UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Miguel Sanches Rocha

Identificação de ataques não-conhecidos em sistemas de detecção de intrusão baseados em anomalia

Uberlândia, Brasil

2023

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Miguel Sanches Rocha

Identificação de ataques não-conhecidos em sistemas de detecção de intrusão baseados em anomalia

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciência da Computação.

Orientador: Rodrigo Sanches Miani

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2023

Miguel Sanches Rocha

Identificação de ataques não-conhecidos em sistemas de detecção de intrusão baseados em anomalia

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciência da Computação.

Rodrigo Sanches Miani
Orientador

Professor

Professor

Uberlândia, Brasil
2023

Resumo

Os Sistemas de Detecção de Intrusão (*IDS*) se tornaram um dos mais importantes controles de segurança devido à capacidade de detectar ataques cibernéticos por meio da inspeção do tráfego de rede. Durante a última década, as propostas de *IDS* cada vez mais usam técnicas de aprendizado de máquina (*IDS* baseado em Aprendizado de Máquina) para criar modelos de detecção de ataques. À medida que essa tendência ganha força, os pesquisadores discutem se esses *IDS* podem detectar ataques desconhecidos (*zero-day*). A maioria dos Sistemas de Detecção de Intrusão baseados em aprendizado de máquina, são desenvolvidos com aprendizado supervisionado, o que significa que eles são treinados com uma coleção limitada de exemplos de ataque e portanto, detectar ataques que não foram cobertos durante a fase de treinamento pode ser um desafio para esses sistemas. Este trabalho avalia a capacidade do *IDS* baseado em Aprendizado de Máquina em detectar ataques desconhecidos. A ideia geral é entender o que acontece quando um modelo de detecção treinado com um ataque específico A, recebe dados de entrada de um ataque desconhecido B. Usando o conjunto de dados *CIC-IDS2017*, foi demonstrado que modelos supervisionados de detecção de intrusão, na maioria dos casos, não podem detectar ataques desconhecidos. A única exceção ocorre com ataques *DoS* (Negação de serviço). Por exemplo, um modelo de detecção de intrusão treinado com amotras de ataque *HTTP Flood DoS (GoldenEye)*, pode detectar um tipo diferente de ataque *HTTP DoS*, o tipo *Slowloris*.

Palavras-chave: Segurança da informação, Sistemas de detecção de intrusão, Detecção de anomalias, Aprendizado supervisionado.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

IDS	*Intrusion Detection System* (Sistema de Detecção de Instrusão)
CPU	*Central Processing Unit* (Unidade central de processamento)
DoS	*Denial of Service* (Negação de serviço)
DDoS	*Distributed Denial of Service* (Negação de serviço distribuído)
HWIDS	*Hybrid Web Intrusion Detection System* (Sistema Híbrido de Detecção de Intrusão na Web)
UNAD	*Unknown Network Attack Detector* (Detector de ataque de rede desconhecido)
CART	*Classification and Regression Trees* (Árvores de classificação e regressão)
ANN	*Artificial Neural Network* (Rede neural artificial)
CLONALG	*Clonal Selection Algorithm* (Algoritmo de seleção clonal)
HTTP	*Hypertext Transfer Protocol* (Protocolo de transferência de hipertexto)
DNS	*Domain Name System* (Sistema de nomes de domínio) ;
FTP	*File Transfer Protocol* (Protocolo de transferência de arquivo)
HTTPS	*Hypertext Transfer Protocol Secure* (Protocolo de transferência de hipertexto seguro)
SSH	*Secure Shell* (*Shell* seguro)
NTP	*Network Time Protocol* (Protocolo de tempo de rede)
TCP	*Transmission Control Protocol* (Protocolo de controle de transmissão)
IP	*Internet Protocol* (Protocolo da Internet)
UDP	*User Datagram Protocol* (Protocolo de datagrama do usuário)
AW	Ataques web
AUC	*Area Under Curve* (Área sob a curva)

ROC	*Receive Operating Characteristic* (Característica de operação do receptor)
TPR	*True Positive Rate* (Taxa de verdadeiro positivos)
FPR	*False Positive Rate* (Taxa de falsos positivos)
XSS	*Cross-site Scripting* (*Script* entre sites)
CA	Classe de Ataque
TA	Todos os Ataques

Sumário

1 Introdução

A Internet é uma complexa estrutura formada por inúmeras redes conectadas pelas quais os indivíduos transmitem e recebem informações, o que permite uma interação intensa e extremamente rápida entre os usuários. Diante disso, surge a questão da segurança da informação como um dos principais tópicos a serem estudados e fortalecidos nesse ambiente revolucionário conhecido como 'A rede das redes'. (LI; LIU, 2021) afirma que os riscos de ataques cibernéticos de todos os tipos estão sempre em uma crescente, o que pode ocasionar diversos impactos a usuários e corporações de várias formas. Nesse sentido, diversos mecanismos foram criados para mitigar os riscos associados a ciberataques como protocolos criptográficos, *Firewalls*, Antivírus e os chamados Sistemas de Detecção de Intrusões (IDS) (CERT.br, 2021).

O crescimento exponencial de usuários conectados à Internet e o surgimento de novas tecnologias relacionadas ao mundo cibernético, criaram o ambiente perfeito para a disseminação de ataques. Por esta razão, ter uma sólida estrutura de cibersegurança é necessário para manter e sustentar os recursos de computação das organizações. Nesse cenário, ferramentas como os Sistemas de Detecção de Intrusão (*IDS*), são uma das principais alternativas capazes de mitigar as tentativas de ataques e manter os sistemas computacionais seguros. Deste modo, trabalhos que visam o desenvolvimento e a análise de mecanismos de segurança do meio cibernético, se apresentam como uma questão fundamental atualmente e possuem um caráter de importante relevância científica e social.

Existe uma vasta literatura sobre o tema e vários estudos buscam analisar o comportamento de algoritmos de aprendizado de máquina supervisionado na detecção de ataques desconhecidos, porém usando técnicas muito específicas. Por exemplo, Zhang et al. (ZHANG et al., 2021) propôs um modelo específico baseado em técnicas de aprendizado profundo para identificar ataques desconhecidos. A abordagem proposta foi avaliada nos conjuntos de dados "*DARPA KDDCUP 99*" (STOLFO et al., 2000) e *CIC-IDS2017* (SHARAFALDIN; LASHKARI; GHORBANI, 2018). Os resultados foram satisfatórios, mas o trabalho investigou apenas uma pequena amostra de ataques e não avaliou algoritmos mais simples comumente encontrados na literatura, como o Floresta Aleatória, Máquina de Vetores de Suporte e Árvores de Decisão. Além disso, foi usada uma abordagem híbrida para o modelo de aprendizado (supervisionado/não supervisionado) e também um método para atualizar o classificador. Outros estudos como (JONGSUEBSUK; WATTANAPONGSAKORN; CHARNSRIPINYO, 2013), (SONG et al., 2007) e (XU et al., 2019), também propõem modelos de intrusão híbridos e específicos para detecção de ataques desconhecidos.

A principal contribuição deste trabalho é fornecer um estudo para entender o impacto da configuração mais comum de *IDS* baseado em anomalias na detecção de ataques desconhecidos: um modelo supervisionado simples sem técnicas de aprendizado incremental.

Portanto, este trabalho visa avaliar o desempenho de *IDS* baseados em anomalias e desenvolvidos com técnicas de Aprendizado de Máquina com relação à ataques desconhecidos. Desta forma, será possível entender o comportamento dos classificadores ao lidar com dados que não foram apresentados a eles durante a fase de treinamento.

Com isso, será possível adquirir uma sólida base de conhecimento científico acerca do comportamento e das possíveis vantagens do uso de tais sistemas na estrutura de segurança de ambientes computacionais, cada vez mais propensos a invasões e ataques. A ideia do trabalho é justamente fornecer *insights* e, possivelmente, novos direcionamentos sobre o uso de modelos de aprendizado de máquina na área de detecção de intrusão.

O restante do trabalho está organizado da seguinte forma. O Capítulo 2 discute acerca da revisão bibliográfica, englobando o referencial teórico na Seção 2.1, o qual trata de diversos conceitos importantes para esta monografia, e os trabalhos relacionados na Seção 2.2, que destaca o que vem sendo feito na área de detecção de intrusão e ataques desconhecidos. O Capítulo 3 trata do desenvolvimento do trabalho. Nele, é apresentada uma visão geral, bem como o conjunto de dados utilizado juntamente com os métodos de pré-processamento aplicados. Posteriormente neste mesmo capítulo, a partir da Seção 3.3, tem-se a metodologia experimental e todas as etapas de experimentos que foram conduzidas neste trabalho. Finalmente, o Capítulo 4 traz os resultados obtidos em cada rodada de experimentos, de modo a destacar os cenários mais interessantes alcançados na detecção de ataques desconhecidos e o Capítulo 5 apresenta a conclusão e as considerações finais.

2 Revisão Bibliográfica

Este capítulo será reservado para que sejam feitas definições de conceitos importantes utilizados neste trabalho, bem como para citar o que vem sendo realizado na área de detecção de intrusão e ataques desconhecidos. Na Seção 2.1, tem-se a definição de conceitos como ”Ataques Cibernéticos” (Subseção 2.1.1), juntamente com as 3 classes de ataques que foram abordadas nesta monografia (DoS, Força Bruta e Ataques Web). Posteriormente, também é discutido o conceito de ”Ataques Desconhecidos (Dia Zero)”, que possui papel central neste trabalho. Em seguida, a Seção 2.1 ainda contempla e discorre sobre o tópico ”Aprendizado de Máquina” na Subseção 2.1.2, e o aprofunda ao tratar separadamente dos tópicos ”Pré-processamento dos dados”, ”Treinamento e teste dos modelos”, ”Aprendizado supervisionado e não supervisionado” e ”Validação do modelo”. Finalmente, a Subseção 2.1.3 ”Sistemas de Detecção de Intrusão (IDS)” traz a definição de *IDS* e discorre sobre suas características, bem como sobre seus diferentes tipos no tópico ”Tipos de IDS (Host/Rede)”. Finalmente, na Seção 2.2 tem-se informações acerca de alguns trabalhos que foram conduzidos na área de *IDS* e ataques desconhecidos. O objetivo foi destacar a metodologia utilizada pelos autores, os resultados obtidos e realizar uma comparação com este trabalho, de modo que possa ser possível destacar a inovação e o diferencial trazidos nesta monografia.

