# CLASSIFICAÇÃO DE ATAQUES DDOS UTILIZANDO ALGORITMOS DE APRENDIZADO DE MÁQUINA: K-NEAREST NEIGHBORS E RANDOM FOREST

## DDOS ATTACK CLASSIFICATION USING MACHINE LEARNING ALGORITHMS: K-NEAREST NEIGHBORS AND RANDOM FOREST

Hugo Felipe Nogueira Teixeira[1]
Tassio Costa de Carvalho[2]

**Área Temática 07: Tecnologias Sociais, Tecnologia Educacionais e Assistivas e Tecnologia da Informação**
**Modalidade: Artigo Científico**

**Resumo**
Este trabalho realiza uma análise comparativa entre os algoritmos de aprendizado de máquina K-Nearest Neighbors (KNN) e Random Forest para a classificação de ataques DDoS. O estudo incluiu etapas de seleção de atributos, balanceamento de dados e validação cruzada. O KNN foi otimizado com curva de validação variando o número de vizinhos, enquanto o Random Forest utilizou seus parâmetros padrão. A avaliação por métricas como acurácia, precisão, revocação e F1-score demonstrou desempenho satisfatório de ambos os modelos, com destaque para o KNN, que obteve melhores resultados. Além disso, os resultados obtidos foram comparados com trabalhos relacionados à literatura, a fim de contextualizar e validar a eficácia dos modelos propostos. Os experimentos confirmam a eficácia do uso de técnicas de aprendizado de máquina na detecção de ataques DDoS em tráfego de rede.
**Palavras-Chave: Algoritmo, Aprendizado, Classificação, DDos**

**Abstract**
This work performs a comparative analysis between the K-Nearest Neighbors (KNN) and Random Forest machine learning algorithms for DDoS attack classification. The study included feature selection, data balancing, and cross-validation steps. KNN was optimized with a validation curve varying the number of neighbors, while Random Forest used its default parameters. The evaluation by metrics such as accuracy, precision, recall, and F1-score demonstrated satisfactory performance of both models, with emphasis on KNN, which obtained better results. In addition, the results obtained were compared with related works in the literature in order to contextualize and validate the effectiveness of the proposed models. The experiments confirm the effectiveness of using machine learning techniques in detecting DDoS attacks in network traffic.
**Key words: Algorithm, Learning, Classification, DDoS**

[1] (UFPA) Universidade Federal do Pará; elna.hugo.nogueira@gmail.com
[2] (UFPA) Universidade Federal do Pará; tassiocarv@gmail.com

## 1. Introdução

O crescente aumento da conectividade e do volume de informações trafegadas nas redes trouxe consigo ameaças cada vez mais sofisticadas. Dentre elas, destacam-se os ataques de negação de serviço que consistem no envio de pacotes massivos extremamente rápidos. Esses ataques visam aplicações específicas ou recursos do servidor, consumindo recursos de forma sutil e dificultando sua detecção (GOGOI; AHMED, 2022).

Segundo Chaves et al (2021), os ataques de Negação de Serviço distribuídos causam indisponibilidade de serviços essenciais: O impacto mais direto e imediato de um ataque DDoS é a negação do acesso a serviços para usuários legítimos, tornando os serviços de comunicação lentos ou até mesmo indisponíveis. Como consequência, podem causar prejuízos consideráveis tanto em redes privadas quanto em infraestruturas institucionais públicas. Bem como a identificação de ataques DDoS é difícil devido à sua operação na camada de aplicação, que os torna indistinguíveis do tráfego legítimo por métodos de defesa tradicionais (ZHAO; et al, 2010)

Este trabalho tem como objetivo contribuir para a identificação desse tipo de ataque, auxiliando na detecção de anomalias e no aprimoramento das estratégias de mitigação em ambientes de rede. Dentro deste contexto, este artigo propõe a implementação de mecanismos de detecção de ataques DDoS por meio da aplicação dos algoritmos de aprendizado de máquina K-Nearest Neighbors (KNN) e Random Forest. Para isso, foram realizados experimentos com a base de dados pública CIC-DDoS2019, um conjunto de dados amplamente utilizado no treinamento e validação de sistemas de detecção de ataques DoS (SHAFARALDI et al., 2019). Os resultados obtidos demonstraram a eficácia dos modelos KNN e Random Forest na identificação de ataques, reforçando o potencial do uso de aprendizado de máquina nesse tipo de cenário.

