Uma Abordagem em Múltiplos Estágios para Detecção Hierárquica de Intrusões

Luiz Henrique, Rayhene Arruda, Rodrigo Abreu

Centro de Informática, Universidade Federal de Pernambuco (UFPE)

lhbas@cin.ufpe.br

rrda@cin.ufpe.br

raafm@cin.ufpe.br

***Abstract*— A constante digitalização no mundo moderno e o crescimento de dispositivos interconectados representam um aumento no risco de ameaças cibernéticas. Dessa forma, se faz necessário um esforço no aprimoramento dos mecanismos de defesa existentes e no desenvolvimento de abordagens inovadoras para lidar com essas ameaças. Este estudo busca explorar uma abordagem inovadora de detecção hierárquica, validada em conjuntos de dados públicos, que se destaca pela escalabilidade e capacidade de detectar ataques desconhecidos (zero-day) de maneira eficaz e robusta. A proposta supera abordagens atuais, apresentando-se como uma solução promissora diante dos desafios da segurança cibernética.**

***Keywords*— Detecção de intrusão, classificação binária, classificação multi-classe, detecção multiestágio, arquitetura hierárquica.**

I. Introdução

O problema investigado reside na limitação dos sistemas de detecção de intrusão atuais em ambientes digitais. Considerando o aumento de ataques cibernéticos, incluindo os desconhecidos (zero-day), e os desafios de escalabilidade e privacidade na monitorização de redes distribuídas, o estudo propõe o desenvolvimento de um sistema de detecção hierárquica de intrusões como solução para superar essas limitações. Esse problema é relevante devido à crescente digitalização da sociedade moderna, que expõe a sociedade a riscos significativos de ataques cibernéticos. A detecção eficaz de intrusões é crucial para preservar a segurança digital e proteger sistemas críticos.

A solução proposta neste trabalho é necessária para abordar as limitações das abordagens existentes, especialmente a dependência de modelos únicos, a alta carga computacional, a falta de detecção de ataques desconhecidos e a inadequação à escalabilidade e requisitos de privacidade.

Essa solução consiste em uma abordagem de múltiplos estágios inovadora para detecção hierárquica de intrusões, projetada para ser independente da fonte de entrada ou método de detecção. Ela incorpora detecção de outliers, classificação por etapas e otimização de hiperparâmetros utilizando práticas de aprendizado de máquina.

Dessa forma, as principais contribuições deste trabalho são:

- Proposta de abordagem de múltiplos estágios adaptável para detecção hierárquica de intrusões.
- Capacidade de realizar detecção binária e de múltiplas classes, incluindo ataques zero-day.
- Otimização de hiperparâmetros usando algoritmos de aprendizado de máquina.
- Validação extensiva em conjuntos de dados modernos, comparando com modelos de referência e abordagens de múltiplos estágios atuais estado-da-arte.

II. Trabalhos Relacionados

Na literatura, existem alguns estudos sobre sistemas de detecção de intrusão baseados em múltiplas camadas e estágios, alguns deles que se destacam em relação a abordagem proposta no artigo analisado são:

Zhang et al. [2] - IDS em Redes Elétricas Inteligentes (Smart Grids):

- Propõem um IDS em três camadas para detectar ciberataques em smart grids.
- Cada camada monitora um conjunto exclusivo de características para detecção de ataques.
- Comunicação interna entre camadas permite a identificação de tráfego malicioso.

Ali e Yousaf [3] - IDS em Redes Definidas por Software (SDN):

- Apresentam uma abordagem de três camadas para detectar intrusões em SDN.
- Utilizam autenticação de usuário, filtragem difusa e uma Rede Neural Convolucional (CNN) para detecção de tráfego malicioso.

Li et al. [4] - Cascata de Classificadores para Tráfego de Rede:

- Propõem um sistema de cascata para classificar tipos de ataque em tráfego de rede.
- A primeira etapa consiste em classificadores binários cujas saídas são agregadas para formar uma previsão.
- A segunda etapa utiliza o modelo K-nearest neighbors (KNN) para classes incertas.

Umer et al. [5] - Modelo Multiestágio para Redes de Próxima Geração:

- Apresentam um modelo de múltiplos estágios para redes de próxima geração.
- Utilizam OC-SVM na primeira etapa e um mapa auto-organizável na segunda para detecção de anomalias e classificação multi-classe.