2.1 Referencial Teórico

Esta Seção é dedicada para que diversos conceitos fundamentais para a estruturação do trabalho fossem definidos e detalhados, de modo a criar uma sólida base de tópicos que envolvem os temas de ataques cibernéticos, aprendizado de máquina, *IDS* e suas especificidades.

2.1.1 Ataques Cibernéticos

Como dito anteriormente, o amplo acesso e a democratização do uso da Internet fizeram do meio digital um ambiente passível de ataques com diferentes características e finalidades. Biju et al. (BIJU; GOPAL; PRAKASH, 2019) conceitua o ataque cibernético como sendo um tipo de ataque que visa roubar, alterar ou mesmo danificar um dado que esteja presente em um computador ou em uma rede de computadores. Nesse sentido, o trabalho também afirma que o atacante pode se tratar de um indivíduo ou processo que consiga obter acesso não autorizado a um sistema, e que o ataque cibernético faz uso de códigos maliciosos para alterar os dados e/ou a lógica do computador alvo. As Subseções 2.1.1.1, 2.1.1.2 e 2.1.1.3 definem e exemplificam três tipos de ataques cibernéticos comu-

mente presentes em conjuntos de dados públicos, sendo eles: DoS/DDoS, Força Bruta e Ataques Web, respectivamente.

2.1.1.1 DoS / DDoS

Biju et al. (BIJU; GOPAL; PRAKASH, 2019) cita e conceitua diversos tipos de ataques cibernéticos e, dentre eles, o ataque de negação de serviço (DoS) e o de negação de serviço distribuído (DDoS). Segundo os autores, ambos os ataques visam sobrecarregar os recursos de um sistema com solicitações ilegítimas, para que ele não seja capaz de responder a um pedido verdadeiro de serviço. Deste modo, os recursos do sistema ficam indisponíveis para o real usuário.

Apesar de serem muito parecidos, a principal diferença entre os ataques de tipo DoS e os de tipo DDoS é que o primeiro se trata de um único sistema atacando outro, já no segundo tipo podem existir vários sistemas, comandados por um computador, atuando em conjunto para atacar e derrubar determinado alvo.

A Figura 1 ilustra o funcionamento básico de um ataque DDoS, em que o servidor é sobrecarregado com requisições ilegítimas (representadas pelas flechas vermelhas) provenientes dos computadores atacantes, o que resulta na indisponibilidade do serviço para as requisições legítimas (representadas pelas flechas verdes) provenientes de usuários legítimos.

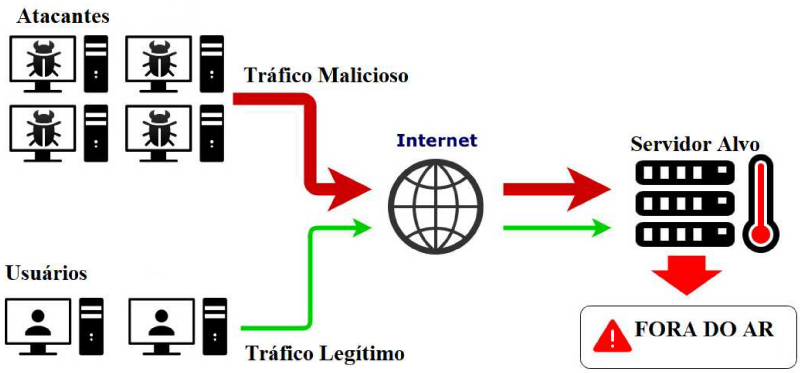

Figura 1 – Funcionamento básico de um ataque DDoS. Adaptado de (MEDIA, 2023).

2.1.1.2 Força Bruta

De acordo com Park et al. (PARK et al., 2021), o entendimento sobre ataques de força bruta pode ser obtido com um exemplo de acesso a uma conta em determinado sistema, em que seja necessário digitar uma senha. Nesse sentido, a força bruta é caracterizada por se tentar todas as combinações possíveis de valores como entrada para a senha requerida, até que se chegue a ela, o que pode ser muito dispendioso dependendo de seu tamanho e complexidade.

Esse processo repetitivo de tentativa e erro geralmente é feito com o auxílio de uma ferramenta ou programa que automatize a geração e submissão das senhas. Segundo os autores, existem algumas formas de se defender contra ataques que utilizam força bruta, como por exemplo controlar o acesso quando senhas erradas são inseridas, definir um número máximo de tentativas e ter uma política que exija uma certa complexidade e tamanho mínimos de senha para os usuários.

A Figura 2 ilustra o funcionamento básico de um ataque de força bruta, em que um agente mal intencionado desenvolve uma ferramenta que é capaz de gerar e submeter senhas a uma página web, até que uma delas seja aceita e o atacante consiga ganhar acesso ao sistema.

Figura 2 – Funcionamento básico de um ataque de força bruta. Adaptado de (ENGINE, 2023).

2.1.1.3 Ataques Web

Sarmah et al. (SARMAH; BHATTACHARYYA; KALITA, 2018) cita o importante papel que a Web possui no cotidiano de grande parte da sociedade e discute a questão da segurança no ambiente Web, o qual é bombardeado com ataques diversos e de diferentes complexidades de forma crescente e contínua. Nesse sentido, pode-se tomar o ataque Web XSS (*Script* entre sites) como um dos tipos de ataques mais comuns no ambiente Web e os autores o definem como sendo aquele em que um código malicioso é introduzido em uma aplicação Web que possua vulnerabilidades de segurança. Assim, ao acessar a aplicação, o usuário terá sua requisição respondida com o conteúdo adulterado pelo atacante disfarçado de conteúdo legítimo. Desta forma, o usuário é infectado pelo atacante que, por sua vez, poderá realizar tarefas como copiar *cookies*, *tokens* e por vezes ter acesso a dados críticos registrados no navegador web da vítima.

A Figura 3 ilustra o funcionamento básico de um ataque Web XSS, em que o agente mal intencionado injeta um código malicioso em uma aplicação Web, e assim, os usuários que a acessam passam a ter dados particulares enviados ao atacante.

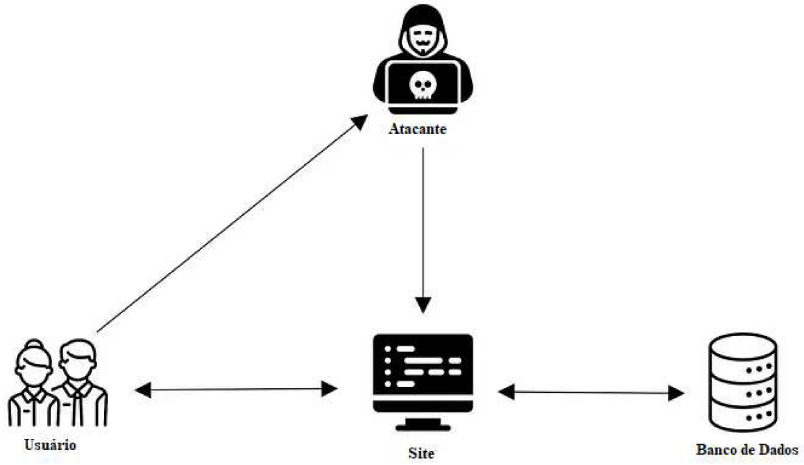

Figura 3 – Funcionamento básico de um ataque Web XSS. Adaptado de (TAKASE, 2021).

2.1.1.4 Ataques Desconhecidos (Dia Zero)

De acordo com (ALI et al., 2022), ataques de dia zero, isto é, ataques até então desconhecidos, representam uma séria ameaça à segurança da Internet como um todo. Estes, segundo os autores, não possuem padrões correspondentes em *malwares* e são geralmente usados em conjunto com outros ataques complexos a fim de evitar serem detectados por sistemas de segurança computacional. Segundo o trabalho, ataques de dia zero podem possuir diversas variantes, como *worms*, vírus, cavalos de Tróia, entre outros.

Outro dado interessante destacado pelos autores, diz respeito ao prolongado tempo que um agente malicioso desconhecido pode permanecer em um sistema até que seja detectado pela equipe de segurança. Esse tempo é de 10 meses em média, o que mostra como esses ataques podem representar um grande desafio para qualquer sistema computacional. Finalmente, o estudo também cita um dado estatístico de quem mais de 62% dos ataques são detectados depois de o sistema já ter sido comprometido e que os ataques de dia zero estão cada vez mais imprevisíveis, à medida que seu número de ocorrências aumenta periodicamente.

A Figura 4 ilustra alguns casos de grande impacto em anos recentes de sistemas robustos que foram alvos de ataques de dia zero. Isso demonstra como esse tipo de ataque pode representar um grande desafio até mesmo para corporações consolidadas e com amplo sistema de segurança cibernética.

2.1.2 Aprendizado de Máquina

O Aprendizado de Máquina é um campo vinculado às Ciências da Computação que tem como objetivo introduzir autonomia de aprendizagem aos sistemas computacionais, sem que precisem ser explicitamente programados por um profissional humano (MAN-

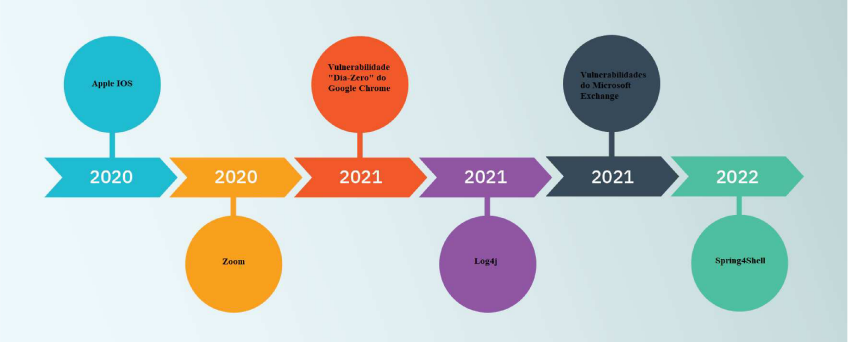

Figura 4 – Casos recentes de ataques de dia zero. Adaptado de (BLOG, 2022).

NILA, 1996). Esta ”autonomia intelectual” é obtida por meio de algoritmos sofisticados capazes de analisar e aprender, podendo assim, fazer rotulações e previsões acerca de dados. Esses algoritmos extraem os principais padrões e características da base de dados e desenvolvem um modelo de classificação que é submetido a testes para ter sua precisão aferida. A Figura 5 ilustra o processo descrito.

