Estudo de Caso de Identificação de Invasão DoS Utilizando Técnicas de Aprendizado de Máquina

Identification of DoS Intrusion Using Machine Learning Techniques Case Study

João Victor Cardoso[1]
orcid.org/0009-0001-9790-1109

Vladimir Homobono[1]
orcid.org/0000-0001-5727-2427

Escola Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil.
E-mail: jvces@poli.br

Artigo recebido em:
Artigo aceito em:

DOI:

Como citar este artigo pela NBR 6023/2018: AUTOR 1; AUTOR 2; AUTOR 3. Título do artigo. Revista de Engenharia e Pesquisa Aplicada, Recife, v. , n. , p. , mês, ano. DOI: Disponível em: . Acesso em: .

RESUMO

Este trabalho propõe um estudo sobre a detecção de ataques cibernéticos do tipo DoS, comparando diferentes métodos de aprendizado de máquina. A base para este estudo é a análise de performance dos algoritmos: Regressão Logística, Florestas Aleatórias, Máquinas de Vetores de Suporte, K-Means e Perceptron Multi-camadas frente ao dataset NSL-KDD que contém dados reais de variados tipos de comportamentos medidos em redes classificados como normais e maliciosos. As metas estabelecidas envolvem o pré-processamento dos dados de tráfego, criando assim conjuntos de treinamento confiáveis e de qualidade, a identificação de parâmetros importantes na detecção deste tipo de intrusão e o que a difere das demais, além da classificação de cada algoritmo utilizando as métricas de Acurácia, Precisão, Recall e F1-Score. Os resultados obtidos nesta análise caracterizam contribuições importantes para a área de cibersegurança e de dados, além de levantar pontos de melhoria para novos estudos, frente à evolução vigente dos algoritmos de inteligência artificial e a gama de possibilidades futuras neste campo de atividade.

PALAVRAS-CHAVE: Aprendizado de Máquina; Cibersegurança; Detecção de Intrusão; DoS

ABSTRACT

This work proposes a study on the detection of DoS-type cyber-attacks, comparing different machine learning methods. The basis for this study is the performance analysis of the algorithms: Logistic Regression, Random Forests, Support Vector Machines, K-Means and Multi-layer Perceptron against the NSL-KDD dataset that contains real data from different types of behaviors measured in networks classified as normal and malicious. The established goals involve the pre-processing of traffic data, thus creating reliable and quality training sets, the identification of important parameters in detecting this type of intrusion and what differentiates it from others, in addition to the classification of each algorithm using the Accuracy, Precision, Recall and F1-Score metrics.
The results obtained in this analysis characterize important contributions to the area of cybersecurity and data, in addition to raising points for improvement for new studies, given the current evolution of artificial intelligence algorithms and the range of future possibilities in this field of activity.

KEY-WORDS: Machine Learning; Cybersecurity; Intrusion Detection; DoS

1 INTRODUÇÃO

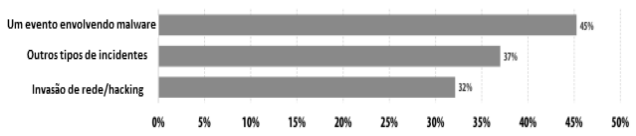
Com a evolução tecnológica exponencial e a democratização do acesso à internet, hoje temos um cenário mundial de grande disponibilidade de dados, um dos ativos mais valiosos da atualidade, por conseguinte é notável a crescente sofisticação dos ataques cibernéticos, utilizando desde métodos mais disseminados de engenharia social até algoritmos robustos de inteligência artificial, como o deepfake, por exemplo, para obter ganhos mediante dados sensíveis de uso pessoal e até corporativo. Devido a estes fatores, a segurança de redes de computadores é um desafio constante, já que os métodos tradicionais de detecção de intrusão podem não acompanhar a alta volatilidade do mercado e não ser eficazes neste novo cenário. A identificação de novas ameaças deve ser cada vez mais ágil e inteligente, visando acompanhar a evolução destes ataques. Neste contexto, o uso de técnicas de aprendizado de máquina tem-se mostrado bastante atrativo e promissor no combate a esses riscos, já que são métodos inteligentes e bastante adaptativos, permitindo que estes sistemas de detecção evoluam conforme novos ataques são desenvolvidos, proporcionando uma abordagem dinâmica e resiliente.

1.1 RELEVÂNCIA DO TEMA

Neste contexto em que a interconectividade digital se tornou a essência da nossa vida cotidiana e das operações organizacionais, os ataques DoS tornaram-se uma ferramenta eficaz e muito comum para desestabilizar serviços críticos, segundo Saravanan e Bama **[1]**, 24% das empresas sofreram um ataque deste tipo no ano de 2018. A falta de identificação e mitigação rápida destas intrusões afeta não só a disponibilidade dos dados, mas também a sua integridade e confidencialidade, impactando diretamente a confiança dos utilizadores e a reputação das entidades afetadas.

Além disso, a grande maioria dos ataques DoS são coordenados com vários outros tipos de ataques, conforme mostra a figura 1, justamente para explorar a fragilidade do sistema.

Figura 1 – Tipos de ataques em conjunto ao DoS



Fonte: [2]

Desta forma, identificar um ataque de negação de serviço se torna essencial para a prevenção de ataques maiores e, que trazem perdas ainda mais significativas à rede.

1.2 PROPOSTA

Este artigo se destaca pela abordagem prática e aplicada, através da realização de estudos de caso para verificar a eficácia da tecnologia proposta num ambiente controlado e avaliar a sua implementabilidade em situações da realidade. Isto ajudará a construir conhecimento aplicável e fornecerá informações valiosas para profissionais, pesquisadores e formuladores de políticas de segurança cibernética.

Deste modo este estudo foca em realizar análises utilizando a base de dados NSL KDD, pré-processar os dados contidos na base, para obter os melhores resultados e evitar análises enviesadas, e validar cada tipo de algoritmo de aprendizado de máquina selecionado junto aos parâmetros de avaliação escolhidos.

No decorrer deste trabalho serão apresentados os conceitos necessários para a aplicação dos modelos propostos, além da explanação do passo a passo praticado para chegar nos resultados alcançados.

2 APRENDIZADO DE MÁQUINA (AM)

Segundo Mitchell **[3]**, um computador aprende com a experiência "E" a respeito de alguma classe de tarefas "T" e desempenho medido por "P", se seu desempenho nas tarefas em "T", conforme medido em "P", melhora com a experiência "E", portanto se pode dizer que o aprendizado de máquina, que é um subcampo da inteligência artificial, se concentra no desenvolvimento de modelos computacionais que podem aprender padrões e tomar decisões sem serem explicitamente programados. A abordagem básica do AM é expor os sistemas aos dados para que o aprendizado de padrões seja automático, utilizando funções estatísticas e analisando dependências e relacionamentos entre as variáveis, e generalizar para dados novos. Essa adaptabilidade é de particular relevância para a detecção de ameaças cibernéticas, onde formas perigosas podem desenvolver-se rapidamente.