### 1.2. Trabalhos Relacionados

A literatura abrangente demonstra que a utilização de aprendizado de máquina tem se tornado uma ferramenta poderosa na detecção de ataques DoS e também auxiliar na segurança de redes de computadores. A seguir, alguns trabalhos que exploram semelhanças com este presente artigo.

O primeiro trabalho de Dias (2023), aborda a segurança em Redes Definidas por Software (SDN), destacando os ataques de Negação de Serviço Distribuído (DDoS) como um

desafio crítico. A pesquisa propõe uma abordagem baseada em microsserviços para detectar esses ataques usando aprendizado de máquina. Foram avaliados cinco modelos de aprendizado de máquina: SVM (Support Vector Machine), RL (Regressão Logística), KNN (K-Nearest Neighbors), DT (Decision Tree) e RF (Random Forest). Os resultados identificaram o Random Forest (RF) como o modelo mais eficaz, atingindo um F1-Score de 98,65% e acurácia de 98,50%. O Decision Tree (DT) também obteve bons resultados, com 98,06% de acurácia e 98,25% de F1-Score. A abordagem de microsserviços permitiu a utilização de modelos mais complexos, como o RF, sem comprometer o desempenho do controlador SDN, que utilizou em média 0,25 CPU durante os ataques, enquanto o modelo RF isoladamente consumiu até 0,5 CPU

O segundo artigo, "Detecção de Ataques de Negação de Serviço Distribuídos com Algoritmos de Aprendizado de Máquina", por Silva et al (2024), propõe uma metodologia para detectar e classificar ataques DDoS, que inclui técnicas de balanceamento de dados, pré-processamento e seleção de atributos. O conjunto de dados utilizado para treinamento, validação e avaliação foi o CIC-DDoS2019. Foram avaliados cinco algoritmos de aprendizado de máquina: Naive Bayes (NB), MultiLayer Perceptron (MLP), Árvore de Decisão (Decision Tree - DT), Random Forest (RF) e Support Vector Machine (SVM). Os experimentos mostraram que o algoritmo Random Forest (RF) obteve os melhores resultados tanto na classificação binária quanto na multiclasse. No cenário binário sem dados sintéticos, o RF alcançou 99,8% de acurácia. Na classificação multiclasse, o RF atingiu uma taxa de detecção de 100% para ataques SYN e 98% ou superior para outros tipos de ataques (SILVA; et al, 2024).

O artigo de Costa, Portela e Gomes (2021), propõe um sistema inteligente para detectar ataques DDoS em ambientes inteligentes, integrando aprendizado de máquina com computação em névoa e nuvem. Foram utilizados os algoritmos KNN, Naive Bayes, Regressão Logística, Árvore de Decisão, Floresta Aleatória e SVM. Árvore de Decisão e Floresta Aleatória se destacaram tanto na segmentação de tráfego quanto na detecção de DDoS, alcançando acurácias próximas a 99%. A seleção de características (mRMR, Lasso, SVC, Extra-Árvore e Baixa Variância) foi fundamental para reduzir o tempo de treinamento, melhorar a acurácia e diminuir o volume de dados transmitidos da névoa para a nuvem.

A análise desses artigos é importante para verificar a eficácia de algoritmos de aprendizado de máquina como KNN e Random Forest, na detecção de ataques DDoS em redes

de computadores. Este trabalho dá ênfase no desempenho de hiperparâmetros e no pré-processamento robusto de dados do dataset CIC-DDoS2019, contribuindo para a validação de técnicas em cenários reais de tráfego de rede!

## 2. Metodologia

Esta metodologia descreve as etapas implementadas no código para análise e classificação de ataques de negação de serviço distribuídos (DDoS) utilizando o dataset DrDoS_DNS.csv do CIC-DDoS2019. Com o objetivo de construir, treinar e avaliar modelos de machine learning para classificar tráfego de rede como DDoS ou Benigno, com foco em balanceamento de dados, seleção de características e avaliação de desempenho.

### 2.1. Pré-processamento de dados

Inicialmente foi realizada a instalação de bibliotecas essenciais como pandas, numpy, matplotlib, seaborn, scikit-learn, imblearn e optuna para manipulação de dados, visualização, aprendizado de máquina e otimização de hiperparâmetros. Carregamento do Dataset DrDoS_DNS.csv, limitado a 300.000 linhas para reduzir o custo computacional e ignorando linhas inválidas. Seleção de colunas específicas (Fwd IAT Total, Flow IAT Max, Flow Packets/s) para análise, reduzindo a dimensionalidade e focando em atributos relevantes. O critério de seleção se deu pela utilização do algoritmo MRMR (Mínima Redundância Máxima Relevância), Ao selecionar um novo atributo, a técnica MRMR avalia dois aspectos principais Máxima relevância em relação ao rótulo (ou classe) Mínima redundância em relação ao subconjunto de características que já foram selecionadas. A relevância e a redundância são calculadas utilizando as pontuações do teste de Fisher e a correlação de Pearson, respectivamente (COSTA; et al, 2021). Assim foram selecionadas as colunas Fwd IAT Total, Flow IAT Max, Flow Packets/s como as mais indicadas para se realizar a análise.