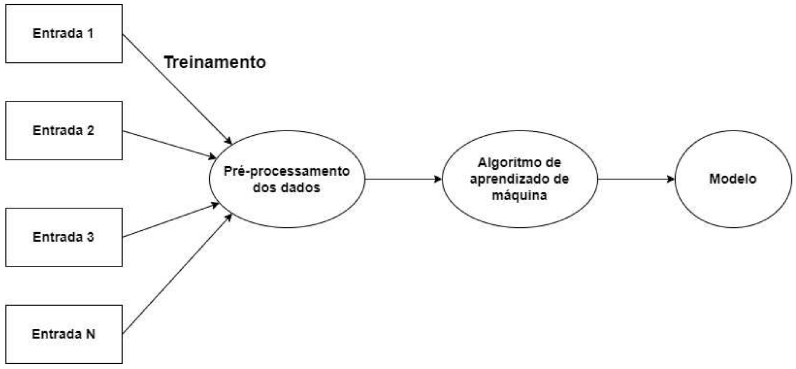

Figura 5 – Processo simplificado de aprendizado de máquina. Fonte: Do Autor.

2.1.2.1 Pré-processamento dos dados

O pré-processamento dos dados corresponde à fase inicial do processo de desenvolvimento de um classificador de aprendizado de máquina. Nesta etapa, os dados brutos sofrem transformações a fim de que adquiram um formato mais fácil e eficaz para serem utilizados em etapas posteriores de processamento. Uma das estratégias mais tradicionais de pré-processamento é a normalização dos dados, em que eles têm seus valores redimensionados usando algum método matemático como o clássico método chamado ”MinMax”, o que resulta em dados com uma mesma escala, por exemplo no intervalo de 0 a 1. Essas transformações realizadas no pré-processamento muitas vezes melhoram o tempo de treinamento do modelo de classificação e evitam interferências negativas no desempenho dos algoritmos. (AHMAD; AZIZ, 2019).

Abaixo, tem-se a fórmula do método ”MinMax”, em que o valor normalizado (Xnormalizado) é obtido por meio do valor original do dado ”X” subtraído do valor mínmo que ”X” assume para esse atributo (”Xmin”) no conjunto de dados. O resultado dessa subtração deve ser divido pelo valor máximo que ”X” assume para esse atributo (”Xmax”) no conjunto de dados, subtraído novamente pelo valor mínimo (”Xmin”).

$$Xnormalizado = \frac{(X - Xmin)}{(Xmax - Xmin)} \tag{2.1}$$

Outra manipulação frequentemente realizada nos conjuntos de dados e que faz parte da etapa de pré-processamento, é o tratamento de dados que possuem valores infinitos ou nulos. Essas ocorrências, caso não tratadas, podem inviabilizar o processo de treinamento e validação do modelo de classificação pois muitas vezes irão enviesar o classificador, resultando em um desempenho ruim na etapa de teste do modelo. Diante disso, algumas alternativas podem ser empregadas visando inibir essa interferência negativa, mas geralmente esses dados com valores infinitos ou nulos são simplesmente excluídos da base de treino dos modelos (como foi feito neste trabalho).

2.1.2.2 Treinamento e teste dos modelos

(UÇAR et al., 2020) et. al discorre sobre o processo de treinamento e teste de um modelo de classificação desenvolvido com técnicas de aprendizado de máquina. Os autores afirmam que esta etapa representa a parte mais crítica capaz de afetar o desempenho do algoritmo, à medida que um processo de treinamento eficaz e bem conduzido melhora a qualidade do modelo desenvolvido. O trabalho ainda cita que os pesquisadores geralmente dividem o conjunto de dados que está sendo utilizado em duas partes, uma para treinamento do modelo e outra para o teste. Segundo os autores, a relação de tamanho entre essas duas bases varia de acordo com a estrutura e a correlação dos dados, mas que usar

menos de 50% do conjunto para a etapa de treinamento não é recomendado, pois existe grande chance de afetar negativamente o desempenho do classificador.

Após a fase de treinamento do modelo utilizando o conjunto de dados específico para treino, o modelo é então testado usando os dados de teste, que até então nunca foram vistos pelo classificador, com o objetivo de determinar o quanto dos dados foi realmente aprendido pelo algoritmo. Finalmente, com o término da fase de treinamento e teste, o desempenho do modelo pode ser analisado pelos profissionais envolvidos, a fim de realizar eventuais ajustes necessários no conjunto de dados ou no próprio algoritmo de classificação, de modo a conduzir esta etapa novamente se necessário, buscando o melhor desempenho possível do modelo.

A Figura 6 traz uma representação básica de como o conjunto de dados original pode ser subdivido em conjunto de treino (para se treinar o modelo) e conjunto de teste (para se testar/avaliar o modelo previamente treinado).

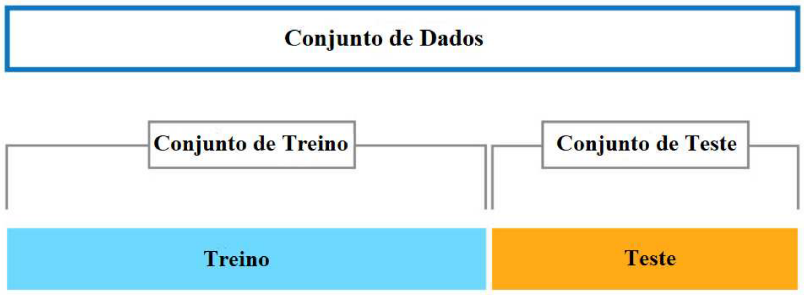

Figura 6 – Divisão básica de um conjunto de dados em conjunto de treino e conjunto de teste. Adaptado de (AMIN, 2022).

2.1.2.3 Aprendizado supervisionado e não supervisionado

(BENVENUTO et al., 2018) et. al traz os conceitos de aprendizado supervisionado e aprendizado não supervisionado, destacando as características principais e as diferenças entre essas duas abordagens. Segundo os autores, no caso do aprendizado supervisionado, o conjunto de dados utilizado possui seus atributos rotulados, representando o resultado das observações. Nesse sentido, um classificador supervisionado de aprendizado de máquina, possui a tarefa de realizar uma previsão na tentativa de determinar o rótulo associado ao conjunto de dados fornecido na etapa de teste.

O funcionamento da abordagem não supervisionada é caracterizado por não fazer uso de nenhum conjunto de dados rotulados. Assim, há apenas a análise pura dos dados, que são agrupados em diferentes conjuntos de acordo com a semelhança de suas características. Uma das técnicas mais comuns de aprendizado não supervisionado é o chamado *Clustering*, em que basicamente o conjunto de dados é organizado em grupos de amostras (*clusters*) com base em sua similaridade. Nesse sentido, dados parecidos estarão agrupados em um mesmo *cluster*.

De modo mais específico e prático, (MURPHY; KAISER; HU, 2008) et. al trata da etapa de treinamento e teste para os modelos supervisionados e não supervisionados de aprendizado de máquina e destaca as principais diferenças entre as abordagens. Para o primeiro tipo, os autores conceituam que na fase de treinamento, o algoritmo analisa o conjunto de dados rotulados específico para treino, e o resultado dessa análise é um modelo que irá tentar realizar generalizações sobre como os atributos se relacionam com o rótulo dado a eles (Classificação). Já na fase de teste, o classificador é aplicado ao conjunto de dados específico para teste que, por sua vez, possui seus rótulos desconhecidos e o algoritmo tentar prever o rótulo de cada exemplo da base de dados.

Para algoritmos de aprendizado de máquina não supervisionados, as fases de treinamento e teste também estão presentes, porém, o conjunto de treinamento necessariamente não é rotulado. Nesse sentido, os autores discorrem que o modelo irá tentar aprender propriedades por si só, tais como distribuição numérica dos atributos que compõe a base de dados e como esses atributos se relacionam entre si.

A Figura 7 ilustra o processo de ”Classificação” dos algoritmos de aprendizado supervisionado do lado esquerdo, em que o modelo consegue determinar o rótulo das instâncias dos dados e assim, classificá-los. Já do lado direito da imagem, tem-se a representação da abordagem não supervisionada conhecida como ”Clusterização”, em que os dados são subdivididos e realocados em grupos de amostras a depender de suas características e similaridades, sem levar em conta qualquer noção de rótulo previamente conhecida.

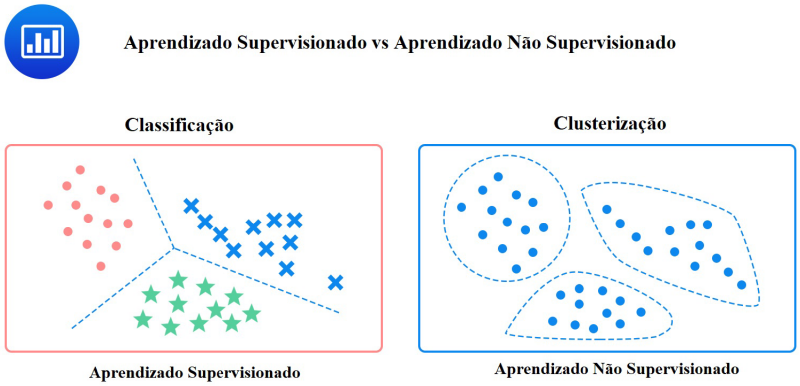

Figura 7 – Representação da abordagem supervisionada (Classificação) e não supervisionada (Clusterização). Adaptado de (ANALYSTPREP, 2021).

2.1.2.4 Validação do modelo

A etapa de validação do modelo é a parte em que o classificador terá seu desempenho analisado por meio do uso de diversas métricas amplamente utilizadas na área de aprendizado de máquina. (DALIANIS; DALIANIS, 2018) et. al destaca duas das principais métricas usadas para medir o desemprenho de um classificador, sendo elas a precisão e a revocação. A fim de se chegar aos valores das métricas, é preciso primeiro gerar a chamada "Matriz de Confusão" do modelo de classificação. Essa estrutura é basicamente uma tabela que contém os erros e acertos do classificador, ao se comparar com o que era o resultado esperado de fato.

A Figura 8 ilustra a estrutura de uma matriz de confusão. Caso o valor predito seja "Sim", isto é, pertence à classe, e o valor real também seja "Sim", tem-se então um valor classificado como "Verdadeiro Positivo". O mesmo raciocínio pode ser aplicado para as demais opções de resultados na matriz de confusão.

		Valor Obtido	
		Sim	**Não**
Valor Real	**Sim**	Verdadeiro Positivo (TP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (TN)

Figura 8 – Matriz de Confusão. Adaptado de (NOGARE, 2020).

2.1.2.5 Métricas utilizadas

Esta subseção traz a definição das métricas de validação utilizadas neste trabalho, sendo elas precisão, revocação, *F1-Score*, curva *ROC* e AUC.