No contexto da segurança cibernética, a AM tem sido aplicada em diversas áreas, incluindo detecção de invasões, análise do comportamento do usuário

e identificação de ameaças em tempo real. Algoritmos de classificação, regressão e agrupamento são frequentemente usados para classificar dados, identificar anomalias e prever comportamentos suspeitos.

Podemos subdividir os modelos de algoritmos de aprendizado de máquina em cinco grupos principais: aprendizado supervisionado, não-supervisionado, deep learning, por reforço, semi-supervisionado. Para este trabalho, vamos focar nos três primeiros (supervisionado, não-supervisionado e deep learning).

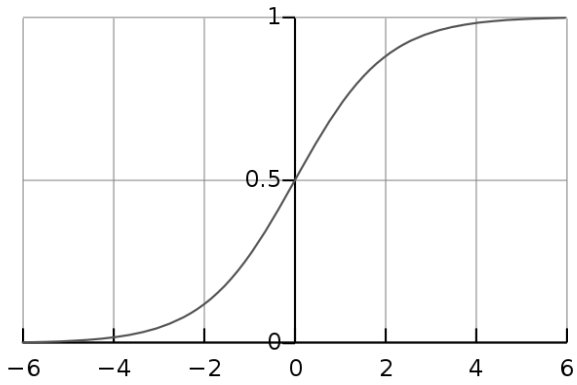
2.1 *APRENDIZADO SUPERVISIONADO*

O aprendizado supervisionado é um método em que o algoritmo é treinado com uma base que contém dados devidamente rotulados de entradas e suas respectivas saídas, parafraseando Bishop **[4]**, os dados de treinamento têm exemplos dos vetores de entrada juntamente com seus vetores alvo correspondentes, e visa aprender a relação entre as variáveis para que o modelo faça novas predições ou classificações.

2.1.1 Regressão Logística

A regressão logística é um método estatístico usado para classificação binária, onde a variável de saída é uma variável categórica com dois resultados possíveis, geralmente rotulados como 0 e 1. Apesar do nome, a regressão logística é um algoritmo de classificação e não um algoritmo de regressão. Utiliza-se da função sigmoide mostrada na figura 3 para modelar a probabilidade do resultando estar entre 0 e 1 conforme o limiar definido.

Figura 2 – Função Sigmóide



Fonte: O autor

$$\emptyset(z) = \frac{1}{1+e^{-z}} \quad \textbf{(1)}$$

$$z = w^T x = w_0 + w_1 x + w_2 x_1 + \cdots + w_n x_{n-1} \quad \textbf{(2)}$$

A fórmula 2 demonstra que z representa a combinação linear dos dados de entrada x (denominados variáveis preditoras) com os pesos w, os quais são determinados pelo modelo para maximizar a verossimilhança em relação aos resultados conhecidos. Essa combinação linear é então submetida à função sigmoide para o cálculo da probabilidade final..

2.1.2 Florestas Aleatórias (Random Forests)

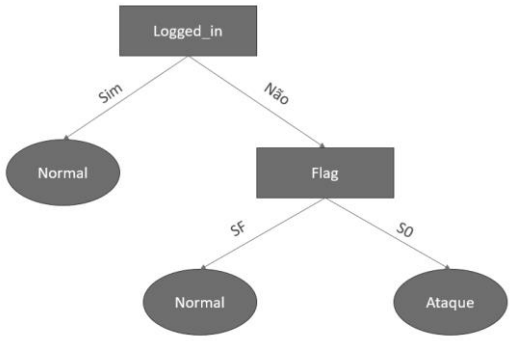
Este algoritmo é, na verdade, um conjunto de algoritmos de Árvores de Decisão (Decision Trees) que são modelos de aprendizado supervisionados. As DTs são métodos que se assemelham a um fluxograma, pois são feitos testes no atributo (característica presente na base de dados) em questão, cada nó representa uma verificação, cada ramo um resultado possível do teste feito e cada folha seria o resultado (valor numérico ou categórico) **[3]**.

Na figura 4 temos uma representação gráfica das DTs, com o exemplo da base de dados utilizada por este estudo, o nó raiz realiza o teste da característica "Logged_in" do dataset que indica se o indivíduo realizando a conexão está conectado ou não na rede, e em seguida, através dos ramos (resultados) os nós secundários realizam outra etapa de verificação na coluna "Flag", que retorna uma indicação se a conexão foi completada com sucesso (S0) ou se existiu algum erro (SF), até que os nós terminem nas folhas que indicam o resultado daquela predição.

Portanto, os algoritmos de Florestas Aleatórias, são o agrupamento de vários modelos mais fracos de Árvores de Decisão com atributos distintos durante a etapa de treinamento e têm o resultado da classificação ou predição definido por uma técnica de votação e distribuição de pesos para as Árvores geradas.

DOI: 10.xxxx/s11468-014-9759-3

Figura 3 – Exemplo de árvore de decisão



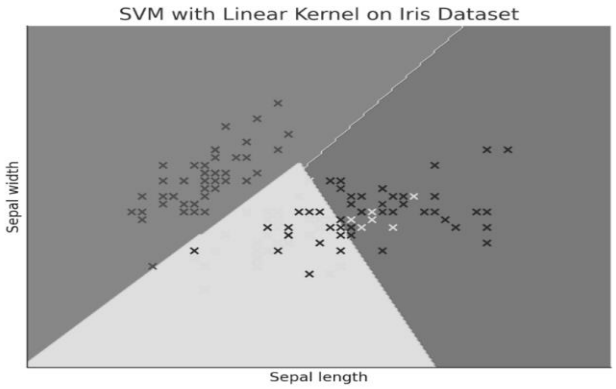
Fonte: O autor

2.1.3 Máquina de Vetores de Suporte (SVM)

O Support Vector Machine (SVM) é um algoritmo de aprendizado de máquina supervisionado usado principalmente para tarefas de classificação de conjuntos de dados, mas também pode ser empregado para regressão. Sua ideia fundamental é encontrar o melhor limite (ou hiperplano) que separa as classes de dados em um espaço de características, conforme bem explicado por Lorena **[5]**, o objetivo do algoritmo é maximizar a distância dos dados mais próximos de cada lado do hiperplano, estes dados são chamados de Vetores de Suporte.