Para garantir que os dados estejam limpos e prontos para serem analisados, foi necessário fazer um pré-processamento, Realização da limpeza e tratamento de valores faltantes e infinitos, bem como o balanceamento de classes, como demonstra a tabela 1, para igualar o número de amostras de cada classe, com base na classe com menor quantidade de instâncias, a conversão de valores categóricos, em “0” e “1”, isto é essencial, pois os modelos utilizados apenas trabalham com valores numéricos. Os dados balanceados são embaralhados para garantir aleatoriedade.

Tabela 1 - Dados balanceados

| Amostras | Antes do balanceamento | depois do balanceamento |
|---|---|---|
| DrDoS_DNS (0) | 498095 | 1905 |
| BENIGN (1) | 1900 | 1905 |

Fonte - Autores

Para poder treinar o modelo é necessário fazer uma divisão, separar dados para treino que são os dados que realmente serão postos aos experimentos do modelo, para treiná-lo, e dados para teste, tais dados serão um comparativo, onde o modelo será posto a prova, com avaliação de desempenho a partir dos dados de teste. Logo a divisão foi feita com 70% dos dados para treinamento e os restantes, 30%, para teste.

**2.2 Modelagem**

Na etapa de modelagem, foram implementados dois modelos de classificação, o K-Nearest Neighbors (KNN) e o Random Forest. O KNN é um dos algoritmos não-paramétricos mais importantes no campo de reconhecimento de padrões, sendo um algoritmo de classificação de aprendizado supervisionado. As regras de classificação do KNN são geradas pelas próprias amostras de treinamento sem nenhum dado adicional (COSTA; et al, 2021). Ja o Random Forest é um modelo de árvores de decisão: O Random Forest é um modelo composto por várias árvores de decisão. Em suma, uma árvore de decisão é baseada na extração de regras dos dados que serão analisados para que com isso possa gerar a árvore de decisão daquele modelo treinado e testado. O Random forest é visto como um sucessor do algoritmo Random Tree, pois utiliza múltiplas árvores para melhorar o desempenho (Nascimento, 2022).

O processo de modelagem com o classificador K-Nearest Neighbors (KNN) começa com a inicialização do modelo utilizando os parâmetros padrão da biblioteca *scikit-learn*. Em seguida, é realizada uma busca em grade com o *GridSearchCV* para testar diferentes valores do hiperparâmetro número de vizinhos (*n_neighbors)*, variando de 1 a 29 com incremento de 2, empregando validação cruzada do tipo *StratifiedShuffleSplit* e utilizando a métrica *f1_macro* para avaliar o desempenho. Para analisar como o desempenho do modelo varia em função de *n_neighbors*, uma curva de validação é gerada, comparando os scores de treino e validação. O

treinamento do modelo é conduzido com os dados de treinamento após a imputação de valores faltantes, utilizando a estratégia de média com *SimpleImputer*, e a normalização dos dados. Na etapa de avaliação, o melhor valor de *n_neighbors* é selecionado com base no maior *f1_macro* obtido na busca em grade. As métricas de desempenho, incluindo acurácia, precisão, revocação e *F1-score*, são calculadas por classe, bem como nas médias macro e micro, fornecendo uma visão detalhada do desempenho do modelo. Por fim, uma matriz de confusão é gerada para visualização, permitindo a análise das previsões corretas e incorretas do modelo.

A modelagem do classificador Random Forest inicia com a configuração do modelo, utilizando 100 estimadores, profundidade máxima ilimitada e parâmetros padrão para *min_samples_split* e *min_samples_leaf*, parâmetros que controlam como as árvores de decisão são construídas, influenciando o crescimento e a complexidade do modelo. A divisão dos. O modelo é então treinado com os dados de treinamento. Na etapa de avaliação, as métricas de desempenho, incluindo acurácia, precisão, revocação e *F1-score*, são calculadas da mesma forma que no classificador KNN, considerando tanto os valores por classe quanto às médias macro e micro. Por fim, uma matriz de confusão é gerada para visualização, permitindo a análise das previsões corretas e incorretas do modelo.

