



**CURSO DE BACHARELADO EM
CIÊNCIA DA COMPUTAÇÃO**

IGOR MATHEUS TEIXEIRA

TRABALHO DE CONCLUSÃO DE CURSO

Deteccção de Ataques de Negação de Serviço Utilizando Aprendizado de Máquina.

Presidente Epitácio – SP

2024

IGOR MATHEUS TEIXEIRA

TRABALHO DE CONCLUSÃO DE CURSO

Detecção de Ataques de Negação de Serviço Utilizando Aprendizado de Máquina.

Trabalho de Conclusão de Curso apresentado à banca examinadora do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Campus Presidente Epitácio – Curso de Bacharelado em Ciência da Computação, como requisito parcial à obtenção do grau de Bacharel em Ciência da Computação sob a orientação do professor Me. Kleber Manrique Trevisani.

Presidente Epitácio – SP

2024

FICHA CATALOGRÁFICA

Ficha catalográfica elaborada pela Coordenadoria de Biblioteca, IFSP Campus de Presidente Epitácio, com dados fornecidos pelo autor.

T266d Teixeira, Igor Matheus.

Detecção de ataques de negação de serviço utilizando aprendizado de máquina / Igor Matheus Teixeira. – Presidente Epitácio : I. M. Teixeira, 2025.

27 f. ; il.

Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Presidente Epitácio/SP, 2025.

Bibliografia: f. 26.

Orientador: Prof. Dr. Kleber Manrique Trevisani.

1. Algoritmo. 2. Aprendizado de Máquina. 3. Negação de Serviço. I. Autor.

CDD – 005.1

FOLHA DE APROVAÇÃO

FOLHA DE APROVAÇÃO

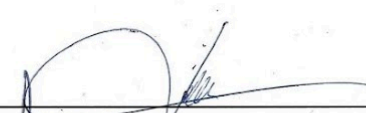
IGOR MATEUS TEIXEIRA

Deteccção de ataques de negação de serviço utilizando aprendizado de maquina

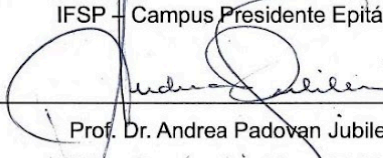
Trabalho de Conclusão de Curso para obtenção do título de Bacharel em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Câmpus Presidente Epitácio.

Aprovado pela banca examinadora em Segunda-feira, 09 de dezembro de 2024.


BANCA EXAMINADORA:



Prof. Me. Kleber Manrique Trevisani
IFSP – Campus Presidente Epitácio



Prof. Dr. Andrea Padovan Jubileu
IFSP – Campus Presidente Epitácio



Prof. Dr. Cesar Alberto da Silva
IFSP – Campus Presidente Epitácio

Dedico este trabalho à minha mãe Eidimara e ao meu pai Djalma por todo apoio e carinho.

Obrigado!

AGRADECIMENTOS

Expresso minha mais profunda gratidão, primeiramente, a Deus, pela oportunidade e pela força concedida ao longo de toda essa jornada até a conclusão deste trabalho. Agradeço ao Instituto Federal de Educação, Ciência e Tecnologia, especialmente ao campus de Presidente Epitácio, pelo suporte acadêmico e pelas oportunidades que me permitiram chegar até aqui. Meu sincero reconhecimento também a todos os professores que contribuíram para minha formação, em especial ao meu orientador, Prof. Me. Kleber Manrique Trevisani, por sua orientação e apoio tanto neste trabalho quanto ao longo do curso.

“Nada na vida deve ser temido, apenas compreendido. Agora é a hora de compreender mais, para temer menos. ”

Marie Curie.

RESUMO

Este trabalho descreve detalhes sobre o desenvolvimento de um modelo de aprendizado de máquina para detecção de ataques de negação de serviço. As técnicas empregadas no desenvolvimento deste foram embasadas em uma revisão bibliográfica e na experiência de trabalhos relacionados recentes e relevantes. Os experimentos realizados utilizaram um conjunto de dados público e já existente, muito utilizado pelos trabalhos relacionados em questão. Contudo, para avaliar o desempenho do modelo em uma situação real, também foram utilizados dados capturados de ataques reais em um ambiente controlado. Por fim, são apresentados os resultados obtidos e o desempenho do modelo desenvolvido é comparado com o desempenho dos modelos descritos nos trabalhos relacionados.

Palavras-chave: Negação de Serviço; Segurança Cibernética; Aprendizado de Máquina.

Detection of Denial of Service Attacks Using Machine Learning

ABSTRACT

This work describes details about the development of a machine learning model for detecting denial-of-service attacks. The techniques employed in this development were based on a literature review and the experience of recent and relevant related works. The experiments conducted used an existing public dataset, widely utilized by the related studies in question. However, to evaluate the model's performance in a real-world scenario, data captured from real attacks in a controlled environment were also used. Finally, the obtained results are presented, and the performance of the developed model is compared with the performance of the models described in related works.

Keywords: Denial of Service; Dataset; Machine Learning; Attacks.

LISTA DE ILUSTRAÇÕES

Figura 1. Colunas dos dados brutos extraídos.....	22
Figura 2. Colunas dos dados transformados.....	22
Figura 3. Resultados do modelo utilizando dados do NSL-KDD.....	23
Figura 4. Resultados do modelo utilizando dados de ataques reais.....	24

LISTA DE TABELAS

Tabela 1. Resultados dos experimentos realizados por Taher et al (2019).....	16
Tabela 2. Resultados dos experimentos de Andrade, Santos e Freitas (2023).....	17
Tabela 3. Taxas de valores apresentados pelos classificadores.....	18
Tabela 4. Resultados dos experimentos de AL-Khassawneh (2023).....	18

SUMÁRIO

Capítulo 1 - Introdução.....	12
Capítulo 2 - Artigo Científico.....	13
1. Introdução.....	13
1.1 Metodologia.....	14
2. Trabalhos Relacionados.....	15
3. Desenvolvimento.....	19
3.1 Seleção do Conjunto de Dados.....	19
3.2 Criação do Conjunto de Dados Real.....	19
3.3 Seleção das Técnicas de Aprendizado de Máquina.....	20
3.4 Implementação do modelo de aprendizado de máquina.....	21
3.4.1 Preparação do conjunto de dados real.....	21
3.4.2 Desenvolvimento da técnica de Florestas Aleatórias.....	23
4. Considerações finais.....	25
Referências.....	26

Capítulo 1 - Introdução

Este documento apresenta um Trabalho de Conclusão de Curso (TCC) que é um projeto de pesquisa científica do curso superior de Bacharelado em Ciência da Computação. Além disso, tem-se como entrega final um artigo científico redigido de acordo com o modelo da Sociedade Brasileira de Computação (SBC) que será apresentado no Capítulo 2 deste documento.