Na figura 5 podemos ver um exemplo de algoritmo SVM utilizando a base de dados de flores íris, nela temos dados referentes a largura e comprimento das sépalas, as cores de fundo são definidas pelos limites impostos pelo hiperplano, classificando cada espécie de acordo com as características das sépalas.

Figura 4 – Exemplo de SVM



Fonte: O autor

2.2 APRENDIZADO NÃO-SUPERVISIONADO

Algoritmos de aprendizado não supervisionado são uma classe de técnicas de aprendizado de máquina usadas para descobrir padrões ocultos ou estruturas intrínsecas em dados não rotulados. Em contraste com os algoritmos de aprendizado supervisionado, os algoritmos de aprendizagem não supervisionada inferem padrões sem referência a resultados conhecidos ou rotulados, por não terem uma base de treinamento prévia. Um dos tipos mais comuns de algoritmos não supervisados são os de agrupamento (clustering), esse tipo de algoritmo identifica grupos de indivíduos que são similares uns aos outros, também presente na obra de Bishop **[4]**.

2.2.1 K-Means

K-Means é um dos algoritmos de cluster amplamente utilizados em aprendizado de máquina não supervisionado. É conhecido por sua simplicidade e eficiência, especialmente ao agrupar grandes conjuntos de dados. Nele coloca-se como entrada o número de agrupamentos (clusters) de dados pretendido, que pode ser obtido por métodos como o Elbow method, a partir disso criam-se centroides em diversos pontos aleatórios do espaço dimensional correspondente aos dados.

O algoritmo calcula a menor distância entre cada ponto de dado e o centroide mais próximo. Ao descobrir o centroide mais próximo, este ponto de dado será associado a ele. Após contabilizar todos os pontos de um agrupamento, o algoritmo ajustará a posição do centroide do agrupamento conforme a média dos pontos que o compõem. Se houver um novo ponto pertencente a esse grupo, por conta do ajuste do centroide, este será adicionado. O algoritmo irá parar quando não houver mais ajustes de centroides e nem novas ligações entre amostras de dados e grupos **[6]**.

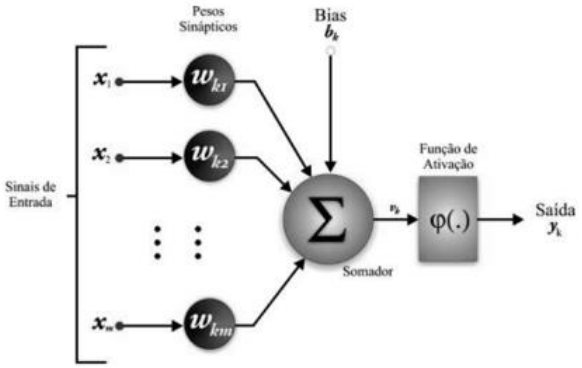
2.3 REDES NEURAIS ARTIFICIAIS (RNA)

Em sua essência, uma Rede Neural Artificial é um modelo computacional inspirado na forma como as redes neurais biológicas no cérebro humano processam informações. Segundo Sawyer **[7]**, O verdadeiro poder destes sistemas, porém, está na forma como a rede adquire esta capacidade de prever o resultado, a capacidade de automodificar

o cálculo com base no treino com dados nos quais a variável de saída é conhecida.

Um dos componentes principais de uma Rede Neural é o Neurônio, esta unidade basicamente simula um neurônio biológico, que processa informações e passa adiante para o próximo neurônio da cadeia de forma não-linear e paralela, como pode ser visto pela figura 7, ele é composto pelo valor de entrada "x" e pelo peso correspondente "w", sempre que um valor passa pelo neurônio é multiplicado pelo peso, a rede "aprende" modificando seus pesos. Além disso, introduz-se um terceiro elemento chamado de viés (bias), que atua como um tipo especial de peso. Esse elemento permite que o neurônio ajuste seu resultado independente dos dados de entrada, contribuindo assim para a flexibilidade do modelo em capturar padrões nos dados.

Figura 5 – Modelo de um neurônio

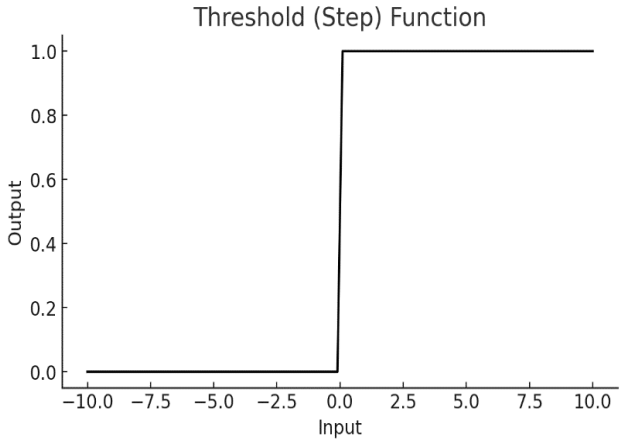


Fonte: [8]

Outra característica muito importante das RNAs são as chamadas Funções de Ativação, elas que definem qual será o valor de saída daquele neurônio conforme o valor recebido na sua entrada. Existem vários tipos de funções de ativação, as principais sendo:

- **Função Limiar**: com este tipo de função, a saída do neurônio assume o valor de 0 para valores de entrada negativos e 1 para valores de entrada positivos. A representação gráfica desta função encontra-se na figura 8.

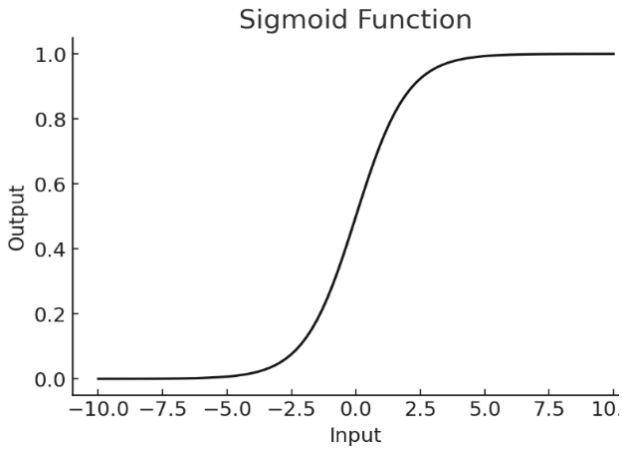
Figura 6 – Função Limiar



Fonte: O autor

- **Função Sigmoide:** neste caso, os valores de saída são contínuos, mas variam entre 0 e 1, como mostrado na figura 9, trazendo uma característica não linear ao resultado de saída, muito utilizado no cálculo de probabilidades.