A precisão mede o taxa de instâncias corretamente classificadas como positivo, dentre todas as que foram classificadas como positivo (o que inclui falsos positivos).

$$Precisão = \frac{TP}{TP + FP} \tag{2.2}$$

A revocação mede o taxa de instâncias corretamente classificadas como positivo, dentre todas que realmente eram positivas (o que inclui falsos negativos).

$$Revocação = \frac{TP}{TP + FN} \tag{2.3}$$

Outra métrica amplamente utiliza é a chamada *F1-Score* que representa basicamente uma maneira de unir os valores de precisão e revocação em uma única métrica. O *F1-Score* é calculado por meio da média harmônica entre precisão e revocação, portanto, caso se tenha um baixo valor para esta métrica, pode ser um indicativo de que o valor da precisão ou da revocação do classificador está baixo.

$$F1\text{-}Score = \frac{2 * Precisão * Revocação}{Precisão + Revocação} \tag{2.4}$$

Finalmente, outra ferramenta bastante empregada e que foi largamente usada neste trabalho é a curva *ROC* (Característica de operação do receptor). Esta, é uma representação gráfica em que no eixo X (horizontal), tem-se a taxa de falsos positivos e no eixo Y (vertical), tem-se a taxa de verdadeiros positivos. A curva *ROC* ilustra como esses valores variam na etapa de teste do modelo e mostra o quão bom é a capacidade de distinção do modelo.

Já a AUC (Área sob a curva) é o valor obtido ao se calcular a área sob a curva *ROC*, e varia de 0.0 até 1.0. Uma AUC de 0,5 representa um teste sem capacidade de distinção, isto é, o modelo seria praticamente aleatório, enquanto que um valor de AUC igual a 1.0, representa uma capacidade de distinção perfeita do modelo (HOO; CANDLISH; TEARE, 2017).

A Figura 9 traz um exemplo de curva *ROC* e *AUC* que foi gerada durante o desenvolvimento deste trabalho. Quanto maior o valor de *AUC* para um modelo de classificação, melhor seu desempenho na classificação do conjunto de teste.

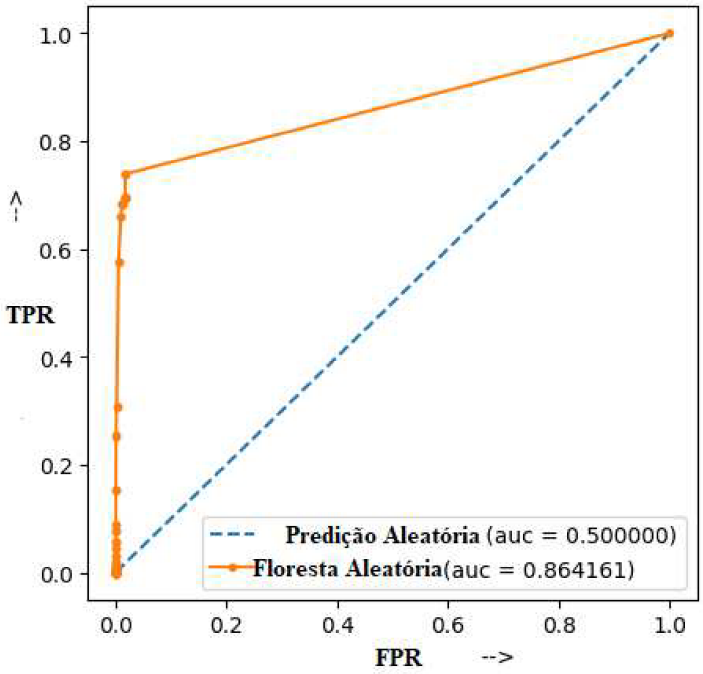

Figura 9 – Curva *ROC* e *AUC*. Fonte: Do Autor.

2.1.3 Sistemas de Detecção de Intrusão (*IDS*)

(DUQUE; OMAR, 2015) et. al define intrusão como sendo qualquer atividade que possa violar as políticas de segurança de uma rede. Nesse sentido, os autores conceituam *IDS* como sendo um mecanismo de *software* e *hardware* que é implantado com a finalidade de realizar a detecção de uso não autorizado ou ataques a um computador ou a uma rede. O uso desses sistemas tem por objetivo preencher lacunas de segurança deixadas por outras ferramentas como *firewalls* e antivírus. O trabalho também define uma vantagem obtida com o uso de *IDS*, que seria sua capacidade de documentar a intrusão e fornecer bases sobre padrões de ataques mais recentes por meio de *logs* dos sistemas.

De acordo com (ASHOOR; GORE, 2011) et. al, o principal papel dos sistemas de detecção de intrusão é ajudar os sistemas computacionais a lidar com ataques. Diante disso, autores trazem uma lista com as principais funções que os *IDS* podem desempenhar, que inclui:

- Monitoramento de atividades do sistema;
- Análise de vulnerabilidades;
- Reconhecimento de padrões típicos de ataques;
- Análise de comportamentos anormais de atividade;

2.1.3.1 Tipos de *IDS* (*Host*/Rede)

Os sistemas de detecção de intrusão podem ser divididos em 2 principais grupos: baseados em *host* e baseados em rede. O *IDS* baseado em *host* é caracterizado basicamente por monitorar características do *host* como uso da *CPU* (Unidade central de processamento), arquivos de sistema acessados e aplicativos. Já o *IDS* baseado em rede se concentra basicamente no monitoramento de tráfego e na identificação de comportamento malicioso na rede (MOLINA-CORONADO et al., 2020). Este segundo tipo é o foco de análise deste trabalho.

A Figura 10 traz uma representação genérica do funcionamento de sistemas de detecção de intrusão baseados em rede.

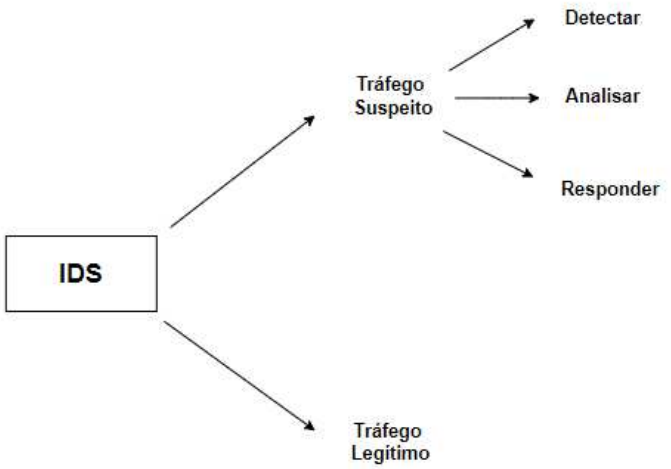

Figura 10 – Funcionamento básico de um IDS genérico baseado em rede. Adaptado de (NAKAMURA; GEUS, 2007).

Pode-se subdividir os *IDS* basicamente em duas categorias distintas, o *IDS* baseado em assinaturas e o *IDS* baseado em anomalias. (OTOUM; NAYAK, 2021).

O *IDS* baseado em assinaturas é um sistema que requer um banco de dados contendo informações sobre assinaturas de ataques já conhecidos. Desta forma, o sistema realiza comparações entre essas assinaturas e os pacotes de rede coletados periodicamente. Quando as características de um pacote correspondem a uma assinatura de ataque, o *IDS* gera um alerta e o operador humano pode tomar as ações necessárias.

(ASHOOR; GORE, 2011) et. al cita a eficácia deste tipo de *IDS* contra os ataques, mas também destaca a necessidade de atualizações regulares no banco de assinaturas e também a sua incapacidade em lidar com ataques desconhecidos (dia zero) e novas versões de ameaças.

A segunda categoria de *IDS* é o chamado *IDS* baseado em anomalias. Este, possui

como principal característica, tentar distinguir o comportamento considerado ”normal” de um comportamento considerado ”malicioso” em determinado sistema, por meio da análise dos pacotes de rede.

Esses sistemas geralmente são desenvolvidos com uso de aprendizado de máquina e são criados usando um conjunto de dados rotulado no modo de ”lotes”. Isso significa que o algoritmo de aprendizado de máquina é aplicado uma vez a um conjunto de dados de treinamento estático e, depois disso, o modelo produzido é usado para fazer previsões para os dados recebidos.

Uma vantagem do *IDS* baseado em anomalias em relação aos baseados em assinaturas, é que o primeiro tipo pode ser usado para detectar ataques novos ou desconhecidos, devido a sua característica de analisar o comportamento do sistema e não depender de uma base de dados prévia com assinaturas de ataques para determinar se um evento é legítimo ou não. (ASHOOR; GORE, 2011) et. al destaca essa característica desse tipo de *IDS*, mas informa que a taxa de resultados falsos positivos possui maior chance de se elevar, quando comparado com os sistemas baseados em assinaturas.

A capacidade de ”detecção de ataque desconhecido” do *IDS* baseado em anomalias depende muito da qualidade dos dados de treinamento e das semelhanças de ataques desconhecidos com os já conhecidos. Também é comum que o classificador seja criado de forma supervisionada, isto é, usando uma base de dados rotulada (tráfego normal e malicioso, por exemplo) de modo a adquirir as informações mais importantes a respeito das características ou atributos do ambiente de rede em que está inserido. O modelo gerado à partir das entradas representa o conjunto de regras que possibilitarão identificar um comportamento malicioso. Ao fim da fase de treinamento, o modelo é testado no conjunto de testes para ter seu desempenho analisado pelos profissionais envolvidos. O alvo deste trabalho é o *IDS* baseado em anomalias.

A Figura 11 ilustra o funcionamento básico dos *IDS* baseados em rede. No caso do *IDS* baseado em assinaturas, tem-se a etapa de verificação de padrões, a qual é realizada comparando o evento a ser detectado com uma base de dados contendo as assinaturas (representada pelo objeto ”*Security Rules*”). Posteriormente, com uma combinação correta de padrões de ataque, seria gerado o alerta para os profissionais responsáveis.

Já no caso do *IDS* baseado em anomalias, tem-se um perfil de atividade que é considerado como o padrão normal aceitável do sistema e, caso um pacote de rede seja interceptado e possua um comportamento diferente do padrão, o alerta de aviso será gerado.

