Detecção de Intrusão em Múltiplos Domínios com NIDS Baseado em Saídas Antecipadas

Lucas Sotomaior A. Pereira[1] **e Eduardo K. Viegas**[1]

[1]Programa de Pós-Graduação em Informática (PPGIa)
Pontifícia Universidade Católica do Paraná (PUCPR)
80.215-901 – Curitiba – PR

{lucas.apereira, eduardo.viegas}@ppgia.pucpr.br

Resumo. *Este artigo propõe uma arquitetura de Network Intrusion Detection System (NIDS) que combina treinamento cross-dataset e um mecanismo dinâmico de saídas antecipadas para equilibrar o desempenho de detecção e a eficiência computacional. A abordagem utiliza um ramo leve para classificar amostras de menor complexidade e um ramo de maior capacidade computacional para processar eventos que exigem representações mais profundas. Experimentos realizados nos conjuntos de dados BoT-IoT, UNSW-NB15 e CICIDS2018 demonstram que a estratégia proposta melhora a capacidade de generalização em diferentes domínios, alcançando ganhos de até* $0,25$*,* $0,25$ *e* $0,34$ *na pontuação F1, respectivamente, quando comparada a arquiteturas convencionais. Além disso, o mecanismo de saída antecipada reduz o custo médio de inferência em até 22%, mantendo alto desempenho de classificação. Os resultados evidenciam que a combinação entre adaptação multidomínio e inferência hierárquica constitui uma alternativa eficiente para a implantação de NIDS em ambientes de rede heterogêneos e com restrições computacionais.*

1. Introdução

Sistemas de Detecção de Intrusão em Rede (NIDS) baseados em comportamento têm empregado extensivamente métodos de Aprendizado de Máquina, com destaque para as Redes Neurais Profundas (Deep Neural Network (DNN)), que vêm demonstrando desempenho superior em tarefas de detecção de ameaças [Pawlicki et al. 2023, Espindola et al. 2026b]. A obtenção desses resultados normalmente depende do treinamento supervisionado, com conjuntos de dados separados para treinamento, validação e avaliação. Entretanto, a capacidade de generalização desses modelos está diretamente relacionada à representatividade dos dados utilizados durante o treinamento, que devem refletir as características do ambiente de rede onde o sistema será aplicado. Essa representatividade é difícil de ser alcançada na prática, uma vez que ambientes de rede são altamente dinâmicos e sofrem constantes alterações decorrentes da evolução dos ataques, da inclusão de novos serviços e da variação nos padrões de comunicação [Simioni et al. 2025a]. Diante desse cenário, estudos recentes têm investigado estratégias para aumentar a capacidade de adaptação dos modelos entre diferentes domínios de rede, permitindo sua aplicação em ambientes heterogêneos. Contudo, a identificação de características generalizáveis, capazes de evitar dependências excessivas de particularidades presentes nos dados de origem, permanece como um dos principais desafios para a área [Cantone et al. 2024].

A maioria dos trabalhos presentes na literatura ainda trata a generalização como um aspecto secundário, priorizando a obtenção de elevadas taxas de detecção em ambientes específicos e avaliados de forma isolada [Cantone et al. 2024]. Como consequência, é comum que as soluções propostas adotem arquiteturas progressivamente mais complexas, com o aumento da profundidade das redes e da quantidade de parâmetros treináveis, visando melhorar o desempenho preditivo [Simioni et al. 2025b]. Entretanto, essa estratégia pode comprometer a capacidade de generalização dos modelos, favorecendo o sobreajuste aos padrões particulares dos domínios utilizados durante o treinamento [Barik and Misra 2025]. Além disso, a construção de NIDS aplicáveis em dispositivos com restrições computacionais requer um compromisso adequado entre capacidade de adaptação e custo de processamento [Wang et al. 2024]. Embora arquiteturas sofisticadas possam apresentar resultados expressivos quando avaliadas em conjuntos de dados específicos, suas demandas computacionais frequentemente inviabilizam a execução em ambientes com recursos limitados. Nesse contexto, o desenvolvimento de modelos mais eficientes, capazes de preservar elevado desempenho de detecção em cenários *cross-dataset*, torna-se fundamental para a adoção prática de sistemas de detecção baseados em comportamento.