Figura 7 – Função Sigmoide

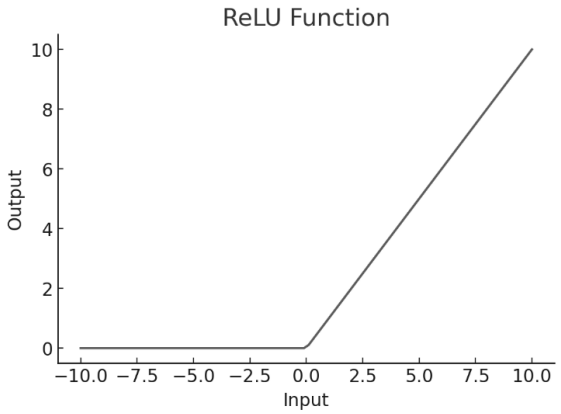


Fonte: O autor

- **Função ReLU:** aqui a saída do neurônio é a própria entrada, caso este valor seja positivo e caso seja negativo, o valor de saída é 0 (figura 10).

DOI: 10.xxxx/s11468-014-9759-3

Figura 8 – Função ReLU



Fonte: O autor

2.3.1 Perceptron Multi-camadas (MLP)

No campo das Redes Neurais Artificiais (RNAs), os Perceptrons Multicamadas (MLP) ocupam um lugar importante. Muitas vezes considerados os alicerces da aprendizagem baseada em redes neurais, os MLPs são um tipo específico de RNA em que existem 3 tipos de camadas, a camada de entrada, camadas ocultas e camada de saída, todos os neurônios de uma camada são conectados com todos os neurônios da próxima camada **[4]**. As camadas de entrada propagam os dados para as camadas ocultas onde acontecem os cálculos dos pesos dos neurônios e então os dados são propagados para a camada de saída onde temos os resultados calculados. Este processo é chamado de propagação direta.

Na etapa de treinamento o MLP utiliza o algoritmo de retro propagação para minimizar os erros da rede, nesta etapa calculam-se os erros dos resultados obtidos, comparando-os com os resultados esperados, utilizando uma função de perda, como o erro quadrático médio e a entropia cruzada, a partir desta informação os pesos dos neurônios são recalculados utilizando a regra delta, repetindo o processo até que o nível do erro calculado atinja um limiar aceitável, tornando a rede exclusivamente de propagação direta.

3 DETECÇÃO DE INTRUSÃO E SEGURANÇA DE REDES

No cenário em evolução da tecnologia digital, os dados vêm se tornando ativos cada vez mais valorosos, tanto em termos corporativos quanto pessoais. Empresas podem falir e famílias podem ser destruídas pelo simples comprometimento de algum tipo de informação presente no mundo online. Por esse motivo, a segurança cibernética permanece como um pilar crítico que protege a informação, a privacidade e a integridade dos sistemas em todo o mundo. No cerne deste campo está a tarefa imperativa de detectar e mitigar uma infinidade de ameaças cibernéticas. Este capítulo investiga o domínio da segurança cibernética e dos métodos de detecção de intrusões, além das ameaças cibernéticas atuais, com foco nos ataques DoS.

3.1 DETECÇÃO DE INTRUSÃO

Como falado anteriormente, a detecção de intrusões é uma das barreiras iniciais para variados tipos de ataques em nossos sistemas, este tipo de tarefa refere-se à capacidade de identificar e alertar sobre atividades maliciosas na rede, podendo reforçar os três pilares da cibersegurança, a confidencialidade, integridade e disponibilidade.

3.1.1 Detecção por Anomalias

Existem várias formas e rotinas que podem ser vistas como métodos de detecção de intrusão, aqui vamos focar em duas principais. A detecção por anomalias trabalha com um perfil pré-definido de comportamento que pode ser considerado normal, qualquer tipo de ação que fuja desse escopo pode ser enquadrada como maliciosa.

Sendo assim, parafraseando Lee [10], este tipo de detecção funciona bem com ataques que nunca foram registrados anteriormente, porém, em alguns casos, podem ser gerados falsos positivos, já que podem existir comportamentos que são considerados anômalos de usuários não-maliciosos e vice-versa.

3.1.2 Detecção por Assinatura

Neste método a detecção é feita por meio de parâmetros já conhecidos pela rede como maliciosos **[9]**, por essa característica, este tipo de detecção é bastante assertiva para ataques já conhecidos, a problemática é que a rede fica completamente vulnerável para ataques

desconhecidos pelo sistema de detecção, por isso que se precisa ter uma manutenção regular da base de assinaturas.

3.2 ATAQUES DE NEGAÇÃO DE SERVIÇO (DOS)

Existem diversos tipos de ameaças a serem combatidas pelos sistemas de detecção, aqui iremos focar no ataque de negação de serviço (DoS), o principal objetivo deste tipo de intrusão é sobrecarregar um recurso ou sistema específico com uma quantidade enorme de tráfego, tornando-o indisponível e, como visto no início deste artigo, muitas vezes este tipo de ataque visa vulnerabilizar a rede para que outras ameaças tirem proveito disso, ou apenas trazer prejuízos financeiros ou de reputação à empresa alvo.

Muitas vezes, os atacantes utilizam redes formadas por diversos hosts infectados por bots controlados pelo invasor para efetuar ataques distribuídos e sincronizados, os chamados DDoS, conceito bem definido no livro de Bhattacharyya **[10]**.

3.2.1 Método de Ataque por Volume

Esta técnica consiste em sobrecarregar a rede com requisições de conexão ICMP e UDP, esgotando a largura de banda dedicada. Nela, o atacante por muitas vezes obtém uma botnet, a qual ele falsifica os endereços de IP desta rede para o endereço de IP da vítima do ataque. Assim todas as respostas que o servidor DNS iria mandar para os hosts infectados acabam indo para a vítima, inundando a largura de banda.

3.2.2 Método de Ataque por Protocolo

Neste modelo, o atacante geralmente utiliza uma brecha do método de conexão three way handshake presente no protocolo TCP. Para que uma conexão seja estabelecida no protocolo TCP é necessário que o host envie um pacote SYN, que nada mais é do que uma solicitação de conexão, para o servidor, a partir do pacote SYN o servidor envia para o host um pacote SYN-ACK, e por fim, para que a conexão seja estabelecida, um pacote ACK deve ser enviado pelo host para o servidor.

Sendo assim, para realizar o ataque, o hacker envia diversos pacotes SYN para o servidor, o servidor responde com SYN-ACK, porém o hacker nunca responde com o ACK, deixando o servidor aguardando pela resposta, consumindo recursos da rede que poderiam ser destinados a conexões legítimas.