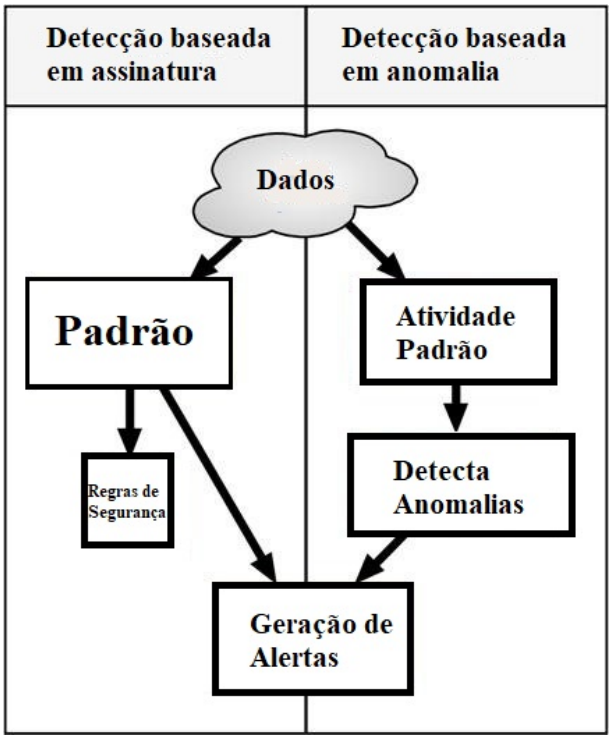

Figura 11 – Funcionamento básico dos tipos de *IDS* baseados em rede. Adaptado de (FEKOLKIN, 2015).

2.2 Trabalhos Relacionados

Louvieris, Clewley e Liu (2013) apresenta uma técnica para detectar ataques desconhecidos através da identificação de recursos de ataque. Essa técnica se baseia em agrupamento *k-means*, seleção de recursos Bayes ingênuo e métodos de classificação de árvore de decisão C4.5. Potnis et al. (POTNIS et al., 2022) foca na detecção de ataques do tipo *DDoS*. Os autores propõem um sistema híbrido de detecção de intrusão na web (*HWIDS*) (Sistema Híbrido de Detecção de Intrusão na Web) para detectar essa classe de ataque (CA). Cinco tipos de *DDoS* que visam a camada de aplicação são abordados, bem como suas variantes desconhecidas. O sistema proposto tem uma precisão de 93,48% e uma taxa de falso negativo de 6,52% na detecção de ataques desconhecidos. Alzubi et al. (ALZUBI; STAHL; GABER, 2021) apresenta um novo sistema baseado em conjunto chamado *"Unknown Network Attack Detector" (UNAD)* (Detector de ataque de rede desconhecido), que propõe um fluxo de trabalho de treinamento composto por técnicas de detecção de anomalias heterogêneas e não supervisionadas. De acordo com o estudo, essa abordagem funciona melhor ao detectar ataques desconhecidos e alcança resultados promissores.

Shin et al. (2021) aponta que, embora a detecção de anomalias seja uma boa abordagem para detectar ataques desconhecidos, os falsos positivos são altamente prováveis. O trabalho propõe uma forma híbrida de detecção de intrusão (baseada em assinatura e

baseada em anomalia) usando a técnica de Agrupamento Difuso, juntamente com outras como árvores de classificação e regressão (CART) para evitar esse problema.

Al-Zewairi, Almajali e Ayyash (2020) reitera que o problema de detectar ataques completamente desconhecidos em um sistema ainda é um campo de pesquisa em aberto, pois esses ataques representam um desafio complexo para qualquer *IDS*. O trabalho enfatiza que algumas definições para ataques desconhecidos são inconsistentes e propõe uma categorização em dois tipos de ataques (Tipo-A e Tipo-B). Neste caso, o primeiro tipo representa novos ataques, e o segundo representa ataques desconhecidos, mas em categorias já conhecidas. Experimentos foram realizados com *IDS* baseado em redes neurais para detectar ataques Tipo-A e Tipo-B como um problema de classificação binária. Os resultados em dois conjuntos de dados (*UNSW-NB15* e *Bot-IoT*) mostraram que os modelos avaliados (classificadores de rede neural artificial profundos e rasos) tinham medidas gerais de erro de generalização ruins e a taxa de erro de classificação para vários tipos de ataques desconhecidos foi de cerca de 50%.

Serinelli, Collen e Nijdam (2021) usa uma abordagem supervisionada para investigar a detecção de ataques desconhecidos nos seguintes conjuntos de dados: *KDD99*, *NSL-KDD* e *CIC-IDS2018*. Para simular um ataque desconhecido, os autores realizaram dois ataques simples (*DoS* usando a ferramenta *Hping* e *PortScan* usando a ferramenta *Nmap*) em um ambiente *VirtualBox*. Em seguida, eles gravaram os arquivos de captura de pacotes e os inseriram em três modelos de detecção de intrusão treinados com os seguintes algoritmos: Máquina de Vetor de Suporte, Floresta Aleatória e *XGBoost*. Os resultados mostram que, na maioria dos cenários, ambos os ataques apresentam erros de classificação.

Em alguns casos, quando o ataque desconhecido exibe um perfil de tráfego de rede semelhante aos ataques conhecidos, o modelo pode identificar o tipo de ataque corretamente. Em nosso trabalho, foi investigado a similaridade de diferentes tipos de ataque para entender se alguns ataques têm perfis de tráfego de rede semelhantes.

Ferreira e Antunes (2020) propõem uma avaliação do conjunto de dados *CIC-IDS2018* e comparam o desempenho de alguns algoritmos supervisionados e bioinspirados, são eles: *CLONALG* (Algoritmo de seleção clonal), Sistema imunológico artificial, Quantização Vetorial de Aprendizado e *Perceptron* Multicamadas com *Back-Propagation*. Eles também investigaram como essa abordagem pode lidar com alguns ataques desconhecidos. Os autores trabalham apenas com dois cenários: a) detectar diferentes tipos de ataque *DoS* (treinar com *GoldenEye* e testar com *Slowloris*), e b) treinar com dados de um tipo de ataque e testar com outro tipo (tráfego *DoS* x tráfego *DDoS*). Segundo os autores, o IDSproposto teve melhor desempenho para o cenário "b", quando se trata de identificar ataques desconhecidos.

Em comparação com os estudos mencionados acima, as seguintes diferenças podem ser encontradas em nosso trabalho: i) apenas alguns tipos de ataque específicos foram

investigados (*DoS/DDoS* e *PortScan*), ii) alguns trabalhos propuseram métodos específicos (técnicas híbridas - supervisionado/não supervisionado, por exemplo) adaptados para identificar ataques desconhecidos, iii) alguns trabalhos ainda usam conjuntos de dados desatualizados, como *KDD99* e *NSL-KDD* e iv) ausência de um *benchmark* para detecção de ataques desconhecidos usando modelo *IDS* supervisionado, construído com classificadores de Aprendizado de Máquina tradicionais, em vez de Aprendizado Profundo.

3 Desenvolvimento

Neste capítulo, serão apresentadas e detalhadas todas as etapas que compõe o desenvolvimento deste trabalho.

3.1 Visão Geral

As etapas do trabalho foram divididas da seguinte forma:

- Seleção do conjunto de dados a ser usado;
- Pré-processamento e tratamento dos dados;
- Seleção dos algoritmos de aprendizado de máquina;
- Elaboração dos experimentos de acordo com os diferentes tipos de ataques;
- Criação dos modelos de *IDS* usando os algoritmos selecionados;
- Avaliação e comparação dos modelos;

Na Seção 3.2, são discutidas questões acerca do conjunto de dados utilizado neste trabalho, bem como as técnicas de pré-processamento que foram empregadas. A Seção 3.3 trata da seleção de algoritmos que foi realizada e traz a lista de algoritmos que foram inicialmente utilizados. Finalmente, a Seção 3.4 contempla a metodologia experimental empregada, a fim de detalhar cada rodada de experimento conduzida neste trabalho, bem como a forma de avaliação de desempenho dos modelos que foi empregada.

3.2 Conjunto de dados e pré-processamento

Ao selecionar o conjunto de dados para os experimentos, vários fatores foram considerados como a disponibilidade de rótulos, diversidade de tipos de ataque e dados de ataques recentes (KENYON; DEKA; ELIZONDO, 2020). Diante disso, foi selecionado o conjunto de dados *CIC-IDS2017* (SHARAFALDIN; LASHKARI; GHORBANI, 2018), (BRUNSWICK, 2017), por ser uma das opções mais completas e consolidadas entre as opções para *IDS* (RING et al., 2019).

O conjunto de dados contém 2.830.743 registros compreendendo mais de 78 atributos de fluxo de rede referentes aos protocolos *TCP/IP* (Protocolo de controle de transmissão/Protocolo da Internet), com tráfego normal e ataques observados durante uma

semana (segunda a sexta). O conjunto de dados abrange sete classes de ataque e 16 tipos de ataque, incluindo também dados brutos de pacotes de rede (formato PCAP) e fluxos de rede rotulados como ataques normais (benignos). Há apenas um dia (segunda-feira) que conta apenas com tráfego normal.

Para melhor desempenho e correto funcionamento dos algoritmos de classificação, foram realizados os seguintes procedimentos de pré-processamento utilizando a linguagem Python, versão 3.9.10:

- Codificação de variáveis categóricas;
- Tratamento de valores infinitos e nulos;
- Normalização.

Em relação às variáveis categóricas, primeiro os rótulos benignos e de ataque foram convertidos em "0" e "1", respectivamente. Em seguida, o atributo "*DestinationPort*" foi manipulado. Este, representa a porta de destino solicitada pelo invasor, que exigiu tratamento adequado. Como resultado, foi criada uma coluna para as seguintes portas: 80, 53, 21, 22, 123 e 443. Essas portas são frequentemente visadas por invasores e associadas aos protocolos *HTTP* (Protocolo de transferência de hipertexto), *DNS* (Sistema de nomes de domínio), *FTP* (Protocolo de transferência de arquivo), *SSH* (*Shell* seguro), *NTP* (Protocolo de tempo de rede) e *HTTPS* (Protocolo de transferência de hipertexto seguro). Também foram criadas outras colunas para as seguintes portas *TCP/UDP*(Protocolo de controle de transmissão/Protocolo de datagrama do usuário): i) abaixo de 1024, mas não nos sete valores anteriores e ii) acima de 1024. Não há atributos para a porta de origem. Fizemos ambos os tratamentos para variáveis categóricas usando codificação *one-hot*, e então excluímos a coluna original.

Os fluxos de rede com valores infinitos e nulos foram removidos do modelo pois eram muito poucos em relação ao tamanho do conjunto de dados original. Por fim, foi aplicada a normalização de divisão por máximo em todos os atributos, exceto o "*DestinationPort*", dividindo o valor do atributo pelo valor mais alto em sua categoria.