Capítulo 2 - Artigo Científico

Detecção de Ataques de Negação de Serviço utilizando Aprendizado de Máquina

Igor Matheus Teixeira¹, Kleber Manrique Trevisani²

^{1,2}Instituto Federal de Ciência e Tecnologia de São Paulo (IFSP)
Presidente Epitácio – SP – Brasil.

i.teixeira@aluno.ifsp.edu.br¹, kleber@ifsp.edu.br²

Abstract. *This work describes details about the development of a machine learning model for detecting denial-of-service attacks. The techniques employed in this development were based on a literature review and the experience of recent and relevant related works. The experiments conducted used an existing public dataset, widely utilized by the related studies in question. However, to evaluate the model's performance in a real-world scenario, data captured from real attacks in a controlled environment were also used. Finally, the obtained results are presented, and the performance of the developed model is compared with the performance of the models described in related works.*

Resumo. *Este trabalho descreve detalhes sobre o desenvolvimento de um modelo de aprendizado de máquina para detecção de ataques de negação de serviço. As técnicas empregadas no desenvolvimento deste foram embasadas em uma revisão bibliográfica e na experiência de trabalhos relacionados recentes e relevantes. Os experimentos realizados utilizaram um conjunto de dados público e já existente, muito utilizado pelos trabalhos relacionados em questão. Contudo, para avaliar o desempenho do modelo em uma situação real, também foram utilizados dados capturados de ataques reais em um ambiente controlado. Por fim, são apresentados os resultados obtidos e o desempenho do modelo desenvolvido é comparado com o desempenho dos modelos descritos nos trabalhos relacionados.*

1. Introdução

A evolução tecnológica e a expansão da conectividade por meio da Internet provocou uma proliferação de dispositivos conectados em rede e uma grande dependência humana dos sistemas de informação e comunicação. No entanto, essa grande quantidade de sistemas computacionais interconectados também chamou a atenção de pessoas com interesses em subvertê-los. Desde então, os ataques cibernéticos têm aumentado em quantidade e complexidade, incluindo os ataques de Negação de Serviço, também conhecido como DoS (*Denial of Service*), causando prejuízos para pessoas e organizações que dependem dos sistemas atingidos por esses ataques.

Um ataque de DoS é uma atividade não autorizada que compromete um sistema computacional tornando-o indisponível para os usuários por meio da sobrecarga do sistema alvo com um volume excessivo de solicitações. A indisponibilidade de sistemas pode causar prejuízos a pessoas e instituições de diversos tipos, conforme relatado em (Menscher, 2020) e (Kottler, 2018). Por essa razão, a detecção de ataques DoS desempenha um papel fundamental

na proteção de recursos computacionais, pois pode contribuir para mitigar esse tipo de ciberataque de forma mais rápida e, conseqüentemente, diminuir os prejuízos causados.

De acordo com CERT.br (2025) , o Brasil registrou aproximadamente 106 mil notificações de ataques DoS entre 2023 e 2024, demonstrando que esses valores ressaltam a necessidade de aprimorar as técnicas de detecção e prevenção para lidar com ameaças cada vez mais sofisticadas.

Diferentes métodos têm sido utilizados para identificar ataques de DoS. Métodos tradicionais, baseados em assinaturas e com regras predefinidas, muitas vezes não conseguem acompanhar a evolução e a complexidade dos ataques modernos. Collins (2014), relata que os sistemas de detecção de intrusão baseados em assinatura apresentam uma alta taxa de falsos negativos, pois um grande número de ataques não é identificado, especialmente os ataques de dia-zero, que não possuem assinaturas disponíveis no momento de sua execução.

No contexto dos ataques de DoS, o aprendizado de máquina surge como abordagem interessante para realizar sua detecção. De acordo com (Xin, et al, 2018), o aprendizado de máquina é um ramo da inteligência artificial que está relacionado à estatística computacional, focando na predição utilizando computadores, com uma conexão com a otimização matemática, que fornece métodos e teorias aplicáveis ao campo. O pioneiro do aprendizado de máquina, Arthur Samuel, definiu como um campo de estudo que dá aos computadores a habilidade de aprender sem a necessidade de ser explicitamente programado (Xin, et al, 2018). Ele permite que um sistema de segurança consiga aprender a partir de dados que são fornecidos e, a partir disso, identificar padrões ou anomalias que possam indicar atividades maliciosas.

O objetivo geral deste trabalho foi selecionar e avaliar o desempenho de uma técnica de aprendizado de máquina na detecção de ataques de DoS. Para tanto, foi implementado um modelo de aprendizado de máquina utilizando a técnica selecionada, cuja seleção foi embasada em experiências relatadas em trabalhos relacionados relevantes. Para facilitar a comparação dos resultados deste trabalho com os resultados dos trabalhos relacionados, as métricas utilizadas para avaliação da implementação realizada também foram selecionadas considerando as métricas utilizadas por esses trabalhos.

Seguindo a lógica dos trabalhos relacionados, este trabalho também treinou o modelo de aprendizado de máquina implementado utilizando um conjunto de dados público e já existente. Contudo, a avaliação deste modelo, além de ser realizada utilizando parte do conjunto de dados selecionado, utilizou também dados provenientes de tráfego real de rede capturados em um ambiente controlado. O objetivo de utilizar dados reais, foi avaliar o desempenho do modelo implementado em uma situação real.

1.1 Metodologia

Para o desenvolvimento deste trabalho, inicialmente foi realizada uma revisão bibliográfica sobre aprendizado de máquina e detecção de intrusão em redes de computadores, com o objetivo de compreender os conceitos, métodos, desafios e técnicas relevantes na área. A partir dessa revisão, definiu-se o tipo específico de atividade maliciosa que seria detectada pelo modelo, sendo selecionada a detecção de ataques DoS. A partir disso, é feita a escolha de uma técnica de aprendizado de máquina para realizar a predição dos ataques, com essa escolha sendo embasada nos trabalhos relacionados.

Para a realização dos experimentos, foram selecionados ambientes de desenvolvimento e execução específicos para a geração e captura dos dados. A geração dos ataques foi realizada em uma máquina com sistema operacional Ubuntu, com a utilização da ferramenta Hping3 para disparar pacotes em larga escala. Já os computadores-alvo operavam com Windows 10 e foram configurados para capturar e armazenar os pacotes de dados utilizando o *software* Wireshark.