Contribuição. Em resposta a esses desafios, este artigo apresenta um NIDS leve e com capacidade de generalização, baseado em aprendizado de máquina e desenvolvido a partir de uma abordagem composta por duas estratégias principais. Primeiramente, a inferência é realizada por uma rede neural com saídas antecipadas, permitindo a classificação eficiente de eventos menos complexos sem a necessidade de executar todas as camadas do modelo. Essa estratégia reduz o custo computacional associado ao processamento de amostras simples, preservando a capacidade de detecção para padrões de tráfego mais complexos. Em segundo lugar, a generalização do modelo é promovida por meio de uma configuração de treinamento multidomínio, na qual diferentes ambientes de rede são considerados durante o desenvolvimento. Essa abordagem incorpora uma função de perda específica para incentivar o aprendizado de características invariantes entre domínios distintos, aumentando a robustez do modelo frente a variações nos cenários de aplicação.

As principal contribuição do trabalho envolve a proposição de um novo NIDS leve, com capacidade de generalização baseada em aprendizado de máquina, que melhora a pontuação F1 em até $0,34$ em contextos *cross-dataset*, reduz os custos computacionais em até 22%.

2. Trabalhos Relacionados

Nos últimos anos, diversas pesquisas envolvendo aprendizado de máquina aplicado a NIDS têm concentrado esforços na maximização do desempenho de detecção em ambientes específicos, frequentemente deixando em segundo plano aspectos relacionados à capacidade de generalização e aos requisitos computacionais necessários para sua implantação [Molina-Coronado et al. 2020, Espindola et al. 2026a]. Lidholm *et al.* [Lidholm et al. 2024], por exemplo, realizaram uma análise comparativa de algoritmos e técnicas de pré-processamento baseadas em fluxo em diferentes cenários industriais e acadêmicos. Embora os resultados tenham demonstrado a viabilidade da utilização de mecanismos de detecção em dispositivos de borda, o estudo não explorou de forma aprofundada os impactos da variação entre conjuntos de dados e dos custos associados à inferência. Essa limitação também foi destacada por Cantone *et al.* [Cantone et al. 2024], que identificaram reduções significativas no desempenho de modelos quando aplicados a cenários

cross-dataset [Filho et al. 2025]. Em paralelo, pesquisas direcionadas à redução do custo computacional têm investigado alternativas mais eficientes, incluindo representações leves de pacotes sem o uso de aprendizado de máquina [Zohourian et al. 2024, de Oliveira et al. 2025], bem como técnicas de redução de dimensionalidade e estratégias de amostragem para minimizar a sobrecarga durante a inferência [Roy et al. 2022]. Além disso, mecanismos de saída antecipada vêm sendo explorados como uma alternativa para acelerar o processo de classificação, como demonstrado pela rede neural proposta por Simioni *et al.* [Simioni et al. 2025a], que busca combinar inferência eficiente e capacidade de detecção. Entretanto, essa abordagem ainda apresenta limitações relacionadas à avaliação em ambientes restritos a um único domínio. De forma geral, os trabalhos existentes demonstram avanços relevantes em termos de precisão e eficiência computacional; contudo, a generalização entre diferentes ambientes, a redução dos custos de execução e a confiabilidade em cenários reais permanecem como desafios importantes para a adoção de NIDS baseados em aprendizado de máquina.

3. Definição do Problema

Esta seção apresenta uma análise aprofundada do desempenho de abordagens atuais de NIDS baseadas em Machine Learning (ML), considerando cenários nos quais a capacidade de generalização é incorporada de forma explícita ao processo de avaliação dos modelos.

3.1. Conjunto de Dados

A avaliação de NIDS baseados em aprendizado de máquina em cenários *cross-dataset* apresenta desafios significativos devido à falta de padronização entre as representações dos vetores de características utilizados pelos diferentes conjuntos de dados. Para minimizar esse problema e estabelecer uma base de comparação consistente, este trabalho adota conjuntos de dados de intrusão normalizados, nos quais o tráfego de rede é representado de maneira uniforme utilizando o formato NetFlowV3 composto por 44 características.

O cenário de avaliação considera três domínios de dados distintos. O conjunto de dados **BoT-IoT** [Koroniotis et al. 2019] é utilizado para representar ambientes relacionados à Internet das Coisas (*IoT*) e padrões de comportamento associados a botnets, incluindo ataques como DoS, *backdoors* e *exploits*, gerados em um ambiente simulado de cibersegurança. O conjunto **UNSW-NB15** [Moustafa and Slay 2015] contém amostras de tráfego produzidas pela ferramenta IXIA PerfectStorm, abrangendo nove categorias de ataque, incluindo *Fuzzers*, *Shellcode* e *Worms*. Por fim, o conjunto **CICIDS2018** [Sarhan and Layeghy 2023] contempla protocolos tradicionais de redes de tecnologia da informação e cenários de ataques mais complexos, como infiltrações e combinações de DDoS com *PortScan*, gerados em um ambiente controlado de testes em larga escala.