**2.3 Avaliação de desempenho**

Para avaliar o desempenho dos modelos, são analisadas as métricas, acurácia, precisão, revocação e F1-score. A acurácia representa a proporção de previsões corretas feitas pelo modelo em relação ao total de previsões, a precisão é a proporção de previsões positivas corretas em relação ao total de previsões positivas feitas pelo modelo. A revogação (Recall), a proporção de casos positivos corretamente identificados em relação ao total de casos positivos reais. E, por fim, o F1-score é a média harmônica entre precisão e revocação, balanceando essas duas métricas. Para avaliar o desempenho das métricas são plotadas relatórios que calculam métricas detalhadas e imprime um relatório com valores por classe e médias e a matriz de confusão que demonstra que organiza as previsões do modelo em relação aos valores reais, permitindo visualizar os acertos e erros. Também é avaliado as métricas dos modelos, bem como feito um comparativo entre as acurácias de cada modelo. Por fim, este experimento prioriza o pré-processamento e o balanceamento, garantindo que não ocorram erros nos modelos, bem como a escolha de KNN e Random Forest permite comparar um modelo baseado em distância (KNN) com um modelo baseado em árvores (Random Forest).

Por fim será comparado os resultados dos trabalhos relacionados, especificamente os resultados da métrica de acurácia para os algoritmos KNN e Random Forest, com os resultados obtidos no presente trabalho, com o objetivo de avaliar o desempenho do experimento em relação à demais semelhantes artigos.

## 3. Resultados/Discussões

Os experimentos dos algoritmos KNN e Random Forest foram realizados na plataforma do Google Colab e apresentaram resultados satisfatórios, as imagens apresentadas nesta seção demonstram os resultados das métricas de acurácia, precisão, revocação e *F1-score*. A seguir, discutiremos os resultados obtidos, com ênfase na precisão e consistência dos dados.

### 3.1. K-Nearest Neighbors (KNN)

O modelo K-Nearest Neighbors (KNN) apresentou um desempenho expressivo na classificação de tráfego benigno e ataques do tipo DrDoS_DNS, com todas as métricas superando a marca de 95%. Para a classe BENIGN a revocação foi de 96,3%, a precisão de 98,2%, F1-score de 97,3%. Já para a classe DrDoS_DNS, os resultados foram ainda mais altos, com revocação de 98,2%, precisão de 96,4% e F1-score de 97,3%. As médias macro e micro mantiveram-se equilibradas, ambas com revocação de 98,6%, precisão de 97,0% e F1-score de 97,7%, o que evidencia a robustez do modelo mesmo em um cenário com múltiplas classes. O destaque vai para a revocação e o F1-score da classe DrDoS_DNS, indicando uma capacidade significativa do modelo em identificar ataques com alta efetividade. A acurácia geral do modelo KNN foi de 97,3%, representado pelo F1-score ou F-medida,

Imagem 1 - relatório de métricas do KNN

| Revocacao | Precisao | F-medida | Classe |
|---|---|---|---|
| 0.963 | 0.982 | 0.973 | BENIGN |
| 0.982 | 0.964 | 0.973 | DrDoS_DNS |
| 0.973 | 0.973 | 0.973 | Média macro |
| 0.973 | 0.973 | 0.973 | Média micro |

