

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
PRÓ-REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Paulo Ericson Valentim Silva

**AI/IDS Um framework para detecção de intrusão híbrido utilizando
Transformer em redes emuladas**

Fortaleza – CE

21 de janeiro de 2025

Paulo Ericson Valentim Silva

AI/IDS Um framework para detecção de intrusão híbrido utilizando Transformer em redes emuladas

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Linha de Pesquisa: Inteligência Artificial

Orientador: Reinaldo Bezerra Braga

Coorientador: Antonio Wendell de Oliveira Rodrigues

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação (PRPI)

Programa de Pós-Graduação em Ciência da Computação (PPGCC)

Fortaleza – CE

21 de janeiro de 2025

Dados Internacionais de Catalogação na Publicação
Instituto Federal do Ceará - IFCE
Sistema de Bibliotecas - SIBI
Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

- V155a Valentim Silva, Paulo Ericson.
 AI/IDS Um framework para detecção de intrusão híbrido utilizando Transformer em redes emuladas /
 Paulo Ericson Valentim Silva. - 2025.
 83 f. : il. color.
- Dissertação (Mestrado) - Instituto Federal do Ceará, Mestrado em Ciência da Computação, Campus
 Fortaleza, 2025.
 Orientação: Prof. Dr. Reinaldo Bezerra Braga.
 Coorientação: Prof. Dr. Antonio Wendell de Oliveira Rodrigues.
1. Sistemas de Detecção de Intrusão. 2. Ataques cibernéticos. 3. Deep Learning. I. Título.
- CDD 004
-



Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)
Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação (PRPI)
Programa de Pós-Graduação em Ciência da Computação (PPGCC)

Paulo Ericson Valentim Silva

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, sendo aprovada pela Coordenação do Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará e pela banca examinadora:

Orientador: Reinaldo Bezerra Braga

Instituto Federal de Educação, Ciência e Tecnologia do
Ceará (IFCE)

**Coorientador: Antonio Wendell de Oliveira
Rodrigues**

Instituto Federal de Educação, Ciência e Tecnologia do
Ceará (IFCE)

Membro da Banca: Diego Aguiar Sousa

Instituto Federal de Educação, Ciência e Tecnologia do
Ceará (IFCE)

Membro da Banca: Claudio de Castro Monteiro

Instituto Federal de Educação, Ciência e Tecnologia do
Tocantins (IFTO)

Fortaleza – CE
21 de janeiro de 2025

Resumo

Recentemente, é notável o aumento no volume de tráfego na Internet, impulsionado pela pandemia global e pela crescente importância das interações online. Esse crescimento tem acarretado um aumento significativo no número de crimes cibernéticos. O presente trabalho realiza uma revisão sistemática da literatura no âmbito dos sistemas de detecção de intrusão, abordando os principais conjuntos de dados e técnicas empregadas para detectar intrusões em redes de computadores. O objetivo é fornecer uma visão abrangente do estado atual da detecção de intrusão, destacando abordagens eficazes e desafios persistentes. Um framework completo é proposto para detecção de intrusão híbrido utilizando transformer em redes emuladas. Além disso, são apresentadas orientações para pesquisas futuras na área de detecção de intrusão, visando aprimorar ainda mais a segurança cibernética em face das crescentes ameaças.

Palavras-chave: Ataques cibernéticos. Detecção de intrusão. *Deep Learning*.

Abstract

Recently, the volume of Internet traffic has increased significantly, driven by the global pandemic and the growing importance of online interactions. This growth has led to a significant increase in the number of cybercrimes. This search performs a systematic review of the literature in the field of intrusion detection systems, addressing the main datasets and techniques used to detect intrusions in computer networks. The objective is to provide a comprehensive overview of the current state of intrusion detection, highlighting strategic approaches and persistent challenges. A framework is proposed for hybrid intrusion detection using transformer in emulated networks. Furthermore, directions for future research in the area of intrusion detection are presented, further innovations in cybersecurity in the face of increasing novelties.

Keywords: Cyber attacks. Machine learning. Deep learning.

Lista de ilustrações

Figura 1 – Subdomínios da Inteligência Artificial.	20
Figura 2 – Modelo de arquitetura de um Transformer.	23
Figura 3 – Modelo de descoberta de conhecimento em bases de dados.	24
Figura 4 – Diagrama Prisma adaptado pelo autor.	34
Figura 5 – Conjuntos de dados mais utilizados.	38
Figura 6 – Etapas adaptadas do processo KDD utilizado na metodologia.	42
Figura 7 – Topologia do experimento.	43
Figura 8 – Fluxo de execução para gerar o <i>dataset</i>	45
Figura 9 – Etapas de pré-processamento.	46
Figura 10 – Fluxograma do experimento.	47
Figura 11 – Matriz de confusão normalizada do <i>AI/IDS Transformer</i> com o CSE- CIC-IDS2018.	52
Figura 12 – Variação da perda ao longo das épocas durante o treinamento do modelo.	53
Figura 13 – Interface gráfica desenvolvida em <i>Python</i>	57
Figura 14 – Tela de configurações do emulador de rede GNS3.	71
Figura 15 – Ferramenta <i>Kali Linux</i>	72
Figura 16 – Fluxo de criação de pacotes no <i>Scapy</i>	72
Figura 17 – Tela inicial do <i>CICFlowMeter</i>	73
Figura 18 – Fluxo geral do experimento.	80

Lista de tabelas

Tabela 1	– Ataques e recursos do conjunto de dados CSE-CIC-IDS2017.	24
Tabela 2	– Estatísticas de registros redundantes no KDD Train.	27
Tabela 3	– Estatísticas de registros redundantes no KDD Test.	27
Tabela 4	– Tabela PICO com referências para protocolo de busca.	31
Tabela 5	– Protocolo de busca utilizados nos motores de busca.	32
Tabela 6	– Quantidade de estudos por ano de publicação.	35
Tabela 7	– Quantidade de publicações por tipo no indexador IEEE Xplore.	35
Tabela 8	– Técnicas de aprendizagem mais utilizadas nos estudos.	36
Tabela 9	– Conjuntos de dados mais utilizados nos estudos.	38
Tabela 10	– Comparativo dos trabalhos relacionados.	41
Tabela 11	– Comparação entre os trabalhos discutidos na revisão sistemática.	55

Sumário

1	INTRODUÇÃO	10
1.1	Motivação para a Dissertação	10
1.2	Descrição do Problema	13
1.3	Objetivos Geral e Específicos	14
1.4	Contribuições	14
1.5	Estrutura da Dissertação	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Sistemas de Detecção de Intrusão (IDS)	16
2.1.1	Sistema de Rede de Detecção de Intrusão	17
2.1.2	Sistema Hospedeiro de Detecção de Intrusão	18
2.1.3	Híbridos	18
2.1.4	Tipos de Detecção	18
2.2	Aprendizagem de Máquina	19
2.3	Transformer	21
2.4	Processo de Descoberta de Conhecimento	22
2.5	Conjunto de Dados CSE-CIC-IDS2017	24
2.6	Conjunto de Dados CSE-CIC-IDS2018	25
2.7	Conjunto de Dados KDD Cup 99	26
2.8	Conjunto de Dados NSL-KDD	27
2.9	Conjunto de Dados UNSWNB15	28
3	TRABALHOS RELACIONADOS	29
3.1	Pesquisa de trabalhos similares	30
3.1.1	Planejamento	30
3.1.2	Protocolo de busca	31
3.1.3	Critérios de Inclusão/Exclusão	32
3.1.4	Condução	33
3.2	Respostas das Questões de Pesquisa	34
3.3	Ameaças à Validade	39
3.4	Comparativo dos Trabalhos Relacionados	39
4	FRAMEWORK DE TESTES	42
4.1	Visão Geral	42
4.2	Rede de Ataque	43
4.3	IDS	44

4.4	DMZ e LAN	44
4.5	Base de Dados	45
4.6	Pré-processamento	45
5	MODELO AI/IDS TRANSFORMER	47
5.1	Transformer Layer	47
5.2	Métricas de avaliação	49
6	RESULTADOS E DISCUSSÃO	50
6.1	Experimentos	50
6.2	Avaliação do Framework	50
6.3	Comparativo dos trabalhos	51
6.4	Experimento no Ambiente Virtual	55
6.5	Experimento com Interface Gráfica	56
7	CONCLUSÃO	58
	REFERÊNCIAS	59
	APÊNDICE A – FERRAMENTAS DE SOFTWARE	70
A.1	Emulador GNS3	70
A.2	Kali Linux	70
A.3	Scapy	71
A.4	CICFlowMeter	73
	ANEXOS	74
	ANEXO A – RECURSOS CICFLOWMETER	75
	ANEXO B – FLUXO GERAL DE EXECUÇÃO	79
	ANEXO C – RESULTADOS DOS ARTIGOS SELECIONADOS	81
	ANEXO D – TIPOS DE ATAQUES	82
D.0.1	Ataque de Força Bruta	82
D.0.2	Ataque Heartbleed	82
D.0.3	Botnet	82
D.0.4	Negação de Serviço	82
D.0.5	Negação de Serviço Distribuída	82
D.0.6	Ataques WEB	83
D.0.7	Infiltração da Rede	83

1 Introdução

Neste primeiro capítulo, são apresentados os principais elementos e discussões que contextualizam, fundamentam e motivam o desenvolvimento deste trabalho. Além disso, são apresentados o objetivo geral e os objetivos específicos a serem alcançados e, ao final, como a dissertação está estruturada.

1.1 Motivação para a Dissertação

O uso crescente da Internet, das aplicações em rede e dos serviços em nuvem torna os sistemas mais vulneráveis a ameaças externas que podem comprometer os principais princípios de segurança: confidencialidade, integridade, disponibilidade e autenticidade (LEON; MARKOVIC; PUNNEKKAT, 2022).

Como resultado, houve também um aumento significativo no número de crimes cibernéticos, exercendo uma influência notoriamente prejudicial em várias esferas. Conforme indicado por um estudo realizado pela *McAfee*, uma empresa especializada em segurança digital, em 2020, as invasões e delitos cibernéticos causaram danos que excederam a marca de US\$ 1 trilhão de dólares (MCAFEE, 2020).

Nessa perspectiva, o relatório anual de tendências de rede da *Cisco* destaca a segurança dos dispositivos de rede como um aspecto crucial e cada vez mais indispensável na atual paisagem tecnológica. A empresa ressalta a importância de priorizar a proteção das redes diante de um cenário em constante transformação (CISCO, 2022).

Esse crescimento, impulsionado pelo contínuo avanço da Internet e das tecnologias de rede, fez da segurança cibernética um campo de pesquisa essencial. Isso abrange a implementação de soluções como software antivírus, *firewalls* e sistemas de detecção de intrusão (IDS) para salvaguardar sistemas contra ameaças tanto internas quanto externas (SURAKHI et al., 2021). Uma intrusão ocorre quando um invasor tenta obter entrada ou interromper as operações normais de um sistema de informação, quase sempre com o intuito de fazer o mal (AMBATI; VIDYARTHI, 2013).

Uma ferramenta importante nesse contexto é o IDS, um tipo de sistema de gerenciamento de segurança para computadores e redes. Ele funciona coletando e registrando dados dos usuários, analisando aspectos como comportamentos padrão, fluxo de informações, horários de acesso e outras variáveis, a fim de identificar possíveis atividades suspeitas ou não autorizadas. Usando essas informações, em conjunto com o conhecimento prévio de padrões de ataques conhecidos, é possível distinguir se o evento em questão possui ou não intenções maliciosas (MEHRA, 2012).

Em relação aos IDSs, o trabalho desenvolvido por (VINOD; PRASAD, 2023) cita duas variações comuns: o Sistema de Detecção de Intrusão em Hospedeiros (HIDS, *Host Intrusion Detection System*), que se concentra na análise de ações específicas realizadas em sistemas individuais, como acessos a arquivos, alterações em configurações e registros de atividades. Por outro lado, o sistema de detecção de intrusão em rede (NIDS), ou *Network Intrusion Detection System*, tem sua ênfase na análise do fluxo de informações que circula pela rede, com o objetivo de identificar padrões de comportamento suspeitos.

Ambas as abordagens desempenham papéis cruciais na segurança cibernética, protegendo contra intrusões e ameaças em diferentes níveis. Em geral, existem duas técnicas principais de detecção para IDS, ou seja, baseada em assinatura e baseada em anomalia (TAYYEBI; BHILARE, 2018).

Um IDS baseado em assinatura tenta definir um conjunto de regras (ou assinaturas) que podem ser usadas para decidir se um determinado padrão é o de um intruso. Como resultado, os sistemas baseados em assinaturas são capazes de atingir altos níveis de precisão e um número mínimo de falsos positivos na identificação de intrusões. Pouca variação nos ataques conhecidos também pode afetar a análise se um sistema de detecção não estiver configurado adequadamente (BROWN; SUCKOW; WANG, 2002).

Em contraste, a detecção de anomalias, concentra-se na identificação de eventos que desviam do padrão de comportamento normal de um sistema. A abordagem baseada em anomalias implica a coleta de dados relacionados ao comportamento de usuários legítimos ao longo de um período de tempo, seguida pela aplicação de testes estatísticos ao comportamento observado para determinar sua legitimidade. Esta abordagem possui a vantagem de detectar ataques que não foram previamente identificados.

O elemento crucial para empregar essa abordagem com eficácia é a criação de regras de tal maneira que possam reduzir a taxa de alarmes falsos tanto para ataques desconhecidos quanto para aqueles já conhecidos (MODI et al., 2013). A detecção de comportamento anômalo é uma tarefa de análise de dados que busca identificar padrões emergentes e suspeitos (MARIR; WANG; FENG, 2019).

O IDS baseado em aprendizado estabelece um sistema para identificar categorias de ataques com base no comportamento normal e aprendido dos ataques. Além disso, as técnicas de Inteligência Artificial (IA) se destacam por sua capacidade de identificar semelhanças em vastos conjuntos de dados. Essas abordagens prometem a automação da detecção com base em dados, permitindo a inferência de informações sobre tráfego de rede malicioso a partir do extenso volume de registros de tráfego disponíveis (Mishra et al., 2019).

No entanto, a maioria das técnicas usadas nos IDS atuais não conseguem lidar com a natureza dinâmica e complexa dos ataques cibernéticos em redes de computadores.

Devido ao desenvolvimento diário desses ataques maliciosos, os sistemas de segurança de rede que são utilizados atualmente permanecem insuficientes para proteger os sistemas de computador (OZKAN-OKAY et al., 2021).

A análise de grandes quantidades de dados requer técnicas que sejam rápidas, eficientes e eficazes. Alcançar monitoramento e granularidade em profundidade é crucial para aumentar a eficácia e a precisão da análise NIDS. A capacidade de atribuir mudanças comportamentais a elementos de rede específicos, como usuários individuais, versões de sistema operacional ou protocolos, é essencial. A multiplicidade de protocolos diferentes e a natureza diversa dos dados que atravessam redes modernas amplificam a dificuldade em estabelecer um NIDS confiável (ELSAYED; MOHAMED; MADKOUR, 2024a).

Adicionalmente, a IA pode ser empregada para detectar anomalias nos dados sem a necessidade de conhecimento prévio desses dados. Ao combinar as capacidades das técnicas de IA com unidades de processamento avançadas, é possível criar uma poderosa defesa contra ameaças à segurança da rede. Além disso, considerando a vasta quantidade de dados disponíveis atualmente, a expertise humana é limitada e dispendiosa. Portanto, ao utilizar a IA, o *software* pode automaticamente identificar padrões que podem escapar da percepção humana devido à escala dos dados. Sem o uso da IA, especialistas humanos seriam obrigados a criar regras manualmente, o que não seria prático em grande escala (MAHFOUZ et al., 2020).

Muitas pesquisas na área de detecção de comportamento anormal têm usado o aprendizado de máquina e técnicas de aprendizado profundo para classificar o tráfego de rede anormal do tráfego normal. No entanto, muitos IDSs ainda sofrem com uma alta taxa de alarmes falsos, gerando muitos alertas para situações de baixa ameaça. Assim, muitos pesquisadores se concentraram no desenvolvimento de IDSs com taxas de detecção mais altas e taxas reduzidas de alarmes falsos. Outro problema com os IDSs existentes é a falta de capacidade para detectar ataques desconhecidos (Marir; Wang; Feng, 2019).

Como os ambientes de rede mudam rapidamente, os ataques variantes e novos ataques surgem constantemente. Assim, é necessário desenvolver IDSs que possam detectar ataques desconhecidos. Para resolver os problemas acima, os pesquisadores começaram a se concentrar na construção de sistemas de rede de detecção de intrusão (NIDSs) usando métodos de aprendizado de máquina (LIU; LANG, 2019; LI et al., 2019).

Encontramos trabalhos relacionados nesta linha de pesquisa que contam com conjuntos de dados conhecidos por apresentar deficiências para treinar modelos NIDS. Exemplos de problemas com muitos conjuntos de dados existentes incluem, entre outros: são limitados devido à falta de diversidade e volumes de tráfego, informações anônimas de pacotes e carga útil, restrições à variedade de ataques e falta do conjunto de recursos e metadados (RING et al., 2019; RANJIT; SAMARJEET, 2018). Além disso, uma taxonomia e uma pesquisa sobre IDSs e conjunto de dados dos últimos anos mostram que mais de

70% usavam conjuntos de dados ultrapassados como por exemplo o KDD-Cup99 (HINDY et al., 2018).

Diante disso, diversas organizações estão preocupadas em desenvolver sistemas de detecção de intrusão que são capazes de identificar possíveis ataques em suas infraestruturas representando um enorme desafio por conta da constante evolução das técnicas utilizadas nos ataques. Consequentemente, não restam dúvidas da necessidade contínua do desenvolvimento de soluções que apontem caminhos, métodos e ferramentas que auxiliem os pesquisadores no enfrentamento desse problema. Neste contexto, sugere-se aqui como proposta uma Revisão Sistemática da Literatura sobre sistemas de detecção de intrusão, conjunto de dados e um *framework* que seja capaz de fornecer mecanismos eficientes para construção e avaliação de sistemas de detecção de ataques cibernéticos modernos.

1.2 Descrição do Problema

O foco principal da Segurança da Informação é proteger o ativo mais importante para o negócio, as informações, devendo assim ser colocado em prática os requisitos responsáveis pela segurança que conforme a (NBR 27002, 2013), são eles: confidencialidade, integridade, disponibilidade e autenticação; apesar de não ser os únicos pontos importantes da segurança, ao atender os critérios necessários aos pilares da segurança, grande parte da proteção estará garantida (LAUREANO, 2012).

Ataques costumam ocorrer na Internet com diversos objetivos, visando diferentes alvos e usando variadas técnicas. Qualquer serviço, computador ou rede que seja acessível via Internet pode ser alvo de um ataque, assim como qualquer computador com acesso à Internet pode participar de um ataque (CERTBR, 2024).

Um ataque consiste de uma atividade maliciosa que explora uma vulnerabilidade em uma rede ou em um dispositivo na rede. O que define as chances de um ataque na Internet ser ou não bem-sucedido é o conjunto de medidas preventivas tomadas pelos usuários, desenvolvedores de aplicações e administradores dos computadores, serviços e equipamentos envolvidos.

Esses incidentes de segurança são violações ou ameaças às políticas de segurança definido como tentativas de comprometer a confiabilidade, a integridade ou a disponibilidade dos recursos do sistema. Uma ameaça de segurança é considerada como a possibilidade ou tentativa potencial de um intento “não autorizado” de acessar e/ou manipular a informação ou ainda de tornar-se um sistema não confiável ou não utilizável. Os IDS inspecionam e analisam informação relativa aos eventos observados e emitem alarmes para ao administrador de rede ou são armazenados em arquivos para realização de uma análise (LOPEZ, 2014).

As técnicas de *Machine Learning* e *Deep Learning* estão se tornando comuns nos sistemas de detecção de intrusão, pois permitem uma resposta em tempo real e têm a capacidade de aprender e se adaptar. Um IDS supera esses desafios introduzindo o aprendizado para identificar padrões de ataque a partir de bases de dados (CHKIRBENE et al., 2020) (GAO et al., 2021).

1.3 Objetivos Geral e Específicos

O objetivo geral desta dissertação consiste na condução de uma revisão teórica com a análise dos métodos empregados para representar a eficiência dos IDS propostos por diversos autores. Os resultados deste levantamento teórico são de grande valia para pesquisas posteriores no campo da detecção de intrusão. Neste contexto, são exploradas e relacionadas as preferências na utilização de conjuntos de dados e diferentes abordagens de detecção.

As principais contribuições deste trabalho são as seguintes:

- Introduzir o contexto atual e as deficiências de IDS e conjunto de dados;
- Mostrar quais são as principais formas de detecção utilizadas na literatura atual para IDS;
- Apresentar quais algoritmos de IA e conjuntos de dados são mais utilizados na literatura;
- Proporcionar uma visão geral sistemática dos sistemas de detecção de intrusões e métodos para estudos posteriores;
- Propor um *framework* completo e escalável para realização de testes de detecção de intrusão;
- Propor um modelo híbrido baseado em *Transformer* para detecção de intrusão;
- Realizar uma avaliação do modelo proposto em comparação com outras abordagens encontradas na literatura;
- Disponibilizar ao final do projeto o *framework* através do repositório GitHub, que pode ser acessado em: GitHub - AI/IDS.

Até onde se sabe, este trabalho de mestrado é pioneiro ao propor uma revisão abrangente dos últimos 4 anos e apresentar um *framework* para a simulação de ataques cibernéticos, tornando o processo de desenvolvimento e teste de sistemas de detecção de intrusão mais eficiente e acessível.

1.4 Contribuições

Os resultados obtidos ao longo da elaboração desta dissertação geraram uma publicação, incluída nos Anais do VIII Congresso sobre Tecnologias na Educação —

Ctrl+e 2023 (VALENTIM; BRAGA; OLIVEIRA, 2023). A segunda contribuição está relacionada à execução de um projeto PIBIC, coordenado pelo autor no Instituto Federal do Maranhão, com foco em segurança da informação, *firewalls* e sistemas de detecção de intrusão. A terceira contribuição foi a submissão de um artigo intitulado: *Hybrid Intrusion Detection System: Transformer with Convolutional Neural Networks* no NOMS 2022 IEEE/IFIP Network Operations and Management Symposium (NOMS 2022). O artigo não foi aceito pelo limite de publicações onde recebeu duas avaliações para aceitação e duas para rejeição. Por fim, a quarta contribuição refere-se à submissão do artigo intitulado: Uma Revisão Sistemática da Literatura sobre Sistemas de Detecção de Intrusão que se encontra em processo de finalização da escrita.

1.5 Estrutura da Dissertação

O restante do trabalho está estruturado da seguinte forma: no Capítulo 2, é apresentada a fundamentação teórica; no Capítulo 3, são discutidos os trabalhos relacionados mais recentes na área de sistemas de detecção de intrusão; o Capítulo 4 traz a proposta do *framework* de testes desenvolvida nesta dissertação de mestrado; no Capítulo 5, o modelo *AI/IDS Transformer* é apresentado; no Capítulo 6 são expostos os resultados obtidos e a discussão correspondente; por fim, o Capítulo 7 apresenta as conclusões do estudo e sugestões para trabalhos futuros.

2 Fundamentação Teórica

Este capítulo visa oferecer a fundamentação teórica necessária para compreensão de um IDS que vai além de conhecer sua aplicação prática e seus benefícios; exige uma compreensão aprofundada dos princípios que fundamentam essa solução de segurança e das técnicas que são utilizadas. É crucial entender por que esse tipo de sistema é necessário, o que o define como um detector de intrusão, como é implementado e como sua eficiência pode ser avaliada.

2.1 Sistemas de Detecção de Intrusão (IDS)

A detecção de Intrusão pode ser conceituada como a tecnologia de segurança que auxilia na identificação e isolamento de intrusões contra sistemas de computadores (PTACEK, 1998). (Buschkes; Kesdogan; Reichl, 1998) usa a seguinte definição: "Detecção de Intrusão é o ato de identificar indivíduos que estão utilizando um sistema computacional sem autorização e também indivíduos legítimos, ou seja, autenticados, que abusam de seus privilégios".

De acordo com (LAZAREVIC; KUMAR; SRIVASTAVA, 2005), um sistema de detecção de intrusos pode ser definido como uma combinação de componentes de *software* e/ou *hardware* que monitoram o sistema e emitem um alarme na ocorrência de um ataque, após a classificação de um evento.

Quando um IDS é desenvolvido adequadamente, segundo (McHugh; Christie; Allen, 2000), este consegue prover alertas indicando que um sistema está sendo atacado, mesmo este não sendo vulnerável a um ataque específico. Esses alertas podem ajudar os usuários na resistência a ataques e um IDS pode confirmar uma configuração segura e a operação de outros mecanismos de segurança.