Bovenzi et al. [6] - Abordagem Hierárquica com DAE e Classificadores Suaves:

- Propõem uma abordagem hierárquica em dois estágios com um Autoencoder Profundo (DAE) na primeira etapa.
- O classificador suave na segunda etapa detecta ataques desconhecidos usando um limiar de confiança na predição.
- Essa abordagem é a que mais se assemelha à solução proposta no artigo.

Injadat et al. [7] - Múltiplos Estágios em um Pipeline de Processamento de Dados:

- Descrevem um modelo multiestágio com vários passos em um pipeline de processamento de dados.
- Não abordam uma cadeia de classificadores, mas referem-se a diferentes etapas no processamento de dados.

A tabela abaixo apresenta uma comparação entre os estudos relacionados a abordagem proposta levando em consideração parâmetros como detecção, classificação, número de estágios, detecção de ataques zero-day, hierarquia de estágios e redução de custos computacionais.

TABLE I
TAXONOMY OF THE RELATED WORK

Study	Detection	Classification	Stages	Zero-Day	Hierarchical	Computational Reduction
Zhang et al. [5]	±	✓	3	✗	✓	✗
Ali and Yousaf [6]	±	✓	3	✗	✓	✗
Li et al. [7]	✗	✓	2	✗	✗	✗
Pajouh et al. [8] [9]	✗	✓	2	✗	✗	✗
Al-Yaseen et al. [10]	✗	✓	5	✓	✗	✗
Ji et al. [11]	✓	✓	2	✗	✗	✗
Khan et al. [12]	✓	✓	2	✗	✗	✗
Divyatmika and Sreekesh [13]	✓	✓	3	✓	✗	±
Umer et al. [14]	✓	✓	2	✗	✗	✗
Abuadlla et al. [15]	✓	✓	2	±	✗	✗
Bovenzi et al. [16]	✓	✓	2	✓	✓	±
Verkerken et al.	✓	✓	3	✓	✓	✓

III. SISTEMA PROPOSTO PELO ARTIGO DE REFERÊNCIA

O sistema proposto pelo artigo de referência é uma abordagem hierárquica de múltiplos estágios para a detecção de intrusões, destinada a oferecer eficiência computacional e precisão na classificação. A arquitetura é composta por três estágios distintos:

Estágio 1 - Detecção de Anomalias
Entrada: Vetor de características (X)
Saída: Pontuação de Anomalia (λ B)

O primeiro estágio utiliza um modelo binário treinado exclusivamente com dados benignos para aprender o comportamento normal da rede monitorada. Durante o treinamento, são selecionados hiperparâmetros ótimos usando a AUROC como métrica de validação. A pontuação de anomalia (λB) é gerada e comparada a um limiar (τ B) para decidir se uma amostra é benigna ou suspeita. O objetivo desse estágio é filtrar amostras maliciosas de maneira computacionalmente eficiente. Dessa forma, o número de amostras que

precisam ser analisadas nos estágios subsequentes é significativamente reduzido.

Estágio 2 - Classificador de Ataque
Entrada: Pontuação de Anomalia (λB)
Saída: Probabilidades para classes de ataques conhecidos ([PATK1, PATK2,...,PATKN])

Este estágio tenta classificar amostras previstas como maliciosas pelo primeiro estágio em classes de ataques conhecidas. Um classificador multi-classe é treinado exclusivamente com dados maliciosos. O limiar (τM) é definido para determinar se uma amostra pertence a uma classe conhecida ou é desconhecida. A saída inclui as probabilidades para cada classe conhecida, e a classe prevista é determinada pela maior probabilidade, a menos que seja inferior ao limiar (τ M), caso em que a amostra é considerada desconhecida.

Estágio 3 - Extensão
Entrada: Pontuação de Anomalia (λB)
Saída: Classificação final (Benigno, Ataque Conhecido, Ataque Desconhecido)

Este estágio não implementa outro classificador, mas reutiliza a pontuação de anomalia do primeiro estágio para ajustar as previsões. O limiar (τU) é definido para reduzir falsos positivos, e a amostra é classificada como benigna se a pontuação for inferior a τU ou como um ataque desconhecido se for superior. É eficaz apenas se τU for maior que τB, garantindo maior precisão.