4 METODOLOGIA E RESULTADOS

Neste capítulo, serão evidenciadas as etapas do estudo e as ferramentas utilizadas para realizar a análise de dados diante da base escolhida, além de detalhar quais foram as métricas para avaliação de cada algoritmo e as características deste tipo de ataque que podemos observar na base de dados. Os resultados de cada algoritmo serão analisados e comparados, a fim de encontrar um modelo ótimo para a detecção de ataques DoS.

A análise dos algoritmos escolhidos foi realizada através da linguagem de programação Python, onde para o tratamento e manuseio de dados, foram utilizadas as bibliotecas Pandas e Numpy, para as visualizações de gráficos mostrados ao longo deste artigo, Matplotlib e Seaborn, e para as etapas de pré-processamento, indicadores de desempenho e os próprios algoritmos de aprendizado de máquina, Sklearn. Estas bibliotecas estão presentes no projeto em Python criado para a execução das etapas citadas neste capítulo.

4.1 BASE DE DADOS

A base de dados utilizada para este estudo foi a NSL-KDD, este conjunto de dados é uma versão aprimorada do KDD Cup de 1999, quando a Agência Avançada de Defesa e Projetos de Pesquisa (DARPA) em conjunto com o Laboratório de Pesquisas de Força Aérea dos Estados Unidos foram patrocinadores da criação desta base que utilizou dados simulados de trafégo em redes militares, além disso, a contrução da base contou com o apoio do MIT para adicionar dados de tráfego de rede adicionais capturados pela ferramenta tcpdump. A NSL-KDD conta com uma variedade de dados simulados de invasão militares em formato de tabela, dentro da base temos quatro tipos de ataques:

- Ataques de Negação de Serviço (DoS)
- Ataques de Sondagem (Probe)
- Ataques de Privilégio (U2R)
- Ataques de Acesso (R2L)

DOI: 10.xxxx/s11468-014-9759-3

Colunas binárias foram adicionadas à base de dados para indicar a presença (1) ou ausência (0) de cada tipo de ataque, com cada coluna representando um tipo específico de ataque.

4.2 ANÁLISE EXPLORATÓRIA

Para que os algoritmos sejam treinados de forma coerente e que não ocorra o enviesamento deles, é necessário avaliar os dados como um todo para que características redundantes ou discrepantes possam ser desconsideradas, ou substituídas, portanto, alguns métodos podem ser aplicados para avaliar estas características na base de dados.

4.2.1 Correlação

Neste método é utilizado o cálculo do Coeficiente de correlação de Pearson, para identificar o relacionamento entre duas variáveis, Segundo Miot **[11]**, o cálculo deste coeficiente é um teste estatístico que explora a intensidade e o sentido do comportamento mútuo entre variáveis, limitando-se a -1 e 1.

$$r = \frac{\sum(x_i-\bar{x})*(y_i-\bar{y})}{\sqrt{\sum(x_i-\bar{x})^2*\sum(y_i-\bar{y})^2}} \quad \textbf{(2)}$$

O cálculo, como se pode observar na fórmula 2, é feito dividindo o somatório do produto entre a subtração dos valores individuais de cada variável, (x_i, y_i) pelas suas médias $(\bar{x}, \bar{y})$, pelo produto das raízes quadradas das somas dos quadrados das diferenças, isso resulta no coeficiente de correlação (r). Diante disso, a correlação entre duas variáveis pode ser classificada de três formas:

- **Correlação positiva:** as variáveis são diretamente proporcionais
- **Correlação negativa:** as variáveis são inversamente proporcionais
- **Correlação nula**: não existe relação linear entre as variáveis.

Na NSL-KDD podemos ver algumas variáveis extremamente correlacionadas com as outras e, por isso, podem se tornar redundantes, estas colunas fortemente correlacionadas estão dispostas na tabela 1.

Tabela 1 – Colunas fortemente correlacionadas

Variável 1	Variável 2	Correlação
num_root	num_compromised	0,9988
serror_rate	srv_serror_rate	0,9953
rerror_rate	srv_rerror_rate	0,9933
dst_host_srv_serror_rate	dst_host_serror_rate	0,9876
dst_host_srv_serror_rate	srv_serror_rate	0,9857
serror_rate	dst_host_srv_serror_rate	0,9827
srv_serror_rate	dst_host_serror_rate	0,9798
serror_rate	dst_host_serror_rate	0,9796
dst_host_srv_rerror_rate	rerror_rate	0,9604
dst_host_srv_rerror_rate	srv_rerror_rate	0,9588
dst_host_srv_rerror_rate	dst_host_rerror_rate	0,9557
dst_host_rerror_rate	rerror_rate	0,9494
srv_rerror_rate	dst_host_rerror_rate	0,9446
dst_host_srv_count	dst_host_same_srv_rate	0,9321
same_srv_rate	dst_host_same_srv_rate	0,8235
is_guest_login	hot	0,8212

Fonte: O autor

Comparando cada variável com suas respectivas correlações junto a variável alvo monta-se a tabela 2.

Tabela 2 – Correlação das colunas da figura 17 com a coluna alvo

Coluna	Correlação (Alvo)
num_root	-0,01
num_compromised	-0,01
serror_rate	0,78
srv_serror_rate	0,79
rerror_rate	0,19
srv_rerror_rate	0,19
dst_host_srv_serror_rate	0,79
dst_host_serror_rate	0,79
dst_host_srv_rerror_rate	0,19
dst_host_rerror_rate	0,2
dst_host_srv_count	-0,72
dst_host_same_srv_rate	-0,76
same_srv_rate	-0,87
is_guest_login	-0,07
hot	-0,05

Fonte: O autor

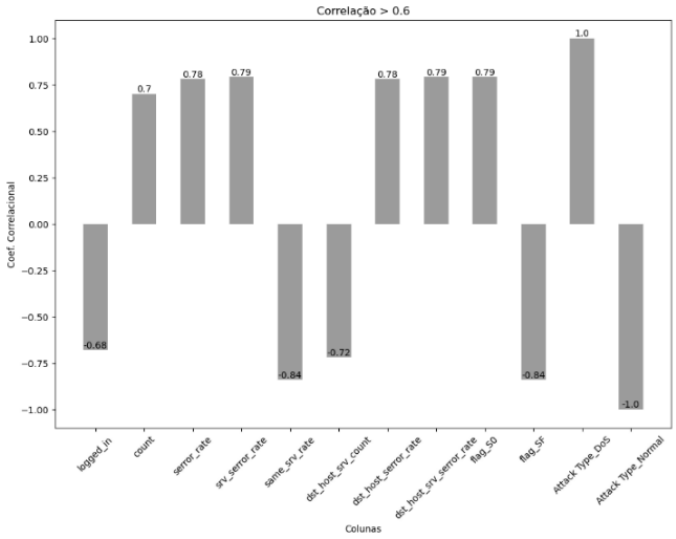
Dessa forma, as seguintes colunas foram descartadas da base por serem redundantes e serem menos correlacionadas com a coluna alvo (ataque DoS):

- **Serror_rate:** variável que representa o percentual de conexões com erro SYN, altamente correlacionado com Srv_serror_rate, que identifica o mesmo, mas para um mesmo número de porta;
- **Dst_host_srv_rerror_rate:** percentual de conexões que tiveram erro de RST para um mesmo número de porta, está correlacionado com rerror_rate que é o percentual de erros RST no geral;
- **Dst_host_srv_count:** número de conexões para um mesmo host de destino e mesmo serviço, correlacionado com dst_host_same_srv_rate, o percentual de conexões para um mesmo host e serviço.