Um IDS tem por finalidade analisar o comportamento em uma rede de computadores. Ele trabalha com um sistema que alerta o administrador ou o próprio sistema quando da ocorrência de tentativas de intrusões na rede, com uso de conhecimentos listados em uma base de dados (como por exemplo assinaturas de ataques e comportamento diferente do esperado) (NAKAMURA, 2007).

Para (SILVA, 2003) um sistema de detecção de intrusão pode ser um *software* ou *hardware*, que automatize o processo de detecção através de técnicas de monitoramento, visando identificar possíveis alterações nos padrões ou atividades previamente definidas como maliciosas.

Estes sistemas podem ser classificados quanto ao tipo da informação analisada e

também em relação ao modo como os sistemas analisam as informações coletadas. Quanto às informações analisadas, os sistemas são classificados em:

- NIDS (*Network Intrusion Detection System*): monitoram os pacotes que trafegam na rede;
- HIDS (*Host Intrusion Detection System*): monitoram o comportamento dos sistemas, analisando dados tais como arquivos de log, chamadas de sistema, dados de equipamentos como *firewalls* e roteadores;
- Híbridos: sistemas que monitoram tanto os pacotes da rede, quanto às informações dos sistemas.

Técnicas diferentes usadas na detecção acarretam em diferentes tipos de benefícios, aplicáveis em uma grande variedade de ambientes sendo, portanto, necessário selecionar aquele tipo que mais possa se adequar às necessidades do ambiente a ser protegido. A forma mais correta de definir o tipo a ser utilizado consiste em um conhecimento pleno de quais são as requisições específicas do sistema para, então, executar a implantação de um IDS (PROCTOR, 2000).

2.1.1 Sistema de Rede de Detecção de Intrusão

O NIDS deriva seu nome pelo fato de monitorar um segmento inteiro da rede, ou da sub-rede. Isto é feito alterando o modo de funcionamento da placa de interface de rede. Normalmente, uma placa de rede opera em modo não promíscuo, ouvindo apenas pacotes destinados a seu próprio endereço MAC (Controle de Acesso ao Meio). Outros pacotes que não são encaminhados a pilha são ignorados.

Para monitorar todo o tráfego no segmento da rede e não apenas os pacotes endereçados a própria máquina, o NIDS deve aceitar todos os pacotes e enviá-los até o banco de dados de tráfego coletado. Esta ação é conhecida como modo promíscuo. Neste modo, a ferramenta de detecção pode escutar todas as comunicações do segmento da rede. No entanto, isso não é tudo o que é necessário para garantir que o NIDS seja capaz de escutar todo o tráfego na sub-rede (Caswell; Beale, 2004).

A forma de implementar um IDS centralizado é usando um NIDS. As vantagens do NIDS encontram-se em analisar todo o fluxo de informações de uma rede de computadores, monitorando pacotes, serviços, portas, e requisições, bem como prover maior segurança, já que normalmente roda em uma máquina (física ou virtual) específica e configurada para aquela aplicação. A principal desvantagem de um sistema NIDS é que ele é incapaz de analisar o tráfego criptografado, como é o caso de VPNs, e pode ter limitações para analisar redes com grande fluxo de dados (GTA-UFRJ, 2016).

2.1.2 Sistema Hospedeiro de Detecção de Intrusão

O IDS baseado em *Host*, ou HIDS, protege somente o sistema no qual ele reside, ou seja, apenas o *host* onde se encontra instalado e não a sub-rede inteira (Caswell; Beale, 2004). Esta ferramenta utiliza a placa de rede da máquina onde está inserida, funcionando normalmente e operando em modo não promíscuo.

Esta é uma vantagem do HIDS, visto que, em alguns casos as interfaces de rede não são capazes de operar em modo promíscuo, embora a maioria das interfaces de rede modernas execute esta funcionalidade. Outra vantagem do HIDS é a capacidade de adequar seu conjunto de regras para o sistema *host* em particular. A capacidade de ajustar suas regras irá melhorar seu desempenho e diminuir os falsos positivos.

A grande vantagem de um HIDS, no entanto, reside na sua capacidade de detectar alterações específicas em arquivos no sistema operacional do seu hospedeiro. Ele pode controlar o tamanho dos arquivos e verificar os *checksums* para garantir que os arquivos essenciais do sistema não sejam modificados de forma maliciosa, sem alguém perceber. Além disso, pode ver o tráfego dentro de um sistema que nunca passa a rede e, portanto, nunca seria visto pelo NIDS. Existem algumas desvantagens para a eleição de usar um HIDS. Você terá que escolher aquele que é adaptado ao seu sistema operacional. Um HIDS irá adicionar carga para o *host* em que está configurado, pois o processo HIDS irá consumir recursos.

2.1.3 Híbridos

Os IDS que utilizam a aplicação das técnicas de assinatura e anomalia em conjunto, são denominados de IDS Híbrido ou IDS-H. Este IDS visa aumentar o poder de detecção de intrusos, pois detectam tanto ataque conhecidos, quanto desconhecidos.

2.1.4 Tipos de Detecção

- Anomalia: Um sistema baseado em anomalia é um sistema que monitora as ações que ocorrem numa rede e, através de técnicas de inteligência computacional (como, por exemplo, redes neurais e classificadores estatísticos), aprende o comportamento padrão do sistema. Com treinos específicos e a definição dos padrões obtida pode-se monitorar a rede e, dependendo do tráfego dos pacotes e uso, é possível identificar as atividades da rede e classificá-las como normais ou anômalas. O maior problema com esta metodologia que se deve primeiro definir o que é normal. Se durante a fase de aprendizagem a rede for vítima de um ataque não identificado, os sistemas IDS baseados em anomalias interpretarão esse tráfego malicioso como normal e nenhum alarme será acionado na próxima vez que ocorrer um ataque similar (GTA-UFRJ, 2016).

- Assinatura: Um IDS baseado em assinaturas utiliza um banco de dados com os ataques já conhecidos e compara-os com as ações, utilizando algoritmos estatísticos para reconhecer algum desses ataques. Nesse caso, é de extrema importância que o banco de dados com as assinaturas esteja sempre atualizado para garantir a segurança do ambiente. Uma assinatura é um padrão do que buscar no fluxo de dados (GTA-UFRJ, 2016).

2.2 Aprendizagem de Máquina

O aprendizado de máquina, ou *Machine Learning* (ML), é uma área da Inteligência Artificial (IA) dedicada ao estudo de algoritmos que buscam identificar padrões em conjuntos de dados, produzindo um modelo adaptado que pode ser utilizado em um novo conjunto de informações. Por essas características, muitas são as aplicações das técnicas de inteligência artificial por meio do aprendizado de máquina. Essas técnicas são aplicadas há algum tempo em várias áreas como mineração de dados, classificação, projeto, controle, modelagem, otimização, diagnóstico e previsão de máquinas (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

Segundo (BISHOP, 2006) o ML é um sistema capaz de adquirir e acumular conhecimento e, assim, melhorar a performance em tarefas específicas, é um dos tipos de algoritmos usados na IA, que desenvolve programas que aprendem a fazer previsões com base em dados, sem necessitar do auxílio de um programador. O ML é utilizado em aplicações como processamento de imagens, filtro de *spam*, recomendações de música, e detecção de fraudes entre outras.

Basicamente o aprendizado de máquina busca padrões para classificar conjuntos de dados segundo esses padrões aprendidos. São diversas as técnicas que possibilitam, a partir do aprendizado, classificar um determinado conjunto de dados. A análise preditiva tem como atividade principal estimar o risco de eventos futuros com base em experiências passadas, para orientar a tomada de decisão atual. A operacionalização desse processo envolve um conjunto de ferramentas, ou algoritmos, utilizados para compreender os dados existentes e gerar regras de predição. De modo geral, essas ferramentas podem ser alocadas em uma das seguintes categorias de aprendizagem: a supervisionada e a não supervisionada (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

No aprendizado supervisionado, os padrões de treinamento, bem como seus padrões de saída desejados associados, são alimentados na rede durante a fase de treinamento. (Hsieh; Chen, 1993). O supervisionamento é feito quando a partir de um conjunto de dados rotulados previamente definido deseja-se encontrar uma função que seja capaz de prever rótulos desconhecidos. Nesse tipo de aprendizagem pode-se estimar números reais ou valores dentro de um conjunto finito (SENAI, 2018)

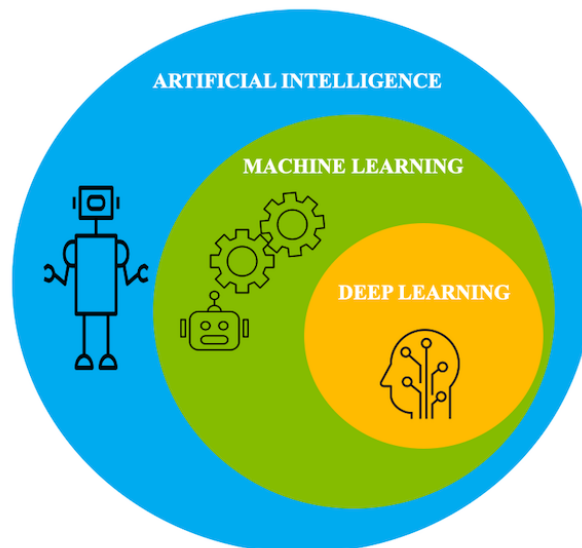
Na aprendizagem não supervisionada, apenas os padrões de treinamento são fornecidos, e a rede geralmente os categoriza de acordo com uma medida especificada de similaridade entre os padrões. (Hsieh; Chen, 1993). Nesse tipo de aprendizagem o conjunto de dados utilizado não possui nenhum tipo de rótulo. O objetivo desse tipo de aprendizagem é descobrir similaridades entre os objetos analisados a fim de detectar similaridades e anomalias (SENAI, 2018).

Em um esforço para criar sistemas que aprendem de forma semelhante à maneira como os humanos aprendem, a arquitetura subjacente para o aprendizado profundo foi inspirada pela estrutura do cérebro humano. Por esse motivo, algumas terminologias fundamentais no aprendizado profundo podem ser mapeadas de volta para a neurologia. Semelhante à como os neurônios formam os blocos de construção fundamentais do cérebro, a arquitetura de aprendizado profundo contém uma unidade computacional que permite a modelagem de funções não lineares chamadas *perceptron*.

A magia do aprendizado profundo começa com o humilde *perceptron*. Semelhante a como um “neurônio” em um cérebro humano transmite pulsos elétricos por todo o nosso sistema nervoso, o *perceptron* recebe uma lista de sinais de entrada e os transforma em sinais de saída (IBM, 2020).

A Figura 1 mostra os subdomínios da IA, divididos em IA, ML e DL.

Figura 1 – Subdomínios da Inteligência Artificial.



Fonte – (IBM, 2020)

Segundo (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) DL é uma abordagem relacionada com aprendizado de máquina e redes neurais. O DL pode ser entendido como uma classe de técnicas de aprendizado de máquina que explora muitas camadas de informação não lineares. Basicamente, o DL trata do aprendizado de múltiplos níveis de

representação e abstração que ajudam a construir sentido em dados, tais como imagens, sons e texto. Em suas diversas definições, duas características estão sempre presentes e permitem a representação sucessivamente mais alta de camadas mais abstratas, sendo um modelo que consiste de múltiplas camadas de processamento de informação não linear e também um método de aprendizado supervisionado e não supervisionado.

Para (ARROYO-FIGUEROA; SUCAR; VILLAVICENCIO, 2000) DL é construído sobre as redes neurais, sendo um tipo de modelo de ML que se parece com os neurônios cerebrais humanos. Utiliza modelo bioinspirado, como uma rede neural artificial, cujos neurônios artificiais estão organizados em camadas interconectadas. É uma das técnicas de aprendizado de máquina que utiliza as redes neurais artificiais para processar as informações e aprendizagem, sendo capaz de trabalhar com análise de dados brutos, o DL propicia a classificação de informações em diferentes formatos como áudio no reconhecimento de fala, imagens no reconhecimento facial, entre outros.

Há uma camada de entrada que recebe dados externos e uma de saída que diz como o sistema vai responder a uma determinada informação. Entre essas camadas, há outras adicionais “escondidas” de neurônios que processam dados dando um peso numérico à informação que recebem de cada camada anterior, transmitindo essa informação para a próxima camada da rede. Uma rede neural possibilita a solução de problemas complexos pois há uma enorme quantidade de neurônios trabalhando juntos.

2.3 Transformer

A arquitetura do *Transformer* se tornou a pedra angular de vários modelos de sequência de *Natural Language Processing* (PNL), impulsionando resultados de última geração para várias tarefas, como classificação de texto e marcação de sequências, e superando modelos recorrentes e convolucionais. A arquitetura foi projetada para processar sequências discretas de forma mais eficiente, em parte devido ao mecanismo de autoatenção com múltiplas cabeças integradas. A arquitetura permitiu acelerações significativas para pré-treinamento em grandes volumes de dados e tornou possível usar conjuntos de dados maiores para realizar o treinamento do modelo (VASWANI et al., 2023).

Um *transformer* é basicamente composto por um codificador e um decodificador.

- *Encoder*: O codificador é composto por uma pilha de 6 camadas idênticas. Cada camada tem duas subcamadas. O primeiro é um mecanismo de autoatenção com múltiplas cabeças (*multi-head self-attention*), e a segunda é uma rede *feed-forward* simples, totalmente conectada em termos de posição.
- *Decoder*: O decodificador também é composto por uma pilha de 6 camadas idênticas. Além das duas subcamadas em cada camada do codificador, o decodificador insere uma terceira subcamada, que realiza atenção multi-cabeças sobre a

saída da pilha do codificador. Semelhante ao codificador, empregamos conexões residuais em torno de cada uma das subcamadas, seguidas pela normalização da camada. Também modificamos a subcamada de autoatenção na pilha do decodificador para impedir que posições atendam a posições subsequentes.

O *Transformer* segue essa arquitetura geral usando autoatenção empilhada e camadas totalmente conectadas por ponto para o codificador e o decodificador, mostradas nas metades esquerda e direita da Figura 2, respectivamente.

2.4 Processo de Descoberta de Conhecimento

Em diversas aplicações que exigem a manipulação de grandes volumes de dados, o processo de descoberta de conhecimento em bases de dados (*Knowledge Discovery in Database-KDD*) visa extrair novas informações desses dados. No estudo apresentado por (FAYYAD et al., 1996), KDD é definido como "um processo iterativo e iterativo, não trivial, composto por diversas etapas, com o objetivo de extrair informações implícitas, previamente desconhecidas e potencialmente úteis, a partir dos dados armazenados".

O termo iterativo refere-se à necessidade da atuação do homem como responsável pelo controle do processo, ou seja, analisar e interpretar os resultados obtidos ao longo do processo. Já o termo iterativo sugere a necessidade de repetições do processo de KDD, a fim de buscar os melhores resultados por meio de sucessivos refinamentos.

O processo de descoberta de conhecimento, ilustrado na Figura 3, é iterativo e iterativo, consistindo de cinco etapas. O processo é iterativo em cada etapa, o que significa que a movimentação para ajustar etapas anteriores podem ser necessárias. A seguir, uma breve descrição das etapas do processo de KDD é apresentada de acordo com os autores.

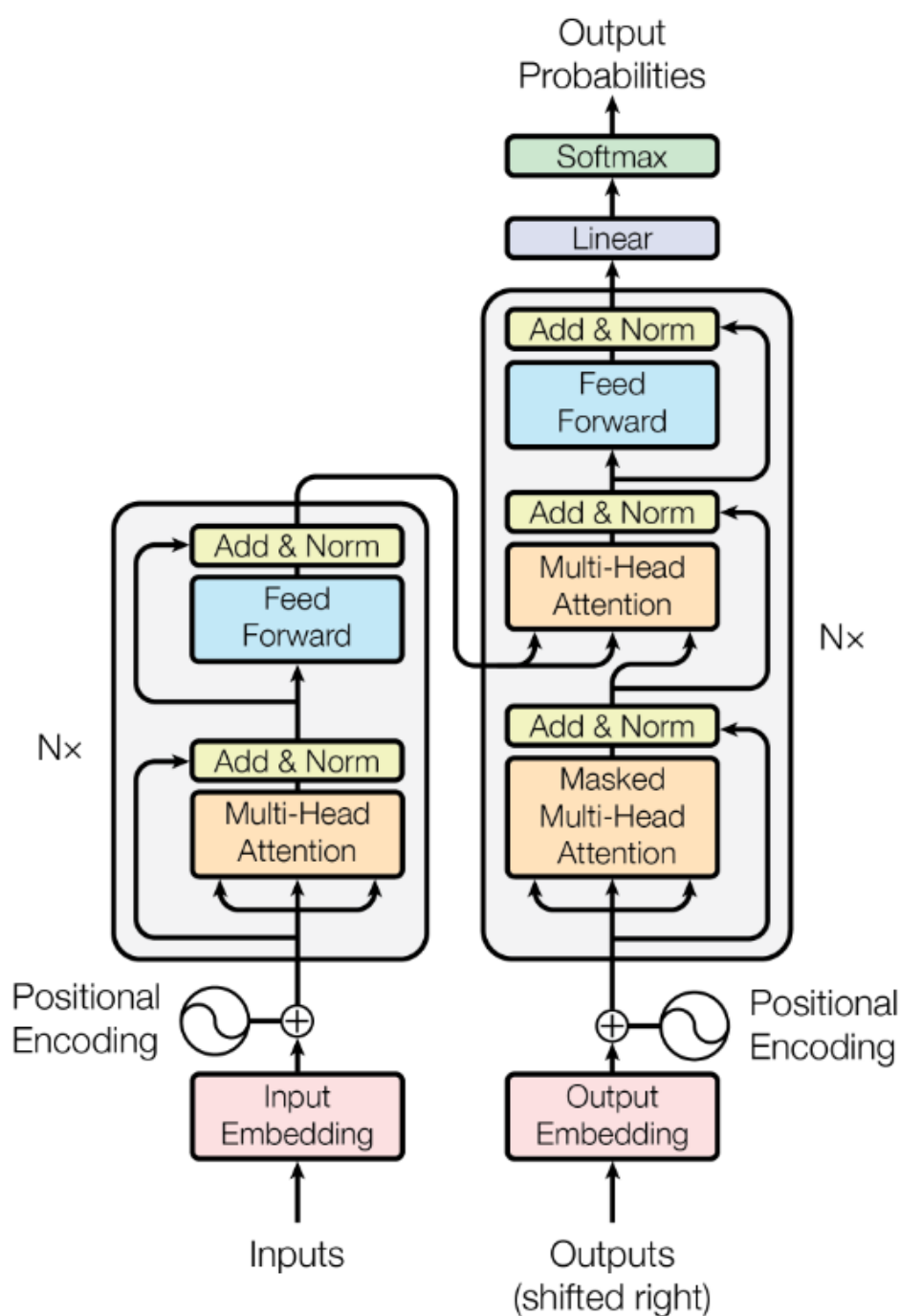
Seleção: esta etapa consiste em selecionar um conjunto ou subconjunto de dados que farão parte da análise. As fontes de dados podem ser variadas (planilhas, sistemas gerenciais, *data warehouses*) e possuir dados com formatos diferentes (estruturados, semiestruturados e não-estruturados).

Processamento: esta etapa consiste em fazer a verificação da qualidade dos dados armazenados. A base passa por um processo de limpar, corrigir ou remover dados inconsistentes, verificar dados ausentes ou incompletos, identificar anomalias.

Transformação: esta etapa consiste em aplicar técnicas de transformação como: normalização, agregação, criação de novos atributos, redução e sintetização dos dados. Aqui os dados ficam disponíveis agrupados em um mesmo local para a aplicação dos modelos de análise.

Mineração de Dados: esta etapa consiste em construir modelos ou aplicar técnicas de mineração de dados. Essas técnicas têm por objetivo (1) verificar uma hipótese, (2)

Figura 2 – Modelo de arquitetura de um Transformer.

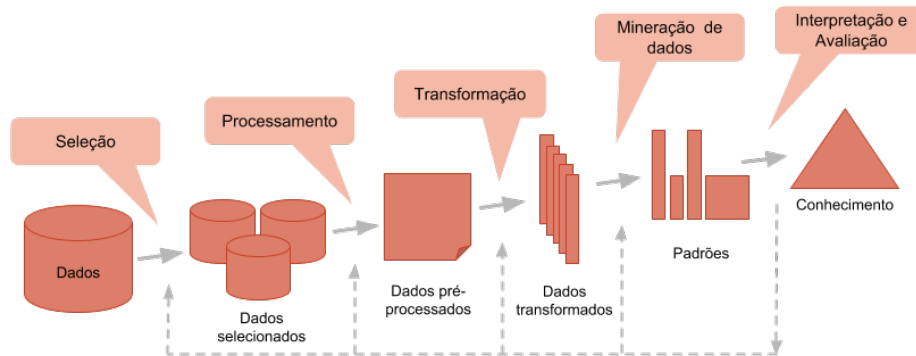


Fonte – Adaptado de (VASWANI et al., 2023)

descobrir novos padrões de forma autônoma. Além disso, a descoberta pode ser dividida em: preditiva e descritiva. Esses modelos geralmente são aplicados e refeitos inúmeras vezes dependendo do objetivo do projeto.

Interpretação e Avaliação: esta etapa consiste em avaliar o desempenho do modelo,

Figura 3 – Modelo de descoberta de conhecimento em bases de dados.



Fonte – Adaptado de (FAYYAD et al., 1996)

aplicando em cima de dados que não foram utilizados na fase de treinamento ou mineração. A validação pode ser feita de diversas formas, algumas delas são: utilizar medidas estatísticas, passar pela avaliação dos profissionais de negócio.

2.5 Conjunto de Dados CSE-CIC-IDS2017

O conjunto de dados CSE-CIC-IDS2017 contém ataques comuns e tráfego de dados benignos, que se assemelham aos verdadeiros dados do mundo real (PCAPs). Também inclui os resultados da análise de tráfego de rede usando o *CICFlowMeter* com fluxos rotulados com base nos IPs de carimbo de data/hora, fonte e destino, portas de origem e destino, protocolos e ataques (arquivos CSV) (SHARAFALDIN; LASHKARI; GHORBANI, 2018).

Tabela 1 – Ataques e recursos do conjunto de dados CSE-CIC-IDS2017.

Label	Feature
Benign	B.Packet Len Min, Subflow F.Bytes, Total Len F.Packets, F.Packet Len Mean
DoS GoldenEye	B.Packet Len Std, Flow IAT Min, Fwd IAT Min, Flow IAT Mean
Heartbleed	B.Packet Len Std, Subflow F.Bytes, Flow Duration, Total Len F.Packets
DoS Hulk	B.Packet Len Std, Flow Duration, Flow IAT Std

DoS Slowhttp	Flow Duration, Active Min, Active Mean, Flow IAT Std
DoS slowloris	Flow Duration, F.IAT Min, B.IAT Mean, F.IAT Mean
SSH-Patator	Init Win F.Bytes, Subflow F.Bytes, Total Len F.Packets, ACK Flag Count
FTP-Patator	Init Win F.Bytes, F.PSH Flags, SYN Flag Count, F.Packets/s
Web Attack	Init Win F.Bytes, Subflow F.Bytes, Init Win B.Bytes, Total Len F.Packets
Infiltration	Subflow F.Bytes, Total Len F.Packets, Flow Duration, Active Mean
Bot	Subflow F.Bytes, Total Len F.Packets, F.Packet Len Mean, B.Packets/s
PortScan	Init Win F.Bytes, B.Packets/s, PSH Flag Count
DDoS	B.Packet Len Std, Avg Packet Size, Flow Duration, Flow IAT Std

2.6 Conjunto de Dados CSE-CIC-IDS2018

A detecção de anomalias tem sido o foco principal de muitos pesquisadores devido ao seu potencial na detecção de novos ataques. No entanto, sua adoção para aplicativos do mundo real foi prejudicada devido à complexidade do sistema, pois esses sistemas exigem uma quantidade substancial de testes, avaliações e ajustes antes da implantação. A execução desses sistemas em tráficos de rede rotulados reais, com um conjunto abrangente e extenso de intrusões e comportamentos anormais, é a metodologia mais ideal para testes e avaliações.

Para superar essas deficiências, uma abordagem sistemática foi concebida para gerar conjuntos de dados para analisar, testar e avaliar sistemas de detecção de intrusão, com foco em detectores de anomalias baseados em rede. O principal objetivo do projeto *CSE-CIC-IDS2018* é desenvolver uma abordagem sistemática para gerar conjuntos de dados de referência diversificados e abrangentes para detecção de intrusão com base na criação de perfis de usuário que contêm representações abstratas de eventos e comportamentos vistos na rede. Os perfis serão combinados para gerar um conjunto diversificado de conjuntos de dados, cada um com um conjunto exclusivo de recursos, que cobre uma parte do domínio de avaliação (CYBERSECURITY, 2018).