3.3 Seleção dos Algoritmos

Foram escolhidos os seguintes algoritmos de classificação como candidatos potenciais para o desenvolvimento de um sistema de detecção de intrusão: Árvores de Decisão, Gradiente Descendente Estocástico, *Perceptron* Multicamadas, Bayes Ingênuo Gaussiano, K-vizinhos mais próximos, Floresta Aleatória, Máquina de Vetores de Suporte, Regressão Logística e Árvores Extremamente Aleatórias (*Extra-Trees*). Esses algoritmos represen-

tam alguns dos mais usados na literatura de *IDS* baseados em aprendizado de máquina supervisionado.

3.4 Elaboração dos Modelos de Detecção de Intrusão Supervisionados

O principal objetivo desta etapa do trabalho é avaliar empiricamente o desempenho de modelos de detecção de intrusão supervisionados quando apresentados a tipos de ataque desconhecidos. Em seguida, é apresentada a metodologia experimental e os experimentos conduzidos no conjunto de dados *CIC-IDS2017*.

3.4.1 Metodologia experimental

A metodologia experimental consiste nas seguintes etapas:

1. Organização de classes de ataque e estabelecimento de um *Baseline*;
2. Seleção de algoritmos de classificação;
3. Criação de diferentes conjuntos de treino/teste de acordo com as classes e tipos de ataque;
4. Avaliação de desempenho de modelos de detecção de intrusão supervisionados em diferentes cenários.

Cada instância de dados no conjunto de dados terá um rótulo B (benigno) ou A (ataque). As instâncias de ataque podem ser agrupadas em classes $C_1, C_2, ..., C_i$ onde i denota o número de classes de ataque. Cada classe é composta por um tipo $t_{i,j}$ onde j denota o número de tipos de ataque. Neste trabalho, foram investigadas três classes de ataques: *DoS*, Força Bruta e Ataques Web. Dentro de cada classe, temos vários tipos de ataque. A classe *DoS* é composta pelos ataques do tipo *DoS GoldenEye*, *DoS Hulk*, *DoS SlowHTTPTest*, *DoS SlowLoris* e *DDoS Loit*. A classe de Força Bruta consiste nos seguintes ataques: *FTP-Patator* e *SSH-Patator*. Finalmente, a classe de Ataques Web contém os tipos Ataque Web Força Bruta e Ataque Web XSS. O conjunto de dados *Baseline* consiste em amostras de ambos os rótulos B (benigno) e A (ataque). Foram selecionadas amostras aleatórias do *CIC-IDS2017* para compor cada turma. A Tabela 1 detalha a composição do conjunto de dados *Baseline*.

3.4.2 1ª rodada de experimentos: Utilização do *Baseline*

Foi avaliado o desempenho de cada algoritmo de classificação usando a amostra de dados *Baseline*, utilizando 90% do conjunto para treino e 10% para teste. O objetivo

Tabela 1 – Composição do *Baseline*

Tipo	Classe	Número de Amostras
Benigno	Benigno	174421
FTP-Patator	Força Bruta	794
SSH-Patator	Força Bruta	590
DDoS Loit	DoS	12803
DoS GoldenEye	DoS	1043
DoS Hulk	DoS	23107
DoS Slowhttptest	DoS	550
DoS slowloris	DoS	580
AW Força Bruta	Ataque Web	151
AW XSS	Ataque Web	65

era classificar corretamente amostras de ataque e amostras benignas (tarefa binária). Essa primeira rodada foi conduzida para se obter uma visão geral do desempenho dos algoritmos selecionados, para que fosse possível destacar o classificador com melhores resultados e usá-lo nas demais rodadas de experimentos.

Os resultados preliminares mostraram que o algoritmo Floresta Aleatória teve um bom desempenho em termos de Precisão, Revocação, *AUC* e tempo de treinamento. Seus números foram melhores do que aqueles obtidos por modelos mais sofisticados e complexos como as redes neurais e, por esse motivo, o classificador Floresta Aleatória foi selecionado para o restante dos experimentos. Vale ressaltar que o ambiente de desenvolvimento era formado pela linguagem de programação Python na versão 3.9.10, a biblioteca *scikit-learn* e parâmetros padrão para todos os algoritmos em todos os experimentos.

3.4.3 2ª rodada de experimentos: Classes de ataques individuais

Esta próxima etapa envolve a criação de diferentes conjuntos de treinamento/teste de acordo com as classes de ataque. Primeiro, foram criados seis conjuntos de dados para avaliar o desempenho do modelo de detecção de intrusão na detecção de classes de ataque desconhecidas. A regra de formação para esses conjuntos pode ser resumida da seguinte forma:

- Seleção de uma classe de ataque C_k;
- O conjunto de treinamento será composto de todas as amostras da classe de ataque selecionada C_k juntamente com amostras benignas;
- O conjunto de teste será composto por uma classe de ataque diferente C_j e amostras benignas.
- Repita o processo até que todas as classes de ataque sejam avaliadas nos conjuntos de treinamento e teste.

Por exemplo, se for selecionada a classe de ataque $C_1 = DoS$, teremos o conjunto de treinamento composto por todas as amostras *DoS* e o conjunto de teste composto pelas amostras de C_2 = Ataques Web, por exemplo. Em ambos os conjuntos (treinamento/teste), foi adicionado um número de amostras benignas proporcional ao número de amostras de ataque.

3.4.4 3ª rodada de experimentos: Tipos de ataques de mesma classe

Em seguida, foram criados outros 24 conjuntos de dados para observar os resultados de uma perspectiva mais detalhada, mudando o foco das classes de ataque para os tipos de ataques. A regra de formação desses conjuntos é semelhante à anterior, e pode ser resumida da seguinte forma:

- Selecione um tipo de ataque $t_{k,j}$;
- O conjunto de treinamento será composto de todas as amostras do tipo de ataque selecionado $t_{k,j}$ e também de amostras benignas;
- O conjunto de teste será composto por um tipo de ataque diferente da mesma classe de ataque $t_{k,l}$ e também por amostras benignas.
- Repita o processo até que todos os tipos de ataque dentro de cada classe de ataque sejam avaliados nos conjuntos de treinamento e teste.

Por exemplo, se for selecionado o tipo de ataque $t_{1,1} = DoS\ GoldenEye$, tem-se o conjunto de treinamento composto por todas as amostras *DoS GoldenEye* e o conjunto de teste composto pelas amostras de $t_{1,2} = DoS\ Hulk$, por exemplo. Em ambos os conjuntos (treinamento/teste), foi adicionado um número de amostras benignas proporcional ao número de amostras de ataque.

3.4.5 4ª e 5ª rodadas de experimentos: Classes de ataques combinadas e tipos de ataques de todas as classes

Finalmente, fez-se necessário a realização de duas novas rodadas de experimentos. Na 4ª etapa, foram criados novos conjuntos de dados de acordo com o seguinte raciocínio: 1) o que acontece se o conjunto de treinamento tiver várias classes/tipos de ataque e o conjunto de teste tiver apenas uma classe/tipo de ataque que não fazia parte do conjunto de treinamento? e 2) o que acontece se o conjunto de treinamento tiver apenas uma classe/tipo de ataque e o conjunto de treinamento tiver várias classes/tipos de ataque que não faziam parte do conjunto de treinamento?

A ideia aqui é entender se alguns ataques não vistos podem ser detectados usando o conhecimento prévio de outros ataques. No caso i), por exemplo, se o conjunto de treinamento for composto por $C_1 = DoS$ e $C_2 =$ Ataques Web, o conjunto de teste terá apenas amostras de ataque de $C_3 =$ Força Bruta (ou seja, a única classe de ataque que não fazia parte do conjunto de treinamento). No caso ii), por exemplo, se o conjunto de treinamento for composto por $C_1 = DoS$, o conjunto de teste terá amostras de ataque de $C_2 =$ Ataques Web e $C_3 =$ Força Bruta (ou seja, as classes de ataque que não fazem parte do conjunto de treinamento). A mesma lógica se aplica para a 5º rodada de experimentos, em que são considerados os tipos de ataque individualmente. Neste caso, pode-se ter por exmeplo, o conjunto de treino composto pelo tipo de ataque *DoS GoldenEye* e o conjunto de treino contendo todos os demais tipos de ataques de todas as 3 classes abordadas. O experimento é então realizado novamente invertendo-se os conjunto de treino e teste e, conforme discutido anteriormente, amostras benignas também fazem parte dos conjuntos.

3.4.6 Avaliação do modelo

Em todos os cenários, o modelo do classificador Floresta Aleatória foi avaliado usando as seguintes métricas: Precisão, Revocação e AUC. Nos experimentos, a classe positiva representa um ataque e a classe negativa representa um fluxo de rede benigno. Revocação representa a proporção de amostras positivas classificadas corretamente (ataques) para o total de exemplos positivos no conjunto de dados. A precisão quantifica o número de previsões de classes positivas que pertencem à classe positiva.

Finalmente, foram plotadas as curvas ROC de cada experimento. Como dito anteriormente, a curva ROC mostra a taxa de verdadeiros positivos (TPR) versus a taxa de falsos positivos (FPR) de um determinado modelo. A área sob a curva ROC (AUC) é um valor entre 0 e 1 e modelos eficientes têm valores de AUC mais próximos de 1.

4 Resultados e discussão

Neste capítulo, serão apresentados os resultados obtidos em cada rodada de experimentos, destacando-se casos em que resultados interessantes foram alcançados. A Tabela 2 resume todas as rodadas de experimentos que foram conduzidos neste trabalho. Como dito no capítulo anterior, a primeira rodada de experimentos foi conduzida para que fosse possível identificar o classificador com melhor desempenho, dentre todos os que foram citados anteriormente. O objetivo agora é analisar o desempenho do algoritmo selecionado (Floresta Aleatória) em todos os demais cenários abordados posteriormente.

Tabela 2 – Rodadas de experimentos conduzidas

Treino	Teste
90% Baseline	10% Baseline
1 classe de ataque	Outra classe de ataque
1 tipo de ataque de uma classe	Todos os outros ataques da classe
1 classe de ataque	Outras 2 classes de ataque (e vice-versa)
1 tipo de ataque de uma classe	Todos os outros ataques de todas as classes (e vice-versa)

4.1 Resultados da 2ª rodada de experimentos

A Tabela 3 mostra o desempenho do modelo supervisionado Floresta Aleatória ao detectar classes de ataque desconhecidas. As taxas de precisão, revocação e *AUC* para todos os experimentos diferem significativamente do experimento conduzido no conjunto *Baseline*. Por exemplo, um modelo de detecção de intrusão, utilizando o classificador Floresta Aleatória e treinado apenas com amostras de *DoS*, não pode detectar ataques desconhecidos relacionados a Ataques Web e Força Bruta. O mesmo ocorre ao analisar modelos de detecção de intrusão treinados com Ataques Web e Força Bruta. Os baixos valores de revocação para todos os experimentos indicam que os modelo supervisionado treinados com uma determinada classe de ataque não pode identificar uma nova classe de ataque (ainda não vista).