Foram analisadas também as variáveis fortemente correlacionadas com a variável alvo de tipo de conexão, todas elas estão evidenciadas no gráfico da figura 11.

- **Logged_in:** Representa se o usuário que está solicitando a conexão está logado (0 se não, 1 se sim);
- **Count:** Número de conexões para o mesmo host nos últimos 2 segundos;
- **Srv_serror_rate:** Percentual de conexões com erro de SYN para um mesmo número de porta;
- **Flag S0:** Tentativa de conexão sem resposta, indica positivamente a possibilidade de ataque;
- **Flag SF:** Conexão realizada e terminada, indica negativamente a possibilidade de ataque.

Figura 9 – Variáveis fortemente correlacionadas com a variável dependente



Fonte: O autor

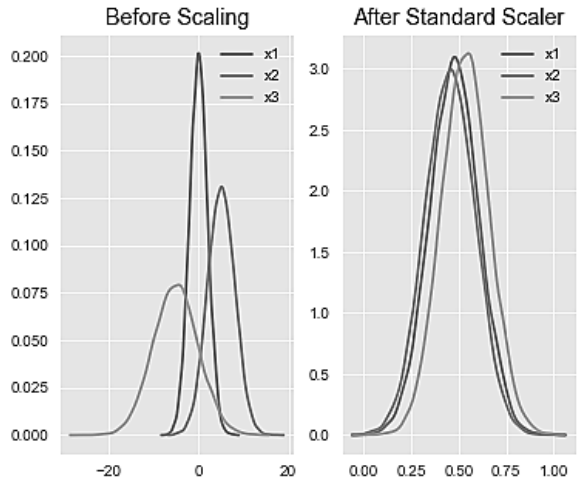
4.3 PRÉ-PROCESSAMENTO

A etapa de pré-processamento é onde modificamos os dados identificados anteriormente como prejudiciais para o treinamento e modelagem do algoritmo de aprendizado de máquina, esta fase é essencial e pode ter impactos relevantes no desempenho do método se não for bem executada, para isso, existem algumas técnicas que podem ser aplicadas na base de dados que não foi tratada ainda

4.3.1 Transformação de Dados

Este tipo de técnica é utilizada para normalizar e padronizar as variáveis presentes no conjunto de dados. Para isso, são aplicados métodos de dimensionamento, tornando as colunas comparáveis entre si e facilitando o treinamento dos métodos de aprendizado de máquina. A figura 12 deixa claro o antes e depois de 3 variáveis de exemplo ao serem normalizadas.

Figura 10 – Dados antes e depois do redimensionamento



Fonte: [12]

$$z = \frac{x-\mu}{\sigma} \quad \textbf{(3)}$$

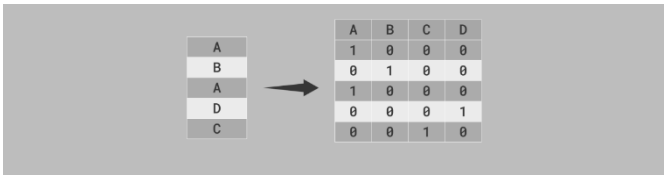
Na fórmula 3 podemos ver que a variável é normalizada (z) quando passa pela subtração da média das amostras (μ) pela amostra a ser normalizada (x) e então é dividida pelo desvio padrão de todas as amostras (σ).

DOI: 10.xxxx/s11468-014-9759-3

Outro ponto importante na transformação dos dados é a codificação e colunas categóricas. Como os algoritmos apenas interpretam números em seus cálculos, qualquer variável que não seja considerada numérica (texto, booleana, entre outras) deve ser codificada em numerais. Existem diversas técnicas para isso, a utilizada neste estudo foi a "codificação one-hot".

Neste tipo de codificação, cada variável única presente na coluna a ser codificada é transformada em uma nova coluna separada, onde esta é preenchida com 0s e 1s, 0 indica a ausência da característica e 1 indica a presença dela.

Figura 11 – One-hot encoding



Fonte: [13]

4.4 MÉTODOS DE AVALIAÇÃO

A avaliação de modelos de Aprendizado de Máquina tem um papel crucial no desenvolvimento de sistemas de alta qualidade e na tomada de decisões informadas. Neste capítulo, serão explorados diferentes métodos e métricas de avaliação, tais como acurácia, precisão, recall e F1-score, que são essenciais para mensurar o desempenho e a eficácia dos modelos de classificação.

4.4.1 Acurácia

A acurácia é uma métrica básica e amplamente utilizada para avaliar modelos de classificação. Ela representa a proporção de previsões corretas em relação ao número total de exemplos no conjunto de dados de teste **[14]**. Matematicamente, a acurácia é calculada como:

$$Acurácia = \frac{Número\ de\ previsões\ corretas}{Número\ total\ de\ registros} \quad \textbf{(4)}$$

No contexto deste artigo, a acurácia refere-se ao percentual de ataques previstos corretamente, em comparação ao total de conexões da base de dados.

4.4.2 Precisão

A medida de precisão avalia a proporção de registros positivos corretamente previstos em relação ao total de exemplos positivos previstos pelo modelo **[14]**. Essa métrica é importante quando o objetivo é reduzir os falsos positivos. A fórmula da precisão é:

$$Precisão = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positivos + Falsos\ Positivos} \quad \textbf{(5)}$$

Em relação à nossa base, a métrica de precisão calcula a proporção de ataques previstos corretamente, em comparação ao total de ataques (previstos corretamente e incorretamente).