3.2. Construção do Modelo

Para avaliar o impacto da complexidade arquitetônica na capacidade de generalização do modelo, este trabalho considera uma única configuração de Perceptron Multicamadas (Multilayer Perceptron (MLP)) em um cenário *cross-dataset*. A arquitetura selecionada corresponde ao modelo computacionalmente mais exigente, seguindo uma abordagem de maior capacidade representacional para investigar o comportamento de redes neurais

Tabela 1. Desempenho do modelo computacionalmente exigente em cenários dentro do conjunto de dados e entre diferentes conjuntos de dados.

Ambiente de Treinamento	BoT-IoT			UNSW-NB15			CICIDS2018		
	TVP	**TVN**	**F1**	**TVP**	**TVN**	**F1**	**TVP**	**TVN**	**F1**
BoT-IoT	0.89	0.96	0.92	0.95	0.02	0.65	0.68	0.40	0.60
UNSW-NB15	0.98	0.02	0.66	0.99	0.98	0.99	0.70	0.13	0.54
CICIDS2018	0.32	0.41	0.33	0.95	0.26	0.71	0.92	0.99	0.95

profundas quando aplicadas a diferentes domínios de tráfego de rede, conforme a metodologia de benchmarking para NIDS proposta em [Cantone et al. 2024]. O modelo é composto por duas camadas ocultas iniciais com 88 neurônios cada, seguidas por quatro camadas densas adicionais contendo 1024, 2048, 2048 e 1024 neurônios, respectivamente. Todas as camadas utilizam função de ativação ReLU e dropout com taxa de $0,2$, enquanto a camada final possui 2 neurônios com função de ativação sigmoide para realizar a classificação binária.

O modelo foi implementado utilizando PyTorch $v.2.10$. O treinamento foi realizado durante 500 épocas, empregando uma taxa de aprendizado de 1×10^{-4}, o otimizador Adam e lotes contendo 128 amostras. Com o objetivo de reduzir o risco de sobreajuste, foi aplicado um mecanismo de parada antecipada, interrompendo o processo de treinamento quando a perda de validação não apresentava melhoria durante 15 épocas consecutivas. A função de perda utilizada foi a Entropia Cruzada Binária (BCE), responsável por orientar a atualização dos parâmetros durante o processo de otimização. Cada domínio de dados foi particionado utilizando uma divisão estratificada de 40% para treinamento, 30% para validação e 30% para teste, garantindo a preservação da distribuição original das classes em cada subconjunto.

A avaliação do desempenho foi realizada a partir de métricas obtidas da matriz de confusão, considerando Verdadeiros Positivos (VP) e Verdadeiros Negativos (VN) como classificações corretas, enquanto Falsos Positivos (FP) e Falsos Negativos (FN) representam erros de classificação. Nesse contexto, amostras contendo atividades de intrusão são definidas como a classe positiva, enquanto o tráfego legítimo corresponde à classe negativa. A capacidade geral de classificação do modelo é mensurada utilizando a métrica F1-score, que permite avaliar conjuntamente os efeitos de falsos positivos e falsos negativos.

3.3. O Desafio da Generalização

O conjunto de experimentos conduzidos neste estudo tem como objetivo responder às seguintes Questões de Pesquisa (QP):

- **QP1.** *Qual é o desempenho da taxa de acerto de um NIDS baseado em aprendizado de máquina em um cenário intra-dataset?*
- **QP2.** *Como um cenário cross-dataset afeta o desempenho de classificação?*

O primeiro experimento investiga a QP1, analisando o desempenho da classificação do modelo selecionado em um cenário de domínio único. Para essa avaliação, o modelo foi treinado individualmente utilizando a partição de treinamento de cada conjunto de dados e posteriormente avaliado na respectiva partição de teste do mesmo domínio. Conforme apresentado na Tabela 1, o modelo computacionalmente exigente apresenta