Com os ambientes configurados, foi selecionado um conjunto de dados público, que contém exemplos de tráfego de rede normais e de ataques. Em seguida, os dados foram submetidos a uma etapa de pré-processamento, para em seguida realizar a análise. Para isso, algumas técnicas de aprendizado de máquina foram escolhidas para avaliação, sendo elas selecionadas com base nos trabalhos relacionados, e baseadas em aprendizado supervisionado. As técnicas foram treinadas com o conjunto de dados de treinamento e passaram por um processo contínuo de ajustes, e em seguida, foram avaliadas utilizando métricas relevantes para a análise de seus desempenhos.

Após a experimentação, os resultados foram analisados, comparando o desempenho das técnicas avaliadas na detecção de intrusões em redes de computadores. Foram discutidas as vantagens e limitações de cada abordagem, bem como possíveis direcionamentos futuros para a pesquisa.

Além da Introdução, o presente trabalho está organizado em 3 seções. A seção 2 apresenta os trabalhos relacionados selecionados que embasaram as decisões tomadas para a realização deste trabalho. A seção 3 fornece detalhes sobre o processo de desenvolvimento do modelo de aprendizado de máquina. Já a seção 4 apresenta os resultados observados. Por fim, a seção 5 apresenta as considerações finais e sugestões de possíveis trabalhos futuros.

2. Trabalhos Relacionados

Devido a existência de diversas abordagens para realizar a detecção de ataques de negação de serviço utilizando aprendizado de máquina, cada uma com suas vantagens e desvantagens, foi realizado um levantamento sobre as abordagens mais utilizadas em trabalhos relacionados e recentes que pudessem auxiliar os autores em decidir quais delas seriam empregadas neste trabalho. Nesse contexto, foram estabelecidos critérios para garantir a relevância e a qualidade das fontes utilizadas, sendo a pesquisa conduzida utilizando bases de dados acadêmicas reconhecidas, como *IEEE*, *Science Direct* e *Google Scholar*. Além disso, foi estabelecido um período de publicação de até cinco anos (a partir de 2019), de modo a priorizar estudos e pesquisas atualizadas que refletissem os resultados mais relevantes na área. Ademais, somente foram selecionados trabalhos que abordassem diretamente o tema de técnicas de aprendizado de máquina e que apresentassem resultados embasados em experimentos ou análises detalhadas de trabalhos relacionados. A partir desse ponto esta seção apresenta uma análise sobre cada um dos trabalhos selecionados.

O estudo de Taher, Jisan e Rahman (2019) propõe um sistema de detecção de intrusões em redes baseado em técnicas de aprendizado supervisionado, enfatizando a relevância da seleção de atributos para aprimorar a acurácia dos classificadores. O método proposto integra algoritmos como Redes Neurais Artificiais e Máquinas de Vetor Suporte com técnicas de redução dimensional, utilizando tanto abordagens de filtro quanto de *wrapper*. Essa combinação visa extrair os atributos mais significativos dos dados, contribuindo para a

melhoria do desempenho na identificação de tráfego malicioso, conforme evidenciado pelo uso do conjunto de dados NSL-KDD.

A avaliação experimental foi conduzida com base na métrica de taxa de sucesso da acurácia, permitindo uma análise comparativa entre os diferentes modelos. Os resultados demonstraram que o modelo baseado em Redes Neurais Artificiais, quando associado à técnica *wrapper* de seleção de atributos, obteve uma acurácia de 94,02%, superando significativamente os modelos de Máquinas de Vetor Suporte, que apresentaram taxas em torno de 81,78% a 82,34%. Esses resultados ressaltam que a correta identificação e remoção de atributos redundantes ou irrelevantes podem potencializar a capacidade de aprendizado das técnicas apresentadas.

Além disso, o trabalho destaca a importância de se considerar a qualidade dos dados de treinamento e teste, evidenciando que o sucesso da detecção de intrusões não depende apenas da escolha do algoritmo, mas também da integridade e relevância dos conjuntos de dados utilizados. A implementação realizada no ambiente Weka possibilitou a realização de experimentos controlados, nos quais diferentes configurações de cada técnica foram testadas para que pudesse identificar a combinação ideal de parâmetros, contribuindo para a redução de falsos positivos e para o aprimoramento geral do sistema.

A Tabela 1 apresenta os resultados obtidos por Taher et al. (2019), as técnicas utilizadas e a quantidade de características utilizadas. Os experimentos foram conduzidos utilizando Redes Neurais Artificiais e Máquinas de Vetor Suporte, variando a quantidade de características selecionadas para o treinamento dos modelos. Foi observado que a melhor performance foi alcançada pelas Redes Neurais Artificiais utilizando uma quantidade de 17 características, atingindo uma acurácia de 94,02%. No entanto, ao utilizar 35 características, a acurácia dessa técnica foi reduzida para 83,68%, indicando que a seleção adequada de atributos pode melhorar significativamente o desempenho do modelo. Por outro lado, as Máquinas de Vetor Suporte apresentaram uma variação menor na acurácia, com 81,78% ao utilizar 17 características e 82,34% com 35 características, sugerindo que essa abordagem é menos sensível à quantidade de atributos escolhidos. Esses resultados destacam a importância da seleção de características no aprimoramento da detecção de intrusões, principalmente em modelos mais complexos, como Redes Neurais Artificiais e destacando a necessidade de encontrar a combinação ideal para aumentar a detecção de novos ataques.

Tabela 1. Resultados dos experimentos realizados por Taher et al (2019).

Técnica	Número de Características	Resultados
Redes Neurais Artificiais	17	94.02%
	35	83.68%
Máquinas de Vetor Suporte	17	81.78%
	35	82.34%

Fonte: Elaborada pelo autor.

O trabalho de Andrade, Santos e Freitas (2023) apresenta um estudo detalhado sobre a aplicação de técnicas de aprendizagem de máquina na detecção de ameaças em redes de computadores, com o objetivo principal de detectar intrusões e prevenir ataques cibernéticos por meio da análise de tráfego de rede. As técnicas de classificação utilizadas no referido trabalho foram as Árvores de Decisão, Tabelas de Decisão e Naive Bayes.