Tabela 3 – Desempenho do Floresta Aleatória de acordo com as classes de ataques

Treino	Teste	Precisão	Revocação	*AUC*
Baseline	Baseline	0.9257	0.9454	0.9791
DoS	Ataques Web	0.2303	0.0973	0.5663
DoS	Força Bruta	0	0	0.4742
Ataques Web	DoS	0.7442	0.0001	0.6361
Ataques Web	Força Bruta	0.0185	0.0001	0.4966
Força Bruta	DoS	0	0	0.4836
Força Bruta	Ataques Web	0	0	0.8366

4.2 Resultados da 3ª rodada de experimentos

A análise subsequente refere-se a investigar o comportamento do modelo de detecção de intrusão quando ele é treinado com ataques específicos dentro de uma classe, e o que acontece quando uma nova variante desse tipo de ataque é apresentada ao classificador. A Tabela 4 resume o desempenho do modelo de detecção de intrusão em relação ao tipo de ataque.

O cenário aqui é um pouco diferente em comparação com os experimentos que envolvem detectar uma nova classe de ataque. Ao analisar os 20 cenários dos tipos de ataque *DoS*, temos valores médios de 0,83, 0,41 e 0,91 para precisão, revocação e AUC, respectivamente. Este resultado indica que um modelo treinado com um tipo de *DoS* pode ser capaz de identificar um tipo de ataque *DoS* novo. Em alguns casos, o desempenho do modelo de detecção de intrusão é ainda melhor do que o *Baseline*, por exemplo, em um cenário onde temos *DoS Slowhttptest* no conjunto de treinamento e *DoS Slowloris* no conjunto de teste. Ambos os ataques estão relacionados à Camada de Aplicação (protocolo *HTTP*), neste caso. No entanto, em outros casos, o modelo de detecção de intrusão não conseguiu detectar um tipo de ataque *DoS* desconhecido, por exemplo, *DoS Slowloris* (treinamento) e *DDoS LOIT* (teste).

Foram encontrados resultados semelhantes para Ataques Web (AW). Em ambos os casos, AW Força Bruta e AW XSS, o modelo supervisionado pode detectar um ataque novo desse tipo. A única classe de ataque onde os modelos treinados não foram capazes de identificar ataques desconhecidos foi a classe Força Bruta. Os valores de precisão e revocação foram nulos em ambos os casos (*FTP-Patator e SSH-Patator*). Uma vez que ambos os ataques visam protocolos de aplicativos diferentes (*FTP* e *SSH*), a semelhança do tráfego de rede entre esses dois ataques é supostamente baixa.

Tabela 4 – Desempenho do Floresta Aleatória de acordo com os tipos de ataques

Treino	Teste	CA	Precisão	Revocação	*AUC*
Baseline	Baseline	-	0.925	0.945	0.979
DoS GoldenEye	DoS Hulk	DoS	0.988	0.336	0.947
DoS GoldenEye	DoS Slowhttptest	DoS	0.994	0.830	0.983
DoS GoldenEye	DoS Slowloris	DoS	0.997	0.830	0.994
DoS GoldenEye	DDoS LOIT	DoS	0	0	0.778
DoS Hulk	DoS GoldenEye	DoS	0.743	0.999	0.971
DoS Hulk	DoS Slowhttptest	DoS	1	0.022	0.981
DoS Hulk	DoS Slowloris	DoS	0.996	0.587	0.985
DoS Hulk	DDoS LOIT	DoS	0.999	0.168	0.999
DoS Slowhttptest	DoS GoldenEye	DoS	0.999	0.747	0.957
DoS Slowhttptest	DoS Hulk	DoS	0.979	0.209	0.979
DoS Slowhttptest	DoS Slowloris	DoS	0.997	0.995	0.997
DoS Slowhttptest	DDoS LOIT	DoS	0.972	0.138	0.972
DoS Slowloris	DoS GoldenEye	DoS	0.994	0.742	0.992
DoS Slowloris	DoS Hulk	DoS	0.976	0.352	0.968
DoS Slowloris	DoS Slowhttptest	DoS	0.988	0.952	0.993
DoS Slowloris	DDoS LOIT	DoS	0	0	0.946
DDoS LOIT	DoS GoldenEye	DoS	0	0	0.628
DDoS LOIT	DoS Hulk	DoS	0.978	0.034	0.742
DDoS LOIT	DoS Slowhttptest	DoS	0.991	0.044	0.838
DDoS LOIT	DoS Slowloris	DoS	0.996	0.214	0.653
FTP-Patator	SSH-Patator	FB	0	0	0.637
SSH-Patator	FTP-Patator	FB	0	0	0.746
AW Força Bruta	AW XSS	AW	0.934	0.998	0.983
AW XSS	AW Força Bruta	AW	0.852	0.983	0.975

4.3 Resultados da 4ª e 5ª rodada de experimentos

Os resultados desses novos experimentos mostraram que, mesmo quando o modelo de detecção de intrusão era treinado com várias classes/tipos de ataque, ainda era um desafio para o modelo detectar ataques não vistos. Apenas dois cenários estavam próximos do *Baseline*. No primeiro, tem-se um conjunto de treinamento composto por todos os tipos de ataque de todas as 3 classes, exceto o tipo *DoS Slowloris*, e o conjunto de teste é composto apenas por ataques *DoS Slowloris*. Neste cenário, o Floresta Aleatória atingiu valores superiores a 90% para precisão, revocação e *AUC*. No segundo, tem-se um conjunto de treinamento composto por todos os tipos de ataques de todas as 3 classes exceto o tipo AW Força Bruta, e o conjunto de teste composto apenas por AW Força Bruta. Neste cenário, tivemos valores superiores a 90% para precisão e *AUC* e 83% para revocação. As Tabelas 5 e 6 trazem os resultados obtidos pelo algoritmo Floresta Aleatória na 4ª e 5ª rodada de experimentos, respectivamente.

Tabela 5 – Desempenho do Floresta Aleatória na 4ª rodada de experimentos

Treino	Teste	Precisão	Revocação	*AUC*
Baseline	Baseline	0.9257	0.9454	0.9791
DoS + Força Bruta	AW	0.0101	0.1183	0.5612
DoS + AW	Força Bruta	0.0001	0.0000	0.5495
AW + Força Bruta	DoS	0.0001	0.7347	0.8642
AW	DoS + Força Bruta	0.0026	0.9562	0.6504
Força Bruta	DoS + AW	0.0000	0.0000	0.4455
DoS	AW + Força Bruta	0.0089	0.0135	0.5695

Tabela 6 – Desempenho do Floresta Aleatória na 5ª rodada de experimentos

Treino	Teste	Precisão	Revocação	*AUC*
Baseline	Baseline	0.9257	0.9454	0.9791
TA – DoS GoldenEye	DoS GoldenEye	0.2152	0.9741	0.9849
DoS GoldenEye	TA – DoS GoldenEye	0,3290	0.9777	0.8377
TA – DoS Hulk	DoS Hulk	0.2447	0.9986	0.8570
DoS Hulk	TA – DoS Hulk	0.1928	0.8644	0.7119
TA – DoS Slowhttptest	DoS Slowhttptest	0.2862	0.9308	0.9699
DoS Slowhttptest	TA – DoS Slowhttptest	0.0381	0.8617	0.6958
TA – DoS Slowloris	DoS Slowloris	0.9379	0.9560	0.9875
DoS Slowloris	TA – DoS Slowloris	0.0397	0.7925	0.9380
TA – DDoS LOIT	DDoS LOIT	0.6333	0.9808	0.8887
DDoS LOIT	TA – DDoS LOIT	0.0206	0.9856	0.7393
TA – FTP-Patator	FTP-Patator	0.0000	0.0000	0.8954
FTP-Patator	TA – FTP-Patator	0.0000	0.0000	0.5808
TA – SSH-Patator	SSH-Patator	0.0000	0.0000	0.8842
SSH-Patator	TA – SSH-Patator	0.0000	0.0000	0.4855
TA – AW Força Bruta	AW Força Bruta	0.9005	0.8371	0.9594
AW Força Bruta	TA – AW Força Bruta	0.0062	0.5413	0.8504
TA – AW XSS	AW XSS	0.1288	0.0000	0.4868
AW XSS	TA – AW XSS	0.0059	0.5334	0.7600

5 Conclusão

Neste trabalho, foi proposto um estudo de *IDS* baseado em anomalias e desenvolvido com técnicas de aprendizado de máquina sobre sua capacidade de detectar ataques desconhecidos. As descobertas mostram que um modelo supervisionado treinado com um tipo de ataque específico pode identificar amostras não presentes na base de treinamento do mesmo tipo de ataque em algumas situações. Por exemplo, modelos supervisionados treinados com *DoS GoldenEye* podem identificar amostras de ataque do tipo *DoS Slowloris*. Foi concluído também que os modelos supervisionados não detectaram com eficiência classes de ataque não vistas. Dois experimentos diferentes indicam isso: i) quando um conjunto de treinamento é composto por uma classe de ataque e um conjunto de teste possui uma classe de ataque diferente, e ii) quando um conjunto de treinamento é composto por várias classes de ataque e um conjunto de teste possui um ataque ainda não visto.

Os resultados do trabalho indicam que: i) o desempenho dos modelos *IDS* supervisionados está diretamente relacionado aos ataques apresentados no conjunto de treinamento, ii) os modelos IDS supervisionados podem ser bem sucedidos na detecção de variantes desconhecidas/invisíveis dos mesmos tipos de ataque, por exemplo, *DoS* e alguns Ataques Web e iii) devido à potencial similaridade entre alguns tipos de ataque, pode ser interessante desenvolver modelos de detecção de intrusão sob medida para eles. Tais resultados podem servir como um norte para novos trabalhos que adotam sistemas de detecção supervisionados.