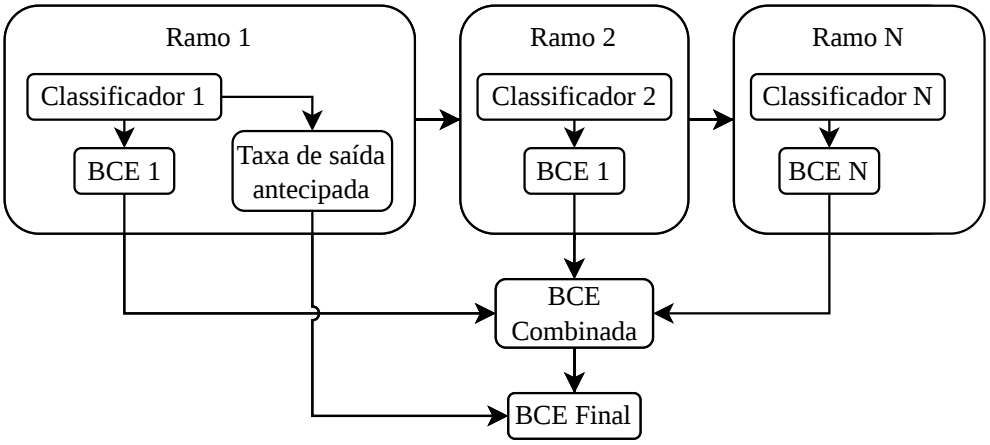


Figura 1. Arquitetura NIDS proposta com generalização *cross-dataset*.

elevado desempenho de detecção quando avaliado em cenários *intra-dataset*. Por exemplo, ao utilizar o conjunto UNSW-NB15 para treinamento e teste, o modelo alcançou uma pontuação F1 de $0,99$, demonstrando alta capacidade de classificação quando as características do ambiente de treinamento e avaliação permanecem consistentes. Esses resultados indicam que, em cenários controlados dentro do mesmo domínio, o aumento da complexidade arquitetônica permite obter ganhos de desempenho, embora tais melhorias sejam limitadas quando não há variação entre os dados de treinamento e teste.

O segundo experimento avalia a QP2, investigando a capacidade de generalização do modelo quando aplicado a conjuntos de dados pertencentes a domínios distintos daqueles utilizados durante o treinamento. Para essa análise, o modelo foi treinado individualmente em cada conjunto de dados e posteriormente avaliado utilizando as partições de teste dos demais domínios, caracterizando um cenário *cross-dataset*. Os resultados apresentados na Tabela 1 evidenciam uma redução significativa no desempenho de classificação quando o modelo é aplicado a ambientes não observados durante o treinamento. Essa degradação é observada pelas menores pontuações de F1 obtidas nas avaliações entre diferentes conjuntos de dados. Por exemplo, o modelo treinado no conjunto UNSW-NB15 apresentou uma pontuação F1 de $0,66$ quando avaliado no BoT-IoT e de $0,54$ no CICIDS2018, indicando uma perda significativa de desempenho em relação ao cenário *intra-dataset*.

4. Estratégia de Saída Antecipada Confiável *Cross-Dataset*

Para abordar os desafios mencionados, este trabalho propõe um novo NIDS, leve e com capacidade de generalização, baseado em aprendizado de máquina e utilizando uma arquitetura com saídas antecipadas. A Figura 1 apresenta a estrutura proposta, organizada em duas etapas principais.

Inicialmente, a inferência é realizada por meio de uma rede neural com saídas antecipadas, permitindo que a classificação de eventos menos complexos seja concluída em camadas intermediárias do modelo. Essa estratégia tem como objetivo reduzir a sobrecarga computacional associada ao processamento de amostras simples, que normalmente representam uma parcela significativa do tráfego de rede, enquanto preserva a capacidade de análise de padrões de ataque mais complexos que demandam maior profundidade de

processamento. Ao eliminar operações desnecessárias durante a inferência, o sistema reduz o custo computacional sem comprometer significativamente o desempenho de detecção. Além disso, a capacidade de generalização do modelo é promovida por meio de uma configuração de desenvolvimento *cross-dataset*. Essa abordagem permite que o modelo seja treinado considerando diferentes domínios de rede, incentivando o aprendizado de características relacionadas ao comportamento das intrusões em vez de padrões específicos presentes em um único conjunto de dados. Dessa forma, busca-se aumentar a robustez do NIDS diante de ambientes heterogêneos e reduzir a degradação de desempenho quando aplicado a cenários distintos daqueles observados durante o treinamento.

4.1. Saídas Antecipadas e Generalização

Este trabalho busca estabelecer um equilíbrio entre desempenho de detecção e custo computacional em NIDS por meio da integração de saídas antecipadas e da avaliação explícita da capacidade de generalização em diferentes conjuntos de dados. A proposta visa manter elevada acurácia de classificação reduzindo a sobrecarga computacional durante o processo de inferência.