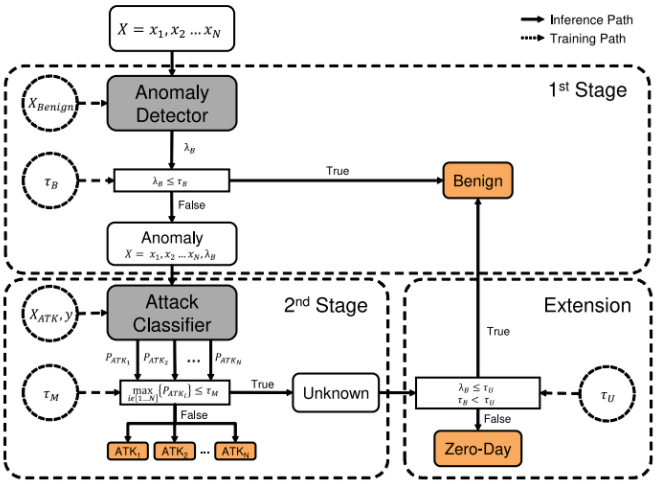
Os tempos de execução para cada estágio são definidos como:

Testágio1 = O(TB)
Testagio2 = O(TB+TM)
Testágio3 = O(TB+TM+TU)
Ttotal = O(TB+αTM+αβTU)

Isso pode ser simplificado para O(TB+TM), onde:

TB = Complexidade Temporal do Detector de Anomalias
TM = Complexidade Temporal do Classificador de Ataques
TU = Complexidade Temporal do Estágio de Extensão
α = Fração de fluxos encaminhados pelo 1º estágio
β = Fração de fluxos encaminhados pelo 2º estágio

A figura abaixo representa a arquitetura proposta do sistema hierárquico de detecção de intrusão de múltiplos estágios.



IV. Solução proposta para melhorar a solução do artigo de referência

O sistema proposto é o mesmo sistema que o sistema proposto pelo artigo, porém com modelos de machine learning distintos. Os modelos de machine learning foram escolhidos visando um aumento da velocidade de processamento do sistema proposto.

O artigo original usava (one class support vector machine) uma OCSVM lenta como detector de anomalias e uma random forest como classificador de anomalias. Como sugerido pelo próprio artigo, é conveniente que o primeiro estágio seja veloz uma vez que todos os dados passarão por eles. O segundo estágio recebe uma quantidade menor de dados e por isso pode ser mais lento, porém melhor (ter maior acurácia). Seguindo esta sugestão, trocamos o primeiro modelo por uma autoencoder mais rápido e dobramos a quantidade de árvores do classificador (mantivemos uma random forest).

Dois algoritmos para auxiliar a escolha dos thresholds foram testados. O primeiro busca um threshold que forneça determinado recall para um modelo no dataset de validação fornecido para o algoritmo (a entrada é o modelo e um dataset, tanto as features com os labels). Este algoritmo é ideal para regular a quantidade de dados que passam para o segundo estágio. O segundo algoritmo tem como entrada e saída os mesmos parâmetros que o primeiro algoritmo, porém ele encontra o threshold com a melhor acurácia possível para o dataset fornecido. Como os dados são muito desbalanceados, em geral, a acurácia não é a única métrica buscada e a maximização dela pode causar uma piora na classificação de uma classe pouco representativa (com poucos dados no dataset), como é o caso dos dados "zero-day".

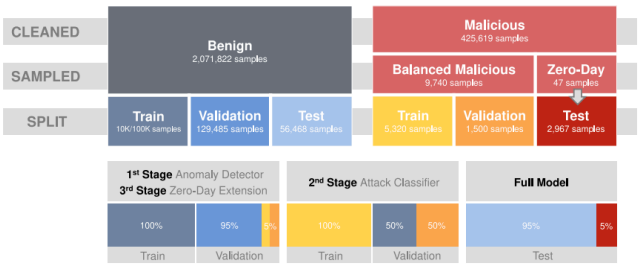
V. Metodologia

Os dados usados pelo artigo de referência incluem o conjunto de dados CIC-IDS-2017 e o conjunto de dados sucessor CSE-CIC-IDS-2018. Esses conjuntos de dados foram utilizados para avaliar o sistema proposto no artigo.

