Um Método para Detecção Online Não Supervisionada de Ataques DDoS em Fluxos de Dados*

Juliane A. Guaracy Ruas Dutra[1], Fernando Nakayama[2], Michele Nogueira[1]

[1]Departamento de Ciência da Computação - Universidade Federal de Minas Gerais

[2]Departamento de Ciências Exatas e Econômicas ITR - Universidade Federal Rural do Rio de Janeiro

{julianeguaracyruas, michele}@dcc.ufmg.br,nakayama@ufrrj.br

Abstract. *The digital infrastructure faces growing challenges from Distributed Denial of Service (DDoS) attacks, whose detection in data streams demands high accuracy under single-pass and memory-limited constraints. This paper presents MODUS (**M**ethod for **O**nline **D**etection **U**nsupervised **S**treams), an online and unsupervised method for volumetric DDoS attack detection under Data Stream Processing (DSP) constraints, which monitors statistical changes in traffic to incorporate concept drift information into anomaly detection. Evaluation on the CTU-13, Edge-IIoT, and ASEADOS-SDN-IoT datasets demonstrates that MODUS maintains low computational overhead, strictly adhering to the memory and processing constraints inherent to the data stream learning paradigm.*

Resumo. *A infraestrutura digital enfrenta desafios crescentes com ataques de Negação de Serviço Distribuído (DDoS), cuja detecção em fluxos de dados exige alta precisão sob restrições de passagem única e memória limitada. Este artigo apresenta o MODUS (**M**ethod for **O**nline **D**etection **U**nsupervised **S**treams), um método online e não supervisionado para detecção de ataques DDoS volumétricos sob as restrições de Processamento de Fluxo de Dados (DSP), que monitora mudanças estatísticas no tráfego para incorporar informações de mudança de conceito à detecção de anomalias. A avaliação com os conjuntos CTU-13, Edge-IIoT e ASEADOS-SDN-IoT demonstra que o MODUS mantém reduzido custo computacional, aderindo estritamente às restrições de memória e processamento inerentes ao paradigma de aprendizado em fluxos de dados.*

1. Introdução

A geração e a transmissão de dados em alta velocidade constituem a base da infraestrutura digital moderna, introduzindo desafios arquiteturais sem precedentes e expandindo a superfície de ataques cibernéticos [García et al. 2016]. Em meio a esse fluxo de dados, o tráfego de rede atua como o principal vetor para ameaças cibernéticas sofisticadas [Khraisat et al. 2019]. Os ataques de Negação de Serviço Distribuído (do inglês, *Distributed Denial of Service* – DDoS) se destacam como um dos desafios mais críticos na segurança de redes devido à sua natureza evolutiva [Bhatia et al. 2018]. A multidimensionalidade e a escala dessas ameaças registraram crescimento substancial recente;

*Este trabalho foi apoiado pelo CNPq, por meio da bolsa nº 111604/2026-5; pela CAPES, por meio da bolsa nº 88887.980395/2024-00; pela FAPESP, por meio das bolsas nº 2024/09341-0, nº 2025/06753-8 e nº 2025/02406-1; pelo MCTI/FAPESP, por meio da bolsa nº 2018/23098-0; e pelo Instituto Kunumi.

relatórios de ameaças DDoS da Cloudflare indicam que o volume de ataques na camada de rede mais que triplicou em 2025, culminando em incidentes hipervolumétricos que estabeleceram o recorde de 31,4 Tbps no respectivo ano [Cloudflare 2026]. O Brasil figura como um alvo proeminente, registrando 470.677 ataques DDoS no período de julho a dezembro de 2025 [NETSCOUT Systems, Inc. 2026]. À medida que essas ameaças aumentam em frequência e multidimensionalidade, há uma demanda urgente por soluções de detecção que prezem não apenas por acurácia, mas também por melhor velocidade de processamento e menor latência.