Considere x como um vetor de características que representa um fluxo de rede submetido ao processo de classificação pela arquitetura proposta. Para permitir uma inferência mais eficiente, a rede neural é estendida com N saídas intermediárias de classificação, nas quais cada saída i corresponde a um ramo auxiliar conectado a uma camada oculta h_l da rede, conforme apresentado na Figura 1. Cada ramo intermediário gera uma previsão local $\hat{y} = h_i(x)$, permitindo obter uma decisão de classificação antes da propagação do fluxo até a camada final do modelo.

Durante a etapa de inferência, a confiança associada a cada previsão intermediária é comparada a um limiar de confiabilidade previamente definido. Quando a previsão atende ao critério estabelecido, o processamento é encerrado antecipadamente e o resultado fornecido pelo ramo intermediário é utilizado como classificação final. Caso contrário, a amostra continua percorrendo as camadas subsequentes da rede até alcançar a camada de saída final.

Para promover a capacidade de generalização em ambientes de rede heterogêneos, o treinamento do modelo utiliza uma estratégia multidomínio, na qual cada conjunto de dados padronizado é considerado um domínio independente $d \in D$. Durante esse processo, cada ramo de classificação intermediário h_i da arquitetura é otimizado considerando a perda de Entropia Cruzada Binária calculada individualmente para cada domínio presente no conjunto de treinamento, conforme ilustrado na Figura 1.

A perda para um ramo específico i no domínio d é formulada como:

$$\mathcal{L}_{i,d} = -\frac{1}{M}\sum_{j=1}^{M}[y_j \log(\hat{y}_{i,j}) + (1 - y_j)\log(1 - \hat{y}_{i,j})] \tag{1}$$

onde M representa o número de amostras no domínio d e y_j é o rótulo verdadeiro. Para promover o aprendizado de características invariantes ao domínio e evitar que o modelo se ajuste demais a um único ambiente de rede, a perda conjunta para cada ramo h_i do classificador é calculada pela média dessas perdas de domínio individuais:

$$\mathcal{L}_{BCE^i} = \frac{1}{|D|} \sum_{d \in D} \mathcal{L}_{i,d} \tag{2}$$

A função objetivo total para todo o sistema é obtida pela média das perdas conjuntas em todas as N saídas antecipadas e na camada de saída final, da seguinte forma:

$$\mathcal{L}_{combinada} = \frac{\sum_{i=0}^{N} \mathcal{L}_{BCE^i}}{N} \tag{3}$$

Para otimizar a eficiência computacional, nosso objetivo de treinamento incorpora uma taxa alvo de saída antecipada. Definimos uma taxa desejada de eventos, ρ_{alvo}, que devem ser classificados e encerrados no primeiro ramo de saída, para minimizar os custos de inferência do tráfego direto. Durante a fase de treinamento, monitoramos a taxa de saída real, ρ_{atual}, que representa a proporção de amostras de um lote que atendem aos requisitos de confiança da primeira saída antecipada. Para um determinado limite de confiabilidade τ, um evento é considerado como tendo saída prematura se a confiança da previsão $C(\hat{y}_1)$ exceder esse limite τ. A taxa de saída atual é calculada como:

$$\rho_{atual} = \frac{1}{M} \sum_{j=1}^{M} \mathbb{1}(C(\hat{y}_{1,j}) > \tau) \tag{4}$$

onde $\mathbb{1}(\cdot)$ é a função indicadora e M é o tamanho do lote. Para alinhar o comportamento do modelo com o desempenho desejado, introduzimos uma perda de eficiência auxiliar, $\mathcal{L}_{saida}$, definida como a diferença absoluta entre as taxas de saída observadas e desejadas:

$$\mathcal{L}_{saida} = |\rho_{atual} - \rho_{alvo}| \tag{5}$$

Como resultado, ao minimizar a saída, o modelo aprende a priorizar a extração de características de alta confiança no estágio mais inicial possível, abordando diretamente a relação entre a profundidade de representação e os custos computacionais.

A perda final do modelo é calculada da seguinte forma:

$$\mathcal{L} = \frac{\mathcal{L}_{combinada} + \mathcal{L}_{saida}}{2} \tag{6}$$

5. Avaliação

Para avaliar a eficácia da arquitetura proposta, realizamos um segundo conjunto de experimentos projetados para abordar as seguintes questões de pesquisa:

- **QP4.** *Qual é o desempenho computacional e de detecção da nossa abordagem de saída antecipada proposta?*

Tabela 2. Avaliação de desempenho *cross-dataset* de dados em diversas taxas de saída-alvo (ρ_{alvo}), com um limiar de confiança do primeiro ramo de $0,85$.