4.4.3 Recall (Sensibilidade)

O recall, também conhecido como sensibilidade ou taxa de verdadeiros positivos, mede a proporção de exemplos previstos como positivos (ataques) corretamente, em relação ao número total de exemplos realmente positivos no conjunto de dados (tanto os ataques previstos como os que não foram previstos) **[14]**. É uma métrica relevante quando se deseja minimizar os falsos negativos. A fórmula do recall é dada por:

$$Recall = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positivos + Falsos\ Negativos} \quad \textbf{(6)}$$

4.4.4 F1-Score

O F1-score é representado pelo dobro da média harmônica entre precisão e recall, uma métrica que combina estas duas outras medidas em uma única, fornecendo um equilíbrio entre ambas as métricas. É particularmente útil quando se deseja encontrar um ponto de equilíbrio entre a minimização de falsos positivos e falsos negativos **[14]**. O F1-score é calculado usando a seguinte fórmula:

$$F1 - score = 2 \times \frac{(Precisão \times Recall)}{Precisão + Recall} \quad \textbf{(7)}$$

4.5 ANÁLISE DE COMPONENTES PRINCIPAIS (PCA)

Um conjunto de dados de dimensão inferior pode ser criado usando PCA, uma técnica estatística de redimensionamento de dados, que preserva a maioria da variação contida nos dados originais

[15]. Para conseguir isso, é identificado um novo conjunto de eixos conhecidos como componentes principais, que representam as direções ao longo das quais os dados variam mais.

Este processo é feito com alguns passos, primeiro calcula-se a matriz de covariância de todas as combinações possíveis de dados, através dela que definimos a relação de variação entre duas variáveis.

$$Cov(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y}) \quad \textbf{(8)}$$

Onde, conforme a fórmula 8, X e Y são as variáveis a serem calculadas e $\overline{X}$ e $\overline{Y}$ são suas respectivas médias. Com a matriz definida, podemos inferir as direções do conjunto de dados, cada componente principal é representado por uma direção de variação dos dados, sendo o primeiro componente a direção de maior variação e assim por diante.

4.6 RESULTADOS

Aqui serão apresentados os resultados obtidos na aplicação dos modelos vistos anteriormente junto à base de dados escolhida. Foram utilizadas 4 sub-bases na etapa de treinamento dos algoritmos, sendo elas:

- **Conjunto FULL:** Considerou-se a base completa de dados sem nenhuma alteração;
- **Conjunto CORR-:** Aqui foram retiradas as variáveis consideradas como redundantes na subseção 4.2.1;
- **Conjunto CORR+:** Nesta base foram retiradas quaisquer colunas que tenham o coeficiente de correlação menor que 0,6;
- **Conjunto PCA:** O resultado da aplicação do PCA na base completa, utilizando 2 componentes principais, definidos através da variância acumulada.

E finalmente, para os métodos de avaliação definidos, os resultados de cada algoritmo, para cada subconjunto de dados estabelecidos, podem ser vistos na tabela 3:

Tabela 3 – Resultados dos métodos de avaliação

Model	Base	Acurácia	Precisão	Recall	F1-score
Florestas Aleatórias	FULL	99,98%	99,98%	99,97%	99,98%
	CORR-	99,98%	99,98%	99,98%	99,98%
	CORR+	98,02%	98,59%	96,49%	97,53%
	PCA	99,68%	99,68%	99,53%	99,60%
Regressão Logística	FULL	99,66%	99,57%	99,59%	99,58%
	CORR-	99,60%	99,43%	99,58%	99,51%
	CORR+	97,09%	98,89%	93,88%	96,32%
	PCA	42,06%	40,97%	97,98%	57,78%
KMeans	FULL	49,91%	20,41%	50,00%	28,99%
	CORR-	40,91%	40,64%	100,00%	57,80%
	CORR+	93,72%	99,81%	84,63%	91,59%
	PCA	41,17%	40,75%	100,00%	57,90%
MLP	FULL	99,97%	99,94%	99,98%	99,96%
	CORR-	99,96%	99,93%	99,97%	99,95%
	CORR+	97,93%	99,43%	95,45%	97,39%
	PCA	92,06%	88,90%	91,83%	90,34%
SVM	FULL	40,64%	40,64%	100,00%	57,79%
	CORR-	40,46%	40,46%	100,00%	57,61%
	CORR+	40,46%	40,46%	100,00%	57,61%
	PCA	40,46%	40,46%	100,00%	57,61%

Fonte: O autor

Observou-se que os algoritmos de Florestas Aleatórias e MLP alcançaram níveis de acurácia e precisão próximos a 99%, detectando 99% dos ataques na base de dados com uma taxa muito baixa de falsos positivos, além de taxas de recall e F1-score também bastante altas, mostrando a sensibilidade destes algoritmos para a diferenciação de ataques e tráfego normal. Por outro lado, os algoritmos de KMeans e SVM apresentaram uma baixa eficácia, caracterizada por uma previsão de ataques significativamente menor (em torno de 40%) e um alto índice de falsos positivos, evidenciado pela taxa de Recall em 100%.

O algoritmo de Regressão Logística também obteve resultados muito bons quando trabalhando com subconjuntos mais densos, todos na faixa de 90%, porém ao aplicar a análise de componentes principais, seu desempenho cai significativamente, indicando uma fragilidade de bases de dados menores.

5 CONCLUSÕES

Este estudo de caso mostrou a eficiência de modelos diferentes de aprendizagem de máquina na tarefa de detecção de ataques de negação de

DOI: 10.xxxx/s11468-014-9759-3

serviço. Se bem aplicados, estes algoritmos são ferramentas muito poderosas e bastante eficientes.

Ainda na etapa de análise podemos observar que o método aplicado nos trouxe informações valiosas e condizentes com a literatura estudada, o algoritmo identificou como assinaturas importantes as variáveis de quantidade de conexões para um mesmo host, percentual de erros de pacote SYN e o flag de tentativa de conexão sem resposta como indicadores importantes na classificação de uma atividade maliciosa característica de um ataque de negação de serviço.

A partir dos resultados indicados pelos métodos de avaliação podemos concluir que os algoritmos que se saíram melhor para este objetivo foram o MLP e as Florestas Aleatórias, muito devido aos seus bons desempenhos quando tratamos de dados não lineares e com grandes números de parâmetros, sendo o caso da base de dados estudada.

As técnicas que obtiveram os piores resultados foram a SVM e o KMeans, pois ambos não lidam bem com uma grande dimensionalidade de dados e com relacionamentos não lineares, isto se reforça quando analisamos o aumento na performance do KMeans quando utilizamos o conjunto de dados CORR+, onde se diminuiu consideravelmente a dimensão da base de dados, além disso, o Recall de 100% unido a uma precisão baixa indica que o algoritmo está retornando muitos falsos positivos (conexões normais identificadas como maliciosas). A presença grande de outliers (pontos fora da curva) também impacta nos cálculos dos centroides no KMeans e na posição do hiperplano do SVM.