O conjunto de dados inclui sete cenários de ataque diferentes: *Brute-force*, *Heart-bleed*, *Botnet*, *DoS*, *DDoS*, *Web attacks* e infiltração na rede de dentro.

2.7 Conjunto de Dados KDD Cup 99

O Programa de Avaliação de Detecção de Intrusão DARPA de 1998 foi preparado e gerenciado pelo *MIT Lincoln Labs*. O objetivo era pesquisar e avaliar pesquisas em detecção de intrusão. Um conjunto padrão de dados a serem auditados, que inclui uma ampla variedade de intrusões simuladas em um ambiente de rede militar, foi fornecido. O concurso de detecção de intrusão KDD de 1999 usa uma versão deste conjunto de dados. O *Lincoln Labs* configurou um ambiente para adquirir nove semanas de dados brutos para uma rede local (LAN) simulando uma LAN típica da Força Aérea dos EUA. Eles operavam a LAN como se fosse um verdadeiro ambiente da Força Aérea, mas a apimentavam com vários ataques (CUP, 1999).

Os dados brutos de treinamento consistiam em aproximadamente quatro gigabytes de dados binários compactados, abrangendo sete semanas de tráfego de rede. Isso foi processado em cerca de cinco milhões de registros de conexão. Da mesma forma, as duas semanas de dados de teste renderam cerca de dois milhões de registros de conexão. Uma conexão é uma sequência de pacotes TCP começando e terminando em alguns momentos bem definidos, entre os quais os dados fluem de e para um endereço IP de origem para um endereço IP de destino sob algum protocolo bem definido. Cada conexão é rotulada como normal ou como um ataque, com exatamente um tipo de ataque específico. Cada registro de conexão consiste em cerca de 100 *bytes*.

O conjunto de dados KDD tem dois problemas críticos concluídos pela análise estatística, que afetam profundamente o desempenho do sistema. O problema mais significativo no conjunto de dados KDD é que ele tem um grande número de registros replicados. Foi descoberto que cerca de 78% e 75% dos registros são duplicados nos conjuntos de dados de treinamento e teste, respectivamente. Um grande número de registros replicados pode levar os algoritmos de aprendizagem a serem parciais em vez de vários registros. Assim, o algoritmo interromperá o aprendizado de registros pouco frequentes (CHOUDHARY; KESSWANI, 2020).

Os ataques se enquadram em quatro categorias principais:

- **DoS (Denial of Service)**: negação de serviço, por exemplo, *SYN flood*.
- **R2L (Remote to Local)**: acesso não autorizado de uma máquina remota, por exemplo, adivinhação de senha.
- **U2R (User to Root)**: acesso não autorizado a privilégios de superusuário local (*root*), por exemplo, vários ataques de estouro de buffer.
- **Probing**: vigilância e outras sondagens, por exemplo, varredura de portas.

Tabela 2 – Estatísticas de registros redundantes no KDD Train.

Category	Original Records	Distinct Records	Reduction Rate
Attacks	3.925.650	262.178	93.32%
Normal	972.781	812.814	16.44%
Total	4.898.431	1.074.992	78.05%

Tabela 3 – Estatísticas de registros redundantes no KDD Test.

Category	Original Records	Distinct Records	Reduction Rate
Attacks	250.436	29.378	88.26%
Normal	60.591	47.911	20.92%
Total	311.027	77.289	75.15%

2.8 Conjunto de Dados NSL-KDD

Para resolver os problemas do conjunto de dados *KDD Cup 99*, eles propuseram um novo conjunto de dados, ou seja, NSL-KDD, que consiste em registros selecionados do conjunto de dados completo *KDD Cup 99*. A seguir estão as vantagens do conjunto de dados NSL-KDD sobre o conjunto de dados KDD Cup 99 (CHOUDHARY; KESSWANI, 2020):

- Ele não inclui registros irrelevantes no conjunto de treinamento, então os classificadores não serão parciais em relação a registros mais repetidos.
- De cada registro de dificuldade, o número de registros escolhidos é inversamente proporcional à porcentagem dos registros no conjunto de dados KDD. Então, isso resulta que a porcentagem de classificação das diferentes técnicas de ML (Machine Learning) diferem em uma ampla faixa. Isso torna a avaliação abrangente estruturada das abordagens de ML.
- No conjunto de dados de treinamento e teste, o número de registros é lógico, o que torna mais confortável fazer os experimentos em todo o conjunto de dados sem a necessidade de escolher pequenos segmentos aleatórios. Portanto, os resultados da avaliação de diferentes trabalhos serão estáveis.

O conjunto de treinamento do NSL-KDD é denominado *KDDTrain+*, enquanto o conjunto de teste é denominado *KDDTest+*. Cada registro no conjunto de dados NSL-KDD possui 42 atributos, dos quais 41 representam as características dos dados e 1 representa o tipo de ataque (TAVALLAEE et al., 2009). As tabelas 2 e 3 ilustram a quantidade de registros redundantes corrigidos na atualização do conjunto de dados *KDD Cup 99* para o NSL-KDD.

2.9 Conjunto de Dados UNSWNB15

Os pacotes de rede brutos do conjunto de dados *UNSW-NB 15* foram criados pela ferramenta *IXIA PerfectStorm* no *Cyber Range Lab* da *UNSW Canberra* para gerar um híbrido de atividades normais modernas reais e comportamentos de ataque contemporâneos sintéticos. A ferramenta *tcpdump* foi utilizada para capturar 100 GB do tráfego bruto (arquivos *.Pcap*). Este conjunto de dados tem nove tipos de ataques, a saber, *Fuzzers*, *Analysis*, *Backdoors*, *DoS*, *Exploits*, *Generic*, *Reconnaissance*, *Shellcode* e *Worms*. As ferramentas *Argus*, *Bro-IDS* são usadas e doze algoritmos são desenvolvidos para gerar um total de 49 recursos com o rótulo de classe. Esses recursos são descritos no arquivo *UNSW NB15-features.csv* (WALES, 2015).

O número total de registros é de dois milhões e 540.044, que são armazenados nos quatro arquivos CSV. Uma partição deste conjunto de dados foi configurada como um conjunto de treinamento e um conjunto de teste, a saber, *UNSWNB15-training-set.csv* e *UNSWNB15-testing-set.csv*, respectivamente. O número de registros no conjunto de treinamento é de 175.341 registros e o conjunto de teste é de 82.332 registros dos diferentes tipos, ataque e normal (MOUSTAFA; SLAY, 2015).

3 Trabalhos Relacionados

A concepção dos IDS teve seu início em 1980 com a proposta de *Jim Anderson* (ANDERSON, 1980). Desde então, houve um aumento na quantidade de IDS propostos para atender às necessidades de segurança de redes. No entanto, à medida que o número de dispositivos conectados e o volume de tráfego de dados aumentaram substancialmente, surgiram diversos tipos de ataques cibernéticos avançados e variados. Como resultado, muitos pesquisadores têm se concentrado em aprimorar os IDS, visando elevar a eficácia na detecção de ataques, tanto os já conhecidos quanto os novos, e ao mesmo tempo reduzir a incidência de falsos alarmes (SURAKHI et al., 2021).

Com o objetivo de compreender as técnicas e os conjuntos de dados utilizados na construção e nos testes de IDSs, foram consultadas pesquisas atuais que revisam a literatura sobre o tema. Durante essa análise, (CHACKO et al., 2023) realizou uma investigação detalhada sobre as técnicas de aprendizado de máquinas aplicadas na criação de IDSs. O estudo abrangeu diversos aspectos, incluindo a avaliação dos conjuntos de dados utilizados e a identificação dos desafios enfrentados pelos pesquisadores ao implementar essas abordagens.

Diferentemente, o trabalho proposto por (NASSIF et al., 2021) oferece uma extensa revisão que abrange um período de duas décadas em relação a IDS que empregam técnicas de aprendizado de máquina. Nesse processo, foram examinados 290 artigos e 22 conjuntos de dados distintos. Como resultado desse estudo, são apresentadas uma série de orientações para futuras pesquisas, que incluem a sugestão de adoção de conjuntos de dados mais atualizados e a necessidade de proporcionar explicações mais claras sobre os métodos de avaliação das técnicas aplicadas em IDS.

Em seu trabalho, (GUPTA; TRIPATHI; GROVER, 2021) enumera diversos aspectos relativos às técnicas e aos conjuntos de dados empregados em IDS. Alguns desses pontos incluem a questão da precisão insatisfatória dos modelos, o que é considerado inaceitável em um IDS contemporâneo. Além disso, aborda a consideração do tempo de processamento dos conjuntos de treinamento e a importância da documentação adequada do modelo para permitir a replicação das técnicas utilizadas.

Uma revisão abrangente das técnicas de aprendizado profundo aplicadas a IDS é apresentada por (LIFANDALI; ABGHOOR, 2021), explorando modelos como *Convolutional Neural Networks (CNN)*, *Recurrent Neural Networks (RNN)*, *Autoencoder*, *Máquina de Boltzmann*, onde foram detalhadas suas aplicações na literatura. Os resultados revelaram que a CNN se destacou como a técnica mais promissora, sugerindo uma direção para pesquisas futuras concentradas nesse método.

Na pesquisa conduzida por (SUTHISHNI; KUMAR, 2022), aborda uma revisão da literatura sobre IDS que fazem uso de técnicas de aprendizado de máquina e aprendizado profundo, acentua-se o desafio enfrentado pelos pesquisadores em encontrar conjuntos de dados recentes que sejam equilibrados e abordem uma ampla variedade de tipos de ataques. A pesquisa também destaca a importância do uso de unidades de processamento gráfico (GPUs) para testar técnicas de aprendizado profundo, uma vez que esses modelos requerem grande capacidade computacional para treinar os algoritmos de maneira eficaz. Por fim, o autor oferece orientações para direcionar pesquisas futuras no campo da detecção de intrusões.

3.1 Pesquisa de trabalhos similares

Com a finalidade de identificar, avaliar e interpretar as pesquisas existentes, a abordagem metodológica adotada para conduzir esta revisão sistemática de literatura baseou-se no guia proposto por (KITCHENHAM; CHARTERS, 2007). Esse guia, composto por três etapas distintas: planejamento, condução e obtenção de resultados. O intuito primordial foi garantir uma compreensão profunda das investigações existentes no campo, seguindo uma estrutura metodológica consolidada e reconhecida.

3.1.1 Planejamento

Criar questões de pesquisa em uma revisão de literatura ajuda a direcionar o foco, identificar lacunas no conhecimento e organizar a análise de trabalhos existentes, garantindo relevância e clareza nos objetivos do estudo.

Seguem as questões de pesquisa dessa revisão:

- Q1:** Como estão distribuídas as publicações sobre sistemas de detecção intrusão ao longo dos anos?
- Q2:** Quais tipos de contribuições têm sido desenvolvidos no contexto de sistemas detecção de intrusões?
- Q3:** Quais são as principais formas de detecção utilizadas na literatura atual para IDS?
- Q4:** Quais são os resultados dos sistemas de detecção dos artigos mais qualificados?
- Q5:** Quais algoritmos de IA estão sendo utilizadas pelos pesquisadores para construção de IDSs?
- Q6:** Quais conjuntos de dados são mais utilizadas nos estudos?
- Q7:** Quais sugestões e diretrizes podem ser propostos para pesquisas futuras na área de IDS?

3.1.2 Protocolo de busca

Para encontrar estudos que respondam às perguntas de pesquisa apresentadas, é essencial estruturá-las antes de desenvolver um protocolo de busca. Isso pode ser realizado utilizando uma tabela PICO (População, Intervenção, Comparação e Resultados - *Outcomes*) (HUANG; LIN; DEMNER-FUSHMAN, 2006).

O protocolo de busca foi estabelecido ao conectar os termos C1, C2 e C3 utilizando o operador lógico (AND) em todas as palavras-chave listadas na Tabela 4. Posteriormente, esse protocolo foi aplicado em três repositórios digitais que contêm as publicações mais relevantes para o contexto deste estudo, a saber: *ACM Digital Library*, sob a gestão da *Association for Computing Machinery* (ACM); *IEEE Xplore Digital Library*, associado ao *Institute of Electrical and Electronics Engineers* (IEEE); e, finalmente, *ScienceDirect*, uma publicação da editora *Elsevier*.

Devido aos critérios de avaliação por revisão dupla e aos critérios utilizados na seleção de artigos, foram considerados esses três grandes repositórios. Diferentemente do repositório *ArXiv*, que, embora contenha pesquisas de excelente qualidade, não adota o processo de revisão dupla.

Tabela 4 – Tabela PICO com referências para protocolo de busca.

Elemento	Descrição	Keywords	Referência
População	Alvo principal de busca, tipo de rede ou sistema onde o sistema de detecção de intrusão será aplicado.	<i>intrusion detection system, intrusion detection, network detection</i>	C1
Intervenção	Tecnologia ou método específico usado para detectar intrusões, baseados em rede (NIDS), baseados em host (HIDS), análise comportamental, baseados em assinaturas, IDS híbridos.	<i>anomaly detection, hybrid, signature</i>	C2
Comparação	Comparar técnicas de aprendizagem utilizadas em IDSs	<i>machine learning, deep learning, neural network, learning.</i>	C3
Resultados	Avaliar a eficácia dos resultados alcançados.	-	-

A Tabela 5 apresenta o protocolo de busca utilizado em cada motor de busca, considerando que cada repositório tem sua própria estrutura de sintaxe.

A revisão proposta neste estudo estabeleceu diversos critérios para a escolha de estudos relacionados à detecção de intrusão. Um importante critério priorizou a data de

Tabela 5 – Protocolo de busca utilizados nos motores de busca.

Indexador	String de busca
<i>ACM Digital</i>	<i>(intrusion detection OR network detection) AND (anomaly detection OR hybrid OR signature) AND (machine learning OR deep learning OR neural network OR learning)</i>
<i>IEEEExplore</i>	<i>("Full Text \& Metadata": intrusion detection OR network detection) AND ("Full Text \& Metadata": anomaly detection OR hybrid OR signature) AND ("Full Text \& Metadata": machine learning OR deep learning OR neural network OR learning)</i>
<i>ScienceDirect</i>	<i>all:(intrusion detection OR network detection) AND (anomaly detection OR hybrid OR signature) AND (machine learning OR deep learning OR neural network OR learning)</i>

publicação, favorecendo artigos mais recentes, publicados entre janeiro de 2021 e dezembro de 2024.

3.1.3 Critérios de Inclusão/Exclusão

Antes de avançar para a fase de execução, é essencial estabelecer os critérios pelos quais os trabalhos serão selecionados, determinando quais serão incluídos ou excluídos da análise.

Critérios de Inclusão:

- Estudos que apresentam uma abordagem orientada à resultados de detecção;
- Possui uma acurácia maior que 80%.

Critérios de Exclusão:

- Trabalhos não escritos em inglês;
- Estudos duplicados;
- Este trabalho é focado em estudos recentes. Logo, esta RS excluirá todas as publicações feitas antes de 2021. Levando em consideração o período 2021 a 2024;
- Estudos fora do contexto;
- Apresenta uma solução sem uma análise comparativa;
- Possui um dataset para o experimento sem URL para *download* ou com menos de 4 tipos de ataques devidamente rotulados;
- Estudos que não atingiram 8 mais pontos quando submetidos aos critérios de qualidade definidos por esta RS.

Critérios de Qualidade: Para a análise de qualidade, os seguintes critérios foram

designados para comparar os estudos selecionados. Foram elaborados 10 critérios de qualidade. Para cada critério, haverá 3 possíveis respostas: Não (escore=0), Parcialmente (escore=0,5), Sim (escore=1). Aqueles que atingirem nota 8 ou mais serão considerados aprovados na avaliação de qualidade desta RS. Eis os critérios:

CQ1: As motivações e os objetivos de pesquisa estão claramente definidos?

CQ2: Os métodos escolhidos são detalhados e justificados?

CQ3: A solução proposta foi detalhada completamente?

CQ4: O pesquisador especificou o dataset selecionado? E qual foi a justificativa para tal seleção?

CQ5: O conjunto de dados foi analisado/interpretado para o experimento?

CQ6: O estudo relata claramente os resultados?

CQ7: As figuras e tabelas estão legíveis e bem desenhadas? O pesquisador discutiu cada uma delas?

CQ8: Os resultados são consistentes com os objetivos apresentados?

CQ9: As limitações e trabalhos futuros são apresentados neste estudo?

CQ10: O artigo foi escrito de maneira clara e coerente?

3.1.4 Condução

Considerou-se o intervalo recente de 2021 a 2024, portanto, trabalhos publicados após esse período não foram incluídos. De qualquer forma, esse número de publicações seria impraticável para realizar uma Revisão Sistemática. Executando o protocolo de pesquisa e considerando o texto completo para cada mecanismo de busca, conforme a Tabela 5, foram coletados 8.171 estudos primários. Vale salientar que esta pesquisa foi realizada em dezembro de 2024.

A primeira etapa envolve a triagem dos títulos e resumos utilizando as palavras-chave mais relevantes desta revisão sistemática (STEINMACHER; CHAVES; GEROSA, 2013). No entanto, mesmo nesta fase, é provável que muitos falsos positivos permaneçam (KITCHENHAM; CHARTERS, 2007). Esta fase de triagem visa remover trabalhos duplicados e realizar uma leitura rápida dos títulos e resumos, procurando as palavras-chave listadas na Tabela 4. Para automatizar esse processo trabalhoso, foi utilizada a ferramenta *Rayyan* (RAYYAN, 2024). Como resultado, 7.853 trabalhos foram excluídos pelos motivos listados na Figura 4, restando 318 estudos selecionados.

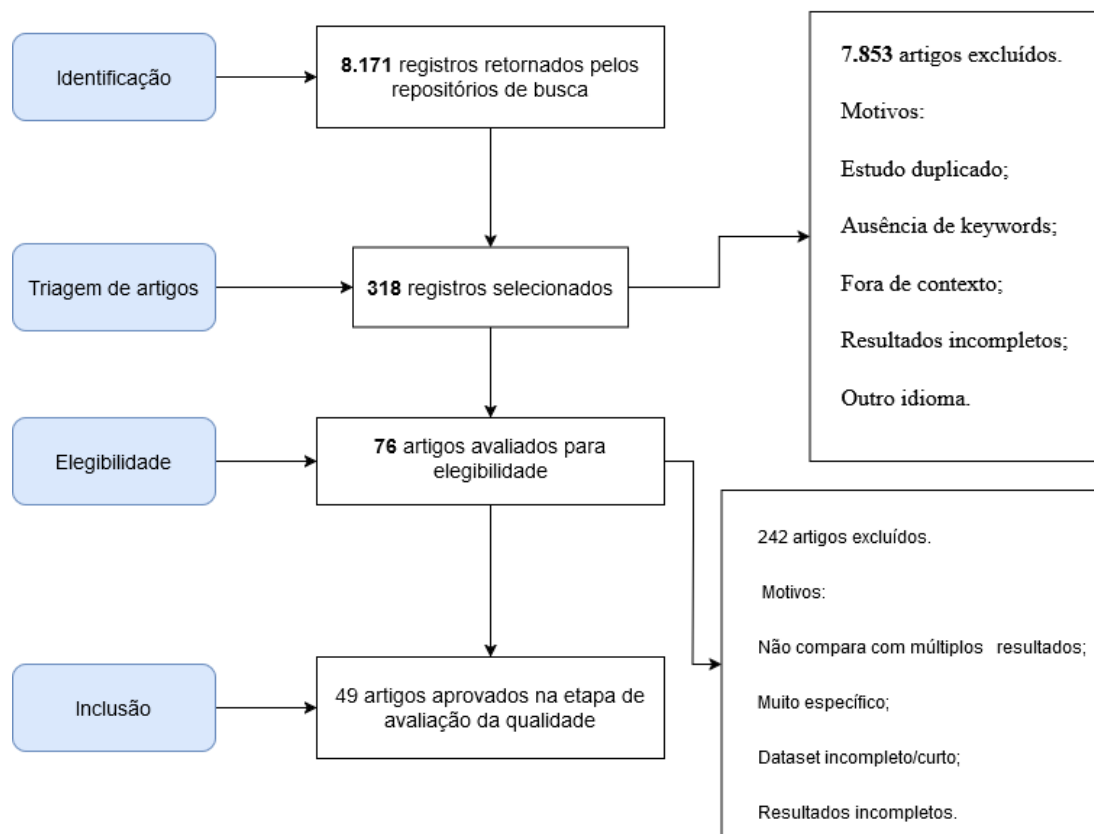


Figura 4 – Diagrama Prisma adaptado pelo autor.

Esses trabalhos avançaram para a fase de elegibilidade, onde foram avaliados em relação a todos os critérios de inclusão e exclusão, resultando em 76 estudos primários. Na etapa final de Condução, conforme ilustrado na Figura 4 (inclusão), os artigos foram lidos na íntegra e avaliados segundo os critérios de qualidade estabelecidos por esta Revisão Sistemática. Ao final, a RS selecionou os 49 estudos de maior qualidade sobre o problema de detecção de intrusões. Esses trabalhos respondem às questões de pesquisa levantadas nesta RS, visando produzir os resultados necessários.

3.2 Respostas das Questões de Pesquisa

Nesta seção, todos os estudos selecionados são analisados em detalhe para abordar as questões propostas neste trabalho, respondendo e comentando as questões de pesquisa. Dessa forma, será possível identificar o panorama atual da pesquisa e apontar as possíveis áreas de investigação futura.

RQ1: Como estão distribuídas as publicações sobre sistemas de detecção intrusão ao longo dos anos?

A Tabela 6 demonstra de maneira clara o crescimento significativo de 115% no número de pesquisas sobre sistemas de detecção de intrusão entre os anos de 2021 e 2024.

Esse aumento é evidente devido ao crescimento do número de ataques, o que reforça a necessidade de investigações para desenvolver soluções mais eficazes para sistemas de detecção de intrusão.

Tabela 6 – Quantidade de estudos por ano de publicação.

Ano da Publicação	Número de Publicações
2021	1.354
2022	1.810
2023	2.394
2024	2.619

RQ2: Quais tipos de contribuições têm sido desenvolvidos no contexto de sistemas de detecção de intrusões?

Uma visão geral dos tipos de estudos primários mais frequentemente publicados no indexador *IEEE Xplore* é mostrado na Tabela 7. Como evidenciado na tabela, as conferências são o principal destino dos trabalhos científicos, com um número de publicações duas vezes maior do que os journals. Essa predominância pode ser explicada pela natureza mais rigorosa e de escopo menor das publicações em *journals* em comparação as conferências. Além disso, as conferências oferecem discussões técnicas entre os pesquisadores nas sessões técnicas, o que enriquece o conhecimento sobre os temas abordados. Outro fator importante é o rápido crescimento dos estudos na área de detecção de intrusão, fazendo com que os trabalhos nessa área geralmente não esperem pelo processo mais longo de revisão e aceitação de *journals* e livros.

Tabela 7 – Quantidade de publicações por tipo no indexador IEEE Xplore.

Tipo de Publicação	Número de Publicações
<i>Conference</i>	4.229
<i>Jornal</i>	1.555
<i>Magazines</i>	20

RQ3: Quais são as principais formas de detecção utilizadas na literatura atual para IDS?

A categorização dos IDS tradicionais é comumente delineada em IDS baseados em assinaturas, IDS baseados em anomalias e híbridos (ALTUNAY; ALBAYRAK, 2023). Durante a minuciosa investigação dos estudos, foram encontrados somente 578 que utilizaram IDSs baseados em assinaturas com destaque notável para (YEOM; NIKNAMI; WU, 2023) e (JIN; CHUNG; XU, 2021). Outros 1.603 utilizaram outras abordagens IDSs baseados em anomalias. Torna-se claro que as pesquisas atuais estão majoritariamente voltadas para a abordagem baseada em anomalias.

As técnicas de detecção baseadas em assinaturas são rápidas e eficazes para detectar ataques conhecidos, mas não conseguem detectar ataques desconhecidos. O IDS baseado em anomalia gera um alarme quando detecta uma atividade que se desvia dos padrões normais de ataque. O IDS baseado em anomalias pode detectar ataques e tipos desconhecidos, mas gera alarmes falsos. Concluímos que cada abordagem de detecção tem suas próprias vantagens e desvantagens e apresenta melhor desempenho em diferentes conjuntos de dados.

RQ4: Quais são os resultados dos sistemas de detecção dos artigos mais qualificados?