Saída Alvo (ρ_{alvo})	BoT-IoT				UNSW-NB15				CICIDS2018			
	TVP	**TVN**	**F1**	**Saída Atual**	**TVP**	**TVN**	**F1**	**Saída Atual**	**TVP**	**TVN**	**F1**	**Saída Atual**
Sem Saída	0.95	0.94	0.94	0.00	0.99	0.99	0.99	0.00	0.92	0.98	0.95	0.00
0.1	0.89	0.96	0.92	0.00	0.99	0.97	0.98	0.17	0.91	0.98	0.95	0.00
0.2	0.89	0.96	0.92	0.08	0.99	0.97	0.98	0.27	0.93	0.98	0.96	0.05
0.3	0.89	0.96	0.92	0.31	0.99	0.98	0.99	0.31	0.93	0.98	0.96	0.19
0.4	0.88	0.96	0.92	0.37	0.98	0.97	0.98	0.54	0.93	0.99	0.96	0.26
0.5	0.95	0.94	0.94	0.39	0.99	0.99	0.99	0.72	0.92	0.98	0.95	0.42
Todas as Saídas	0.86	0.96	0.90	1.00	0.98	0.96	0.97	1.00	0.92	0.98	0.95	1.00

5.1. Construção do Modelo

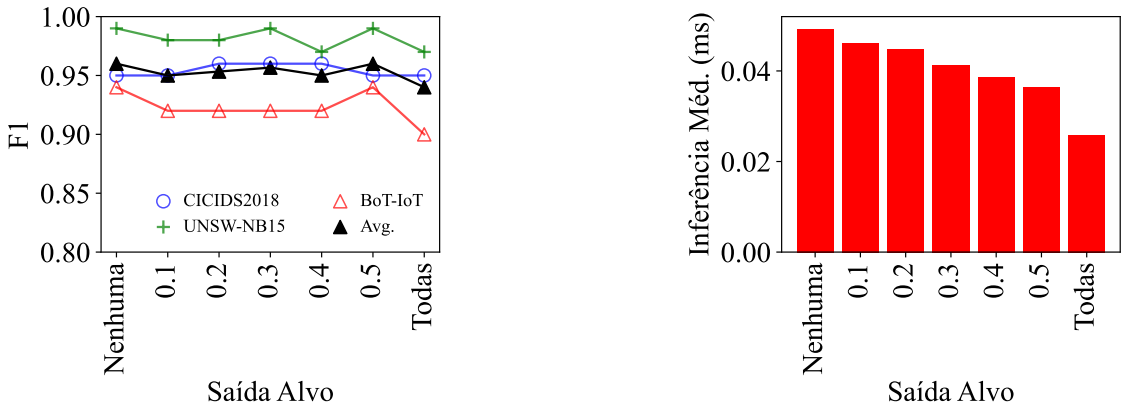
Para implementar a arquitetura proposta, desenvolvemos um modelo hierárquico com uma estratégia de duas saídas em um único pipeline de inferência, conforme descrito a seguir:

- **Primeiro Ramo (Saída Antecipada).** Este ramo implementa a configuração com recursos limitados. A arquitetura é composta por duas camadas ocultas com 88 neurônios cada, ambas utilizando função de ativação ReLU e dropout de $0,2$. A classificação intermediária é realizada por uma camada de saída com 2 neurônios e função de ativação sigmoide. Durante a inferência, caso a confiança da previsão produzida por esse ramo atinja o limite de confiabilidade predefinido, o processamento é encerrado antecipadamente e a classificação intermediária é utilizada como saída final;
- **Segundo Ramo (Saída Final).** Este ramo implementa a configuração Computacionalmente Exigente apresentada na Tabela 1. A entrada desse ramo corresponde à representação latente de 88 neurônios extraída da última camada oculta do primeiro ramo. A partir dessa representação, são adicionadas quatro camadas densas contendo 1024, 2048, 2048 e 1024 neurônios, respectivamente, todas utilizando ativação ReLU e dropout de $0,2$. O classificador final é composto por uma camada de saída com 2 neurônios e função de ativação sigmoide.

Da mesma forma, a arquitetura proposta foi implementada utilizando PyTorch v.2.10. O treinamento foi conduzido durante 500 épocas, empregando uma taxa de aprendizado de 1×10^{-4}, o otimizador Adam e lotes com 128 amostras. Para reduzir o risco de sobreajuste, foi utilizado um mecanismo de parada antecipada, responsável por interromper o treinamento quando a perda de validação não apresentasse melhoria durante 15 épocas consecutivas.