Os ataques DDoS volumétricos ameaçam a disponibilidade da rede ao orquestrar fluxos de tráfego em larga escala para exaurir recursos computacionais e de rede [Bhatia et al. 2018]. O caráter volumétrico e dinâmico desses ataques impõe desafios que os métodos tradicionais de detecção não conseguem suprir. Por exemplo, a análise baseada em lotes pressupõe acesso prévio ao conjunto de dados, enquanto modelos supervisionados se tornam progressivamente obsoletos diante do surgimento de ataques *zero-day* [Hindy et al. 2020]. Diferentemente da análise baseada em lotes, os ambientes *online* processam fluxos contínuos de dados em tempo real, sob as restrições de passagem única e memória limitada impostas pelo Processamento de Fluxo de Dados (*Data Stream Processing – DSP*) [Bifet et al. 2018, Cardellini et al. 2022]. Nesse contexto, o aprendizado não supervisionado surge como uma fronteira central de pesquisa [Zhou 2025], focando na modelagem do comportamento base da rede para detectar os desvios anômalos em tempo real, sem a necessidade de dados históricos rotulados.

A detecção de ataques DDoS em fluxos de dados dinâmicos é um desafio de pesquisa rigoroso, caracterizado por um *trade-off* entre a complexidade do modelo e a responsividade adaptativa [Zhou 2025]. As arquiteturas de aprendizado profundo alcançam alta precisão, mas sua natureza estática inviabiliza adaptação contínua em ambientes não estacionários [Shi et al. 2019]. Em contrapartida, os métodos de agrupamento de fluxos *online* viabilizam a eficiência computacional necessária para o processamento em tempo de execução. Porém, suas capacidades adaptativas são frequentemente implícitas, reagindo a alterações na distribuição estatística dos dados sem isolar ou modelar a mudança subjacente que originou o desvio [Scaranti et al. 2022]. Esforços recentes incorporaram a detecção de mudança de conceito, isto é, alterações ao longo do tempo na distribuição estatística dos dados, para lidar com a evolução dos padrões de tráfego [Selvam and Maheswari Balasubramanian 2024], mas as estratégias vigentes dependem de ajustes fracamente supervisionados ou de retreinamento periódico, introduzindo latência de detecção em cenários não supervisionados [Scaranti et al. 2022].

Este artigo apresenta um método *online* e não supervisionado para detecção de ataques DDoS volumétricos em fluxos contínuos de dados de rede, denominado MODUS (**M**ethod for **O**nline **D**etection **U**nsupervised **S**treams). O método opera sob execução em passagem única e memória estritamente limitada [Cardellini et al. 2022], utilizando a mudança de conceito como principal fonte de informação para aprimorar a detecção de ataques DDoS em redes não estacionárias, em vez de empregá-la apenas como gatilho para retreinamento, por meio de duas etapas online adaptativas: a primeira consiste no monitoramento contínuo da estabilidade estatística do fluxo por meio de um detector de mudança de conceito aplicado à métrica de tamanho de pacote; a segunda utiliza o sinal binário gerado por esse monitoramento como atributo adicional para alimentar um

mecanismo de detecção de anomalias, visando identificar padrões dinâmicos de DDoS.

Para avaliar a eficácia do método proposto, foram conduzidos experimentos com capturas de tráfego de rede do conjunto de dados *Czech Technical University dataset* (CTU-13), amplamente reconhecido por sua representação fiel de tráfego DDoS do mundo real intercalado com atividades legítimas de rede [García et al. 2014]. Além disso, foi empregado o conjunto Edge-IIoT, ampliando a diversidade de cenários e padrões de tráfego avaliados, o qual é amplamente utilizado na literatura [Ferrag et al. 2022]. Para garantir a aderência às arquiteturas de rede contemporâneas, o recém-proposto conjunto de dados ASEADOS-SDN-IoT também foi incorporado aos experimentos [Lakshan Yasarathna and Le-Khac 2026]. O ambiente de avaliação emula um fluxo contínuo de dados, processando cada pacote de rede da camada de transporte individualmente sob um paradigma de passagem única, preservando a sequência temporal dos pacotes. Cada pacote é submetido a uma abordagem intercalada de teste e atualização incremental do modelo, na qual o desempenho é avaliado antes de cada pacote ser incorporado ao processo de aprendizado, permitindo quantificar como a estratégia de enriquecimento baseada em mudança de conceito melhora a acurácia de detecção ao longo do tempo. Os resultados demonstram que o método MODUS mantém um custo computacional reduzido, aderindo estritamente às restrições de memória e processamento inerentes ao paradigma de aprendizado em fluxos de dados.