Acuracia: 0.973

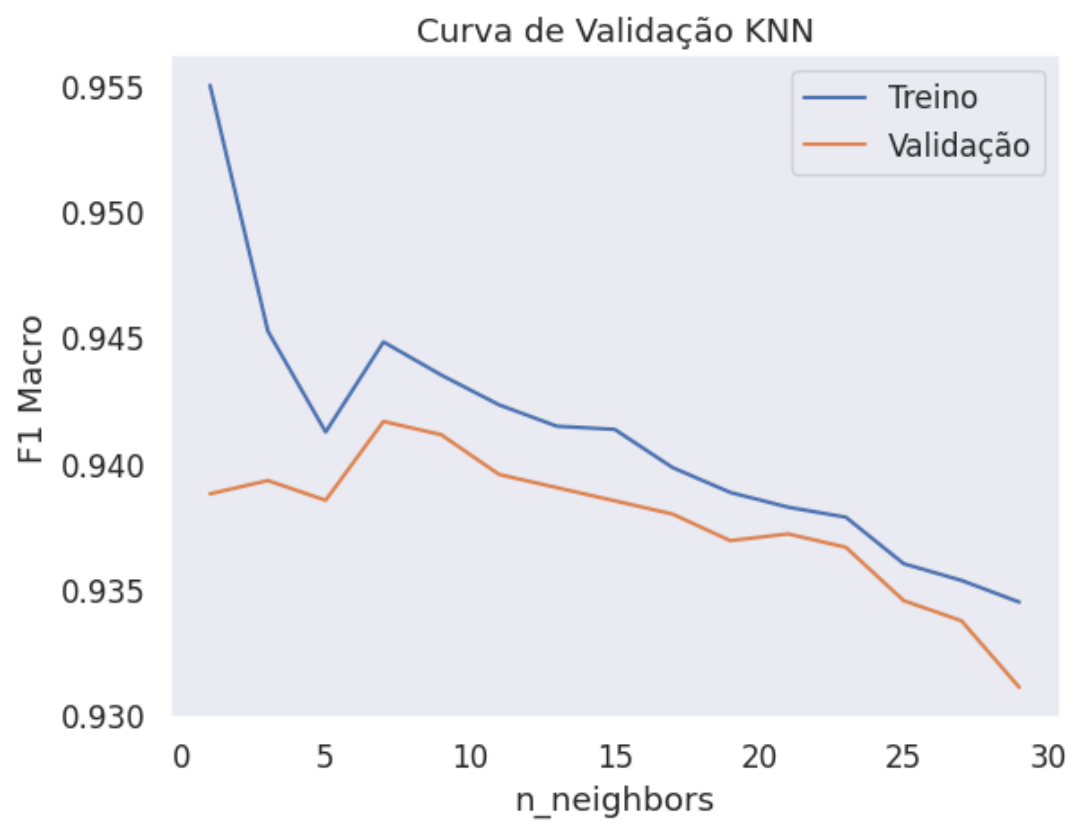
Fonte - Autores

Imagem 2 - Métricas de Desempenho por Classe


Fonte - Autores

Os resultados obtidos refletem a eficácia do modelo após a otimização de seus hiperparâmetros utilizando *GridSearchCV*, complementada por uma análise da curva de validação. Esta curva demonstrou que o F1-score macro no conjunto de treino variou de aproximadamente 0,955 (com *n_neighbors* = 1) até cerca de 0,935 (com *n_neighbors* = 29), enquanto no conjunto de validação os valores oscilaram entre 0,94 e 0,93. Os melhores desempenhos foram observados para valores de vizinhos entre 5 e 15, sugerindo que valores baixos promovem melhor desempenho, enquanto valores altos tendem a causar *underfitting*. A diferença reduzida entre os resultados de treino e validação indica baixa variância, sugerindo que o modelo não sofre de *overfitting*. O valor de vizinhos selecionado foi 3, que mostrou-se adequado para maximizar o desempenho do KNN.