O conjunto de dados CIC-IDS-2017 foi escolhido porque é um conjunto de dados moderno baseado em fluxo de rede intrusiva, desenvolvido especificamente para detecção de intrusões. Ele foi construído do zero para atender a 11 critérios de um conjunto de dados adequado para detecção de intrusões, sendo considerado adequado para fins de benchmark.

44 features foram utilizadas no artigo original, usamos esta mesma quantidade para reproduzir o artigo original. No dataset original, os dados podem ser de 15 classes, há 5 tipos de (D)DOS, por exemplo. Porém o artigo, assim como nós, agregamos em classes mais abrangentes: DDOS, Brute Force, Web Attack, Botnet, Unknown (que serão ataques considerados "zero-day") e Benign. Os dados Unknown são provenientes das classes Infiltration e Heartbleed.

O conjunto de dados original foi limpo em três etapas, removendo informações redundantes, eliminando linhas com valores ausentes ou infinitos e removendo duplicatas. Após a limpeza, os dados foram escalonados normalizando as características usando a implementação StandardScaler do scikit-learn. Um escalonador separado é usado para normalizar os dados para o detector de anomalias e o classificador seguindo as melhores práticas para evitar qualquer viés. O conjunto de dados foi então dividido em três partes: treinamento, validação e teste de acordo com a imagem abaixo.



Inicialmente, empregamos técnicas de limpeza e padronização, incluindo a remoção de espaços desnecessários nos nomes das colunas e a correção de etiquetas redundantes nos conjuntos de dados. Além disso, tratamos os registros duplicados e valores nulos ou não finitos, visando garantir a integridade e qualidade dos dados.

As features foram escolhidas a partir das colunas da tabela de teste fornecida pelos autores. A normalização delas foi feita pelos scalers também fornecidos (no caso da reprodução) e outros scalers usados para melhoria dos dados.

Em seguida, limitamos a quantidade de dados maliciosos para equilibrar as classes e realizamos a divisão dos conjuntos em treino, validação e teste. Por fim, normalizamos os dados e criamos um datagrama específico para o treinamento do primeiro estágio.

Na reprodução, usamos o OCSVM para o treinamento do detector de anomalias do primeiro estágio, utilizando o conjunto de dados de treino e os hiperparâmetros fornecidos pelo autor do artigo, no segundo estágio, foi usado o Random Forest.

Já na melhora, utilizamos o Auto Encoder no primeiro estágio e Random Forest no segundo, utilizando outros hiperparâmetros.

VI. Resultados e discussões

Não conseguimos reproduzir com a mesma qualidade os experimentos do artigo original, isso possivelmente se deu pela falta de rigor na experimentação (repetição de execuções com diferentes seeds) e uma melhor tunagem de hiperparâmetros. Os resultados obtidos no artigo original são muito mais próximos de acertar todos os resultados com quase todas as classes, inclusive zero-day. Os nossos melhores resultados (acurácia, precision, etc) chegaram no máximo em cerca de 90% .

Resultados reproduzidos do artigo original:

test_full_model	precision	recall	f1-score	support
(D)DOS	0.35	0.99	0.51	1076
BENIGN	1.00	0.91	0.95	100000
Botnet	0.24	1.00	0.39	1040
Brute Force	0.45	0.99	0.62	1100
Port Scan	0.95	1.00	0.97	1066
Unknown	0.00	0.00	0.00	0
Web-Attack	0.42	0.99	0.60	1038
accuracy			0.91	105320
macro avg	0.49	0.84	0.58	105320
weighted avg	0.97	0.91	0.94	105320

test_set	precision	recall	f1-score	support
(D)DOS	0.34	0.97	0.50	584
BENIGN	1.00	0.91	0.95	54387
Botnet	0.24	1.00	0.39	477
Brute Force	0.45	0.99	0.62	584
Port Scan	0.95	1.00	0.97	197
Unknown	0.06	0.89	0.11	47
Web-Attack	0.42	0.99	0.60	584
accuracy			0.92	56860
macro avg	0.51	0.96	0.60	56860
weighted avg	0.97	0.92	0.94	56860