Finalmente, a partir deste estudo, pode-se observar quais tipos de algoritmos conseguem ser utilizados em sistemas de detecção de intrusão e quais não devem ser utilizados nos padrões dos dados que foram usados no treinamento. Em trabalhos futuros, tem-se o objetivo de realizar o treinamento destes algoritmos em outras bases de dados com foco na generalização para aplicação destas técnicas em grande escala, além de realizar testes comparativos com novos métodos de redes neurais que requerem maior poder computacional.

REFERÊNCIAS

[1] SARAVANAN, Arumugam; SATHYA, Bama Subramanian. A Review on Cyber Security and the Fifth Generation Cyberattacks. **Oriental Journal of Computer Science and Technology**, v. 12, n. 2, 2019. DOI: 10.13005/ojcst12.02.04 Disponível em:. Acesso em: 03 fev. 2024 às 14h28min.

[2] Kaspersky Labs. **Denial Of Service:** How business evaluate the threat of DDOS attack, IT security risks special report [recurso eletrônico]. Disponível em: https://media.kasperskycontenthub.com/wp-content/uploads/sites/45/2018/03/08234158/IT_Risks_Survey_Report_Threat_of_DDoS_Attacks.pdf. Acesso em: 03 fev. 2024 às 13hs18min.

[3] MITCHELL, Tom M. **Machine Learning.** Nova Iorque: McGraw-Hill Science/Engineering/Math., 1997. Ed. 1, p. 02-53.

[4] BISHOP, Christopher M. **Pattern Recognition and Machine Learning.** Cambridge: Springer, 2006. Ed. 1, p. 03.

[5] LORENA, Ana Carolina; DE CARVALHO, André C. P. L. F. Uma Introdução às Support Vector Machines. **Revista De Informática Teórica E Aplicada**, p. 43-67, 2007. DOI: 10.22456/2175-2745.5690. Disponível em: https://www.researchgate.net/publication/36409205_Uma_Introducao_as_Support_Vector_Machines. Acesso em: 05 fev. 2024 às 22h22min

[6] DUDA, Richard O; HART, Peter E; STORK, David G. **Pattern Classification.** Nova Iorque: John Wiley & Sons, 2000. Ed. 2, p. 610-615.

[7] SAWYER, Mark D. Invited commentary: Artificial neural networks–an introduction. In: Department of Surgery, Minn, vol 127, n 1, 2000, Mayo Clinic, Rochester. DOI: 10.1067/msy.2000.102174. Disponível em: https://pubmed.ncbi.nlm.nih.gov/10660750/ Acesso em: 03 fev. 2024 às 15h13min

[8] HAYKIN, Simon. **Redes Neurais:** Princípios e Prática. Ed 2. p. 36, 2006.

[9] LEE, Wenke; STOLFO, Salvatore J. A Framework for Constructing Features and Models for Intrusion Detection Systems. **ACM Transactions on Information and System Security**, p. 222-229, 2007. DOI: 10.1145/382912.382914. Disponível em: https://dl.acm.org/doi/10.1145/382912.382914. Acesso em: 05 fev. 2024 às 23h54min

[10] BHATTACHARYYA, Dhruba Kumar; KALITA, Jugal Kumar. **DDoS Attacks:** Evolution, Detection, Prevention and Tolerance. Ed 1. p. 03-04, 2016

[11] MIOT, Hélio Amante. Análise de Correlação em Estudos Clínicos e Experimentais. **Jornal Vascular Brasileiro**, 17(4), p. 275-279, 2018. DOI: 10.1590/1677-5449.174118. Disponível em: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6375260/. Acesso em: 06 fev. 2024 às 00h58min

[12] Stackoverflow. **Difference between standardscaler and Normalizer in sklearns.preprocessing** [recurso eletrônico]. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html Acesso em: 03 fev. 2024 às 14hs15min

[13] Alura. **get_dummies vs OneHotEncoder: qual método escolher?** [recurso eletrônico]. Disponível em: https://www.alura.com.br/artigos/get-dummies-vs-onehotencoder-qual-metodo-escolher Acesso em: 03 fev. 2024 às 14hs17min

[14] SILVA, Lucas C et al. Estudo Comparativo de Métodos de Aprendizagem de Máquina Aplicados em Sistemas de Detecção de Intrusão. In: Escola Regional de Computação Do Ceará, Maranhão e Piauí (ERCEMAPI), 7, 2019, São Luís. **RENOTE: Sociedade Brasileira de Computação**, Porto Alegre, p. 135-142 jul. 2019. Disponível em: https://sol.sbc.org.br/index.php/ercemapi/article/view/8855/8756 Acesso em: 03 fev. 2024 às 14h36min

[15] JOLLIFFE, I. T. **Principal Component Analysis.** Springer Series in Statistic, 2002. Ed. 2, p. 02-05.

[16] GOMES, Elias Amadeu de Souza. **Aplicabilidade De Algoritmos De Aprendizado De Máquina Para Detecção De Intrusão e Análise De Anomalias De Rede**. 2019. Dissertação (Pós-graduação em Informática) - Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, 2019. Disponível em: https://repositorio.ufmg.br/handle/1843/SLSC-BC9F4H Acesso em: 03 fev. 2024 às 14h44min.

[17] DASARI, Kishore Babu; NAGARAJU, Devarakonda. Detection of DDoS Attacks Using Machine Learning Classification Algorithms. **International Journal of Computer Network and Information Security**, v. 09, ed. 6, 2022. DOI: 10.5815/ijcnis.2022.06.07 Disponível em: https://www.proquest.com/openview/0494cd320528a417276b2ec7b9ae9564/1?pq-origsite=gscholar&cbl=2026671 Acesso em: 03 fev. 2024 às 14h50min.

[18] LIMA, Igor Vinicius Mussoi de. **Uma Abordagem Simplificada De Detecção De Intrusão Baseada Em Redes Neurais Artificiais.** 2005. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Santa Catarina, 2005, Florianópolis, SC. Disponível em: https://www.inf.ufsc.br/~bosco.sobral/grupo/MestradoIgor.pdf. Acesso em: 03 fev. 2024 às 15h20min

[19] FIGUEIREDO, Bruno et al. **Estudo e Investigação de Técnicas de IA para Detecção de Ataques DDOS**. 2022. Dissertação (Faculdade de Computação e Informática) – Universidade Presbiteriana Mackenzie, 2022, São Paulo, SP. Disponível em: https://dspace.mackenzie.br/items/fcd8e4c8-9b65-4000-b717-e44336b7860e. Acesso em: 03 fev. 2024 às 15h01min

DOI: 10.xxxx/s11468-014-9759-3