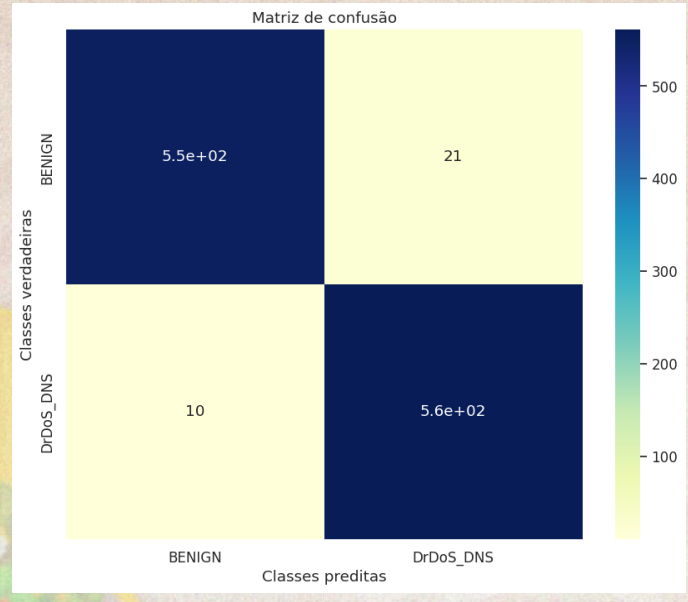
Imagem 3 - Curva de Validação KNN




Fonte - Autores

A matriz de confusão reforça a eficiência do modelo. Foram registrados cerca de 550 acertos na predição de tráfego BENIGN como BENIGN, contra 21 falsos positivos classificados erroneamente como DrDoS_DNS. Para a classe DrDoS_DNS, aproximadamente 560 instâncias foram corretamente identificadas, com apenas 10 falsos negativos classificados como BENIGN. Essa predominância de acertos na diagonal principal da matriz indica alta acurácia e reforça a capacidade do modelo em distinguir corretamente entre tráfego legítimo e malicioso. O número relativamente baixo de erros, tanto falsos positivos quanto falsos negativos, sugere que o modelo é eficaz tanto na detecção de ataques quanto na preservação do tráfego legítimo. Além disso, a simetria observada na matriz indica que o balanceamento do dataset foi bem-sucedido, contribuindo para um desempenho uniforme entre as classes. Ainda assim, os poucos erros observados podem indicar a necessidade de ajustes finos para melhorar a identificação em situações de fronteira entre.

Imagem 4 - Matriz de Confusão KNN




Fonte - Autores

### 3.2. Random Forest

O modelo Random Forest apresentou um desempenho consistente na classificação entre tráfego BENIGN e ataques DrDoS_UDP, com métricas superiores a 91% em todas as avaliações, embora ligeiramente inferiores às obtidas pelo KNN. Como demonstra a imagem, para a classe BENIGN, a revocação foi de 98,6%, a precisão de 92,5% e o F1-score de 95,4%. Já para a classe DrDoS_UDP, a revocação foi de 91,9%, com precisão de 98,5% e F1-score de 95,1%. As médias macro e micro ficaram equilibradas, ambas com revocação de 95,3%, precisão de 95,5% e F1-score de 95,3%, o que indica um desempenho uniforme entre as classes. A acurácia do modelo de Random Forest alcançou os 95,3%.

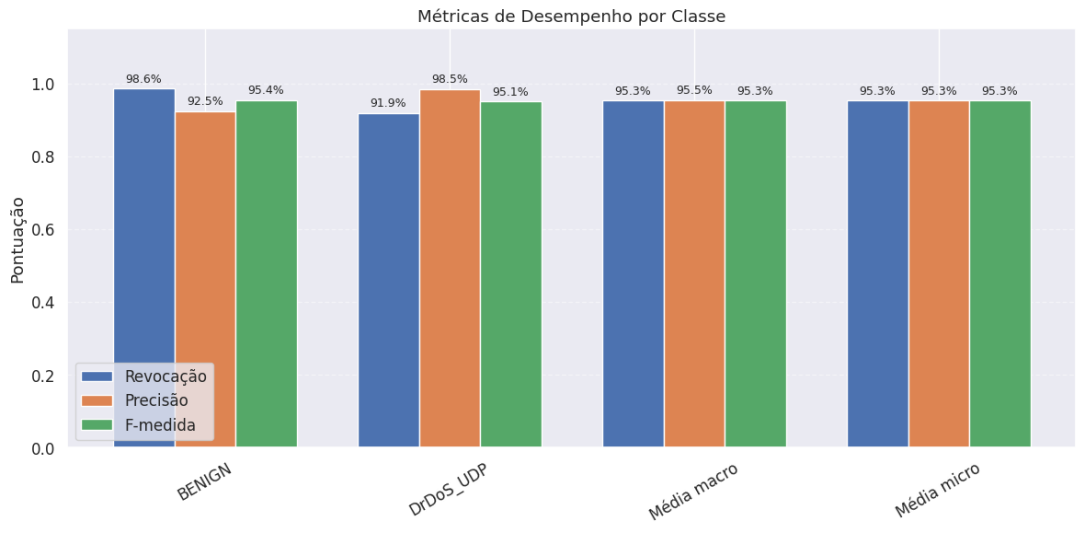
Imagem 5 - relatório de métricas de Classificação Random Forest

```
Relatório de Classificação:

    Revocacao   Precisao   F-medida   Classe
    0.986       0.925      0.954      BENIGN
    0.919       0.985      0.951      DrDoS_UDP
    ------------------------------------------------
    0.953       0.955      0.953      Média macro
    0.953       0.953      0.953      Média micro

    Acuracia: 0.953
```

Fonte - Autores

Apesar da precisão elevada para a classe DrDoS_DNS, a revocação mais baixa nesta classe indica a ocorrência de falsos negativos, ou seja, instâncias de ataque que não foram detectadas pelo modelo. Esse comportamento pode comprometer a eficácia do sistema em aplicações críticas de segurança, onde a falha em identificar um ataque pode trazer consequências severas. Ainda assim, o F1-score equilibrado em ambas as classes mostra que o Random Forest mantém um bom equilíbrio entre precisão e revocação, o que o torna um modelo confiável, embora com espaço para aprimoramento na sensibilidade a ataques.