Os modelos desenvolvidos por Andrade, Santos e Freitas (2023) foram treinados e avaliados utilizando o conjunto de dados KDD Cup 99. Algumas métricas foram utilizadas para avaliar o desempenho desses modelos, incluindo acurácia, precisão, revocação e taxa de verdadeiros e falsos positivos e negativos. Além disso, a matriz de confusão foi empregada para compreender melhor o desempenho do modelo. Os resultados obtidos demonstraram taxas de acurácia, precisão e revocação acima de 89% para os três métodos de classificação avaliados. Além disso, o estudo analisou o tempo de execução de cada modelo, destacando as diferenças no custo computacional entre a Árvore de Decisão, Tabelas de Decisão e Naive Bayes. O estudo observou que as Árvores de Decisão apresentaram um tempo de execução computacional em milissegundos de 13,9%, enquanto o *Naive Bayes* apresentou um custo de 4,7% e por fim, as Tabelas de Decisão apresentaram o maior custo computacional, de 81,4%. A Tabela 2 apresenta as técnicas de aprendizado de máquina utilizadas por Andrade, Santos e Freitas (2023), as métricas utilizadas para avaliação de cada modelo, bem como os resultados obtidos para cada métrica, enquanto a Tabela 3 apresenta as taxas de verdadeiros positivos (VP) e negativos (VN) e falsos positivos (FP) e negativos (FN).

Tabela 2. Resultados dos experimentos de Andrade, Santos e Freitas (2023).

Técnica	Métricas	Resultados	Custo
Árvores de Decisão	Acurácia	99%	13,9%
	Precisão	99%	
	Revocação	99%	
Naive Bayes	Acurácia	91.01%	4,7%
	Precisão	89.97%	
	Revocação	99.94%	
Tabelas de Decisão	Acurácia	99.93%	81,4%
	Precisão	99.98%	
	Revocação	99.94%	

Fonte: Elaborada pelo autor.

Tabela 3. Taxas de valores apresentados pelos classificadores.

Técnica	VP	VN	FP	FN
Árvores de Decisão	396.723	97.271	6	20
Naive Bayes	396.542	53.074	44.203	201
Tabelas de Decisão	396.509	97.202	75	234
Total de amostras: 494.020				

Fonte:Elaborada pelo autor.

O trabalho de AL-Khassawneh (2023) faz um comparativo de técnicas de aprendizado de máquina para um sistema de detecção de intrusão. O estudo aplica as técnicas de Árvores de Decisão, Florestas Aleatórias, *K-Nearest Neighbors (KNN)* e Máquinas de Vetor Suporte (SVM) no conjunto de dados NSL-KDD, sendo a Acurácia, Precisão, Revocação, e *F1-Score* as métricas utilizadas para avaliação. O estudo observou que as técnicas baseadas em assinaturas são mais eficazes na detecção de ataques conhecidos, enquanto as técnicas baseadas em anomalias são mais adequadas para detectar ataques desconhecidos ou novos.

O estudo de AL-Khassawneh (2023) conclui que as Florestas Aleatórias podem acelerar significativamente os processos de treinamento e testes para a detecção de intrusão, visto que elas aumentam a precisão da classificação. Além disso, em comparação com as outras técnicas, elas possuem a menor taxa de erros de classificação e conseguem evitar problemas de *overfitting* (sobreajuste). A Tabela 4 consolida os resultados do trabalho de AL-Khassawneh (2023) associando-os às técnicas e as métricas utilizadas.

Tabela 4. Resultados dos experimentos de AL-Khassawneh (2023).

Técnicas	Métricas	Resultados
Florestas Aleatórias	Acurácia	99%
	Precisão	99.80%
	Revocação	98.70%
	F1-Score	96.60%
K Nearest Neighbors	Acurácia	97.70%
	Precisão	98.80%
	Revocação	98.40%
	F1-Score	95.60%

Fonte: Elaborada pelo autor.

O trabalho de Solanki, Gupta e Rai (2020) propõe uma abordagem híbrida para sistemas de detecção de intrusão baseados em aprendizado de máquina utilizando o conjunto de dados NSL-KDD. Nesse contexto, compara a precisão de diferentes técnicas de aprendizado de máquina sugerindo uma combinação de Máquinas de Vetor Suporte e outras

técnicas de aprendizado de máquina para melhorar a precisão do modelo. O estudo observou que as Máquinas de Vetor Suporte são eficazes na separação de tráfego normal e malicioso, mas podem não capturar as relações complexas entre as características dos dados. Por outro lado, as técnicas de aprendizado de máquina podem capturar essas relações complexas, mas podem não ser tão eficazes na separação de tráfego normal e de ataque. Ao combinar as Máquinas de Vetor Suporte e outras técnicas de aprendizado de máquina, a abordagem híbrida pode capturar tanto as relações simples quanto as complexas nos dados, levando a uma melhoria na precisão da detecção de intrusão.

3. Desenvolvimento

O desenvolvimento deste trabalho envolveu a seleção de um conjunto de dados já existente que contemplasse ataques de DoS, a criação de um conjunto de dados contendo dados de ataques DoS reais, a seleção de técnicas de aprendizado de máquina para implementação de um modelo capaz de identificar ataques de DoS. Além dessas atividades, essa seção também descreve detalhes sobre a implementação do modelo desenvolvido.

3.1 Seleção do Conjunto de Dados

O conjunto de dados selecionado para o treinamento do modelo de aprendizado de máquina foi o NSL-KDD, devido ao fato de ser amplamente utilizado em trabalhos relacionados e por possuir um acervo diverso de tráfego de rede normal e malicioso. Esse conjunto de dados inclui quatro categorias principais de ataques, sendo DoS, R2L (acesso remoto à máquina), U2R (acesso local à raiz) e *Probe* (varredura de rede). No entanto, para este trabalho, será utilizada apenas dados relacionados à categoria DoS, considerando os objetivos deste trabalho.

O NSL-KDD foi desenvolvido como uma melhoria ao KDD *Cup* 99, abordando algumas das limitações desse conjunto de dados, como a alta redundância e o desbalanceamento extremo entre as classes de ataque. Com isso, suas melhorias em relação ao KDD *Cup* 99 consistem na remoção de registros duplicados no conjunto de treinamento, evitando o enviesamento dos classificadores, ausência de registros duplicados no conjunto de teste, além do tamanho do conjunto de treino e teste ser razoável, permitindo a execução dos experimentos no conjunto completo, sem a necessidade de uma amostra reduzida, garantindo que os resultados das avaliações sejam consistentes e comparáveis entre diferentes pesquisas (Tavallaee, *et al*, 2009).