Trabalhos futuros podem incluir a realização de experimentos semelhantes em conjuntos de dados como *CSE-CIC-IDS2018*, *UNSW-NB15* e *UGR'16*. Também pode-se aplicar métodos de transferência de aprendizado para avaliar a capacidade do *IDS* baseado em aprendizado de máquina em detectar ataques desconhecidos coletados à partir de diferentes conjuntos de dados.

Referências

AHMAD, T.; AZIZ, M. N. Data preprocessing and feature selection for machine learning intrusion detection systems. **ICIC Express Lett**, v. 13, n. 2, p. 93–101, 2019. Citado na página 17.

AL-ZEWAIRI, M.; ALMAJALI, S.; AYYASH, M. Unknown security attack detection using shallow and deep ann classifiers. **Electronics**, MDPI, v. 9, n. 12, 2020. Citado na página 26.

ALI, S.; REHMAN, S. U.; IMRAN, A.; ADEEM, G.; IQBAL, Z.; KIM, K.-I. Comparative evaluation of ai-based techniques for zero-day attacks detection. **Electronics**, MDPI, v. 11, n. 23, p. 3934, 2022. Citado na página 15.

ALZUBI, S.; STAHL, F.; GABER, M. M. Towards intrusion detection of previously unknown network attacks. **Communications of the ECMS**, ECMS, v. 35, n. 1, p. 35–41, 2021. Citado na página 25.

AMIN. **Train-Validation-Test**. 2022. Urlhttps://velog.io/@amin/Train-Validation-Test. Citado 2 vezes nas páginas 4 e 18.

ANALYSTPREP. **Supervised Machine Learning, Unsupervised Machine Learning, and Deep Learning**. 2021. Urlhttps://analystprep.com/study-notes/cfa-level-2/quantitative-method/supervised-machine-learning-unsupervised-machine-learning-deep-learning/. Citado 2 vezes nas páginas 4 e 20.

ASHOOR, A. S.; GORE, S. Importance of intrusion detection system (ids). **International Journal of Scientific and Engineering Research**, v. 2, n. 1, p. 1–4, 2011. Citado 3 vezes nas páginas 22, 23 e 24.

BENVENUTO, F.; PIANA, M.; CAMPI, C.; MASSONE, A. M. A hybrid supervised/unsupervised machine learning approach to solar flare prediction. **The Astrophysical Journal**, IOP Publishing, v. 853, n. 1, p. 90, 2018. Citado na página 19.

BIJU, J. M.; GOPAL, N.; PRAKASH, A. J. Cyber attacks and its different types. **International Research Journal of Engineering and Technology**, v. 6, n. 3, p. 4849–4852, 2019. Citado 2 vezes nas páginas 12 e 13.

BLOG, S. **The Ultimate Guide to Zero-Day Vulnerability Exploits Attacks**. 2022. Https://securityboulevard.com/2022/05/the-ultimate-guide-to-zero-day-vulnerability-exploits-attacks/. Citado 2 vezes nas páginas 4 e 16.

BRUNSWICK, U. o. N. B. University of N. **Intrusion Detection Evaluation Dataset (CIC-IDS2017)**. 2017. <https://www.unb.ca/cic/datasets/ids-2017.html>. Citado na página 28.

CERT.br. **Cartilha de segurança para Internet**. 2021. Disponível em: https://cartilha.cert.br, acesso em 29/03/2021. Citado na página 10.

DALIANIS, H.; DALIANIS, H. Evaluation metrics and evaluation. **Clinical text mining: secondary use of electronic patient records**, Springer, p. 45–53, 2018. Citado na página 20.

DUQUE, S.; OMAR, M. N. bin. Using data mining algorithms for developing a model for intrusion detection system (ids). **Procedia Computer Science**, Elsevier, v. 61, p. 46–51, 2015. Citado na página 22.

ENGINE, M. **What is a brute force attack?** 2023. Https://www.manageengine.com/log-management/cyber-security-attacks/what-is-brute-force-attack.html. Citado 2 vezes nas páginas 4 e 14.

FEKOLKIN, R. Intrusion detection & prevention system: overview of snort & suricata. **Internet Security, A7011N, Lulea University of Technology**, p. 1–4, 2015. Citado 2 vezes nas páginas 4 e 25.

FERREIRA, P.; ANTUNES, M. Benchmarking behavior-based intrusion detection systems with bio-inspired algorithms. In: SPRINGER. **International Symposium on Security in Computing and Communication**. [S.l.], 2020. p. 152–164. Citado na página 26.

HOO, Z. H.; CANDLISH, J.; TEARE, D. **What is an ROC curve?** [S.l.]: BMJ Publishing Group Ltd and the British Association for Accident ..., 2017. 357–359 p. Citado na página 21.

JONGSUEBSUK, P.; WATTANAPONGSAKORN, N.; CHARNSRIPINYO, C. Network intrusion detection with fuzzy genetic algorithm for unknown attacks. In: IEEE. **The International Conference on Information Networking 2013 (ICOIN)**. [S.l.], 2013. p. 1–5. Citado na página 10.

KENYON, A.; DEKA, L.; ELIZONDO, D. Are public intrusion datasets fit for purpose characterising the state of the art in intrusion event datasets. **Computers & Security**, Elsevier, p. 102022, 2020. Citado na página 28.

LI, Y.; LIU, Q. A comprehensive review study of cyber-attacks and cyber security; emerging trends and recent developments. **Energy Reports**, Elsevier, v. 7, p. 8176–8186, 2021. Citado na página 10.

LOUVIERIS, P.; CLEWLEY, N.; LIU, X. Effects-based feature identification for network intrusion detection. **Neurocomputing**, Elsevier, v. 121, p. 265–273, 2013. Citado na página 25.

MANNILA, H. Data mining: machine learning, statistics, and databases. In: IEEE. **Proceedings of 8th International Conference on Scientific and Statistical Data Base Management**. [S.l.], 1996. p. 2–9. Citado na página 16.

MEDIA, S. **Ataque de negação de serviço – DDoS**. 2023. Urlhttps://servermedia.com.br/blog/?p=849. Citado 2 vezes nas páginas 4 e 13.

MOLINA-CORONADO, B.; MORI, U.; MENDIBURU, A.; MIGUEL-ALONSO, J. Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process. **IEEE Transactions on Network and Service Management**, IEEE, v. 17, n. 4, p. 2451–2479, 2020. Citado na página 23.

MURPHY, C.; KAISER, G. E.; HU, L. Properties of machine learning applications for use in metamorphic testing. 2008. Citado na página 19.

NAKAMURA, E. T.; GEUS, P. L. de. **Segurança de redes em ambientes cooperativos**. [S.l.]: Novatec Editora, 2007. Citado 2 vezes nas páginas 4 e 23.

NOGARE, D. **Performance de Machine Learning – Matriz de Confusão**. 2020. Urlhttps://diegonogare.net/2020/04/performance-de-machine-learning-matriz-de-confusao/. Citado 2 vezes nas páginas 4 e 20.

OTOUM, Y.; NAYAK, A. As-ids: Anomaly and signature based ids for the internet of things. **Journal of Network and Systems Management**, Springer, v. 29, n. 3, p. 1–26, 2021. Citado na página 23.

PARK, J.; KIM, J.; GUPTA, B.; PARK, N. Network log-based ssh brute-force attack detection model. **CMC-Computers Materials & Continua**, TECH SCIENCE PRESS 871 CORONADO CENTER DR, SUTE 200, HENDERSON, NV 89052 USA, v. 68, n. 1, p. 887–901, 2021. Citado na página 13.

POTNIS, M. S.; SATHE, S. K.; TUGAONKAR, P. G.; KULKARNI, G. L.; DESHPANDE, S. S. Hybrid intrusion detection system for detecting ddos attacks on web applications using machine learning. In: **ICT Analysis and Applications**. [S.l.]: Springer, 2022. p. 797–805. Citado na página 25.

RING, M.; WUNDERLICH, S.; SCHEURING, D.; LANDES, D.; HOTHO, A. A survey of network-based intrusion detection data sets. **Computers & Security**, Elsevier, v. 86, p. 147–167, 2019. Citado na página 28.

SARMAH, U.; BHATTACHARYYA, D.; KALITA, J. K. A survey of detection methods for xss attacks. **Journal of Network and Computer Applications**, Elsevier, v. 118, p. 113–143, 2018. Citado na página 14.

SERINELLI, B. M.; COLLEN, A.; NIJDAM, N. A. On the analysis of open source datasets: validating ids implementation for well-known and zero day attack detection. **Procedia Computer Science**, Elsevier, v. 191, p. 192–199, 2021. Citado na página 26.

SHARAFALDIN, I.; LASHKARI, A. H.; GHORBANI, A. A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: **ICISSp**. [S.l.: s.n.], 2018. p. 108–116. Citado 2 vezes nas páginas 10 e 28.

SHIN, G.-Y.; KIM, D.-W.; KIM, S.-S.; HAN, M.-M. Unknown attack detection: Combining relabeling and hybrid intrusion detection. **CMC-COMPUTERS MATERIALS CONTINUA**, TECH SCIENCE PRESS, 2021. Citado na página 25.

SONG, J.; OHBA, H.; TAKAKURA, H.; OKABE, Y.; OHIRA, K.; KWON, Y. A comprehensive approach to detect unknown attacks via intrusion detection alerts. In: SPRINGER. **Annual Asian Computing Science Conference**. [S.l.], 2007. p. 247–253. Citado na página 10.

STOLFO, S. J.; FAN, W.; LEE, W.; PRODROMIDIS, A.; CHAN, P. K. Cost-based modeling for fraud and intrusion detection: Results from the jam project. In: IEEE. **Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00**. [S.l.], 2000. v. 2, p. 130–144. Citado na página 10.

TAKASE, K. **Você está interessado em Cross-Site Scripting (XSS)?** 2021. Urlhttps://kixoqxq59.medium.com. Citado 2 vezes nas páginas 4 e 15.

UÇAR, M. K.; NOUR, M.; SINDI, H.; POLAT, K. The effect of training and testing process on machine learning in biomedical datasets. **Mathematical Problems in Engineering**, Hindawi, v. 2020, 2020. Citado na página 17.

XU, M.-F.; LI, X.-H.; MIAO, M.-X.; ZHONG, C.; MA, J.-F. An unknown attack detection scheme based on semi-supervised learning and information gain ratio. **Journal of Internet Technology**, v. 20, n. 2, p. 629–636, 2019. Citado na página 10.

ZHANG, Z.; ZHANG, Y.; GUO, D.; SONG, M. A scalable network intrusion detection system towards detecting, discovering, and learning unknown attacks. **International Journal of Machine Learning and Cybernetics**, Springer, p. 1–17, 2021. Citado na página 10.