Imagem 6 - Métricas de Desempenho por Classe Random Forest

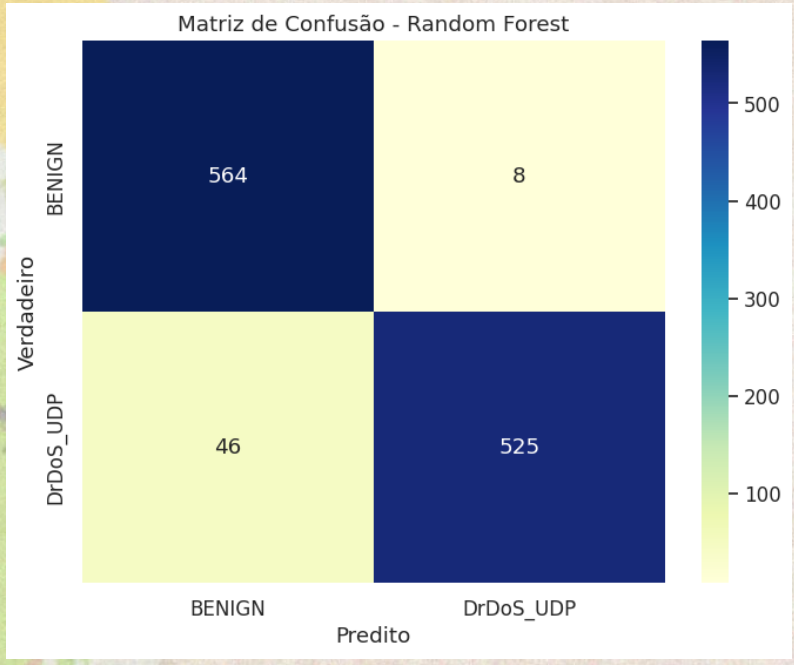



Fonte - Autores

A matriz de confusão reforça essa análise. Foram registrados 564 acertos na identificação correta de tráfego BENIGN e 525 acertos na classificação de DrDoS_UDP, com apenas 8 falsos positivos (BENIGN classificados incorretamente como ataque). Por outro lado, ocorreram 46 falsos negativos, em que ataques DrDoS_UDP foram erroneamente classificados como

BENIGN, representando aproximadamente 8% de erro na detecção de ataques. Isso evidencia a necessidade de ajustar o modelo para melhorar a revocação, sem comprometer a alta precisão já alcançada. No geral, o Random Forest demonstrou desempenho sólido, com acurácia elevada e boa capacidade de generalização, mas com margem para otimização específica na identificação de ataques.

Imagem 7 - Matriz de Confusão Random Forest




Fonte - Autores

Com base nos resultados obtidos, observa-se que o KNN e o Random Forest apresentaram desempenhos satisfatórios na classificação entre tráfego benigno e ataques de negação de serviço. O modelo KNN se destacou com métricas superiores, atingindo acurácia de 97% e revocação ótima, especialmente para a classe de ataque "DrDoS_DNS", o que demonstra sua elevada capacidade de generalização após a otimização por *GridSearchCV*. Sua baixa taxa de falsos negativos e falsos positivos reforça sua eficiência em ambientes onde a detecção precisa de ataques é crucial.

### 3.3. Trabalhos Relacionados

Ao analisar os trabalhos relacionados da Tabela 2 apresenta uma comparação da acurácia dos algoritmos KNN e Random Forest em diferentes estudos relacionados à detecção

de ataques DDoS. Observa-se que, em geral, o Random Forest apresenta desempenho superior, alcançando até 99,99% de acurácia (Silva, 2024). O trabalho de Costa, Portela e Gomes (2021) também destaca o excelente desempenho do Random Forest (99,91%), enquanto o KNN apresenta resultados mais variados. Neste trabalho (2025), ambos os algoritmos obtiveram bons resultados, com o KNN superando o Random Forest, o que pode indicar uma melhor adaptação do modelo KNN aos dados utilizados. Isto demonstra que o experimento realizado está na média dos demais estudos relacionados, obtendo valores em média acima de 91%.