3.2 Criação do Conjunto de Dados Real

Após a seleção do conjunto de dados, o próximo passo foi gerar os ataques em um ambiente controlado, para serem capturados. Nesse sentido, foi utilizada a ferramenta *hping3*, que permite simular ataques DoS ao enviar um grande volume de pacotes configuráveis para um sistema alvo. O *hping3* é uma ferramenta de linha de comando especializada na geração e manipulação de pacotes de rede, oferecendo a capacidade de customizar diferentes parâmetros dos pacotes, como o cabeçalho IP e *flags* TCP, e sua utilização foi fundamental para gerar os ataques DoS e direcionar até os computadores alvo, através do IP da máquina. A seguir o comando utilizado para simular os ataques.

```
hping3 192.168.0.100 -S -p 80 --flood
```

O primeiro elemento do comando representa a ferramenta responsável pelo envio dos pacotes, enquanto em seguida, é definido o endereço IP de destino, na qual será o alvo dos pacotes. Após isso, o parâmetro “-S” indica que os pacotes enviados conterão a *flag* SYN ativada, representando que os pacotes serão de conexão TCP. Em seguida, o parâmetro “-p” indica a comunicação feita através do número da porta, nesse exemplo sendo a 80. E por fim, o parâmetro “- - flood “ ativa o modo de envio contínuo e acelerado de pacotes, sem a necessidade de aguardar respostas do destinatário, gerando um tráfego excessivamente alto, sobrecarregando o servidor e gerando a negação de serviço.

Após configurar a ferramenta, alguns computadores foram ligados para serem os alvos dos ataques e para monitorar e capturar o tráfego dos pacotes durante esses ataques, foi utilizado o *software Wireshark*, um programa que analisa o tráfego da rede e organiza por protocolos (Wireshark, 2025) . Com o intuito de armazenar os ataques e comprimi-los, foram utilizadas duas ferramentas complementares: uma ferramenta integrada ao *Wireshark*, o *Tshark* e uma biblioteca do Python chamada de *Scapy*. O *Tshark* permitiu filtrar e adicionar cabeçalhos relevantes aos pacotes capturados, além de salvar os pacotes da rede após a adição dos filtros. Já o *Scapy* foi utilizado para diminuir o tamanho do conjunto de dados bruto, visto que um ataque DoS gera milhares de pacotes por segundo, a utilização da ferramenta possibilitou comprimir para uma quantidade equivalente a três mil pacotes, o que facilitou a análise posteriormente.

Com os conjuntos de dados selecionados e criados, o próximo passo foi a escolha de uma técnica de aprendizado de máquina para realizar a análise dos pacotes.

3.3 Seleção das Técnicas de Aprendizado de Máquina

Na área de aprendizado de máquina, existe uma variedade de técnicas que são utilizadas para resolver problemas complexos. Diante disso, é essencial entender as características específicas de cada técnica e como elas se adequam aos problemas em questão. A seguir (Pereira, 2022) cita alguns modelos como Florestas Aleatórias, *K -Nearest Neighbor*, Máquinas de Vetor Suporte, *Naive Bayes* e Redes Neurais Profundas, por exemplo. Essas técnicas foram apresentadas com base em sua eficiência comprovada em tarefas de detecção de intrusões.

De acordo com AL-Khassawneh (2023), algoritmos como Florestas Aleatórias são amplamente utilizados devido à sua capacidade de lidar com grandes volumes de dados e sua alta precisão em problemas de classificação. As Máquinas de Vetor de Suporte por sua vez, são conhecidas por seu bom desempenho em espaços de alta dimensionalidade, o que é crucial em cenários de detecção de intrusões, onde os padrões de ataque podem ser complexos e não linearmente separáveis (Shaukat, *et al*, 2020).

Por outro lado, métodos baseados em vizinhos mais próximos, como o *K-Nearest Neighbor*, são eficazes em detectar anomalias em ambientes onde não há suposições claras sobre a distribuição dos dados (AL-Khassawneh, 2023). Já as Redes Neurais Artificiais e as Redes Neurais Profundas são particularmente eficazes quando há um grande volume de dados rotulados e a necessidade de aprender padrões complexos e não lineares, sendo organizadas em camadas: com uma camada de entrada para os dados, camadas ocultas para processar informações e uma camada de saída para classificar os resultados. As Redes Neurais Profundas, por sua vez, possuem mais camadas ocultas, permitindo identificar padrões mais detalhados e profundos, o que as torna ideais para analisar dados temporais ou espaciais, como mostrado por Pereira (2022).

Dessa forma, a seleção dessas técnicas foi fundamentada em suas capacidades comprovadas de lidar com os desafios específicos da detecção de ataques DoS, incluindo grandes volumes de dados, padrões complexos e a necessidade de detectar anomalias em tempo real.

Com base nessas informações, o início do desenvolvimento foi executado com a técnica do *K-Nearest Neighbor*, visto que essa abordagem tem sido utilizada devido à sua simplicidade e capacidade de classificar instâncias com base na proximidade dos dados de treinamento.

Contudo, durante os testes preliminares, os resultados obtidos com o *K-Nearest Neighbor* não foram satisfatórios, apresentando baixas taxas de acurácia e precisão na detecção dos ataques DoS. Esse desempenho inferior mostrou que, para essa implementação em específico, a técnica não era suficientemente robusta para lidar com a variabilidade e o volume dos dados, além de ser sensível a ruídos nas amostras. Em razão disso, a técnica de Florestas Aleatórias foi considerada como uma alternativa mais promissora.

As Florestas Aleatórias por sua vez, são mais robustas quando comparadas com o *K-Nearest Neighbors*, devido à sua capacidade de lidar com variáveis complexas, proporcionando uma tendência menor ao sobreajuste em grandes quantidades de dados quando comparada com outros métodos.

3.4 Implementação do modelo de aprendizado de máquina

O desenvolvimento do modelo foi realizado utilizando a ferramenta Anaconda, *software* que fornece um conjunto de funcionalidades e bibliotecas em um único ambiente. Através do Anaconda foi utilizado o Jupyter *Notebook*, uma interface que permite a criação de blocos de códigos interativos, permitindo combinar códigos, equações, visualizações gráficas e textos descritivos. Nesse contexto, para manipular os dados, foi utilizada a linguagem de programação Python, que é amplamente reconhecida por sua simplicidade e eficácia no processamento de dados, além de seu repertório de bibliotecas para análises de dados.

