frac{1135}{1135+65} ; \\ \text{Recall} &= 0,9792\% ; & \text{Recall} &= 94,58 \end{aligned}$$

4.4 Cálculo da métrica F1-score dos algoritmos de Random e Decision

$$\begin{aligned} F1 &= 2 \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} ; & F1 &= 2 \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} ; \\ F1 &= 2 \frac{0,9866 \cdot 0,9792}{0,9866 + 0,9792} ; & F1 &= 2 \frac{0,9522 \cdot 0,9458}{0,9522 + 0,9458} ; \\ F1 &= 98,29\% ; & F1 &= 94,89\% ; \end{aligned}$$

4.5 Cálculo da métrica Taxa de falso positivo dos algoritmos de Random e Decision

$$\begin{aligned} \text{FPR} &= \frac{FP}{FP+TN} ; & \text{FPR} &= \frac{FP}{FP+TN} ; \\ \text{FPR} &= \frac{16}{16+1784} ; & \text{FPR} &= \frac{57}{57+1743} ; \\ \text{FPR} &= 0,89\% ; & \text{FPR} &= 3,17\% ; \end{aligned}$$

4.6 Cálculo da métrica Taxa de falso negativo dos algoritmos de Random e Decision

$$\begin{aligned} \text{FNR} &= \frac{FN}{TP+FN} ; & \text{FNR} &= \frac{FN}{TP+FN} ; \\ \text{FNR} &= \frac{25}{1200} ; & \text{FNR} &= \frac{65}{1135+65} ; \\ \text{FNR} &= 2,08\% ; & \text{FNR} &= 5,42\% ; \end{aligned}$$

Foram identificados falsos negativos, correspondente a uma taxa de erro de aproximadamente 2,08% e 5,42%, indica que parte dos ataques alcance o controlador sem

bloqueio ou resposta adequada, enquanto os falsos positivos representam fluxos legítimos classificados como ataques que corresponde 0,89% e 3,17%. Pode causar bloqueio indevido e degradação da qualidade do serviço da rede. Este equilíbrio demonstra adequação do Random Forest e Decision Tree, para o cenário proposto.

Tabela 12

Comparação entre Random Forest e Decision Tree

Modelo	Acurácia	Precision	Recall	F1-score
Random Forest	98,63%	98,70%	97,90%	98,30%
Decision Tree	95,93%	95,20%	94,60%	94,90%

Fonte: Autor

O algoritmo de *Decision Tree* apresentou desempenho inferior ao Random Forest em todas as métricas avaliadas devido o fato de que uma única árvore de decisão possui maior sensibilidade às características específicas do conjunto de treinamento, podendo apresentar maior variância e menor capacidade de generalização quando comparada a um conjunto de árvores. A *Random Forest*, por outro lado, combina múltiplas árvores independentes e utiliza mecanismos de agregação dos resultados, reduzindo a influência de ruídos e decisões individuais inadequadas. Essa característica contribui para uma classificação mais estável dos fluxos de rede, especialmente em cenários onde existem padrões complexos, entre as características do tráfego.

5 CONCLUSÃO

Este estudo analisou a utilização de técnicas de *Machine Learning* para a detecção de ataques DDoS direccionados ao controlador de redes definidas por software (SDN), com o objectivo de identificar padrões de tráfego associados à interrupção da comunicação na rede. Num ambiente experimental simulado, foram treinados e avaliados, os algoritmos *Random Forest* e *Decision Tree*. O *Random Forest* apresentou o melhor desempenho, com 98,63% de acurácia, 98,66% de precisão, 97,92% de *recall* e 98,29% de *F1-score*. O *Decision Tree* alcançou 95,93% de acurácia, 95,22% de precisão, 94,58% de *recall* e 94,89% de *F1-score*. Os resultados demonstram elevada capacidade dos modelos para distinguir tráfego legítimo, de tráfego associado a ataques DDoS, no ambiente experimental.