Tabela 2 - Comparação de Acurácias com trabalhos Relacionados

| Trabalho | KNN | Random Forest |
|---|---|---|
| *Dias (2021)* | 97,78% | 98,50% |
| Silva (2024) | _ | 99,99% |
| Costa, Portela e Gomes (2021) | 86,65% | 99,91% |
| Este trabalho (2025) | 97,3% | 95,3% |

Fonte - Autores

## 4. Considerações Finais ou Conclusão

Com base nos experimentos realizados, conclui-se que os algoritmos K-Nearest Neighbors (KNN) e Random Forest apresentaram desempenhos satisfatórios na tarefa de classificação de tráfego de rede, distinguindo eficientemente entre tráfego benigno e de ataques. O modelo KNN obteve métricas superiores, destacando-se com elevados índices de revocação e F1-score, especialmente na detecção de ataques, além de apresentar baixa taxa de falsos positivos e negativos, o que o torna altamente confiável em aplicações onde a sensibilidade e a precisão na detecção são cruciais.

Já o modelo Random Forest demonstrou desempenho robusto, com acurácia e precisão elevadas, embora com maior incidência de falsos negativos, o que pode comprometer sua eficácia em cenários que exigem alta detecção de ameaças. Ainda assim, seu equilíbrio entre

precisão e revocação confirma seu potencial de uso em sistemas de defesa, especialmente em contextos que priorizam a redução de alarmes falsos.

Ambos os modelos beneficiaram-se do pré-processamento cuidadoso, da seleção de atributos relevantes e do balanceamento das classes, o que contribuiu para a estabilidade dos resultados. A análise das curvas de validação e das matrizes de confusão reforçou o entendimento sobre o comportamento de cada algoritmo.

Para trabalhos futuros, pretende analisar o desempenho de mais algoritmos de aprendizado de máquina voltados à classificação, a fim de se ter um experimento mais robusto e abrangente sobre a eficiência de aprendizado de máquina para ataques DDoS.

## 5. Referências Bibliográficas

**GOGOI, Bronjon; AHMED, Tasiruddin.** HTTP Low and Slow DoS attack detection using LSTM based deep learning. In: **IEEE 19th India Council International Conference (INDICON)**, 2022. Anais [...]. IEEE, 2022. DOI: 10.1109/INDICON56171.2022.10039772. Disponível em: https://ieeexplore.ieee.org/document/10039772. Acesso em: 26 jun. 2025.

**SHARAFALDIN, I.; LASHKARI, A. H.; HAKAK, S.; GHORBANI, A. A.** Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In: **2019 IEEE International Conference on Communications,** 2019, [S.l.]. Anais [...]. [S.l.]: IEEE, 2019. p. 1-8. DOI: 10.1109/ICC.2019.8761351. Disponível em: https://ieeexplore.ieee.org/abstract/document/8888419. Acesso em: 26 jun. 2025.

**ZHAO, J.; GUO, S.; ZHENG, K.; NIU, X.; JIANG, Y.** An active defense model for web accessing DoS attacks. In: **2010 IEEE International Conference on Communications,** 2010, Beijing. Anais [...]. [S.l.]: IEEE, 2010. p. 314-318. DOI: 10.1109/ICC.2010.5502051.

**COSTA, Wanderson Leonardo; DE CARVALHO PORTELA, Ariel Lima; GOMES, Rafael Lopes.** Análise de Características do Tráfego de Rede para Detecção de Ataques DDoS em Ambientes IoT. **Anais do Computer on the Beach,** v. 12, p. 217-224, 2021. Disponível em: https://periodicos.univali.br/index.php/acotb/article/view/17404. Acesso em: 26 jun. 2025.

**NASCIMENTO, Juliano Silva do.** Comparação dos algoritmos Random Forest, Random Tree e J48 para detectar ataques DDoS. 2022**. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.** Disponível em: https://riut.utfpr.edu.br/jspui/handle/1/31819. Acesso em: 27 jun. 2025.

DIAS, Victor et al. Detecçao de ataques ddos em redes sdn utilizando aprendizado de máquina: Uma abordagem em microsserviços. In: **Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)**. SBC, 2023. p. 141-152.

SILVA, Rodrigo R. et al. Detecção de Ataques de Negação de Serviço Distribuídos com Algoritmos de Aprendizado de Máquina. In: **Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)**. SBC, 2024. p. 226-241.

COSTA, Wanderson L.; PORTELA, Ariel LC; GOMES, Rafael L. Um Sistema Inteligente para Detecçao de DDoS em Ambientes Inteligentes Baseado em Computaçao em Nuvem e em Névoa. In: **Workshop de Computação Urbana (CoUrb)**. SBC, 2021. p. 237-250.