Além do Python, foram utilizadas bibliotecas para auxiliar na criação, no processamento e nas análises de dados: o NumPy é uma biblioteca que fornece suporte para *arrays* e matrizes, juntamente com uma grande coleção de operações matemáticas para operar com esses *arrays*. O Pandas, por sua vez, é uma biblioteca que oferece estruturas de dados e operações para manipular tabelas numéricas e séries temporais. Já o Scikit-Learn é uma biblioteca de aprendizado de máquina que oferece uma variedade de técnicas de classificação, regressão e agrupamento, além de oferecer funcionalidades de pré-processamento, métricas e técnicas de avaliação. Outra biblioteca utilizada foi Imbalanced-learn, utilizada para lidar com classes desbalanceadas, aplicando a técnica de SMOTE para equilibrar a proporção entre as classes. Por fim, as bibliotecas Seaborn e Matplotlib foram utilizadas para visualização de dados através de histogramas, gráficos e matrizes de confusão, permitindo interpretar os resultados de maneira mais simples.

3.4.1 Preparação do conjunto de dados real

Após a extração dos ataques na rede, foi necessário preparar os dados para serem avaliados pelo modelo. Para isso, primeiramente foi utilizada a biblioteca Scapy para rotular os dados extraídos da rede. Essa biblioteca foi utilizada para identificar e marcar características de cada pacote de dados, distinguindo o tráfego normal dos dados de ataque. O próximo passo

consistiu em transformar as colunas do conjunto de dados real para o mesmo formato das colunas do conjunto de dados do NSL-KDD, porém como ele possui quatro tipos de ataques, não foram todas as colunas que foram transformadas no conjunto de dados real, sendo utilizadas apenas aquelas que têm significado para os ataques DoS. A Figura 1 demonstra algumas colunas do conjunto de dados real antes das transformações feitas.

Figura 1. Colunas dos dados brutos extraídos.

	ip.src	ip.dst	ip.proto	ip.checksum	tcp.srcport	tcp.dstport	tcp.flags	frame.len	frame.time_epoch
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	60	1.717612e+09
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	42	1.717612e+09
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	60	1.717612e+09
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	60	1.717612e+09
4	10.117.77.115	13.85.23.86	6.0	0x0000	61713.0	443.0	0x0018	265	1.717612e+09
...
2995	10.117.77.120	10.117.77.115	6.0	0x6bbe	4850.0	80.0	0x0002	60	1.717612e+09
2996	10.117.77.119	10.117.77.115	6.0	0xa94a	3331.0	80.0	0x0002	60	1.717612e+09
2997	10.117.77.120	10.117.77.115	6.0	0xd617	4810.0	80.0	0x0002	60	1.717612e+09
2998	10.117.77.120	10.117.77.115	6.0	0xb960	4811.0	80.0	0x0002	60	1.717612e+09
2999	10.117.77.120	10.117.77.115	6.0	0x3087	4799.0	80.0	0x0002	60	1.717612e+09

Fonte: Elaborada pelo autor.

Durante as transformações, diversos métodos foram aplicados para tornar a base de dados compatível com o NSL-KDD, o que incluiu o mapeamento de portas de serviço e números de protocolo, a renomeação de colunas, o cálculo de métricas específicas, como taxas de erro de serviço e contagens de conexões recentes, entre outros. A figura 2 mostra as colunas após as transformações feitas.

Figura 2. Colunas dos dados transformados.

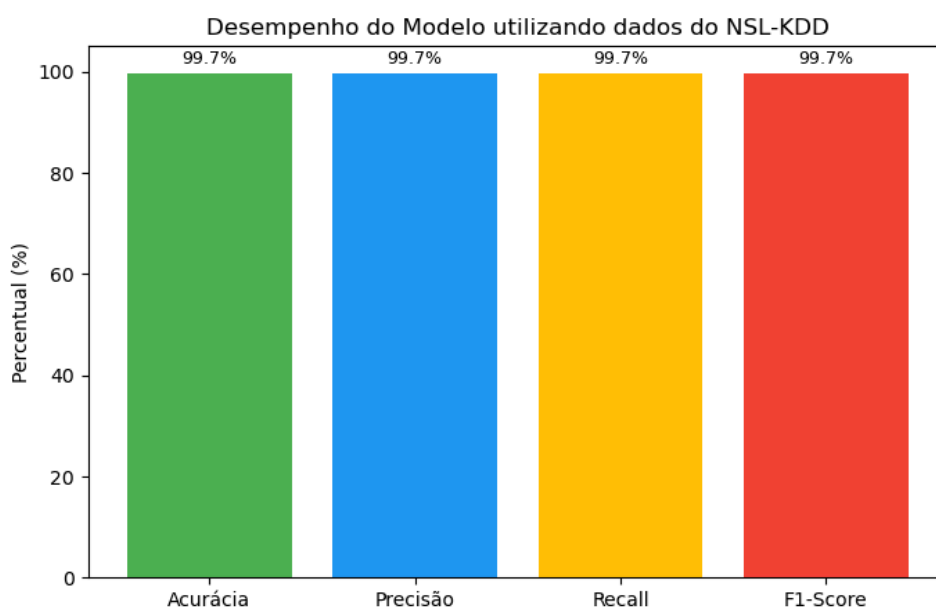
	src_ip	dst_ip	protocol_type	src_port	dst_port	frame_length	timestamp	duration	src_bytes	dst_bytes	...
0	NaN	NaN	Unknown	NaN	NaN	60	2024-06-05 18:22:09.767070976	NaN	NaN	NaN	...
1	NaN	NaN	Unknown	NaN	NaN	42	2024-06-05 18:22:10.803491072	NaN	NaN	NaN	...
2	NaN	NaN	Unknown	NaN	NaN	60	2024-06-05 18:22:10.804153856	NaN	NaN	NaN	...
3	NaN	NaN	Unknown	NaN	NaN	60	2024-06-05 18:22:11.305833216	NaN	NaN	NaN	...
4	10.117.77.115	13.85.23.86	tcp	61713.0	443.0	265	2024-06-05 18:22:11.314750976	0.00000	265.0	265.0	...
...
2995	10.117.77.120	10.117.77.115	tcp	4850.0	80.0	60	2024-06-05 18:22:14.509329152	0.02089	134040.0	134040.0	...
2996	236.103.38.209	10.117.77.115	tcp	3331.0	80.0	60	2024-06-05 18:22:14.509329152	0.00000	60.0	60.0	...
2997	10.117.77.120	10.117.77.115	tcp	4810.0	80.0	60	2024-06-05 18:22:14.509329152	0.02089	134040.0	134040.0	...
2998	10.117.77.120	10.117.77.115	tcp	4811.0	80.0	60	2024-06-05 18:22:14.509329152	0.02089	134040.0	134040.0	...
2999	10.117.77.120	10.117.77.115	tcp	4799.0	80.0	60	2024-06-05 18:22:14.509331968	0.02089	134040.0	134040.0	...