Contudo, o *dataset* foi produzido numa infra-estrutura controlada, utilizando *Mininet* e tráfego sintético, o que limita a generalização dos resultados para redes reais. Nestes ambientes, a maior diversidade de dispositivos, padrões de tráfego e tipos de ataques pode influenciar o desempenho dos modelos. Como trabalhos futuros, recomendam-se a validação da abordagem com *datasets* públicos de tráfego real, a avaliação de algoritmos de *Machine Learning* e *Deep Learning*, a optimização de hiperparâmetros, a implementação de mecanismos de aprendizagem contínua e detecção de anomalias.

6. REFERÊNCIAS

- Ali, T. E., Chong, Y.-W., & Manickam, S. (2023). Machine learning techniques to detect a DDoS attack in SDN: A systematic review. *Applied Sciences*, 13(5), 3183. <https://doi.org/10.3390/app13053183>
- Altamemi, A. J., Abdulhassan, A., & Obeis, N. T. (2022). DDoS attack detection in software defined networking controller using machine learning techniques. *Bulletin of Electrical Engineering and Informatics*, 11(5), 2836–2844. <https://doi.org/10.11591/eei.v11i5.4155>
- Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys & Tutorials*, 16(1), 303–336. <https://doi.org/10.1109/SURV.2013.052213.00046>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>

- Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21–27. <https://doi.org/10.1109/TIT.1967.1053964>
- Douligeris, C., & Mitrokotsa, A. (2004). DDoS attacks and defense mechanisms: Classification and state-of-the-art. *Computer Networks*, 44(5), 643–666. <https://doi.org/10.1016/j.comnet.2003.10.003>
- Dua, S., & Du, X. (2016). *Data mining and machine learning in cybersecurity*. CRC Press.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Han, J., Kamber, M., & Pei, J. (2012). *Data mining: Concepts and techniques* (3rd ed.). Morgan Kaufmann.
- Karthika, P., & Arockiasamy, K. (2023). Simulation of SDN in Mininet and detection of DDoS attack using machine learning. *Bulletin of Electrical Engineering and Informatics*, 12(3), 1797–1805. <https://doi.org/10.11591/eei.v12i3.5232>
- Kim, G., Lee, S., & Kim, S. (2014). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications*, 41(4), 1690–1700. <https://doi.org/10.1016/j.eswa.2013.08.066>
- Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1), 14–76. <https://doi.org/10.1109/JPROC.2014.2371999>
- Lantz, B., Heller, B., & McKeown, N. (2010). A network in a laptop: Rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. Association for Computing Machinery. <https://doi.org/10.1145/1868447.1868466>
- Mansoor, A., Anbar, M., Bahashwan, A. A., Alabsi, B. A., & Rihan, S. D. A. (2023). Deep learning-based approach for detecting DDoS attack on software-defined networking controller. *Systems*, 11(6), 296. <https://doi.org/10.3390/systems11060296>
- Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2), 39–53. <https://doi.org/10.1145/997150.997156>
- Mohsin, M. A., & Hamad, A. H. (2022). Performance evaluation of SDN DDoS attack detection and mitigation based random forest and K-nearest neighbors machine learning algorithms. *Revue d'Intelligence Artificielle*, 36(2), 233–240. <https://doi.org/10.18280/ria.360207>
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81–106. <https://doi.org/10.1007/BF00116251>
- Rossow, C. (2014). Amplification hell: Revisiting network protocols for DDoS abuse. In *Proceedings of the 2014 Network and Distributed System Security Symposium*. Internet Society.
- Scott-Hayward, S., O'Callaghan, G., & Sezer, S. (2016). SDN security: A survey. In *Proceedings of the IEEE SDN for Future Networks and Services (SDN4FNS)*. IEEE.
- Shin, S., & Gu, G. (2013). Attacking software-defined networks: A first feasibility study. In *Proceedings of the 2013 ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking* (pp. 165–166). Association for Computing Machinery. <https://doi.org/10.1145/2491185.2491220>
- Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427–437. <https://doi.org/10.1016/j.ipm.2009.03.002>
- Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE Symposium on Security and Privacy* (pp. 305–316). IEEE. <https://doi.org/10.1109/SP.2010.25>