Tempo de Execução (segundos)

	test_full_model	test_set
Stage1	9.75	5.23
Stage2	1.71	0.76
Stage3	0.00	0.00
Sistema	11.47	5.99

Resultado da melhoria proposta:

test_full_model	precision	recall	f1-score	support
(D)DOS	0.32	0.99	0.49	1076
BENIGN	1.00	0.88	0.94	100000
Botnet	0.24	1.00	0.39	1040
Brute Force	0.46	1.00	0.62	1100
Port Scan	0.96	1.00	0.98	1066
Unknown	0.00	0.00	0.00	0
Web-Attack	0.42	0.99	0.59	1038
accuracy			0.89	105320
macro avg	0.49	0.84	0.57	105320
weighted avg	0.97	0.89	0.92	105320

test_set	precision	recall	f1-score	support
(D)DOS	0.33	0.98	0.49	584
BENIGN	1.00	0.91	0.95	54387
Botnet	0.21	0.99	0.35	477
Brute Force	0.58	0.98	0.73	584
Port Scan	0.94	0.99	0.96	197

Unknown	0.03	0.43	0.06	47
Web-Attack	0.42	0.99	0.59	584
accuracy			0.92	56860
macro avg	0.50	0.90	0.59	56860
weighted avg	0.97	0.92	0.94	56860

Tempo de Execução (segundos)

	test_full_model	**test_set**
Stage1	0.18	0.08
Stage2	1.96	0.78
Stage3	0.00	0.00
Sistema	2.14	0.86

VII. Conclusão

Em conclusão, a abordagem de múltiplos estágios proposta para detecção de intrusões hierárquica apresenta notáveis avanços, principalmente pela sua alta adaptabilidade, capacidade de redução de largura de banda e requisitos computacionais, bem como pela eficácia na detecção de ataques zero-day. Os resultados experimentais indicam um desempenho superior em comparação com abordagens existentes, evidenciando a robustez do modelo. As melhorias para o sistemas propostas pelo nosso trabalho tornaram-no cerca de 7 vezes mais rápido mantendo as métricas de classificação razoavelmente próximas.

As limitações potenciais encontradas incluem a sensibilidade aos thresholds selecionados, que podem ser atenuadas pelos algoritmos de busca de threshold anteriormente mencionados, a necessidade de avaliação em diferentes fontes de entrada (para IDS baseados em host e híbridos, por exemplo) e a constante evolução do cenário de ameaças. Além disso, embora a privacidade tenha sido destacada como preservada durante o treinamento e operação, é interessante realizar análises aprofundadas para garantir a conformidade com regulamentações e normas de segurança.

Referências

[1] Verkerken, M., D'hooge, L., Sudyana, D., Lin, Y.-D., Wauters, T., Volckaert, B., & De Turck, F. (2023). A Novel Multi-Stage Approach for Hierarchical Intrusion Detection. IEEE Transactions on Network and Service Management, 20(3), 1–14. DOI: 10.1109/TNSM.2023.3259474

[2] Y. Zhang, L. Wang, W. Sun, R. C. Green, II, and M. Alam, "Distributed intrusion detection system in a multi-layer network architecture of smart grids," IEEE Trans. Smart Grid, vol. 2, no. 4, pp. 796–808, Dec. 2011.

[3] A. Ali and M. M. Yousaf, "Novel three-tier intrusion detection and prevention system in software defined network," IEEE Access, vol. 8, pp. 109662–109676, 2020

[4] L. Li, Y. Yu, S. Bai, Y. Hou, and X. Chen, "An effective two-step intrusion detection approach based on binary classification and k -NN," IEEE Access, vol. 6, pp. 12060–12073, 2018.

[5] M. F. Umer, M. Sher, and Y. Bi, "A two-stage flow-based intrusion detection model for next-generation networks," PLoS One, vol. 13, no. 1, Jan. 2018, Art. no. e0180945.

[6] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapé, "A hierarchical hybrid intrusion detection approach in IoT scenarios," in Proc. IEEE Global Commun. Conf., Dec. 2020, pp. 1–7.

[7] M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, "Multi-stage optimized machine learning framework for network intrusion detection," IEEE Trans. Netw. Service Manag., vol. 18, no. 2, pp. 1803–1816, Jun. 2021.