Fonte: Elaborada pelo autor.

3.4.2 Desenvolvimento da técnica de Florestas Aleatórias

Para o desenvolvimento do modelo, primeiramente foi necessário pré-processar os dados do NSL-KDD, removendo colunas irrelevantes para a técnica realizar o processamento, escalonando os dados, e dividindo o conjunto de dados em treino e teste. Após isso foi utilizada a técnica de SMOTE, que cria amostras sintéticas da classe minoritária para tratar o desbalanceamento das classes, evitando com que o modelo se tornasse enviesado para a classe de maior pertinência. O próximo passo consistiu em realizar uma busca pelos melhores hiperparâmetros possíveis para as florestas aleatórias utilizando a técnica de *Grid Search*, que consiste em testar diferentes valores para cada parâmetro, como profundidade das árvores e número de estimadores, retornando o valor ideal para o modelo. Em seguida, foi realizado o treinamento do modelo e foram utilizadas métricas de avaliação como Acurácia, Precisão, f1-score e revocação para avaliar o desempenho do modelo, a figura 3 apresenta os resultados do algoritmo ao ser treinado com os dados do NSL-KDD em formato gráfico.

Figura 3. Resultados do modelo utilizando dados do NSL-KDD.



Fonte: Elaborada pelo autor.

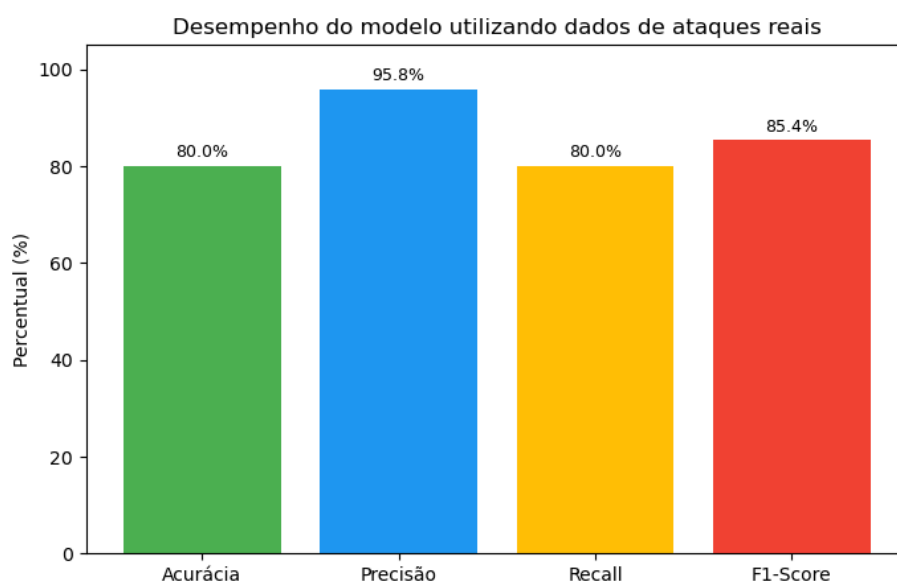
A figura revela que o modelo alcançou um desempenho excepcional ao utilizar os dados do NSL-KDD, com todas as métricas atingindo aproximadamente 99,7%, indicando que o modelo foi altamente eficaz para fazer a distinção entre tráfego normal e anômalo, minimizando tanto os falsos positivos quanto os falsos negativos. A alta precisão sugere que a maioria das predições de ataques feitas pelo modelo estavam corretas, enquanto a revocação igualmente elevada indica que quase todos os ataques foram devidamente identificados. O F1-score por sua vez, que equilibra precisão e revocação, reforça a consistência do modelo. Esses resultados demonstram um desempenho sólido, indicando que a abordagem foi eficiente na detecção dos ataques.

Após o modelo ser treinado e testado com o NSL-KDD, o próximo passo foi replicar o mesmo pré-processamento feito anteriormente para o conjunto de dados real e em seguida avaliar os resultados com a métricas, porém o resultado não foi satisfatório devido à oscilação do modelo, que a cada execução trazia resultados com valores diferentes, gerando diferentes níveis de desempenho, enquanto no treinamento os valores chegavam a 99% de previsibilidade sem variação. Isso sugeriu que o modelo estava instável em seus valores preditivos, com resultados que não poderiam ser confiáveis para serem analisados.

Para contornar isso, foi utilizado um conjunto de diversas técnicas como *pipelines*, curvas de aprendizado, e validação cruzada. A técnica de validação cruzada foi utilizada juntamente com o *Grid Search* que, ao invés de avaliar as combinações dos parâmetros em apenas uma divisão dos dados, aplica uma validação cruzada para cada combinação de parâmetros, ou seja, para cada conjunto de hiperparâmetros, o modelo é treinado e validado em múltiplas partições de dados, resultando uma média com menos variações para estabilizar o modelo. Utilizando essa abordagem, foi possível encontrar uma configuração de parâmetros mais consistente e robusta para essa implementação, o que reduziu risco de sobreajuste. Os *pipelines*, por sua vez, foram utilizados para simplificar o fluxo do pré-processamento e do treinamento, integrando as tarefas de balanceamento e transformação dos dados, o que acabou reduzindo o risco de erros manuais na organização das etapas. Por fim, as curvas de aprendizado foram geradas para visualizar o desempenho do modelo conforme o tamanho do conjunto de treinamento aumentava, o que permitiu identificar se o modelo sofria de sobreajuste, sendo possível ajustá-lo de maneira mais precisa e confirmar se os dados de treinamento estavam com um desempenho estável.

Utilizando essas abordagens, foi possível estabilizar os resultados do tráfego real detectado pelo modelo, elevando significativamente para valores próximos de 80% de acurácia e 96% de precisão. A figura 4 demonstra os resultados obtidos em formato gráfico.

Figura 4. Resultados do modelo utilizando dados de ataques reais.



Fonte: Elaborada pelo autor.

A figura apresentada revela que a acurácia geral do modelo é de aproximadamente 80%, indicando um desempenho satisfatório ao considerar todas as predições, tanto de ataques quanto de tráfego normal. A precisão de 95,8%, por sua vez, mostra que a maioria dos ataques detectados pelo modelo foram corretos, o que acaba reduzindo a ocorrência de falsos positivos. No entanto, a revocação de 80% indica que alguns ataques não foram identificados, resultando em possíveis falsos negativos. Por fim, o F1-Score reflete um equilíbrio razoável entre precisão e revocação, tornando o modelo adequado para a tarefa proposta.