Para abordar esta questão, foi necessário mapear os 49 estudos selecionados conforme descrito no anexo C, onde os resultados de acurácia variam de 0 a 1. Nos casos em que os estudos realizaram experimentos com múltiplos algoritmos, apenas o maior valor de acurácia será apresentado. Além da acurácia, esta tabela oferece informações adicionais que ajudam a responder outras questões de pesquisa, como o algoritmo de IA utilizado pelo autor em sua solução e a nota atribuída na análise de qualidade previamente descrita.

RQ5: Quais algoritmos de IA estão sendo utilizados pelos pesquisadores para construção de IDSs?

A Tabela 8 apresenta as 10 principais técnicas de aprendizagem identificadas nos estudos selecionados, visando determinar quais são mais frequentemente citadas na literatura. Vale notar que essas técnicas fazem uso de *Machine Learning* e *Deep Learning*.

Tabela 8 – Técnicas de aprendizagem mais utilizadas nos estudos.

Técnica Utilizada	Número de Citações
<i>Autoencoder</i> (AE)	6
<i>Decision Tree</i> (DT)	8
<i>Support Vector Machine</i> (SVM)	10
<i>Naive Bayes</i> (NB)	6
<i>K-Nearest Neighbors</i> (KNN)	8
<i>Random Forest</i> (RF)	11
<i>Convolutional Neural Networks</i> (CNN)	17
<i>Recurrent Neural Network</i> (RNN)	6
<i>Long Short-Term Memory</i> (LSTM)	17
<i>Hybrid Model</i> (HM)	14

Uma análise detalhada dos algoritmos de IA mais prevalentes nos estudos é mostrada na Tabela 8. Os algoritmos *Convolutional Neural Networks* e *Long Short-Term Memory* se destaca como os algoritmos mais utilizados, com 17 citações. Modelos de *machine learning* como *Support Vector Machine*, *Autoencoder*, *Random Forest*, *Naive Bayes*, *K-Nearest Neighbors* e *Decision Tree* são utilizados entre 11 e 6 citações. Por último, os modelos híbridos que combinam dois ou mais algoritmos obtêm 14 citações, merecendo destaque

a pesquisa realizada por (HALBOUNI et al., 2022), que desenvolveu um modelo híbrido envolvendo CNN, LSTM. Este estudo não apenas comparou o modelo com outros 4 modelos existentes na literatura, mas também o testou em três conjuntos de dados distintos e alcançou 0.996 de acurácia em classificação binária.

Destacamos o trabalho realizado por (VISHWAKARMA; KESSWANI, 2022), que desenvolveram um IDS baseado em *Deep Learning*, um sistema automatizado e robusto projetado para redes IoT. Esse IDS é capaz de detectar diversos tipos de ataques em tempo real por meio da captura de pacotes. O modelo foi testado utilizando diferentes conjuntos de dados provenientes da literatura (*NF-UNSWNB15*, *NF-BoTIoT*, *NF-ToNIoT*, *NF-CSE* e *CICIDS2018*).

Outro ambiente utilizando IoT foi proposto por (RAMANA et al., 2022). Eles empregaram um algoritmo de otimização de baleias em conjunto com Redes Neurais *Gated Recurrent Unit* (GRU) e conduziram comparações com outros 9 modelos. Os resultados obtidos foram notáveis, alcançando pontuações de 99.79% de acurácia, 99.86% de precisão, 99.82% de recall e 99.8% de F1-score.

O estudo realizado por (LIU; GU; WANG, 2021) empregou um modelo híbrido que combina *K-Means*, *Random Forest* e *Deep Learning* com CNN. Esse modelo foi avaliado usando os conjuntos de dados *NSL-KDD* e *CIC-IDS2017*, alcançando uma acurácia de 0.999. Os resultados desse estudo foram comparados com outros sete trabalhos.

O aspecto desfavorável relacionado aos ambientes de implementação e à disponibilidade dos códigos utilizados é que, muitas vezes, os autores não disponibilizam seus ambientes e códigos para que outros pesquisadores possam utilizá-los ou aprimorar suas próprias pesquisas.

RQ6: Quais conjuntos de dados são mais utilizadas nos estudos?

A Tabela 9 apresenta os principais conjuntos de dados identificados nos estudos selecionados, incluindo o número de citações e as fontes para consultas detalhadas ou download dos arquivos. A Figura 5 apresenta um gráfico com o detalhamento dos conjuntos de dados mais utilizados pelos pesquisadores. É notável a relevância do *NSL-KDD*, mencionado em 29 ocasiões, seguido pelo *CIC-IDS2017*, com 17 citações, o *UNSW-NB15*, com 15 menções, o *CIC-IDS 2018*, com 9 citações, o *KDD CUP 99*, com 8 citações e, por último, o *InSDN*, com 3 citações.

Chamou-nos a atenção o fato de que, mesmo sendo antigo e desatualizado, o *KDD Cup 99* continua sendo empregado em pesquisas atuais. Vale ressaltar que o *NSL-KDD* foi desenvolvido especificamente para corrigir as deficiências do *KDD Cup 99*. Muitos desses conjuntos de dados possuem certo desequilíbrio dos recursos e faltam ataques mais recentes. Além disso, é relevante observar que uma parcela significativa dos estudos utilizou conjuntos de dados do Instituto de Cibersegurança do Canadá (CYBERSECURITY, 2023).

Tabela 9 – Conjuntos de dados mais utilizados nos estudos.

Conjunto de dados	Número de Citações	Fonte
KDD CUP 99	8	(KDD, 1999)
NSL-KDD	29	(CYBERSECURITY, 2009)
UNSW-NB15	15	(WALES, 2015)
CIC-IDS 2017	17	(CYBERSECURITY, 2017)
CIC-IDS 2018	9	(CYBERSECURITY, 2018)
InSDN	3	(ELSAIED; LE-KHAC; JURCUT, 2020)

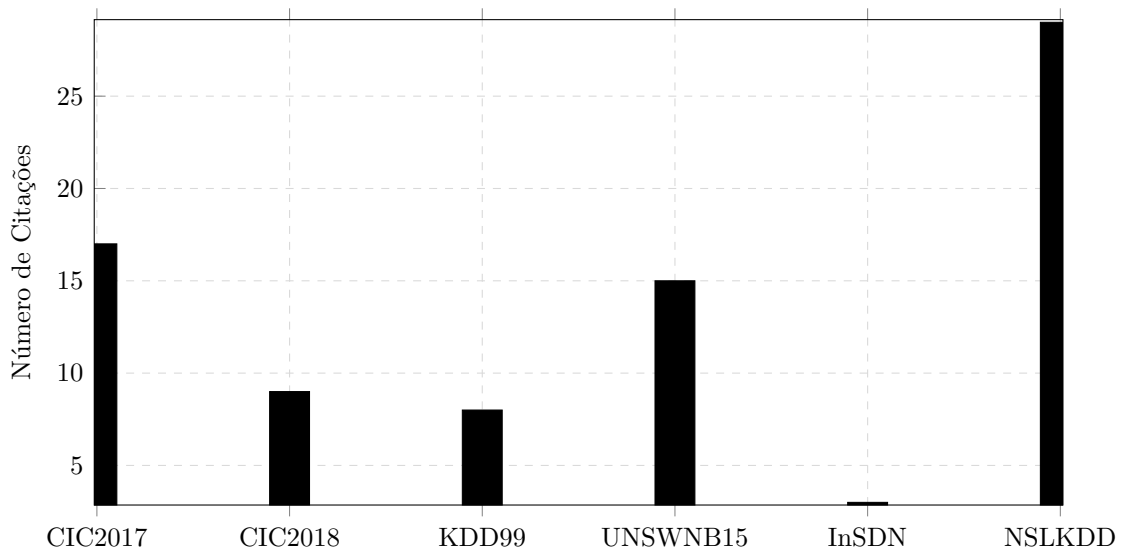


Figura 5 – Conjuntos de dados mais utilizados.

Segundo estudo realizado por (MEZINA; BURGET; TRAVIESO-GONZÁLEZ, 2021) o principal problema dos conjuntos de dados desatualizados é a incapacidade de representar os cenários de ataques modernos. Dado que os padrões comportamentais dos ataques de rede evoluíram, torna-se essencial dispor de conjuntos de dados atualizados para a realização de pesquisas.

RQ7: Quais sugestões e diretrizes podem ser propostas para pesquisas futuras na área de IDS?

Os incidentes de ataques cibernéticos estão crescendo de forma exponencial, e não há um método amplamente reconhecido para evitar todos esses ataques. Os IDS representam uma das abordagens mais significativas para reduzir ou prevenir ataques cibernéticos. Adicionalmente, os invasores estão aproveitando ferramentas e tecnologias de ponta para evadir IDSs, *firewalls* e sistemas antivírus.

Existem diversos aspectos a serem considerados para orientar pesquisas futuras na construção de IDS mais robustos e sofisticados. Em primeiro lugar, é fundamental aprofundar as investigações em técnicas avançadas de *Machine Learning* e *Deep Learning*, ou mesmo de forma híbrida combinando diversas técnicas para aproveitar o que há de

melhor em cada algoritmo.

Outro aspecto crucial envolve a utilização de conjuntos de dados mais recentes, ou até mesmo a criação de novos conjuntos de dados, como exemplificado no estudo conduzido por (ELSAYED; LE-KHAC; JURCUT, 2020). Nessa pesquisa, os cientistas identificaram a carência de conjuntos de dados adequados para aplicação em SDN e, em resposta, desenvolveram um conjunto de dados atualizado. Em seguida, testaram esse conjunto utilizando oito técnicas distintas de aprendizado. Para destacar a relevância desse item, podemos observar o recente estudo conduzido por (KUMAR et al., 2023a). Nesse estudo, os pesquisadores empregaram esse conjunto de dados em um contexto de SDN e o testaram com diversos algoritmos de *Machine Learning*.

Embora os estudos selecionados descritos no C apresentem altas taxas de acurácia, foi possível identificar diferenças significativas em outras métricas amplamente utilizadas na literatura. Além disso, o crescente impacto financeiro causado por ataques cibernéticos ressalta a constante necessidade de evolução das técnicas de aprendizagem, bem como do desenvolvimento de conjuntos de dados mais robustos e atualizados, que abranjam os ataques mais recentes.

3.3 Ameaças à Validade

Como em qualquer Revisão Sistemática, o processo de condução e análise dos resultados deste trabalho pode conter falhas devido a vieses na análise. Outro ponto importante é relacionado aos critérios de qualidade, que podem não ter sido adequados para considerar um estudo como relevante, devido a critérios subjetivos que dependem do perfil de cada pesquisador. Um dos critérios de inclusão/exclusão foi considerar apenas trabalhos escritos em inglês. Isso, de certa forma, limita a contribuição de autores que tenham escrito em outros idiomas.

3.4 Comparativo dos Trabalhos Relacionados

A Tabela 10 apresenta um comparativo resumido de alguns trabalhos relacionados descritos na seção anterior. Utilizamos os seguintes critérios para comparar os trabalhos:

- Conjunto de dados utilizados para treinamento das técnicas de aprendizagem;
- Técnicas de aprendizagem;
- Métrica de acurácia do modelo;
- Possui interface gráfica;
- Utiliza um ambiente virtual para realização do *testbench* das técnicas de aprendizagem;

- Nota máxima atribuída na etapa de avaliação dos trabalhos detalhados no Anexo C.

A abordagem proposta neste trabalho destaca a importância de uma ampla revisão sistemática da literatura, bem como do *framework* desenvolvido para avaliar técnicas de aprendizado na detecção de ataques cibernéticos. Esse *framework*, integrado a um ambiente virtual, oferece suporte ao usuário na execução de *testbed* para modelos de aprendizagem, facilitando o processo de avaliação e comparação.

Tabela 10 – Comparativo dos trabalhos relacionados.

Trabalhos Relacionados	Critérios Avaliados			
	Dataset	Modelo	Acurácia	Interface Gráfica
(XU et al., 2021)	NSL-KDD CICIDS2018	Autoencoder	0.931	Não
(SAID; SABIR; ASKERZADE, 2023)	UNSW-NB15 NSL-KDD	Hybrid CNN-BiLSTM	0.977	Não
(WISANW; THAMMAW, 2021)	NSL-KDD	Hybrid Naive Bayes/SVM	0.889	Não
(ABOBAKER, 2023)	CICIDS2017	KNN DT XGBC GB	0.996	Não
(HALBOUNI et al., 2022)	: CICIDS 2017 UNSW-NB15	Hybrid CNN-LSTM	0.996	Não
(HNAMTE et al., 2023)	CICIDS2017 CICIDS2018	LSTM-Autoencoder	0.999	Não
(LIU; GU; WANG, 2021)	NSL-KDD IDS2017	Hybrid K-Means/RF CNN	0.999	Não
(KANDHRO et al., 2023)	NSL-KDD KDD99 UNSW-NB15	DNN CNN RNN	0.969	Não
(C et al., 2023)	NSL-KDD	CNN XGB RNN LSTM SVM	0.980	Não
(ALHABSHY; HAMEED; ELDAHSHAN, 2022)	CICIDS2017 UNSW-NB15	SVM KNN DT	0.995	Não
(YOO et al., 2021)	NSL-KDD CICIDS2018,	MLP CNN	0.980	Não
(DAS et al., 2022)	NSL-KDD UNSW-NB15 CICIDS2017	ML SVM DT NB RL	0.995	Não
(THAKKAR; LOHIYA, 2023)	NSL-KDD UNSW-NB15 CICIDS2017	Ensemble DNN	0.989	Não
(SAH; SINGH; BANERJEE, 2024)	NSL-KDD CICIDS2017 CIC2019	KNN SVM RNN	0.999	Não
(CHEN et al., 2024)	CICIDS2018	ML XGBoost	0.999	Não
(HU et al., 2024)	UNSW-NB15 CICIDS2017 CICIDS2018	Hybrid LSTM	0.999	Não
(ELSAYED; MOHAMED; MADKOUR, 2024b)	NSL-KDD	Hybrid CNN RNN LSTM	0.999	Não
(SANNIGRAHI; THANDEESWARAN, 2024)	KDD99 NSL-KDD	Hybrid ML SVM NB RF	0.999	Não
Abordagem Proposta AI/IDS	CSE-CIC-IDS2018	Hybrid Transformer	0.999	Sim

4 Framework de Testes

Conforme discutido no Capítulo 3, onde foi realizada uma ampla e exaustiva revisão sistemática da literatura dos últimos quatro anos, identificou-se uma tendência significativa para o aprofundamento em novas técnicas e a exploração de diferentes conjuntos de dados.

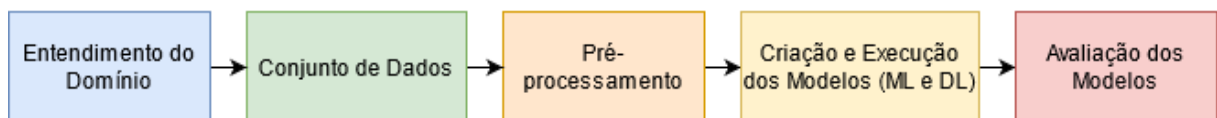
Nesse contexto, este trabalho também se propõe à criação de um *framework* híbrido, que utiliza *Transformer*, com o objetivo de avaliar a eficácia em conjuntos de dados amplamente reconhecidos e utilizados na literatura.

Este capítulo apresenta a proposta de um framework completo para detecção de intrusão híbrido utilizando Transformer para detecção de ataques cibernéticos com o uso de emuladores.

4.1 Visão Geral

Para desenvolver a metodologia proposta aplicamos conceitos de descoberta de conhecimento em bases de dados. Este processo é adaptado e dividido em cinco partes conforme é ilustrado na Figura 6. A seguir, cada etapa é detalhada no contexto da proposta.

Figura 6 – Etapas adaptadas do processo KDD utilizado na metodologia.



Fonte – Autor.

No processo de entendimento do domínio é realizada uma ampla pesquisa na literatura sobre IDS, técnicas de aprendizagem e conjuntos de dados mais utilizados com o objetivo principal de entender quais os principais problemas enfrentados pelos pesquisadores.

Neste trabalho, pode-se equiparar essa primeira etapa do processo aos estudos realizados para construção da motivação e descrição da problemática da proposta.

Para que seja possível a aplicação da metodologia é necessária um ambiente virtual que simula o mundo real utilizando uma série de ferramentas de software detalhadas no Apêndice A.

Dessa forma, é criado um projeto no GNS3 (Apêndice A.1) com configurações de hardware específicas. Neste experimento, é utilizado um computador com as seguintes

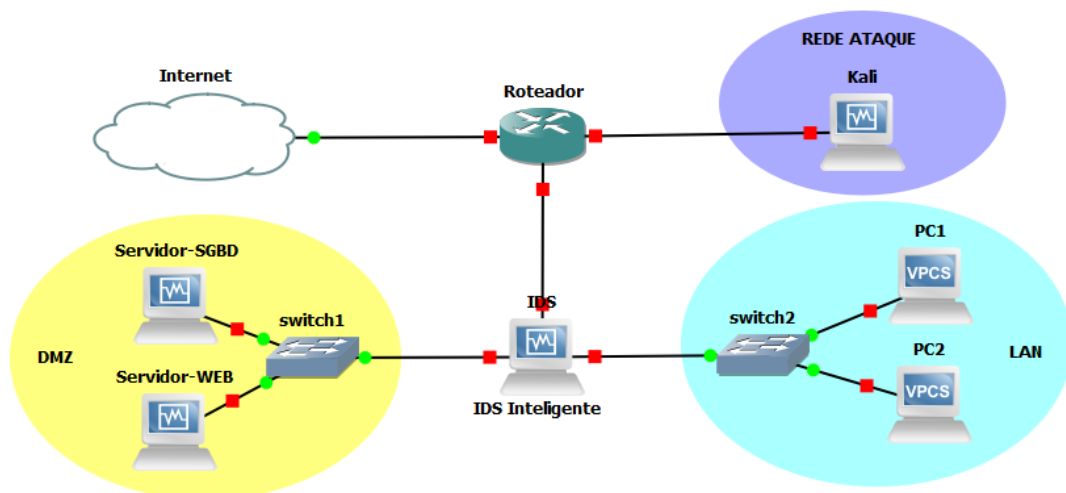
especificações:

- Computador com sistema operacional *Microsoft Windows 10 PRO*;
- Processador *Intel Core i5-11400 CPU 4.4GHz*;
- 32GB de memória RAM;
- GPU NVIDIA GTX 1650;
- Disco rígido de 1TB.

No entanto, por limitações de processamento e de uma GPU para execução dos modelos de aprendizagem profunda, optamos por transferir parte dos experimentos de treinamento desses modelos para a plataforma do *Google Colab* que possui máquinas configuradas para execução de algoritmos e vários modelos avançados de GPU.

A topologia do ambiente virtual é apresentada na Figura 7, sendo dividida em uma conexão com a Internet, um roteador, uma rede de ataque com uma máquina virtual (VM) *Kali Linux* com 4GB de memória, uma VM *Ubuntu 22.04 LTS* com 6GB de memória na figura do IDS inteligente, duas VMs *Ubuntu 22.04 LTS* na zona desmilitarizada (DMZ), sendo uma o servidor WEB e a outra o servidor SGBD, na outra interface a rede de área local (LAN) com dois virtual PC disponíveis no próprio GNS3. Lembrando que novos recursos podem ser instalados no projeto caso o usuário tenha necessidade de executar alguma configuração de ataque específico. Para isso, basta adicionar os dispositivos com as devidas configurações de rede.

Figura 7 – Topologia do experimento.



Fonte – Autor.

4.2 Rede de Ataque

Essa rede será responsável por simular ataques cibernéticos seja para rede LAN ou para algum dos servidores da DMZ (WEB ou SGBD). A VM *Kali* possui instaladas as

ferramentas básicas do sistema e adicionamos algumas outras para facilitar sua utilização como: pacote anaconda *Python3*, *Scapy* para gerar pacotes de dados personalizados de acordo com o tipo de ataque desejado, *script Slowloris* para gerar ataques do tipo negação de serviço e *CICFlowMeeter*.

A rede de ataque está configurada na faixa de rede 40.0.0.0/8, o roteador com endereço IP 40.0.0.1/8 e a VM *Kali Linux* com o endereço 40.0.0.2/8.

4.3 IDS

O principal componente da proposta é o IDS, uma VM *Ubuntu* que é responsável por verificar o tráfego de rede e detectar possíveis intrusões através da ferramenta *CICFlowMeeter*, pacote Anaconda *Python3*, uma interface gráfica desenvolvida em *Python3* para gerenciar os modelos de aprendizagem e um banco de dados *Mysql* para guardar as informações de detecções para posterior avaliação.

O IDS possui 3 interfaces de rede. Sendo uma para ligação com o roteador com o endereço IP 20.0.0.1/8 e 20.0.0.2/8, outra para a DMZ onde ficam os servidores na faixa de rede 50.0.0.0/8, e a última interface para LAN dos virtual PC na faixa de endereços 10.0.0.0/8.

4.4 DMZ e LAN

A norma (NBR 17799, 2005) afirma, em um dos seus controles, que um método de controlar a segurança da informação em grandes redes é dividi-la em domínios de redes lógicas diferentes. De fato, esta é uma prática comum em redes de computadores estruturadas que garante acesso restrito a certos serviços. Neste caso, criamos uma divisão lógica, ou uma sub-rede, chamada de DMZ.

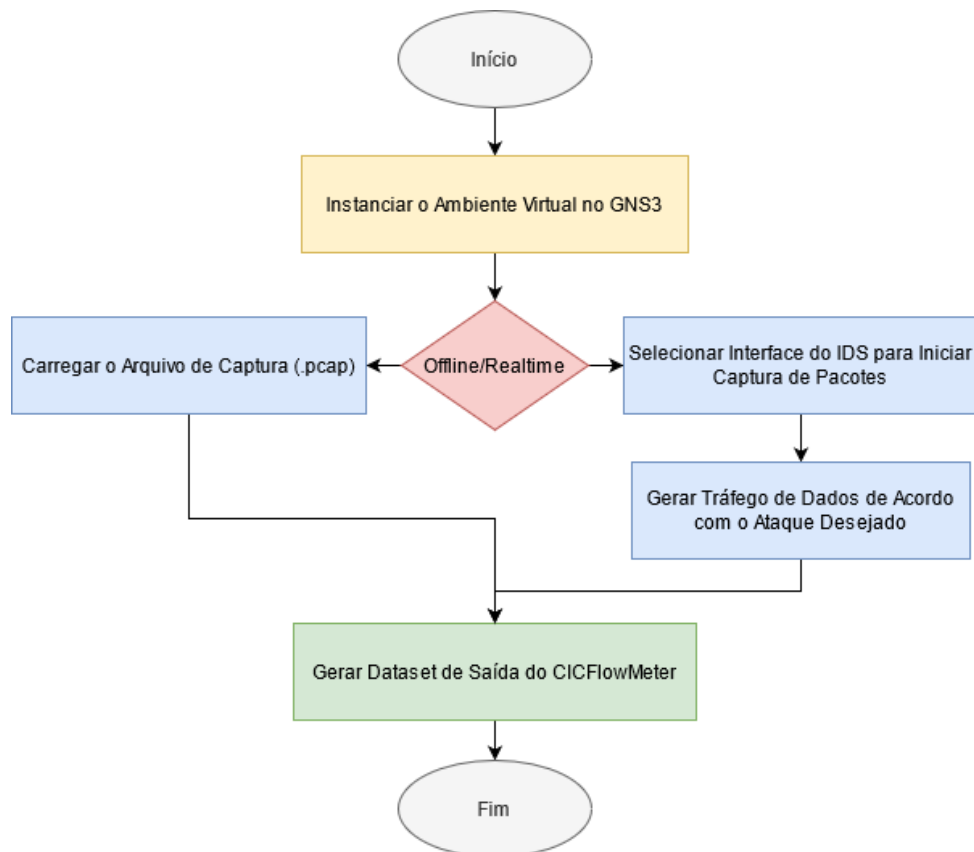
Esta rede é composta por duas VMs, ambas com sistema operacional *Ubuntu 22.04 LTS*. A VM do servidor WEB possui endereço IP 50.0.0.2/8 e o apache2 foi instalado para prover o serviço. A VM do SGBD possui o endereço de rede 50.0.0.3/8. Foram instalados os seguintes serviços nessa VM: *apache2*, *SSH*, *FTP*, *Mysql* e *Phpmyadmin*.

Uma rede que abrange uma área mais ampla, onde pode-se conter vários prédios dentro de um espaço contínuo ligados em rede. Está segundo (TANENBAUM, 2003) é uma LAN, justamente porque esta área dita ampla, abrange 10 quarteirões ou aproximadamente 2.500m quadrados. Para simular este ambiente do mundo real uma LAN foi inserida em umas das interfaces do IDS, sendo conectada a um *switch* do próprio GNS3. A faixa de rede da LAN é 10.0.0.0/8.