5.2. O Desafio de Domínios de Dados Distintos

O primeiro experimento busca responder à QP4, avaliando o desempenho da arquitetura proposta com mecanismo de saída antecipada em um cenário *cross-dataset*. O objetivo é analisar o equilíbrio entre eficiência computacional e capacidade de detecção, considerando a distribuição dos fluxos de entrada entre os dois ramos da rede e o impacto dessa estratégia no desempenho final de classificação. Para essa avaliação, foi definido um limiar de confiança de $0,85$ para a saída do primeiro ramo (conforme a Eq. 4), correspondente à probabilidade média observada no conjunto de validação. Além disso, a taxa de saída alvo foi variada entre $0,1$ e $0,4$ (conforme a Eq. 5), permitindo analisar como diferentes proporções de classificações antecipadas afetam o comportamento do sistema.



(a) F1-Score **(b)** Custo de Inferência

Figura 2. Comparação de desempenho do nosso método proposto de acordo com a *Saída Alvo* (ρ_{alvo}) utilizada.

Como comparação, foram considerados dois cenários de referência. A configuração Sem Saída desabilita o mecanismo de saída antecipada, direcionando todos os eventos para o ramo de maior capacidade computacional. Por outro lado, a configuração Todas as Saídas força todos os eventos a serem classificados pelo primeiro ramo, permitindo avaliar o impacto de utilizar exclusivamente a arquitetura de menor custo computacional.

A Tabela 2 apresenta os resultados obtidos pela abordagem proposta em um cenário *cross-dataset*. Observa-se que o método mantém elevada capacidade de detecção para diferentes valores de saída alvo, preservando taxas consistentes de classificações realizadas antecipadamente. Considerando uma saída alvo de $0,4$, o modelo alcançou valores de F1 de $0,92$ para BoT-IoT, $0,98$ para UNSW-NB15 e $0,96$ para CICIDS2018. Esses resultados indicam que a estratégia proposta é capaz de reduzir os impactos da variação entre domínios, mantendo desempenho elevado mesmo quando aplicada a conjuntos de dados distintos daqueles utilizados no treinamento.

A Figura 2a apresenta a variação da métrica F1 em função dos diferentes valores de saída alvo. O modelo proposto apresenta comportamento estável entre as configurações avaliadas, evidenciando sua capacidade de generalização em cenários *cross-dataset*. As configurações Sem Saída, com Saída Alvo de $0,4$ e Todas as Saídas obtiveram valores médios de F1 de $0,96$, $0,95$ e $0,93$, respectivamente. Esses resultados demonstram que o mecanismo de saída antecipada permite reduzir o custo computacional da inferência sem provocar degradação significativa na capacidade de classificação.

A Figura 2b apresenta o custo computacional de inferência do modelo proposto considerando diferentes valores de Saída Alvo. Apesar da introdução de uma arquitetura com múltiplos ramos e da necessidade de otimização para diferentes domínios, a abordagem mantém elevada eficiência durante a classificação. Especificamente, ao utilizar uma Saída Alvo de $0,4$, o modelo proposto reduz em 22% o tempo médio de inferência quando comparado à configuração tradicional Sem Saída, na qual todos os eventos são processados pelo ramo de maior capacidade computacional. Esses resultados demonstram que a arquitetura proposta estabelece um equilíbrio adequado entre desempenho de detecção e eficiência computacional.

5.3. Discussão

Os resultados obtidos evidenciam que a utilização de uma arquitetura hierárquica com saídas antecipadas representa uma alternativa eficiente para reduzir o custo computacional de NIDS sem comprometer significativamente a capacidade de detecção em cenários *cross-dataset*. Diferentemente de abordagens tradicionais, nas quais todas as amostras são processadas por modelos de alta complexidade independentemente de sua dificuldade de classificação, a estratégia proposta permite direcionar parte dos fluxos para um ramo de menor custo computacional, mantendo o processamento mais profundo apenas para eventos que exigem maior capacidade de representação. Além disso, a estabilidade dos valores de F1 observada entre diferentes taxas de saída antecipada demonstra que a redução do custo de inferência não ocorre às custas da generalização do modelo. Esses resultados reforçam que a combinação entre treinamento multidomínio e mecanismos de saída antecipada pode ser uma estratégia promissora para o desenvolvimento de NIDS mais eficientes e adequados para ambientes heterogêneos, nos quais restrições computacionais e variações no comportamento do tráfego são fatores críticos para a implantação prática.