4. Considerações finais

Inicialmente este trabalho propôs analisar duas técnicas de aprendizado de máquina quanto à sua eficácia na detecção de ataques DoS, utilizando conjuntos de dados públicos para essa avaliação. Conforme o desenvolvimento do trabalho ocorria, os objetivos previstos foram alterados, e de duas técnicas de aprendizado de máquina alternou-se para apenas uma, porém com a criação de um conjunto de dados próprio, gerado em um ambiente controlado para que o modelo pudesse realizar a análise e assim verificar o seu comportamento em um ambiente de tráfego real. A solução proposta visa avaliar as técnicas de aprendizado de máquina em condições de tráfego real, o que proporciona uma avaliação mais precisa de seu desempenho, quando comparadas com modelos testados apenas em conjuntos de dados sintéticos.

Este trabalho apresenta uma contribuição científica relevante para o campo da segurança cibernética, com foco na detecção de intrusões, especificamente nos ataques DoS, em redes de computadores. A implementação da técnica de Florestas Aleatórias visa abordar um dos grandes problemas da área, sendo a identificação eficiente e precisa de ataques em redes de computadores, que comprometem a integridade de sistemas e serviços.

Em comparação com os trabalhos relacionados, esta implementação utilizou um modelo de aprendizado de máquina que se destacou entre os estudos e gerou resultados compatíveis com os intervalos apresentados, porém com a distinção de ser treinado por um conjunto de dados público e avaliado por um conjunto de dados simulado com tráfegos reais, reforçando que o modelo foi capaz de fazer previsões e atingir um nível de satisfação positivo por não se enviesar pelos dados de treinamento.

Além disso, o trabalho passou por algumas limitações, como a criação de um conjunto de dados focado apenas em ataques DoS, a dificuldade em utilizar diferentes técnicas de processamento para melhorar o desempenho do modelo e a limitação em escolher apenas uma técnica de aprendizado de máquina para ser avaliada.

Por fim, este trabalho apresenta uma gama de possibilidades para futuras pesquisas que podem ser realizadas, como a ampliação do conjunto de dados de ataque, incluindo outros tipos de ataques além dos DoS, aumentando a área de intrusões que o modelo pode detectar, e a adoção de técnicas de aprendizado de máquina não supervisionadas, onde os dados não requerem rótulos para serem analisados. Além disso, outra possibilidade seria explorar a implementação de um sistema de detecção de intrusão com a utilização de grupos homogêneos, visando garantir que os dados utilizados para treinar os modelos de aprendizado de máquina atendam a requisitos de qualidade e consistência.

Referências

- AL-Khassawneh, Yazan. A. **An investigation of the Intrusion detection system for the NSL-KDD dataset using machine-learning algorithms**. In: IEEE International Conference On Electro Information Technology(eIT), 2023, Romeoville, IL, USA. Anais... IEEE, 2023. p. 518-523. Disponível em: <https://doi.org/10.1109/eIT57321.2023.10187360>. Acesso em: 18 nov. 2023.
- ANDRADE, Matheus. S; SANTOS, Jean.; FREITAS, Jonathan. **Sistema de detecção de intrusão utilizando métodos de aprendizagem de máquina em redes de computadores**. Revista de Ciência e Inovação, v. 9, n. 1, p. 22, 6 dez. 2023.
- CERT.br – **Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil**. Estatísticas de Incidentes Notificados ao CERT.br. Disponível em: <https://stats.cert.br/incidentes/#tipos-incidente>. Acesso em: 26 fev 2025.
- COLLINS, Michael. **Network Security Through Data Analysis**. O'Reilly Media, 2014.
- KOTTLER, Sam. **February 28th DDoS Incident Report**. Disponível em: <https://github.blog/news-insights/company-news/ddos-incident-report/>. Acesso em: 24 fev. 2025.
- MENSCHER, Damian. **Identifying and protecting against the largest DDoS attacks**. Disponível em: <https://cloud.google.com/blog/products/identity-security/identifying-and-protecting-against-the-largest-ddos-attacks>. Acesso em: 24 fev. 2025.
- PEREIRA, Lucas Fauster L. **Detecção de intrusão em rede por aprendizado de máquina distribuído**. 2022. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Departamento de Ciência da Computação, Universidade Federal Fluminense[S.I.],2022.Disponível em:<https://app.uff.br/riuff/bitstream/handle/1/27941/versão%20final-%20Lucas%20Fauster.pdf>. Acesso em: 19 ago. 2023.
- SHAUKAT, Kamran; LUO, Suhuai; VARADHARAJAN, Vijay; HAMEED, Ibrahim A; XU, Min. **A survey on machine learning techniques for cyber security in the last decade**. IEEE access: practical innovations, open solutions, v. 8, p. 222310–222354, 2020.
- SOLANKI, Surbhi; GUPTA, Chetan; RAI, Kalpana. **A Survey on Machine Learning based Intrusion Detection System on NSL-KDD Dataset**. International Journal of Computer Applications, [S.I.], v. 176, n. 30, p. 36-39, jun. 2020. Disponível em: <https://www.ijcaonline.org/archives/volume176/number30/solanki-2020-ijca-920343.pdf>. Acesso em: 22 nov. 2023.
- TAHER, Kazi Abu; JISAN, Billal Mohammed Yasin; RAHMAN, Md. Mahbubur. **Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection**. In: 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST). [S.I. : s.n.], 2019. p. 643-646. Disponível em: <https://ieeexplore.ieee.org/document/8644161>. Acesso em: 02 mar. 2025.
- TAVALLAEE, M; BAGHERI, Ebrahim; LU, Wei; GHORBANI, Ali A. **A detailed analysis of the KDD CUP 99 data set**. 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, jul. 2009.
- WIRESHARK. **Wireshark Network Protocol Analyzer**. Disponível em: <https://www.wireshark.org/>.

XIN, Yang; KONG, Lingshuang; LIU, Zhi; CHEN, Yuling; LI, Yanmiao; ZHU, Hongliang; GAO, Mingcheng; HOU, Haixia; WANG, Chunhua. **Machine Learning and Deep Learning Methods for Cybersecurity**. IEEE Access, v. 6, p. 35365-35381, 2018. Disponível em: <https://doi.org/10.1109/ACCESS.2018.2836950>. Acesso em: 26 fev. 2025.