4.5 Base de Dados

A etapa de conjunto de dados é detalhada na Figura 8. Essa etapa consiste em gerar um *dataset* através da ferramenta *CICFlowMeter*. Ao instanciar o ambiente virtual no GNS3 e executar o *CICFlowMeter* é necessário escolher um dos dois modos de operação da ferramenta: *Offline* e *Realtime*. O modo *Offline* gera um *dataset* (.csv) a partir de uma captura de pacotes (.pcap) fornecida pelo usuário. Em *Realtime* ela gera um *dataset* a partir da captura de pacotes de uma interface de rede selecionada pelo usuário. Após seleção da interface de rede é necessário gerar tráfego de dados de acordo com o tipo de ataque desejado para realização do *testbench*. No final de ambos os modos a ferramenta produz um *dataset* (.csv) que é utilizado na próxima etapa do experimento.

Figura 8 – Fluxo de execução para gerar o *dataset*.



Fonte – Autor.

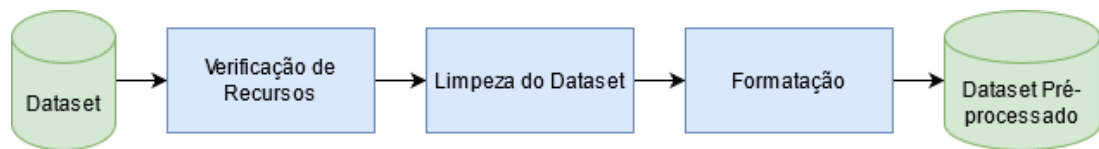
4.6 Pré-processamento

Na etapa de pré-processamento dos dados, realizamos o estudo detalhado dos 84 recursos que a ferramenta *CICFlowMeter* gerou. A Figura 9 detalha as etapas desse processo disponibilizado no script *preprocessing.py* disponibilizado no Github deste trabalho.

O script carrega o dataset e os dados são processados linha por linha para a limpeza de rótulos, remoção de registros duplicados, correção de dados ausentes e erros de formatação.

Os recursos "*Flow ID*", "*Port Dest*", "*Port Src*" e "*Protocol*" foram removidos pois não agregam no processo de detecção de intrusão e podem gerar erros no processo de aprendizagem. Para organizar os dados de forma equilibrada os dados foram submetidos a padronização utilizando o pacote *sklearn*. O recurso "*Timestamp*" que possui registro de data foi convertido para um valor numérico, utilizando a função *datetime*. Após finalizar o processo de limpeza e formatação um novo *dataset* é gerado o (*Dataset Pré-processado*).

Figura 9 – Etapas de pré-processamento.



Fonte – Autor.

5 Modelo AI/IDS Transformer

Para criar o *framework* proposto combinamos o *Transformer* para formar um novo modelo híbrido, que pode atingir melhor desempenho do que algoritmos tradicionais de aprendizado profundo e modelos de aprendizado de máquina. O fluxograma do experimento do modelo híbrido é mostrado na Figura 10.

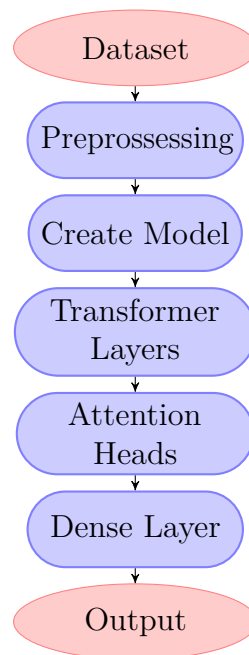


Figura 10 – Fluxograma do experimento.

5.1 Transformer Layer

O modelo proposto é baseado no *TabTransformer* que foi projetado para resolver problemas de classificação binária usando dados tabulares, que consistem apenas em variáveis contínuas. O modelo é configurado para processar as interações entre essas variáveis contínuas através de camadas *Transformer*, que utilizam mecanismos de atenção multi-cabeças para capturar relações complexas nos dados (AMAZON, 2024).

De acordo com o estudos realizados por (HUANG et al., 2020) e (WANG et al., 2024), o modelo *TabTransformer* supera os métodos de aprendizado profundo de última geração para dados tabulares. Por essa razão, foi escolhido como o modelo para a realização dos testes de detecção de intrusão. Além disso, foram utilizados os parâmetros padrão do modelo, conforme especificado no estudo mencionado. Os parâmetros utilizados são detalhados no *Listing* 5.1.

O parâmetro (`num-features`) define a quantidade de variáveis contínuas, enquanto a estrutura do modelo inclui 6 camadas *Transformer* (`depth=6`) com 8 cabeças de atenção em cada camada (`heads=8`), permitindo ao modelo analisar diferentes aspectos dos dados em paralelo. A saída (`dim-out=1`) é um único valor que pode ser interpretado como a probabilidade de pertencer à classe positiva (ataque) em uma tarefa de classificação binária. Os hiperparâmetros, como a dimensão do *embedding* (`dim=64`), dropout para a atenção (`attn-dropout=0.1`) e para o *feedforward* (`ff-dropout=0.1`), são ajustados para garantir que o modelo generalize bem e não sofra de *overfitting*.

```
1 def create-tab-transformer_binary(num_features):
2     model = TabTransformer(
3         categories=(),
4         num-continuous=num-features,
5         dim=64,
6         dim_out=1,
7         depth=6,
8         heads=8,
9         attn_dropout=0.1,
10        ff_dropout=0.1
11    )
12    return model
```

Listing 5.1 – Código em *Python* de criação do modelo proposto.

A função de perda utilizada foi a função *BCEWithLogitsLoss()* que é uma função de perda combinada que é usada para tarefas de classificação binária. Ela combina duas etapas:

- *Sigmoid*: Calcula a probabilidade de uma classe (0 ou 1) a partir da saída do modelo.
- *Binary Cross-Entropy*: Mede a diferença entre as probabilidades previstas pelo modelo e os rótulos reais (0 ou 1).

A vantagem de usar *BCEWithLogitsLoss* é que ela aplica a função sigmoid diretamente à saída do modelo antes de calcular a perda, o que proporciona maior estabilidade numérica durante o treinamento, evitando a necessidade de calcular manualmente a função *sigmoid* e a perda *binary cross-entropy* separadamente.

A otimização do modelo utilizou a função: `optimizer = torch.optim.Adam(model.parameters(), lr=LEARNING-RATE)`. O otimizador usando o algoritmo Adam, que ajusta dinamicamente a taxa de aprendizado de cada parâmetro do modelo durante o treinamento. A variável `LEARNING-RATE` define a taxa de aprendizado, controlando o tamanho dos ajustes feitos nos parâmetros a cada iteração, a taxa utilizada foi 0.001.

No modelo proposto, foram utilizadas 10 épocas de treinamento, o que significa que o modelo passou por todo o conjunto de dados de treinamento 10 vezes. Durante cada uma dessas épocas, o modelo ajustou seus parâmetros (pesos e vieses) com base no erro calculado, com o objetivo de melhorar suas previsões.

5.2 Métricas de avaliação

Existem várias métricas disponíveis para avaliar e comparar o desempenho de classificadores. Neste trabalho utilizamos o que foi apresentado por (SHARAFALDIN; GHORBANI, 2018) e (WANG, 2018):

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (5.1)$$

$$Precision = \frac{TP}{TP + FP} \quad (5.2)$$

$$recall = \frac{TP}{TP + FN} \quad (5.3)$$

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \quad (5.4)$$

Em que verdadeiro positivo (TP) é um resultado em que o modelo prediz corretamente a classe positiva, falso positivo (FP) é um resultado em que o modelo prediz incorretamente a classe positiva, falso verdadeiro (TN) é um resultado em que o modelo prediz corretamente a classe negativa e falso negativo (FN) é um resultado em que o modelo prediz incorretamente a classe negativa.

Geramos também algumas matrizes de confusão para fornecer uma maneira visual de exibir o desempenho detalhado das melhores técnicas, mostrando o número total de amostras classificadas corretamente e incorretamente de acordo com tipo de tráfego de rede benigno ou malicioso. Uma matriz de confusão é muito útil para avaliar não apenas classificadores de classe binária, mas também classificadores de várias classes.

6 Resultados e Discussão

Nesse capítulo, é apresentada a avaliação de desempenho do AI/IDS. O sistema é avaliado comparando as métricas de desempenho de predição binária em termos de Acurácia, Precisão, *Recall* e *F1-score*. Cada um desses indicadores de desempenho do AI/IDS foi comparado com os outros métodos comumente usados em outros trabalhos de IDS. A métrica de acurácia visa verificar em qual rodada ocorre a convergência do AI/IDS em comparação com outros métodos.

6.1 Experimentos

Para testar o framework desenvolvido, foram realizados dois experimentos diferentes. No primeiro, avaliamos o desempenho do sistema usando dados amplamente reconhecidos na literatura, como o conjuntos de dados: *CSE-CIC-IDS2018*. Já no segundo, buscamos algo mais próximo da realidade: simulamos um ataque real do tipo DDoS para verificar como o *framework* se comportaria em situações práticas. Cada etapa desses experimentos será explicada com mais detalhes a seguir. Para simplificar a execução e análise dos modelos previamente treinados, foi desenvolvida uma interface gráfica em *Python* que torna o processo mais intuitivo e acessível.

6.2 Avaliação do Framework

Para avaliar o desempenho das técnicas mencionadas, realizamos dez rodadas de treinamento e testes com diferentes conjuntos de dados. Esses conjuntos simulavam um dia típico de tráfego na rede, incluindo tanto atividades normais quanto ataques. A divisão dos dados foi feita de forma equilibrada, com 80% para o treinamento e 20% para os testes, utilizando a função *"train-test-split"* da biblioteca *"sklearn"*. Após o treinamento, escolhemos a rodada com o melhor desempenho e aplicamos as métricas de avaliação para analisar os resultados obtidos.

A Figura 11 apresenta uma matriz de confusão, que ilustra a quantidade de registros classificados corretamente e incorretamente pelo modelo. Taxas elevadas de falsos positivos (FP) e falsos negativos (FN) indicam que o modelo pode ter dificuldades em identificar as particularidades de diferentes tipos de tráfego de rede, ou que o próprio conjunto de dados tem limitações na captura de certos ataques.

No caso do modelo AI/IDS, ao ser avaliado no conjunto de dados *CSE-CIC-IDS2018*, obteve 98.99% registros classificados corretamente como tráfego benigno (classe binária 0)

e 99.02% registros classificados corretamente como ataque (classe binária 1). No entanto, o modelo cometeu 0.88% falsos positivos (FP), ou seja, classificou erroneamente tráfego benigno como se fosse um ataque. Além disso, foram encontrados 1.01% falsos negativos (FN), onde o modelo classificou tráfego benigno como sendo ataque.

Em testes realizados exclusivamente com a classe de ataque *Infiltration* com o modelo CNN e MLP a acurácia obtida foi de apenas 0.881 e 0.819 e f1-score de 0.881 e 0.819. O que evidencia a dificuldade dos algoritmos em detectar esses tipos de ataques sendo essa uma grande falha do conjunto de dados.

Considerando o estudo realizado por (SILVA, 2024), que fornece uma abordagem minuciosa sobre o desbalanceamento do conjunto de dados *CSE-CIC-IDS2018*. É possível entender algumas dificuldades relacionadas a duas classes de ataques: *WebAttack* e para a classe *Infiltration*. O modelo mais promissor desse trabalho atingiu os seguintes resultados para o algoritmo *LightGBM*: Acurácia de 0.954 e 0.952 para a métrica *f1-score*.

Há um grande número de estudos que utilizam os datasets *CSE-CIC-IDS2018*; contudo, muitos deles não dedicam a devida atenção a aspectos como o desbalanceamento dos dados e o uso de métricas adequadas para avaliar o desempenho com ataques de classes minoritárias. Como resultado, frequentemente apresentam classificadores com acurácias extremamente altas, mas sem investigar a possibilidade de viés ou ocorrência de *overfitting* (SILVA, 2024).

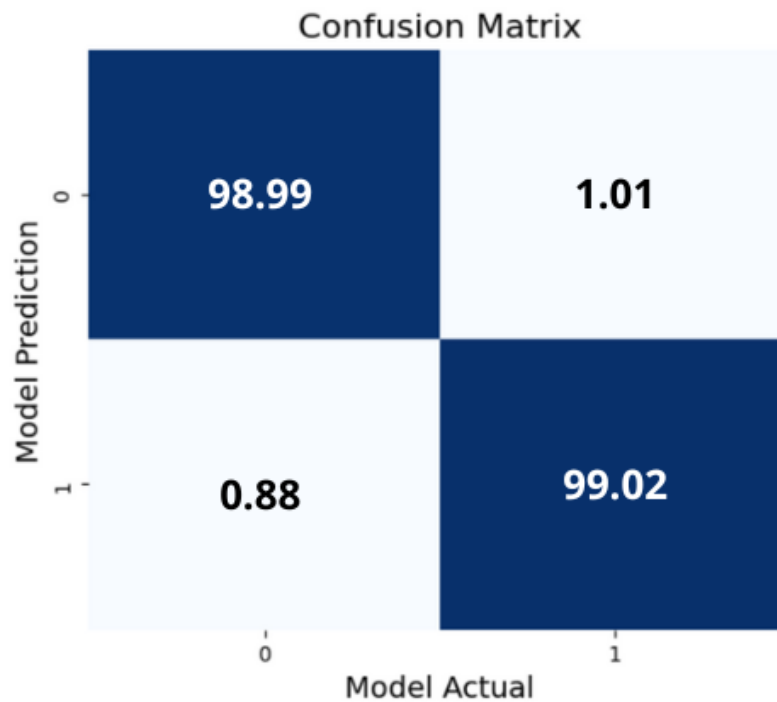
A Figura 12 ilustra a progressão do modelo durante o processo de treinamento. Ela mostra a mudança na perda em cada período, isto é, ao longo do tempo. No eixo horizontal, temos a quantidade de períodos totalizando 50, enquanto no eixo vertical se encontra o valor da perda, que demonstra a adequação do modelo aos dados.

A meta é que, ao longo das épocas, esse valor de perda diminua, sinalizando que o modelo está evoluindo e se aprimorando. Este gráfico nos permite avaliar se o treinamento está sendo realizado de forma eficaz ou se existem indícios de problemas, como quando a perda não diminui como o previsto. É possível observar que, a partir da 40ª época, o modelo se estabiliza, apresentando perdas mínimas. Apesar de ajustarmos diversas vezes a taxa de aprendizado e o número de épocas, não foi possível melhorar a precisão do modelo.

6.3 Comparativo dos trabalhos

O modelo proposto, *AI/IDS Transformer*, obteve resultados satisfatórios, conforme demonstrado na Tabela 11. Nessa tabela, comparamos os trabalhos levando em conta a técnica proposta, o conjunto de dados utilizado e as métricas de acurácia dos modelos. O desempenho do modelo se destaca entre os melhores avaliados, com base nos resultados dos estudos mais relevantes apresentados na revisão sistemática do Capítulo 3, que também

Figura 11 – Matriz de confusão normalizada do *AI/IDS Transformer* com o CSE-CIC-IDS2018.



Fonte – Autor.

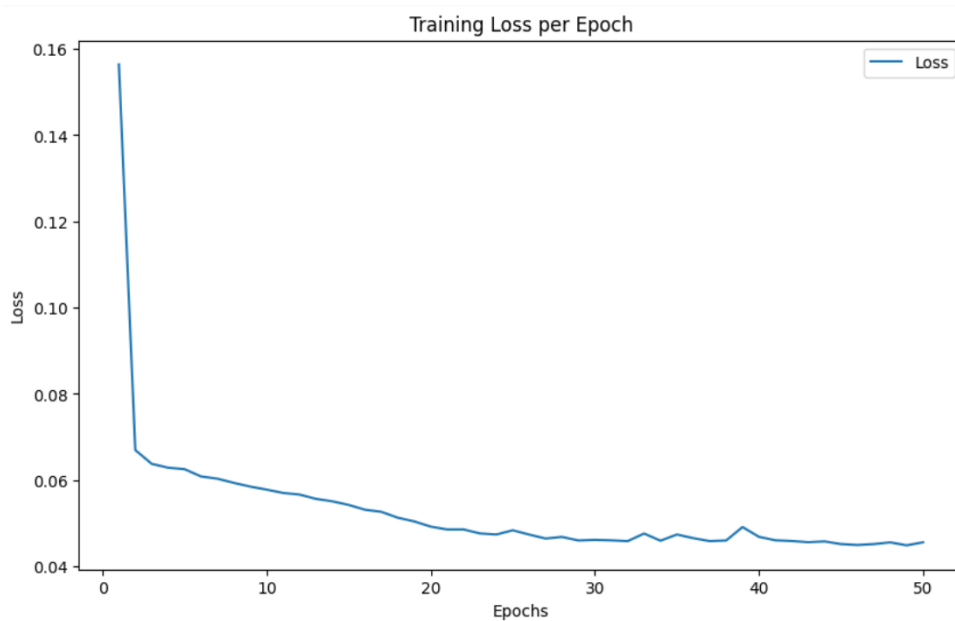
estão disponíveis em mais detalhes no Anexo C.

O *AI/IDS Transformer* demonstrou maior eficiência em comparação com outros seis estudos, obtendo uma acurácia superior a 0,992 pontos ao ser avaliado com o conjunto de dados *CSE-CIC-IDS2018*. Contudo, apresentou desempenho inferior a três estudos. No total, foram identificados, no processo de revisão de literatura, oito estudos que utilizaram, mesmo que parcialmente, o conjunto de dados abordado em nossa proposta. Os estudos apresentados na Tabela 11 serão analisados em maior detalhe a seguir.

O estudo conduzido por (AMALAPURAM et al., 2022) utiliza uma técnica de aprendizado contínuo como uma nova abordagem para detectar ataques modernos, buscando solucionar os problemas de desequilíbrio dos conjuntos de dados. Esse estudo empregou os conjuntos KDD 99, CIC-IDS2017 e CIC-IDS2018. A abordagem apresentou um dos piores resultados gerais. A acurácia do modelo utilizando o algoritmo MLP atingiu 0,932 pontos, com f1-score de 0,753 no conjunto de dados CSE-CIC-IDS2018. No entanto, obteve melhores resultados nos outros conjuntos de dados.

A pesquisa conduzida por (HAI; NAM, 2021) apresenta uma comparação de eficiência entre três abordagens de Deep Learning: RNN (Recurrent Neural Network), LSTM (Long Short-Term Memory) e GRU (Gated Recurrent Unit), aplicadas ao problema de detecção de anomalias em dois conjuntos de dados, CICIDS2017 e CSE-CICIDS2018.

Figura 12 – Variação da perda ao longo das épocas durante o treinamento do modelo.



Fonte – Autor.

Os algoritmos LSTM e GRU alcançaram as melhores pontuações para o conjunto de dados CIC-IDS2018, com acurácia de 0,954, precisão de 0.997 e f1-score de 0.975. Este trabalho destaca a necessidade da construção de novos conjuntos de dados mais robustos e balanceados.

Uma abordagem diferente foi utilizada no estudo realizado por (RATTI; SINGH; NANDI, 2023). Neste artigo, é proposta uma técnica de aprendizado baseada em um codificador automático LSTM com reconhecimento de tempo para detectar ataques em fluxos de rede, treinando o modelo exclusivamente com tráfego normal e utilizando o erro de reconstrução como parâmetro para classificar eventos de ataque. O modelo apresentou acurácia de 0,975 e f1-score de 0,983. O ponto negativo desse trabalho foi a ausência de comparações do modelo proposto com outras abordagens da literatura.

O estudo realizado por (MEZINA; BURGET; TRAVIESO-GONZÁLEZ, 2021) apresenta uma discussão interessante sobre a utilização de conjuntos de dados antigos e mais recentes. No caso do estudo, foram empregados os conjuntos de dados KDD Cup 99 e CIC-IDS2018. Esse trabalho utilizou um modelo híbrido para detecção de intrusão, combinando *LSTM* e *Temporal Convolutional Network*. O modelo alcançou 0,977 pontos de acurácia para o conjunto de dados CIC-IDS2018. Apesar de realizar comparações com outros modelos desenvolvidos pelo autor, o estudo não promoveu discussões com outras pesquisas encontradas na literatura.

Um modelo híbrido utilizando LSTM e Autoencoder foi proposto por (HNAMTE et al., 2023). O modelo foi comparado com outros 26 estudos da literatura, considerando a

quantidade de recursos utilizados, ano de publicação, algoritmo, conjunto de dados e métrica de acurácia. Além disso, o modelo foi testado com outras duas técnicas desenvolvidas pelo autor: DNN e CNN. O modelo alcançou 0,991 pontos de acurácia para o LSTM-AE. Um ponto notável foi a análise do parâmetro de tempo de inferência neste estudo, que se mostrou consideravelmente superior ao de outros modelos reportados na literatura.

O trabalho proposto por (YOO et al., 2021) utiliza um método de pré-processamento de discretização com um classificador baseado em CNN para abordar o problema de desequilíbrio de classes nos dados de tráfego de rede dos conjuntos de dados NSL-KDD e CIC-IDS2018. A abordagem converte o tráfego de rede em imagens, que são aplicadas ao modelo baseado em CNN. O modelo alcançou uma acurácia de 0,980. O estudo realizou quatro formas diferentes de pré-processamento, buscando melhorias nas métricas de precisão, f1-score e recall.

Outra abordagem híbrida foi desenvolvida por (HU et al., 2024). Neste estudo, foi utilizado o BiLSTM juntamente com um módulo de atenção híbrido, que extrai as informações desejadas com maior precisão, melhorando a qualidade das informações de entrada e a precisão do modelo ao capturar os recursos relevantes. O modelo foi testado com os conjuntos de dados UNSW-NB15, CICIDS-2017 e CICIDS-2018. Além disso, foi comparado com outros 15 estudos encontrados na literatura. O modelo proposto atingiu uma acurácia de 0,990 com o CSE-CID-IDS2018.

Na pesquisa conduzida por (CHEN et al., 2024). Um sistema híbrido de detecção de intrusão de rede que combina NIDS e HIDS é proposto para melhorar o desempenho da detecção de intrusão. um modelo de Representações de Codificador Bidirecional de Transformadores (BERT) baseado em Processamento de Linguagem (NLP) de representação textual para numérica, a fim de processar dados do host de forma semelhante aos dados de fluxo de rede por meio de modelos de aprendizado de máquina com o algoritmo XGBoost. Este estudo utilizou os conjuntos de dados CIC-IDS2018 e NDsec-1. os recursos foram combinados baseados em rede e host usando uma abordagem de achatamento de recursos e estudamos o impacto da redução de dimensão em recursos de mensagem nos hosts. Usamos algoritmos de ML tradicionais (por exemplo, XGBoost) e um classificador colaborativo de dois estágios para detectar intrusões. O primeiro estágio de estimativa usou um classificador binário para discriminar tráfego benigno e de ataque, e o segundo estágio usou um classificador multiclasse para discriminar vários tipos de ataque.

Na pesquisa realizada por (CHEN et al., 2024), é proposto um sistema híbrido de detecção de intrusão em rede que combina NIDS e HIDS para melhorar o desempenho na detecção de intrusões. O estudo utiliza um modelo de Representações de Codificador Bidirecional de Transformadores (BERT) baseado em Processamento de Linguagem Natural (NLP) para converter dados textuais em números. Isso permite processar dados dos hosts de forma similar aos dados de fluxo da rede, utilizando modelos de aprendizado com o

algoritmo XGBoost. Este estudo utilizou os conjuntos de dados CICIDS 2018 e NDSec-1. A combinação desses tipos de IDS apresentou excelentes resultados.

Tabela 11 – Comparação entre os trabalhos discutidos na revisão sistemática.

Trabalho	Dataset	Modelo	Acurácia
(AMALAPURAM et al., 2022)	CIC-IDS2018	CNN MLP	0.932
(HAI; NAM, 2021)	CIC-IDS2018	RNN LSTM GRU	0.938
(RATTI; SINGH; NANDI, 2023)	CIC-IDS2018	Hybrid LSTM MLP	0.975
(MEZINA; BURGET; TRAVIESO-GONZÁLEZ, 2021)	CIC-IDS2018	Hybrid LSTM	0.977
(HNAMTE et al., 2023)	CIC-IDS2018	LSTM-Autoencoder	0.991
(YOO et al., 2021)	CIC-IDS2018	CNN	0.980
(HU et al., 2024)	CIC-IDS2018	Hybrid LSTM	0.999
(CHEN et al., 2024)	CIC-IDS2018	ML XGBoost	0.999
Modelo Proposto: AI/IDS Transformer	CIC-IDS2018	Transformer	0.992

6.4 Experimento no Ambiente Virtual

Para simular um ataque real, utilizamos o ambiente virtual para avaliar a eficácia do modelo proposto. No cenário experimental, com a rede de ataque utilizando a VM Kali Linux, onde executamos o script da ferramenta Slowloris, direcionando o ataque ao endereço IP de um servidor *Web Apache* instalado em uma VM na rede interna DMZ. O *Slowloris* é uma ferramenta de ataque de camada de aplicação que utiliza solicitações HTTP parciais. Ele funciona abrindo conexões com o servidor alvo e mantendo-as ativas pelo maior tempo possível. Embora não seja uma categoria de ataque, o *Slowloris* foi projetado para permitir que uma única máquina derrube um servidor utilizando uma quantidade mínima de largura de banda (OKTIVASARI et al., 2022).