6. Conclusão

Este trabalho propôs uma arquitetura de NIDS capaz de manter um desempenho de classificação consistente em diferentes conjuntos de dados, reduzindo simultaneamente o custo associado à inferência. A abordagem combina uma estratégia de treinamento *cross-dataset*, orientada ao aprendizado de características mais independentes do domínio, com um mecanismo dinâmico de saída antecipada para otimizar o processamento dos fluxos de rede. A arquitetura utiliza um ramo inicial de menor complexidade para classificar de forma eficiente padrões de tráfego simples, enquanto amostras que exigem maior capacidade de representação são encaminhadas para um ramo de maior capacidade computacional. Os resultados experimentais demonstram que a estratégia proposta melhora a capacidade de generalização em comparação com abordagens tradicionais, mantendo elevados valores de F1 em diferentes domínios de avaliação e reduzindo significativamente o custo de inferência. Esses resultados evidenciam que a combinação entre treinamento multidomínio e inferência hierárquica é uma alternativa eficiente para o desenvolvimento de NIDS robustos e aplicáveis a ambientes de rede heterogêneos.

Agradecimentos

Este trabalho foi parcialmente financiado pela Fundação Araucária e pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), sob os números de processo 302937/2023-4, 407879/2023-4 e 442262/2024-8.

Referências

Barik, K. and Misra, S. (2025). A comprehensive defense approach of deep learning-based nids against adversarial attacks. *Multimedia Tools and Applications*, 84(31):37745–37791.

Cantone, M., Marrocco, C., and Bria, A. (2024). Machine learning in network intrusion detection: A cross-dataset generalization study. *IEEE Access*, 12:144489–144508.

de Oliveira, V. M., de Oliveira, H. M., Santos, G. M., Geremias, J., and Viegas, E. K. (2025). A big data framework for scalable and cross-dataset capable machine learning in network intrusion detection systems. *IEEE Access*, 13:129419–129431.

Espindola, A. d. S., Casimiro, A., Santin, A. O., Ferreira, P. M., and Viegas, E. K. (2026a). Enhancing intrusion detection generalization via diversity-driven multi-view ensemble learning in industrial systems. *Future Generation Computer Systems*, 182:108458.

Espindola, A. d. S., Santin, A. O., Casimiro, A., Ferreira, P. M., and Viegas, E. K. (2026b). Understanding the adversary: A survey of adversarial machine learning in network intrusion detection. *Computer Science Review*, 62:100995.

Filho, A. G., Viegas, E. K., Santin, A. O., and Geremias, J. (2025). A dynamic network intrusion detection model for infrastructure as code deployed environments. *Journal of Network and Systems Management*, 33(4).

Koroniotis, N., Moustafa, N., Sitnikova, E., and Turnbull, B. (2019). Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems*, 100:779–796.

Lidholm, P., Markovic, T., Leon, M., and Strandberg, P. E. (2024). Network intrusion detection using machine learning on resource-constrained edge devices. In *2024 International Joint Conference on Neural Networks (IJCNN)*, page 1–8. IEEE.

Molina-Coronado, B., Mori, U., Mendiburu, A., and Miguel-Alonso, J. (2020). Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process. *IEEE Transactions on Network and Service Management*, 17(4):2451–2479.

Moustafa, N. and Slay, J. (2015). 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)*, page 1–6. IEEE.

Pawlicki, M., Kozik, R., and Choraś, M. (2023). Improving siamese neural networks with border extraction sampling for the use in real-time network intrusion detection. In *2023 International Joint Conference on Neural Networks (IJCNN)*, page 1–8. IEEE.

Roy, S., Li, J., Choi, B.-J., and Bai, Y. (2022). A lightweight supervised intrusion detection mechanism for iot networks. *Future Generation Computer Systems*, 127:276–285.

Sarhan, M. and Layeghy, S. (2023). Nf-cse-cic-ids2018.

Simioni, J., Viegas, E. K., Santin, A., and Horchulhack, P. (2025a). An early exit deep neural network for fast inference intrusion detection. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*, SAC '25, page 730–737. ACM.

Simioni, J. A., Viegas, E. K., Santin, A. O., and de Matos, E. (2025b). An energy-efficient intrusion detection offloading based on dnn for edge computing. *IEEE Internet of Things Journal*, 12(12):20326–20342.

Wang, C., Xu, D., Li, Z., and Niyato, D. (2024). Effective intrusion detection in highly imbalanced iot networks with lightweight s2cgan-ids. *IEEE Internet of Things Journal*, 11(9):15140–15151.

Zohourian, A., Dadkhah, S., Molyneaux, H., Neto, E. C. P., and Ghorbani, A. A. (2024). Iot-prids: Leveraging packet representations for intrusion detection in iot networks. *Computers & Security*, 146:104034.