O tráfego de pacotes foi capturado utilizando a ferramenta *CICFlowMeter*, permitindo a criação de um dataset específico contendo os pacotes gerados pelo ataque realizado pela *VM Kali Linux* com o *script Slowloris*. Após a captura, o dataset foi pré-processado e submetido ao modelo AI/IDS proposto para avaliar sua capacidade de detectar o ataque. O script foi configurado para estabelecer 2.000 sockets, ou seja, 2.000 conexões entre a máquina virtual de ataque *Kali Linux* e o servidor *Web Apache*, resultando na indisponibilidade do servidor.

O modelo proposto demonstrou eficácia satisfatória na detecção de ataques de Negação de Serviço Distribuído (DDoS), alcançando uma taxa de 96% de acurácia. No entanto, ressalta-se a necessidade de validar o modelo em um ambiente de rede real. Para isso, está em andamento a configuração de testes na rede do Instituto Federal do Maranhão (IFMA) - Campus Grajaú, que conta com mais de 300 usuários e diversos servidores Web em operação. Esses testes estão sendo acordados em parceria com a Coordenação de Tecnologia da Informação do campus, com o objetivo de aprimorar a segurança das regras de firewall do sistema PfSense atualmente em uso.

Outro ponto forte do *framework* proposto é a sua capacidade de escalabilidade, pois novas máquinas virtuais podem ser adicionadas facilmente ao ambiente, sendo a principal limitação o próprio hardware do dispositivo hospedeiro do software GNS3. O ajuste pode ser realizado por meio da configuração dos respectivos endereços IP das máquinas virtuais. Nesse contexto, novos serviços podem ser executados para a realização de testes diferenciados, de acordo com a necessidade do pesquisador.

6.5 Experimento com Interface Gráfica

Propomos inicialmente a utilização de um dos cinco modelos de aprendizagem: *Multi Layer Perceptron* (MLP); *Random Forest* (RF); *Naive Bayes* (NB), *k-Nearest Neighbors* (KNN) e AI/IDS Transformer.

Para construção dos modelos utilizamos inicialmente o conjunto de dados *CSE-CIC-IDS2018* que possui aproximadamente cerca de 230 *gigabytes*. O conjunto de dados está dividido em arquivos em tráfego de rede original no formato (.pcap) e arquivos pré-processados e rotulados no formato (.csv). Usamos em nosso trabalho os arquivos rotulados que são compostos por tráfego de rede benigno e tráfego de rede malicioso. O tráfego malicioso é dividido em 13 tipos de ataques sendo eles: *FTP-BruteForce*; *SSH-Bruteforce*; *DoS attacks-GoldenEye*; *DoS attacks-Slowloris*; *DoS attacks-SlowHTTPTest*; *DoS attacks-Hulk*; *Brute Force-Web*; *Brute Force-XSS*; *SQL Injection*; *Infiltration*; *Bot*; *DDOS attack-LOIC-UDP* e *DDOS attack-HOIC*.

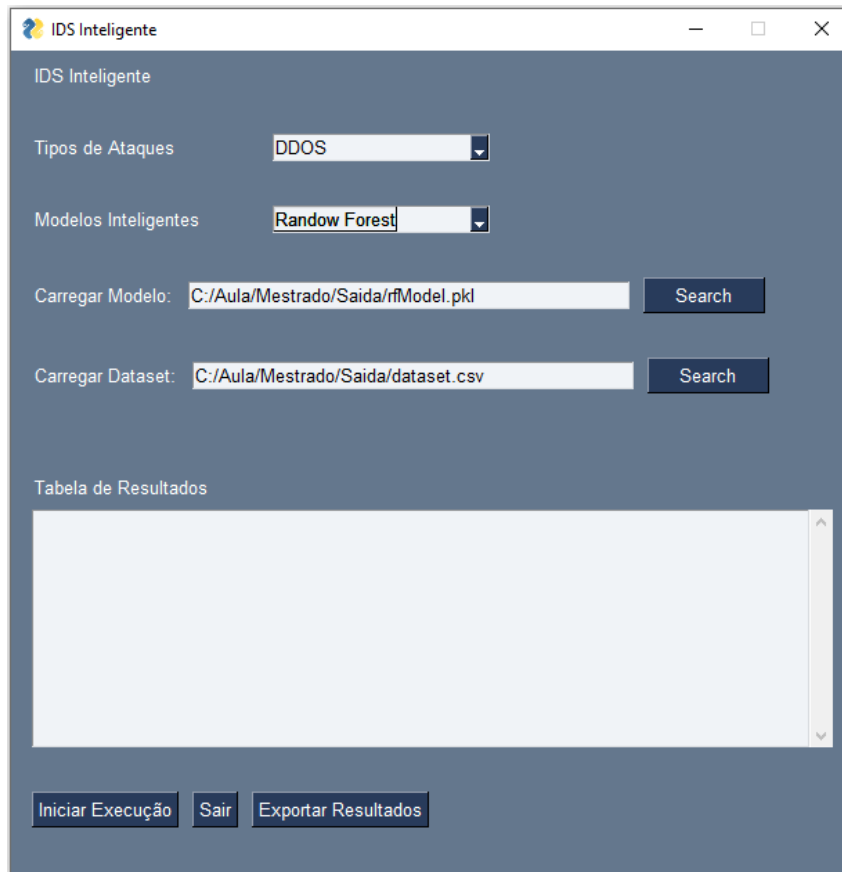
A avaliação de desempenho das técnicas mencionadas anteriormente foi realizada treinando e testando cada técnica de aprendizado 10 (dez) vezes com diferentes conjuntos de dados, correspondendo há um dia de tráfego de rede contendo atividades benignas e maliciosas. Utilizando a função *"train test split"* da biblioteca *"sklearn"* para dividir o conjunto de dados de forma balanceada nas proporções 80% para fase de treinamento e 20% para fase de testes. Após o treinamento dos modelos, a melhor rodada foi selecionada e as métricas de desempenho foram aplicadas. Esses modelos foram criados e treinados utilizando a plataforma *Google Colab* e exportados no formato (.pkl) para facilitar a execução da aplicação.

Para facilitar a execução dos modelos de aprendizagem para o usuário, uma interface gráfica desenvolvida em *Python3* é proposta conforme ilustra a Figura 13. A aplicação encapsula todo o pré-processamento do *dataset* que é gerado pelo *CICFlowMeter* descrito na seção anterior.

Através dessa interface o usuário pode executar *testbenchs* escolhendo um dos tipos de ataques: Força Bruta, DOS, Ataque WEB, Infiltração, *Botnet* e *DDOS*. Mais detalhes desses ataques estão disponíveis no Anexo D. É possível aplicar um dos seguintes modelos de aprendizagem: *Multi Layer Perceptron* (MLP); *Random Forest* (RF); *Naive*

Bayes (NB); *k-Nearest Neighbors* (KNN) e *AI/IDS Transforer*. Logo abaixo, estão as opções de carregar o *dataset* gerado pelo *CICFlowMeter* e carregamento do modelo criado pelo próprio usuário. Todo o fluxo de execução é detalhado no Anexo B, onde é possível visualizar de forma gráfica cada etapa do processo.

Figura 13 – Interface gráfica desenvolvida em *Python*.



Fonte – Autor.

Em nossa proposta a avaliação dos modelos pode ser realizada de duas formas: através da "Tabela de Resultados" na própria interface ou com o auxílio da função "Exportar Resultados".

No campo Tabela de Resultados as métricas são calculadas e exibidas após execução da aplicação. E caso deseje exportar os resultados para posterior avaliação, a função Exportar Resultados cria uma pasta contendo um arquivo no formato (.txt) com o cálculo das métricas e gera uma matriz de confusão para fornecer uma maneira visual de exibir o desempenho detalhado do modelo testado, mostrando o número total de amostras classificadas corretamente e incorretamente de acordo com tipo de tráfego de rede.

7 Conclusão

A adoção de um modelo completo de Revisão Sistemática oferece a vantagem de eliminar artigos fora de contexto que tenham passado inadvertidamente pela fase de triagem. Esses artigos podem ser descartados nas etapas subsequentes antes da avaliação de qualidade. Isso destaca a importância de um processo de revisão sistemática da literatura bem estruturado e abrangente.

Este estudo se concentrou em realizar uma revisão sistemática da literatura sobre sistemas de detecção de intrusão, com o objetivo de identificar os algoritmos de IA e os conjuntos de dados utilizados nos últimos quatro anos. Observou-se que o algoritmo mais prevalente foi o CNN, e apesar da antiguidade, o conjunto de dados mais utilizado foi o NSL-KDD uma evolução do KDD Cup 99. Também foi evidenciada a presença significativa de novos modelos híbridos, que combinam várias técnicas de IA, e a introdução de um novo conjunto de dados voltado para SDN.

Outro aspecto explorado foi a criação do AI/IDS, um *framework* híbrido que utiliza *Transformer* em redes emuladas com o auxílio do software GNS3. O modelo foi treinado e avaliado com o conjunto de dados CSE-CICIDS2018 frequentemente utilizado na literatura, alcançando resultados satisfatórios e corroborando as direções identificadas na revisão sistemática realizada.

Apesar da abundância de pesquisas sobre o tema, a evolução contínua dos ataques demanda dos pesquisadores um esforço mais significativo para aprimorar as técnicas empregadas na construção de IDS mais sofisticados e modernos. Ao mesmo tempo, é crucial a criação de novos conjuntos de dados que abordem as nuances dos novos tipos de ataques, uma vez que esse desafio ainda carece de uma solução definitiva. A aplicação de um IDS em tempo real realmente seguro permanece como um objetivo futuro a ser alcançado.

Referências

- ABOBAKER, I. Hybrid feature selection (rfemi) techniques and intrusion detection systems for web attacks detection using supervised machine learning algorithms. In: *2023 10th International Conference on Future Internet of Things and Cloud (FiCloud)*. [S.l.: s.n.], 2023. p. 78–86. Citado 2 vezes nas páginas 41 e 81.
- ACHARYA, T.; ANNAMALAI, A.; CHOUIKHA, M. F. Efficacy of cnn-bidirectional lstm hybrid model for network-based anomaly detection. In: *2023 IEEE 13th Symposium on Computer Applications Industrial Electronics (ISCAIE)*. [S.l.: s.n.], 2023. p. 348–353. Citado na página 81.
- AKUTHOTA, U. C.; BHARGAVA, L. Evaluation of machine learning models for intrusion detection with the unsw-nb15 dataset. In: *2023 IEEE Silchar Subsection Conference (SILCON)*. [S.l.: s.n.], 2023. p. 1–5. Citado na página 81.
- ALHABSHY, A. A.; HAMEED, B. I.; ELDAHSHAN, K. A. An ameliorated multiattack network anomaly detection in distributed big data system-based enhanced stacking multiple binary classifiers. *IEEE Access*, v. 10, p. 52724–52743, 2022. Citado 2 vezes nas páginas 41 e 81.
- ALTUNAY, H. C.; ALBAYRAK, Z. A hybrid cnn+lstm-based intrusion detection system for industrial iot networks. *Engineering Science and Technology, an International Journal*, v. 38, p. 101322, 2023. ISSN 2215-0986. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2215098622002312>>. Citado na página 35.
- AMALAPURAM, S. K. et al. Continual learning for anomaly based network intrusion detection. In: *2022 14th International Conference on COMmunication Systems NETworkS (COMSNETS)*. [S.l.: s.n.], 2022. p. 497–505. Citado 3 vezes nas páginas 52, 55 e 81.
- AMAZON. *AWS TabTransformer*. 2024. Disponível em: <<https://docs.aws.amazon.com/sagemaker/latest/dg/tabtransformer.html>>. Citado na página 47.
- AMBATI, S. B.; VIDYARTHI, D. A brief study and comparison of, open source intrusion detection system tools. *International Journal of Advanced Computational Engineering and Networking*, v. 1, December 2013. ISSN 2320-2106. Citado na página 10.
- ANDERSON, J. *Computer Security Threat Monitoring and Surveillance*. Fort Washington, PA: James P Anderson Co, 1980. Citado na página 29.
- ARROYO-FIGUEROA, G.; SUCAR, L.; VILLAVICENCIO, A. Fuzzy intelligent system for the operation of fossil power plants. *Engineering Applications of Artificial Intelligence - ENG APPL ARTIF INTELL*, v. 13, p. 431–439, 08 2000. Citado na página 21.
- ASAAD, R. Penetration testing: Wireless network attacks method on kali linux os. *Academic Journal of Nawroz University*, v. 10, n. 1, p. 7–12, 2021. ISSN 2520-789X. Disponível em: <<https://journals.nawroz.edu.krd/index.php/ajnu/article/view/998>>. Citado na página 70.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 17799*: Tecnologia da informação - técnicas de segurança - código de prática para a gestão da segurança da informação. Rio de Janeiro, 2005. Citado na página 44.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 27002*: Tecnologia da informação - técnicas de segurança - código de prática para a gestão da segurança da informação. Rio de Janeiro, 2013. Citado na página 13.

AYAD, A. G.; SAKR, N. A.; HIKAL, N. A. A hybrid feature selection model for anomaly-based intrusion detection in iot networks. In: *2024 International Telecommunications Conference (ITC-Egypt)*. [S.l.: s.n.], 2024. p. 1–7. Citado na página 81.

BAYKARA, M.; ABDULRAHMAN, A.; ALAHMED, A. S. Classification of network data with machine learning methods for intelligent intrusion detection systems. In: *2022 4th International Conference on Advanced Science and Engineering (ICOASE)*. [S.l.: s.n.], 2022. p. 77–82. Citado na página 81.

BISHOP, C. M. *Pattern Recognition and Machine Learning*. [S.l.]: Springer, 2006. Citado na página 19.

BROWN, D. J.; SUCKOW, B.; WANG, T. A survey of intrusion detection systems. *Department of Computer Science, University of California, San Diego*, 2002. Citado na página 11.

Buschkes, R.; Kesdogan, D.; Reichl, P. How to increase security in mobile networks by anomaly detection. In: *Proceedings 14th Annual Computer Security Applications Conference (Cat. No.98EX217)*. [S.l.: s.n.], 1998. p. 3–12. Citado na página 16.

C, D. V. et al. Comparative analysis of deep learning and machine learning models for network intrusion detection. In: *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. [S.l.: s.n.], 2023. p. 1–13. Citado 2 vezes nas páginas 41 e 81.

Caswell, B.; Beale, J. *Snort 2.1 Intrusion Detection, Second Edition*. 1. ed. [S.l.]: Syngress, 2004. Citado 2 vezes nas páginas 17 e 18.

CERTBR. *CERTBR*. 2024. Disponível em: <<https://cartilha.cert.br/>>. Citado na página 13.

CHACKO, A. A. et al. Intrusion detection using machine learning techniques: An exhaustive review. In: *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*. [S.l.: s.n.], 2023. v. 1, p. 1586–1589. Citado na página 29.

CHEN, Z. et al. Machine learning-enabled hybrid intrusion detection system with host data transformation and an advanced two-stage classifier. *Computer Networks*, v. 250, p. 110576, 2024. ISSN 1389-1286. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1389128624004080>>. Citado 4 vezes nas páginas 41, 54, 55 e 81.

CHKIRBENE, Z. et al. Tidcs: A dynamic intrusion detection and classification system based feature selection. *IEEE Access*, v. 8, p. 95864–95877, 2020. Citado na página 14.

CHOUDHARY, S.; KESSWANI, N. Analysis of kdd-cup'99, nsl-kdd and unsw-nb15 datasets using deep learning in iot. *Procedia Computer Science*, v. 167, p. 1561–1573, 2020. ISSN 1877-0509. International Conference on Computational Intelligence and Data Science. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1877050920308334>>. Citado 2 vezes nas páginas 26 e 27.

CISCO. *Relatório de Tendências Globais de Rede de 2022*. 2022. Disponível em: <https://www.cisco.com/c/dam/global/pt_br/solutions/enterprise-networks/nb-06-2022-networking-report-cte.pdf>. Citado na página 10.

CSE-CIC. *CICFlowMeter*. 2018. Disponível em: <<https://www.unb.ca/cic/research/applications.htmlCICFlowMeter>>. Citado na página 73.

CUP, K. *KDD Cup 1999 Tasks*. 1999. Disponível em: <<https://kdd.org/kdd-cup/view/kdd-cup-1999/Tasks>>. Citado na página 26.

CYBERSECURITY, C. I. *NSL-KDD Dataset*. 2009. Disponível em: <<https://www.unb.ca/cic/datasets/nsl.html>>. Citado na página 38.

CYBERSECURITY, C. I. *CSE-CIC-IDS 2017*. 2017. Disponível em: <<https://www.unb.ca/cic/datasets/ids-2017.html>>. Citado na página 38.

CYBERSECURITY, C. I. *CSE-CIC-IDS 2018*. 2018. Disponível em: <<https://www.unb.ca/cic/datasets/ids-2018.html>>. Citado na página 38.

CYBERSECURITY, C. I. *Datasets*. 2023. Disponível em: <<https://www.unb.ca/cic/datasets/index.html>>. Citado na página 37.

CYBERSECURITY, C. I. for. *CSE-CIC-IDS2018*. 2018. Disponível em: <<https://www.unb.ca/cic/datasets/ids-2018.html>>. Citado 2 vezes nas páginas 25 e 82.

DAS, S. et al. Network intrusion detection and comparative analysis using ensemble machine learning and feature selection. *IEEE Transactions on Network and Service Management*, v. 19, n. 4, p. 4821–4833, 2022. Citado 2 vezes nas páginas 41 e 81.

DESAI, R.; VENKATESH, T. Robust network intrusion detection systems for outlier detection. In: *2022 IEEE 27th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. [S.l.: s.n.], 2022. p. 140–146. Citado na página 81.

DISHA, R. A.; WAHEED, S. A comparative study of machine learning models for network intrusion detection system using unsw-nb 15 dataset. In: *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*. [S.l.: s.n.], 2021. p. 1–5. Citado na página 81.

ELSAYED, M. S.; LE-KHAC, N.-A.; JURCUT, A. D. Insdn: A novel sdn intrusion dataset. *IEEE Access*, v. 8, p. 165263–165284, 2020. Citado 2 vezes nas páginas 38 e 39.

ELSAYED, S.; MOHAMED, K.; MADKOUR, M. A. A comparative study of using deep learning algorithms in network intrusion detection. *IEEE Access*, v. 12, p. 58851–58870, 2024. Citado na página 12.

- ELSAIED, S.; MOHAMED, K.; MADKOUR, M. A. A comparative study of using deep learning algorithms in network intrusion detection. *IEEE Access*, v. 12, p. 58851–58870, 2024. Citado 2 vezes nas páginas 41 e 81.
- FAYYAD, U. M. et al. (Ed.). *Advances in Knowledge Discovery and Data Mining*. USA: American Association for Artificial Intelligence, 1996. ISBN 0262560976. Citado 2 vezes nas páginas 22 e 24.
- GAO, J. et al. Omni scada intrusion detection using deep learning algorithms. *IEEE Internet of Things Journal*, v. 8, n. 2, p. 951–961, 2021. Citado na página 14.
- GNS3. *GNS3*. 2021. Disponível em: <<https://www.gns3.com/>>. Citado na página 70.
- GTA-UFRJ. *Sistemas de Detecção de Intrusão*. 2016. Disponível em: <https://www.gta.ufrj.br/grad/16_2/2016IDS/index.html>. Citado 3 vezes nas páginas 17, 18 e 19.
- GUPTA, S. K.; TRIPATHI, M.; GROVER, J. Towards an effective intrusion detection system using machine learning techniques: Comprehensive analysis and review. In: *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. [S.l.: s.n.], 2021. p. 1–6. Citado na página 29.
- HAI, T. H.; NAM, L. H. A practical comparison of deep learning methods for network intrusion detection. In: *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*. [S.l.: s.n.], 2021. p. 1–6. Citado 3 vezes nas páginas 52, 55 e 81.
- HALBOUNI, A. et al. Cnn-lstm: Hybrid deep neural network for network intrusion detection system. *IEEE Access*, v. 10, p. 99837–99849, 2022. Citado 3 vezes nas páginas 37, 41 e 81.
- HASSAN, H. A. et al. An efficient attack detection framework in software-defined networking using intelligent techniques. In: *2023 3rd International Conference on Electronic Engineering (ICEEM)*. [S.l.: s.n.], 2023. p. 1–6. Citado na página 81.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The elements of statistical learning: data mining, inference and prediction*. 2. ed. Springer, 2009. Disponível em: <<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>>. Citado na página 19.
- HINDY, H. et al. *A Taxonomy and Survey of Intrusion Detection System Design Techniques, Network Threats and Datasets*. In: . [S.l.]: *Journal IEEE*, 2018. v. 1. Citado na página 13.
- HNAMTE, V. et al. A novel two-stage deep learning model for network intrusion detection: Lstm-ae. *IEEE Access*, v. 11, p. 37131–37148, 2023. Citado 4 vezes nas páginas 41, 53, 55 e 81.
- Hsieh, K. . R.; Chen, W. . T. A neural network model which combines unsupervised and supervised learning. *IEEE Transactions on Neural Networks*, v. 4, n. 2, p. 357–360, 1993. Citado 2 vezes nas páginas 19 e 20.
- HU, X. et al. An improved algorithm for network intrusion detection based on deep residual networks. *IEEE Access*, v. 12, p. 66432–66441, 2024. Citado 4 vezes nas páginas 41, 54, 55 e 81.

HUANG, X. et al. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020. Available at: <<https://arxiv.org/abs/2012.06678>>. Citado na página 47.

HUANG, X.; LIN, J. J.; DEMNER-FUSHMAN, D. Evaluation of pico as a knowledge representation for clinical questions. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, p. 359–63, 2006. Disponível em: <<https://api.semanticscholar.org/CorpusID:8627913>>. Citado na página 31.

IBM. *An introduction to deep learning*. 2020. Disponível em: <<https://developer.ibm.com/technologies/artificial-intelligence/articles/an-introduction-to-deep-learning/>>. Citado na página 20.

JIN, S.; CHUNG, J.-G.; XU, Y. Signature-based intrusion detection system (ids) for in-vehicle can bus network. In: *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. [S.l.: s.n.], 2021. p. 1–5. Citado na página 35.

KALI. *Kali Linux*. 2021. Disponível em: <<https://www.kali.org/>>. Citado 2 vezes nas páginas 70 e 72.

KANDHRO, I. A. et al. Detection of real-time malicious intrusions and attacks in iot empowered cybersecurity infrastructures. *IEEE Access*, v. 11, p. 9136–9148, 2023. Citado 2 vezes nas páginas 41 e 81.

KANUMALLI, S. S. et al. A scalable network intrusion detection system using bi-lstm and cnn. In: *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*. [S.l.: s.n.], 2023. p. 1–6. Citado na página 81.

KARNA, I. et al. Ensemble-based filter feature selection technique for building flow-based ids. In: *2021 2nd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS)*. [S.l.: s.n.], 2021. p. 324–328. Citado na página 81.

KDD. *Knowledge discovery in data and data-mining*. 1999. Disponível em: <<https://kdd.org/kdd-cup/view/kdd-cup-1999/Data>>. Citado na página 38.

KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering. v. 2, 01 2007. Citado 2 vezes nas páginas 30 e 33.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. v. 25. Disponível em: <<https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>>. Citado 2 vezes nas páginas 19 e 20.

KSHIRSAGAR, D.; KUMAR, S. An efficient feature reduction method for the detection of dos attack. *ICT Express*, v. 7, n. 3, p. 371–375, 2021. ISSN 2405-9595. Citado na página 81.

KUMAR, C. et al. Nature-inspired intrusion detection system for protecting software-defined networks controller. *Computers Security*, v. 134, p. 103438, 2023. ISSN 0167-4048. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167404823003486>>. Citado na página 39.

- KUMAR, N. et al. A smart nids design for sdn-based cloud iot network using optimal path selection and rnn. In: *2023 9th International Conference on Signal Processing and Communication (ICSC)*. [S.l.: s.n.], 2023. p. 33–41. Citado na página 81.
- KUMAR, P.; BANERJEE, I. Attack and anomaly detection in iiot networks using machine learning techniques. In: *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. [S.l.: s.n.], 2023. p. 1–7. Citado na página 81.
- KYE, H.; KIM, M.; KWON, M. Hierarchical detection of network anomalies : A self-supervised learning approach. *IEEE Signal Processing Letters*, v. 29, p. 1908–1912, 2022. Citado na página 81.
- LAUREANO, M. A. P. *Segurança da Informação*. 1. ed. [S.l.]: Editora LT, 2012. Citado na página 13.
- LAZAREVIC, A.; KUMAR, V.; SRIVASTAVA, J. Intrusion detection: A survey. In: _____. *Managing Cyber Threats: Issues, Approaches, and Challenges*. Boston, MA: Springer US, 2005. p. 19–78. ISBN 978-0-387-24230-9. Disponível em: <https://doi.org/10.1007/0-387-24230-9_2>. Citado na página 16.
- LEON, M.; MARKOVIC, T.; PUNNEKKAT, S. Comparative evaluation of machine learning algorithms for network intrusion detection and attack classification. In: *2022 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2022. p. 01–08. Citado 2 vezes nas páginas 10 e 81.
- LI, J. et al. *Machine Learning Algorithms for Network Intrusion Detection*. In: . [S.l.]: Springer International Publishing, 2019. p. 151–179. Citado na página 12.
- LIANG, X.; XING, H.; HOU, T. Network intrusion detection method based on cgan and cnn-bilstm. In: *2023 IEEE 16th International Conference on Electronic Measurement Instruments (ICEMI)*. [S.l.: s.n.], 2023. p. 396–400. Citado na página 81.
- LIFANDALI, O.; ABGHOOR, N. Deep learning methods applied to intrusion detection: Survey, taxonomy and challenges. In: *2021 International Conference on Decision Aid Sciences and Application (DASA)*. [S.l.: s.n.], 2021. p. 1035–1044. Citado na página 29.
- LIU, C.; GU, Z.; WANG, J. A hybrid intrusion detection system based on scalable k-means+ random forest and deep learning. *IEEE Access*, v. 9, p. 75729–75740, 2021. Citado 3 vezes nas páginas 37, 41 e 81.
- LIU, H.; LANG, B. *Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey*. In: . [S.l.]: Journal Applied Sciences, 2019. v. 9. Citado na página 12.
- LOPEZ, M. E. A. *Uma Arquitetura de Detecção e Prevenção de Intrusão para Redes Definidas por Software*. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, mai. 2014. Citado na página 13.
- MAGÁN-CARRIÓN, R. et al. Improving the reliability of network intrusion detection systems through dataset integration. *IEEE Transactions on Emerging Topics in Computing*, v. 10, n. 4, p. 1717–1732, 2022. Citado na página 81.

- MAHFOUZ, A. et al. Ensemble classifiers for network intrusion detection using a novel network attack dataset. *Future Internet*, v. 12, n. 11, 2020. ISSN 1999-5903. Disponível em: <<https://www.mdpi.com/1999-5903/12/11/180>>. Citado na página 12.
- MAHMOUD, M. et al. Ae-lstm: Autoencoder with lstm-based intrusion detection in iot. In: *2022 International Telecommunications Conference (ITC-Egypt)*. [S.l.: s.n.], 2022. p. 1–6. Citado na página 81.
- MAHROUQI, A. et al. Simulating SQL-injection cyber-attacks using GNS3. *International Journal of Computer Theory and Engineering*, IACSIT Press, v. 8, n. 3, p. 213–217, jun 2016. Disponível em: <<https://doi.org/10.7763/2Fijcte.2016.v8.1046>>. Citado na página 70.
- MALIK, M.; SAINI, K. S. Network intrusion detection system using reinforcement learning techniques. In: *2023 International Conference on Circuit Power and Computing Technologies (ICCPCT)*. [S.l.: s.n.], 2023. p. 1642–1649. Citado na página 81.
- MARIR, N.; WANG, H.; FENG, G. Unsupervised feature learning with distributed stacked denoising sparse autoencoder for abnormal behavior detection using apache spark. In: *2019 IEEE 2nd International Conference on Knowledge Innovation and Invention (ICKII)*. [S.l.: s.n.], 2019. p. 473–476. Citado na página 11.
- Marir, N.; Wang, H.; Feng, G. Unsupervised feature learning with distributed stacked denoising sparse autoencoder for abnormal behavior detection using apache spark. In: *2019 IEEE 2nd International Conference on Knowledge Innovation and Invention (ICKII)*. [S.l.: s.n.], 2019. p. 473–476. Citado na página 12.
- MAZIGHI, A.; BALLIHI, L.; ORHANOU, G. Improving intrusion detection using machine learning algorithms with feature selection based on information gain. In: *2023 IEEE International Conference on Advances in Data-Driven Analytics And Intelligent Systems (ADACIS)*. [S.l.: s.n.], 2023. p. 1–6. Citado na página 81.
- MCAFEE. *Report Cyber Crime*. 2020. Disponível em: <<https://www.mcafee.com/enterprise/en-us/assets/reportsrp-hidden-costs-of-cybercrime.pdf>>. Citado na página 10.
- McHugh, J.; Christie, A.; Allen, J. Defending yourself: The role of intrusion detection systems. *IEEE Software*, v. 17, n. 5, p. 42–51, 2000. Citado na página 16.
- MEHRA, P. A brief study and comparison of snort and bro: Open source network intrusion detection systems. *International Journal of Advanced Research in Computer and Communication Engineering*, v. 1, n. 6, p. 383–386, August 2012. Citado na página 10.
- MEZINA, A.; BURGET, R.; TRAVIESO-GONZÁLEZ, C. M. Network anomaly detection with temporal convolutional network and u-net model. *IEEE Access*, v. 9, p. 143608–143622, 2021. Citado 4 vezes nas páginas 38, 53, 55 e 81.
- Mishra, P. et al. *A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection*. In: . [S.l.]: *IEEE COMMUNICATIONS SURVEYS AND TUTORIALS*, 2019. v. 21. Citado na página 11.
- MODI, C. et al. A survey of intrusion detection techniques in cloud. *Journal of Network and Computer Applications*, v. 36, n. 1, p. 42–57, 2013. ISSN 1084-8045. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1084804512001178>>. Citado na página 11.

- MOUSTAFA, N.; SLAY, J. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: *2015 Military Communications and Information Systems Conference (MilCIS)*. [S.l.: s.n.], 2015. p. 1–6. Citado na página 28.
- NAKAMURA, P. L. d. G. E. T. *Segurança de Redes em Ambientes Cooperativos*. 1. ed. [S.l.]: NOVATEC, 2007. Citado na página 16.
- NASSIF, A. B. et al. Machine learning for anomaly detection: A systematic review. *IEEE Access*, v. 9, p. 78658–78700, 2021. Citado na página 29.
- OKTIVASARI, P. et al. Analysis of effectiveness of iptables on web server from slowloris attack. In: *2022 5th International Conference of Computer and Informatics Engineering (IC2IE)*. [S.l.: s.n.], 2022. p. 215–219. Citado na página 55.
- OZKAN-OKAY, M. et al. A comprehensive systematic literature review on intrusion detection systems. *IEEE Access*, v. 9, p. 157727–157760, 2021. Citado na página 12.
- PANUNTUN, L. H. et al. Comparative analysis of machine learning models for data traffic anomaly detection systems in intrusion detection systems. In: *2023 International Conference on Information Technology and Computing (ICITCOM)*. [S.l.: s.n.], 2023. p. 301–306. Citado na página 81.
- PROCTOR, P. E. *Practical Intrusion Detection Handbook*. 1st. ed. USA: Prentice Hall PTR, 2000. ISBN 0130259608. Citado na página 17.
- PTACEK, T. N. N. T. H. Insertion, evasion, and denial of service: Eluding network intrusion detection. p. 63, 1998. Disponível em: <https://www.cs.unc.edu/~fabian/course_papers/PtacekNewsham98.pdf>. Citado na página 16.
- RAMANA, K. et al. Wogru-ids “an intelligent intrusion detection system for iot assisted wireless sensor networks. *Computer Communications*, v. 196, p. 195–206, 2022. ISSN 0140-3664. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0140366422003796>>. Citado na página 37.
- RANJIT, P.; SAMARJEET, B. *A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems*. In: . [S.l.]: *International Journal of Engineering e Technology*, 2018. v. 7, p. 479–482. Citado na página 12.
- RATTI, R.; SINGH, S. R.; NANDI, S. Network based intrusion detection using time aware lstm autoencoder. In: *2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. [S.l.: s.n.], 2023. p. 2570–2578. Citado 3 vezes nas páginas 53, 55 e 81.
- RAYYAN. *Faster Systematic Literature Reviews*. 2024. Disponível em: <<https://www.rayyan.ai/>>. Citado na página 33.
- RING, M. et al. *A survey of network-based intrusion detection datasets*. In: . [S.l.]: *Computers e Security*, 2019. v. 86, p. 147–167. Citado na página 12.
- SAH, G.; SINGH, S.; BANERJEE, S. Intrusion detection system using classification algorithms with feature selection mechanism over real-time data traffic. *China Communications*, v. 21, n. 9, p. 292–320, 2024. Citado 2 vezes nas páginas 41 e 81.

- SAID, R. B.; ASKERZADE, I. Attention-based cnn-bilstm deep learning approach for network intrusion detection system in software defined networks. In: *2023 5th International Conference on Problems of Cybernetics and Informatics (PCI)*. [S.l.: s.n.], 2023. p. 1–5. Citado na página 81.
- SAID, R. B.; SABIR, Z.; ASKERZADE, I. Cnn-bilstm: A hybrid deep learning approach for network intrusion detection system in software-defined networking with hybrid feature selection. *IEEE Access*, v. 11, p. 138732–138747, 2023. Citado 2 vezes nas páginas 41 e 81.
- SALLAM, Y. F. et al. Implementation of network attack detection using convolutional neural network. In: *2021 International Conference on Electronic Engineering (ICEEM)*. [S.l.: s.n.], 2021. p. 1–6. Citado na página 81.
- SANNIGRAHI, M.; THANDEESWARAN, R. Predictive analysis of network-based attacks by hybrid machine learning algorithms utilizing bayesian optimization, logistic regression, and random forest algorithm. *IEEE Access*, v. 12, p. 142721–142732, 2024. Citado 2 vezes nas páginas 41 e 81.
- SCAPY. *Scapy*. 2021. Disponível em: <<https://scapy.readthedocs.io/en/latest/index.html>>. Citado 2 vezes nas páginas 71 e 72.
- SELVARAJAN, P. et al. Networks intrusion detection using optimized hybrid network. In: *2023 International Conference on Smart Computing and Application (ICSCA)*. [S.l.: s.n.], 2023. p. 1–6. Citado na página 81.
- SENAI, I. de inovação T. *Aprendizagem supervisionada e não supervisionada*. 2018. Disponível em: <<https://isitics.com/2018/08/28/aprendizagem-supervisionada-ou-nao-supervisionada/>>. Citado 2 vezes nas páginas 19 e 20.
- SHARAFALDIN, A. L. I.; GHORBANI, A. *Toward generating a new intrusion detection dataset and intrusion traffic characterization*. In: . [S.l.]: In Proc. of the 4th International Conference on Information Systems Security and Privacy, 2018. p. 108–116. Citado na página 49.
- SHARAFALDIN, I.; LASHKARI, A. H.; GHORBANI, A. A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*. Portugal: [s.n.], 2018. Citado na página 24.
- SILVA, A. C. B. D. *Sistema de Detecção de Intrusão baseado em Métodos Estatísticos para Análise de Comportamento*. Tese (Doutorado) — UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL, <https://www.lume.ufrgs.br/bitstream/handle/10183/4112/000407656.pdf?sequence=1>, 6 2003. An optional note. Citado na página 16.
- SILVA, C. L. S. *Abordagens para o problema do desbalanceamento em detecção de intrusão: um estudo de caso aplicando CIC-IDS2018*. Dissertação (Mestrado), 2024. Centro de Engenharias e Ciências Exatas. Disponível em: <<https://tede.unioeste.br/handle/tede/7258>>. Citado na página 51.

- STEINMACHER, I.; CHAVES, A. P.; GEROSA, M. A. Awareness support in distributed software development: A systematic review and mapping of the literature. *Computer Supported Cooperative Work (CSCW)*, Springer, v. 22, n. 2-3, p. 113–158, 2013. Citado na página 33.
- SURAKHI, O. M. et al. A comprehensive survey for machine learning and deep learning applications for detecting intrusion detection. In: *2021 22nd International Arab Conference on Information Technology (ACIT)*. [S.l.: s.n.], 2021. p. 1–13. Citado 2 vezes nas páginas 10 e 29.
- SUTHISHNI, D. N. P.; KUMAR, K. S. S. A review on machine learning based security approaches in intrusion detection system. In: *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*. [S.l.: s.n.], 2022. p. 341–348. Citado na página 30.
- TANENBAUM, A. S. *Redes de Computadores*. 4. ed. [S.l.]: Elieser Campus, 2003. Citado na página 44.
- TAVALLAEE, M. et al. A detailed analysis of the kdd cup 99 data set. In: *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. [S.l.: s.n.], 2009. p. 1–6. Citado na página 27.
- TAYYEBI, Y.; BHILARE, D. A comparative study of open source network based intrusion detection systems. *International Journal of Computer Science and Information Technologies (IJCSIT)*, v. 9, n. 2, p. 23–26, 2018. Citado na página 11.
- THAKKAR, A.; LOHIYA, R. Attack classification of imbalanced intrusion data for iot network using ensemble-learning-based deep neural network. *IEEE Internet of Things Journal*, v. 10, n. 13, p. 11888–11895, 2023. Citado 2 vezes nas páginas 41 e 81.
- TOSSOU, S.; QORIB, M.; KACEM, T. Anomaly based intrusion detection system: A deep learning approach. In: *2023 International Symposium on Networks, Computers and Communications (ISNCC)*. [S.l.: s.n.], 2023. p. 1–6. Citado na página 81.
- VALENTIM, P.; BRAGA, R.; OLIVEIRA, A. Um testbed para o ensino de abordagens de ids baseadas em ia em redes emuladas. In: *Anais do VIII Congresso sobre Tecnologias na Educação*. Porto Alegre, RS, Brasil: SBC, 2023. p. 292–301. Disponível em: <<https://sol.sbc.org.br/index.php/ctrl/article/view/25808>>. Citado na página 15.
- VASWANI, A. et al. *Attention Is All You Need*. 2023. Disponível em: <<https://arxiv.org/abs/1706.03762>>. Citado 2 vezes nas páginas 21 e 23.
- VINOD, D.; PRASAD, M. A novel hybrid automatic intrusion detection system using machine learning technique for anomalous detection based on traffic prediction. In: *2023 International Conference on Networking and Communications (ICNWC)*. [S.l.: s.n.], 2023. p. 1–7. Citado na página 11.
- VISHWAKARMA, M.; KESSWANI, N. Dids: A deep neural network based real-time intrusion detection system for iot. *Decision Analytics Journal*, v. 5, p. 100142, 2022. ISSN 2772-6622. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S277266222200073X>>. Citado na página 37.

- WALES, U. N. S. *UNSW-NB15 Dataset*. 2015. Disponível em: <<https://research.unsw.edu.au/projects/unsw-nb15-dataset>>. Citado 2 vezes nas páginas 28 e 38.
- WANG, X. et al. Advanced network intrusion detection with tabtransformer. *Journal of Theory and Practice of Engineering Science*, v. 4, n. 03, Mar. 2024. Disponível em: <<https://centuryscipub.com/index.php/jtpes/article/view/527>>. Citado na página 47.
- WANG, Z. *Deep Learning-Based Intrusion Detection With Adversaries*. In: . [S.l.]: *SPECIAL SECTION ON CHALLENGES AND OPPORTUNITIES OF BIG DATA AGAINST CYBER CRIME*, 2018. Citado na página 49.
- WELSH, C. Gns3 network simulation guide. In: . [S.l.: s.n.], 2013. Citado na página 70.
- WESTER, P.; HEIDING, F.; LAGERSTRÖM, R. Anomaly-based intrusion detection using tree augmented naive bayes. In: *2021 IEEE 25th International Enterprise Distributed Object Computing Workshop (EDOCW)*. [S.l.: s.n.], 2021. p. 112–121. Citado na página 81.
- WISANW, T.; THAMMAW, M. A double-layered hybrid approach for network intrusion detection system using combined naive bayes and svm. *IEEE Access*, v. 9, p. 138432–138450, 2021. Citado 2 vezes nas páginas 41 e 81.
- XIE, B. et al. Hdcbn: Hybrid neural network for network intrusion detection system. In: *2024 9th International Conference on Computer and Communication Systems (ICCCS)*. [S.l.: s.n.], 2024. p. 427–434. Citado na página 81.
- XU, W. et al. Improving performance of autoencoder-based network anomaly detection on nsl-kdd dataset. *IEEE Access*, v. 9, p. 140136–140146, 2021. Citado 2 vezes nas páginas 41 e 81.
- YEOM, D.; NIKNAMI, N.; WU, J. Evaluating performance of intrusion detection systems under different configurations in sdn. In: *Proceedings of the Twenty-Fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*. New York, NY, USA: Association for Computing Machinery, 2023. (MobiHoc '23), p. 418–423. ISBN 9781450399265. Disponível em: <<https://doi.org/10.1145/3565287.3617614>>. Citado na página 35.
- YOO, J. et al. Study on network intrusion detection method using discrete pre-processing method and convolution neural network. *IEEE Access*, v. 9, p. 142348–142361, 2021. Citado 4 vezes nas páginas 41, 54, 55 e 81.
- ZHANG, Y. et al. Improving the classification effectiveness of network intrusion detection using ensemble machine learning techniques and deep neural networks. In: *2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*. [S.l.: s.n.], 2022. p. 117–123. Citado na página 81.

APÊNDICE A – Ferramentas de Software

A.1 Emulador GNS3

GNS3 é um emulador de rede gráfico de código aberto usado extensivamente por profissionais de rede para simular dispositivos de rede. É ativamente desenvolvido e apoiado, possui uma comunidade crescente de mais de 800.000 membros, onde foi baixado mais de 10 milhões de vezes. Jeremy Grossman, o desenvolvedor original do GNS3, criou originalmente o software para ajudá-lo a estudar para suas certificações CCNP. O GNS3 permitiu que os engenheiros de rede virtualizassem dispositivos de hardware reais por mais de 10 anos. Originalmente apenas emulando dispositivos Cisco usando um software chamado *Dynamips*, ele evoluiu e oferece suporte a muitos dispositivos de vários fornecedores de rede, incluindo *switches* virtuais *Cisco*, *Cisco ASAs*, *Brocade vRouters*, *switches Cumulus Linux*, instâncias *Docker*, *HPE VSRs*, vários dispositivos *Linux* e muitos outros (GNS3, 2021; WELSH, 2013).

Segundo (MAHROUQI et al., 2016) é uma ferramenta de emulação de rede permite que usuários finais e profissionais emulem redes complexas com baixo custo e consumindo menos tempo. Ele nos permite conectar a máquinas virtuais do *Virtual Box* que são usadas para emular diferentes sistemas operacionais, por exemplo, *Linux* e *Microsoft Windows*.

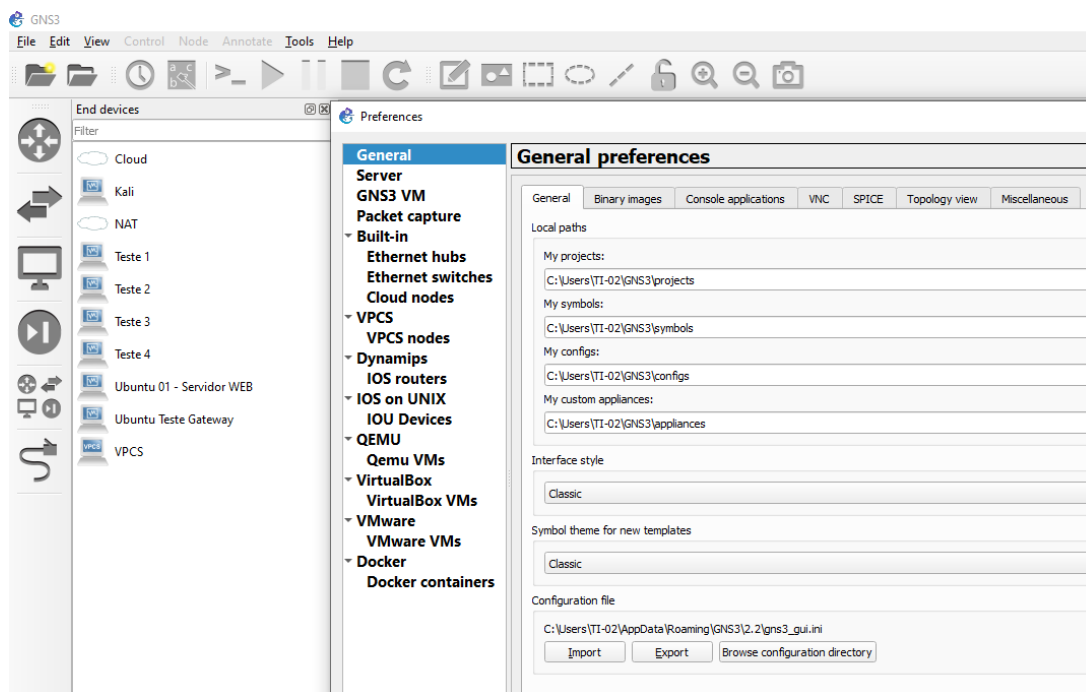
A Figura 14 mostra a tela de configurações do emulador de rede GNS3. Onde é possível adicionar diversos tipos de *hardwares*, configurar a máquina virtual GNS3 e realizar configurações gerais do emulador.

A.2 Kali Linux

O sistema Kali Linux foi desenvolvido por Mati Aharoni e Devon Kearns da *Offensive Security* e era principalmente adequado para testes forenses digitais e de penetração sob *Ethical Hacking*. Tem aproximadamente 300 ou mais ferramentas de hacking que são amplamente categorizadas em coleta de informações, análise de vulnerabilidade, ataques sem fio, aplicativo da web, ferramentas de exploração, ferramentas forenses, ferramentas de detecção e falsificação, ataques de senha, manutenção de acesso, engenharia reversa e ferramentas de *hacking* de *hardware*. Dentre estas, as ferramentas mais comumente utilizadas são o *framework Metasploit*, *dsniff*, *tcp dump*, *Nmap*, *Wireshark*, *Aircrack-ng*, *Armitage*, *Burp suit*, *BeEF* e assim por diante (KALI, 2021).

Segundo (ASAAD, 2021) é a plataforma de teste de penetração mais poderosa e popular do mundo, usada por profissionais de segurança em uma ampla gama de

Figura 14 – Tela de configurações do emulador de rede GNS3.



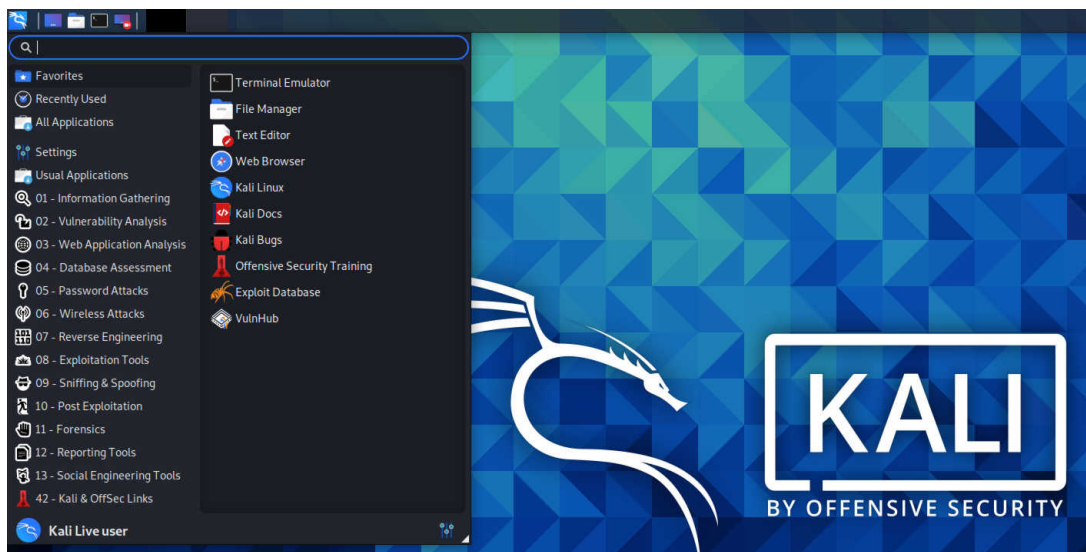
Fonte – Autor

especializações, incluindo teste de penetração, análise forense, engenharia reversa e avaliação de vulnerabilidade. É o resultado de uma evolução contínua da plataforma, de *WHoppiX* a *WHAX*, a *BackTrack* e agora a uma estrutura de teste de penetração completa aproveitando muitos recursos do *Debian GNU / Linux* e a vibrante comunidade de código aberto no mundo todo. O *Kali Linux* não foi construído para ser uma coleção simples de ferramentas, mas sim uma estrutura flexível que testadores de penetração profissionais, entusiastas de segurança, estudantes e amadores podem personalizar para atender às suas necessidades específicas.

A Figura 15 mostra a tela inicial do *Kali Linux* com seu menu de ferramentas. Onde é possível acessar diversas ferramentas que vem instaladas de forma simples e rápida.

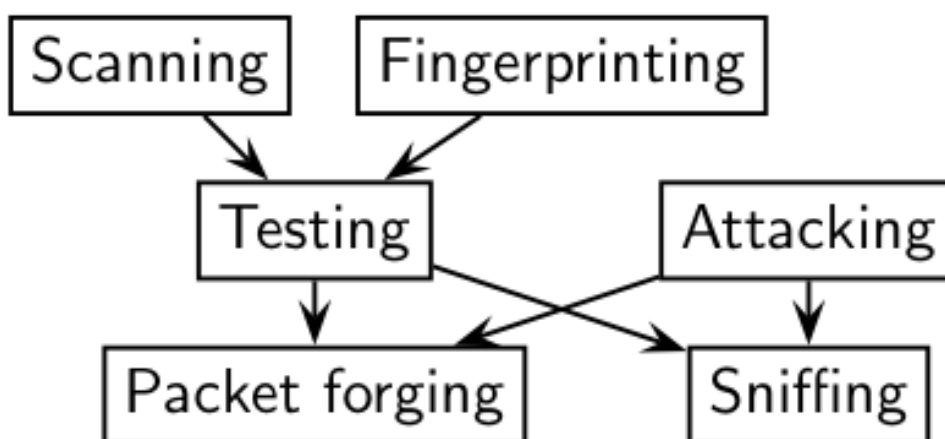
A.3 Scapy

Scapy é um programa *Python* que permite ao usuário enviar, farejar, dissecar e forjar pacotes de rede. Esta capacidade permite a construção de ferramentas que podem sondar, escanear ou atacar redes. Ele é capaz de forjar ou decodificar pacotes de um grande número de protocolos, enviá-los na rede, capturá-los, corresponder a solicitações e respostas e muito mais. Ele pode lidar facilmente com a maioria das tarefas clássicas, como varredura, rastreamento, sondagem, testes de unidade, ataques ou descoberta de rede. Ele pode substituir *hping*, *arp spoof*, *arp-sk*, *arping*, *p0f* e até mesmo algumas partes do *Nmap*, *tcpdump* e *tshark* (SCAPY, 2021).

Figura 15 – Ferramenta *Kali Linux*.

Fonte – Adaptado de (KALI, 2021)

O *Scapy* também executa muito bem em muitas outras tarefas específicas que a maioria das outras ferramentas não consegue lidar, como enviar *frames* inválidos, injetar seus próprios *frames 802.11*, combinar técnicas (salto de VLAN + envenenamento de *cache ARP*, decodificação VOIP em canal criptografado WEP, ...), etc. Além disso, podem ser construídas funções de alto nível, por exemplo, uma que faça traceroutes e forneça como resultado apenas o TTL inicial da solicitação e o IP de origem da resposta. Um que efetua ping em uma rede inteira e fornece a lista de máquinas que atendem. A Figura 16 mostra o fluxo de criação de pacotes com o *Scapy*.

Figura 16 – Fluxo de criação de pacotes no *Scapy*.

Fonte – (SCAPY, 2021)

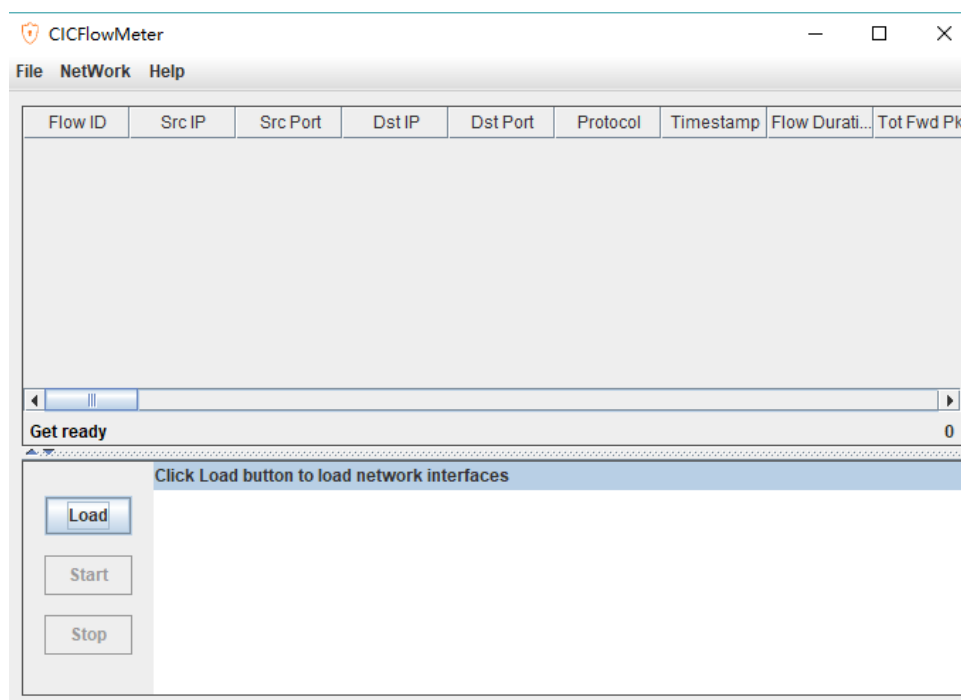
A.4 CICFlowMeter

CICFlowMeter é um gerador e analisador de fluxo de tráfego de rede. Ele pode ser usado para gerar fluxos bidirecionais, onde o primeiro pacote determina as direções para a frente (origem ao destino) e para trás (destino à origem), portanto, mais de 80 recursos de tráfego de rede estatísticos, como duração, número de pacotes, número de bytes, O comprimento dos pacotes, etc. pode ser calculado separadamente nas direções para frente e para trás (CSE-CIC, 2018).

As funcionalidades adicionais incluem a seleção de recursos da lista de recursos existentes, a adição de novos recursos e o controle da duração do tempo limite do fluxo. A saída do aplicativo é o arquivo no formato (.csv) que possui seis colunas rotuladas para cada fluxo (*FlowID*, *SourceIP*, *DestinationIP*, *SourcePort*, *DestinationPort* e Protocolo) com mais de 80 recursos de análise de tráfego de rede. Os fluxos TCP geralmente são encerrados após o término da conexão (pelo pacote FIN), enquanto os fluxos UDP são encerrados por um tempo limite de fluxo. O valor de tempo limite de fluxo pode ser atribuído arbitrariamente pelo esquema individual, por exemplo, 600 segundos para TCP e UDP.

Uma visão geral do *CICFlowMeter* é apresentada na Figura 17 e o detalhamento dos recursos extraídos é apresentado no Anexo A.

Figura 17 – Tela inicial do *CICFlowMeter*.



Fonte – Autor

Anexos

ANEXO A – Recursos CICFlowMeter

Feature Name	Description
fl_dur	Flow duration
tot_fw_pk	Total packets in the forward direction
tot_bw_pk	Total packets in the backward direction
tot_l_fw_pkt	Total size of packet in forward direction
fw_pkt_l_max	Maximum size of packet in forward direction
fw_pkt_l_min	Minimum size of packet in forward direction
fw_pkt_l_avg	Average size of packet in forward direction
fw_pkt_l_std	Standard deviation size of packet in forward direction
Bw_pkt_l_max	Maximum size of packet in backward direction
Bw_pkt_l_min	Minimum size of packet in backward direction
Bw_pkt_l_avg	Mean size of packet in backward direction
Bw_pkt_l_std	Standard deviation size of packet in backward direction
fl_byt_s	flow byte rate that is number of packets transferred per second
fl_pkt_s	flow packets rate that is number of packets transferred per second
fl_iat_avg	Average time between two flows
fl_iat_std	Standard deviation time two flows
fl_iat_max	Maximum time between two flows
fl_iat_min	Minimum time between two flows
fw_iat_tot	Total time between two packets sent in the forward direction
fw_iat_avg	Mean time between two packets sent in the forward direction
fw_iat_std	Standard deviation time between two packets sent in the forward direction
fw_iat_max	Maximum time between two packets sent in the forward direction
fw_iat_min	Minimum time between two packets sent in the forward direction
bw_iat_tot	Total time between two packets sent in the backward direction
bw_iat_avg	Mean time between two packets sent in the backward direction

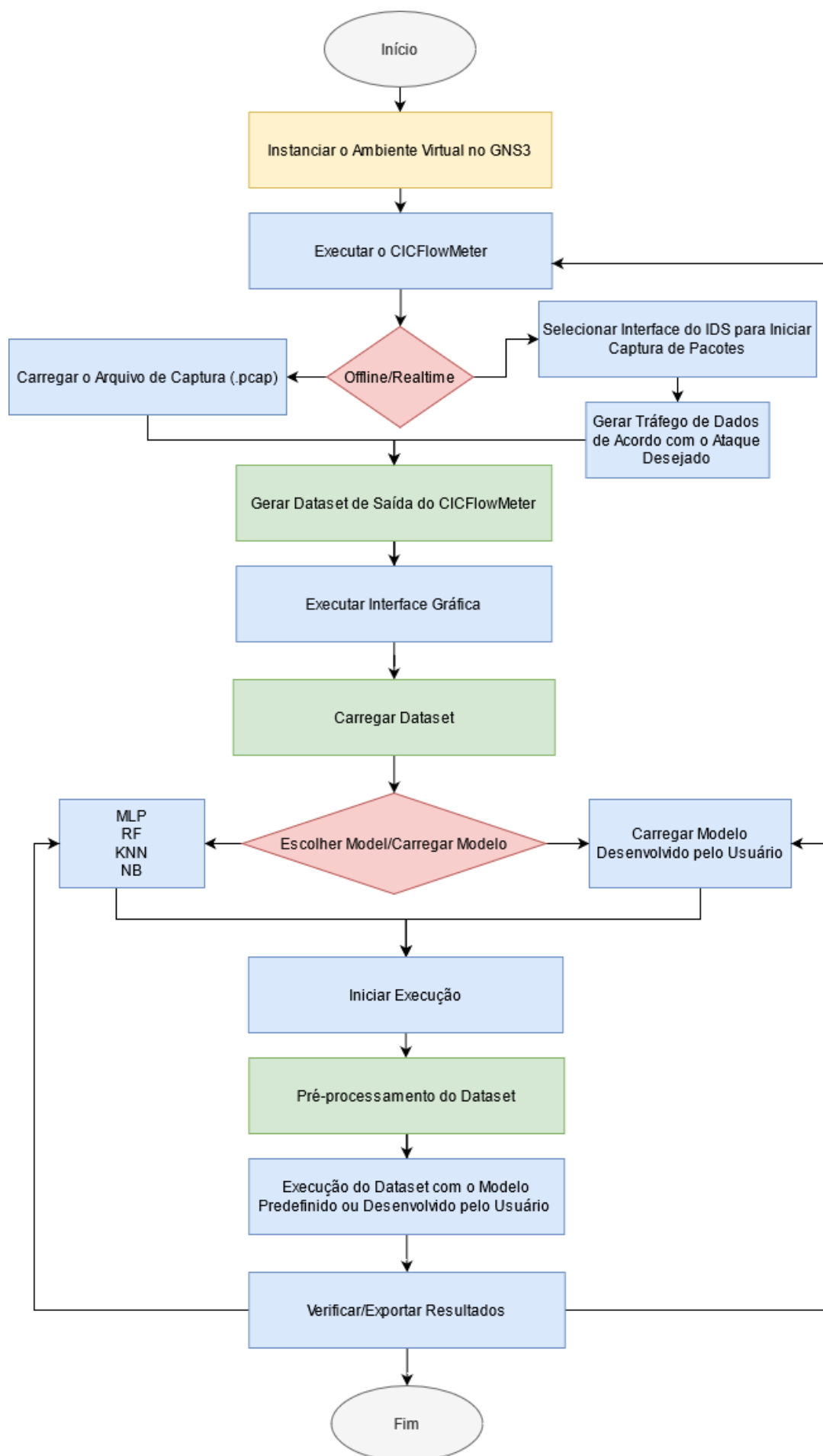
bw_iat_std	Standard deviation time between two packets sent in the backward direction
bw_iat_max	Maximum time between two packets sent in the backward direction
bw_iat_min	Minimum time between two packets sent in the backward direction
fw_psh_flag	Number of times the PSH flag was set in packets travelling in the forward direction (0 for UDP)
bw_psh_flag	Number of times the PSH flag was set in packets travelling in the backward direction (0 for UDP)
fw_urg_flag	Number of times the URG flag was set in packets travelling in the forward direction (0 for UDP)
bw_urg_flag	Number of times the URG flag was set in packets travelling in the backward direction (0 for UDP)
fw_hdr_len	Total bytes used for headers in the forward direction
bw_hdr_len	Total bytes used for headers in the backward direction
fw_pkt_s	Number of forward packets per second
bw_pkt_s	Number of backward packets per second
pkt_len_min	Minimum length of a flow
pkt_len_max	Maximum length of a flow
pkt_len_avg	Mean length of a flow
pkt_len_std	Standard deviation length of a flow
pkt_len_va	Minimum inter-arrival time of packet
fin_cnt	Number of packets with FIN
syn_cnt	Number of packets with SYN
rst_cnt	Number of packets with RST
pst_cnt	Number of packets with PUSH
ack_cnt	Number of packets with ACK
urg_cnt	Number of packets with URG
cwe_cnt	Number of packets with CWE
ece_cnt	Number of packets with ECE
down_up_ratio	Download and upload ratio
pkt_size_avg	Average size of packet
fw_seg_avg	Average size observed in the forward direction
bw_seg_avg	Average size observed in the backward direction

fw_byt_blk_avg	Average number of bytes bulk rate in the forward direction
fw_pkt_blk_avg	Average number of packets bulk rate in the forward direction
fw_blk_rate_avg	Average number of bulk rate in the forward direction
bw_byt_blk_avg	Average number of bytes bulk rate in the backward direction
bw_pkt_blk_avg	Average number of packets bulk rate in the backward direction
bw_blk_rate_avg	Average number of bulk rate in the backward direction
subfl_fw_pkt	The average number of packets in a sub flow in the forward direction
subfl_fw_byt	The average number of bytes in a sub flow in the forward direction
subfl_bw_pkt	The average number of packets in a sub flow in the backward direction
subfl_bw_byt	The average number of bytes in a sub flow in the backward direction
fw_win_byt	Number of bytes sent in initial window in the forward direction
bw_win_byt	Of bytes sent in initial window in the backward direction
Fw_act_pkt	Of packets with at least 1 byte of TCP data payload in the forward direction
fw_seg_min	Minimum segment size observed in the forward direction
atv_avg	Mean time a flow was active before becoming idle
atv_std	Standard deviation time a flow was active before becoming idle
atv_max	Maximum time a flow was active before becoming idle
atv_min	Minimum time a flow was active before becoming idle
idl_avg	Mean time a flow was idle before becoming active

idl_std	Standard deviation time a flow was idle before becoming active
idl_max	Maximum time a flow was idle before becoming active
idl_min	Minimum time a flow was idle before becoming active

ANEXO B – Fluxo Geral de Execução

Figura 18 – Fluxo geral do experimento.



ANEXO C – Resultados dos artigos selecionados

Referência	Algoritmo	Acurácia	Nota
(WESTER; HEIDING; LAGERSTRÖM, 2021)	Naive Bayes	0.998	8.5
(DESAI; VENKATESH, 2022)	CNN K-NearestNeighbors	0.998	8.5
(PANUNTUN et al., 2023)	ML LightGradientBoost	0.999	8.5
(ACHARYA; ANNAMALAI; CHOUIKHA, 2023)	Hybrid CNN-BiLSTM	0.980	8.5
(KUMAR; BANERJEE, 2023)	KNN RL SVM RF AdaBoost	0.980	8.5
(TOSSOU; QORIB; KACEM, 2023)	Hybrid CNN-LSTM	0.993	8.5
(DISHA; WAHEED, 2021)	ML DT RF MLP GB	0.927	8.5
(LIANG; XING; HOU, 2023)	Hybrid CNN-BiLSTM	0.883	8.5
(AMALAPURAM et al., 2022)	ML NB MLP	0.989	8.5
(MAZIGHI; BALLIHI; ORHANO, 2023)	ML NB RF AdaBoost	0.996	8.5
(AKUTHOTA; BHARGAVA, 2023)	ML SVM RF	0.986	8.5
(HAI; NAM, 2021)	LSTM RNN	0.995	8.5
(LEON; MARKOVIC; PUNNEKKAT, 2022)	ANN SVM KNN RF	0.997	8.5
(KARNA et al., 2021)	RF NB DT Adaboost	0.998	8.5
(KSHIRSAGAR; KUMAR, 2021)	IGR ReliefF	0.999	8.5
(AYAD; SAKR; HIKAL, 2024)	ML RF NB SVM	0.987	8.5
(SELVARAJAN et al., 2023)	Hybrid LSTM and CNN	0.978	9.0
(KYE; KIM; KWON, 2022)	Autoencoder	0.992	9.0
(SAID; ASKERZADE, 2023)	Hybrid CNN-BiLSTM	0.980	9.0
(KANUMALLI et al., 2023)	CNN-BiLSTM	0.992	9.0
(MAHMOUD et al., 2022)	Hybrid LSTM Autoencoder	0.988	9.0
(RATTI; SINGH; NANDI, 2023)	LSTM Autoencoder MLP	0.975	9.0
(SALLAM et al., 2021)	CNN	0.984	9.0
(KUMAR et al., 2023b)	RNN	0.950	9.0
(BAYKARA; ABDULRAHMAN; ALAHMED, 2022)	SVM KNN DT	0.997	9.0
(XIE et al., 2024)	Hibrido CNN LSTM	0.998	9.0
(MEZINA; BURGET; TRAVIESO-GONZÁLEZ, 2021)	CNN RNN Autoencoder	0.970	9.5
(ZHANG et al., 2022)	ML DL DT RF XGBoost DNN	0.976	9.5
(HASSAN et al., 2023)	CNN LSTM	0.990	9.5
(MALIK; SAINI, 2023)	Reinforcement Learning	0.987	9.5
(MAGÁN-CARRIÓN et al., 2022)	RF LR	0.994	9.5
(XU et al., 2021)	Autoencoder	0.931	10
(SAID; SABIR; ASKERZADE, 2023)	Hybrid CNN-BiLSTM	0.977	10
(WISANW; THAMMAW, 2021)	Hybrid Naive Bayes/SVM	0.889	10
(ABOBAKER, 2023)	KNN DT XGBC GB	0.996	10
(HALBOUNI et al., 2022)	Hybrid CNN-LSTM	0.996	10
(HNAME et al., 2023)	LSTM-Autoencoder	0.999	10
(LIU; GU; WANG, 2021)	Hybrid K-Means/RF CNN	0.999	10
(KANDHRO et al., 2023)	DNN CNN RNN	0.969	10
(C et al., 2023)	CNN XGB RNN LSTM SVM	0.980	10
(ALHABSHY; HAMEED; ELDAHSHAN, 2022)	SVM KNN DT	0.995	10
(YOO et al., 2021)	MLP CNN	0.980	10
(DAS et al., 2022)	ML SVM DT NB RL	0.995	10
(THAKKAR; LOHIYA, 2023)	Ensemble DNN	0.989	10
(SAH; SINGH; BANERJEE, 2024)	KNN SVM RNN	0.999	10
(CHEN et al., 2024)	ML XGBoost	0.999	10
(HU et al., 2024)	Hybrid LSTM	0.999	10
(ELSAIED; MOHAMED; MADKOUR, 2024b)	Hybrid CNN RNN LSTM	0.999	10
(SANNIGRAHI; THANDEESWARAN, 2024)	Hybrid ML SVM NB RF	0.999	10

ANEXO D – Tipos de Ataques

Ataques e ferramentas utilizadas para construção do conjunto de dados CIC-IDS2018 (CYBERSECURITY, 2018).

D.0.1 Ataque de Força Bruta

Existem muitas ferramentas para conduzir ataques de força bruta e quebra de senha, como Hydra, Medusa, Ncrack, módulos Metasploit e scripts Nmap NSE. Além disso, existem algumas ferramentas como hashcat e hashpump para cracking de hash de senha. Mas uma das ferramentas multithread mais abrangentes é o Patator, que é escrito em Python e parece ser mais confiável e flexível do que outros. Ele também pode salvar todas as respostas em um arquivo de log separado para revisão posterior. Neste conjunto de dados, usamos dois módulos, FTP e SSH na máquina Kali Linux como máquina atacante e um sistema Ubuntu 14.0 como máquina vítima.

D.0.2 Ataque Heartbleed

Uma das ferramentas mais famosas para explorar o Heartbleed é o Heartleech. Ele pode fazer a varredura de sistemas vulneráveis ao bug e pode ser usado para explorá-los e exfiltrar dados. Alguns recursos importantes: Veredictos conclusivos / inconclusivos sobre se o alvo é vulnerável, Download em massa / rápido de dados heartbleed em um arquivo grande para processamento offline usando muitos threads, Recuperação automática de chaves privadas sem etapas adicionais, Alguma evasão IDS limitada, Suporte para STARTTLS, Suporte IPv6, Suporte a proxy Tor / Socks5n e Informações extensas de diagnóstico de conexão.

D.0.3 Botnet

Neste conjunto de dados, foi utilizado o Zeus, que é um pacote de malware cavalo de Tróia que é executado em versões do Microsoft Windows. Embora possa ser usado para realizar muitas tarefas maliciosas e criminosas, é frequentemente usado para roubar informações bancárias por meio de registro de pressionamento de tecla man-in-the-browser e obtenção de formulários. Ele também é usado para instalar o ransomware CryptoLocker. O Zeus se espalha principalmente por meio de downloads drive-by e esquemas de phishing. Além disso, como complemento, foi usado o botnet Ares, que é um botnet de código aberto e tem os seguintes recursos: shell cmd.exe remoto, persistência, upload / download de arquivo, captura de tela e registro de chave. Nesse cenário, foram infectadas máquinas com dois botnets diferentes (Zeus e Ares), também a cada 400 segundos é solicitada capturas de tela dos zumbis.

D.0.4 Negação de Serviço

Slowloris é um tipo de ferramenta de ataque de negação de serviço inventada por Robert Hansen que permite que uma única máquina desligue o servidor da web de outra máquina com largura de banda mínima e efeitos colaterais em portas e serviços não relacionados.

D.0.5 Negação de Serviço Distribuída

O High Orbit Ion Cannon, frequentemente abreviado para HOIC, é um teste de estresse de rede de código aberto e aplicativo de ataque de negação de serviço escrito em BASIC projetado para atacar até 256 URLs ao mesmo tempo. Ele foi projetado para substituir o Canhão de Íons de Órbita Baixa, desenvolvido pela Praetox Technologies.

D.0.6 Ataques WEB

Foi utilizado Damn Vulnerable Web App (DVWA) para conduzir os ataques. DVWA é um aplicativo da web PHP / MySQL que é vulnerável. Seus principais objetivos são ajudar os profissionais de segurança a testar suas habilidades e ferramentas em um ambiente legal, ajudar os desenvolvedores da web a entender melhor os processos de proteção de aplicativos da web e auxiliar professores / alunos a ensinar / aprender a segurança de aplicativos da web em um ambiente de sala de aula . Para automatizar os ataques na seção XSS e Brute-force, foi desenvolvido um código de automação com o framework Selenium.

D.0.7 Infiltração da Rede

Nesse cenário, um aplicativo vulnerável (como Adobe Acrobat Reader 9) deve ser explorado. Primeiro, a vítima recebe um documento malicioso por e-mail. Então, após a exploração bem-sucedida usando a estrutura Metasploit, um backdoor será executado no computador da vítima. Assim, é possível conduzir diferentes ataques na rede da vítima, incluindo varredura de IP, varredura completa de portas e enumerações de serviço usando Nmap.