Este artigo prossegue da seguinte forma. A Seção 2 apresenta a literatura relacionada. A Seção 3 descreve o método MODUS. A Seção 4 discute a avaliação e os resultados. Por fim, a Seção 5 reúne as conclusões e as considerações finais.

2. Trabalhos Relacionados

A detecção de ataques DDoS em tráfego de rede tem sido extensivamente estudada, com abordagens que variam do processamento estático à análise dinâmica de fluxos de dados [Zhou 2025]. Uma vertente de pesquisa foca na aplicação de aprendizado de máquina em datasets estáticos, que utiliza *co-clustering* e *Random Forest* para classificar o tráfego [Garg et al. 2021]. Embora eficazes na validação de modelos, essas abordagens em lote são inerentemente incapazes de operar em tempo real, uma vez que exigem que todo o conjunto de dados esteja disponível *a priori*. Para superar essa limitação, a pesquisa também se direcionou para métodos de processamento *online*. O DeepDDoS, um método de detecção baseado em aprendizado profundo projetado para ambientes de processamento de fluxo de dados em larga escala, representa essa transição, utilizando *Long Short-Term Memory* (LSTM), que são arquiteturas recorrentes especializadas na captura de dependências temporais, para modelar pacotes em micro-janelas temporais [Shi et al. 2019]. Contudo, apesar de processar dados sequencialmente, o modelo de detecção do DeepDDoS é estático, não possuindo mecanismos de adaptação caso a distribuição estatística do tráfego mude ao longo do tempo.

Para lidar explicitamente com a mudança de conceito, surgiram abordagens adaptativas. O UASDAC propõe um *pipeline* que utiliza um detector de mudança não supervisionado para identificar mudanças na distribuição dos dados [Selvam and Maheswari Balasubramanian 2024]. Ao detectar uma mudança, o sistema aciona um retreinamento do classificador para se adaptar ao novo conceito [Selvam and Maheswari Balasubramanian 2024]. A desvantagem central desta

abordagem é o retreinamento, especialmente em fluxos maciços de IoT, que introduz latência computacional significativa, deixando o sistema vulnerável a novos ataques. Buscando contornar essa latência, outras soluções exploraram formas de tornar o modelo inerentemente robusto a mudanças. O ReCDA propõe uma estratégia auto-supervisionada de adaptação à mudança de conceito com aprimoramento de representação [Yang et al. 2025]. Este método utiliza a perda contrastiva e a perturbação de características para que o modelo aprenda representações de tráfego invariantes às mudanças, mantendo a capacidade de classificação mesmo diante de variações na distribuição, porém foca em tornar o modelo resistente às mudanças em vez de explorá-las como fonte de informação, além de se basear em técnicas auto-supervisionadas que diferem fundamentalmente de uma abordagem não supervisionada [Yang et al. 2025].

A literatura, portanto, demonstra uma clara evolução de métodos estáticos em lote para métodos *online* estáticos e, finalmente, para abordagens complexas cientes de mudanças [Garg et al. 2021, Shi et al. 2019]. Contudo, as soluções atuais enfrentam limitações, seja pela alta latência de estratégias reativas de retreinamento, seja pela complexidade de aprender representações invariantes [Selvam and Maheswari Balasubramanian 2024, Yang et al. 2025]. Nesse contexto, este trabalho propõe um novo método *online* e não supervisionado. Ao invés de reagir às mudanças com retreinamento ou tentar ignorá-las, o método apresentado trata a mudança de conceito como um indicador-chave, utilizando-a para informar diretamente o detector de anomalias e, assim, aprimorar a detecção de ataques.

3. Método

Esta seção apresenta o método MODUS para a detecção *online* e não supervisionada de ataques DDoS volumétricos em fluxos contínuos de dados de rede. O método fundamenta-se no paradigma de aprendizado em fluxo de dados, no qual a unidade básica de processamento é a instância, que designa um único ponto de dados no fluxo. No contexto deste trabalho, cada instância corresponde diretamente a um pacote de rede individual, capturado e processado no momento de sua chegada. Dessa forma, os termos instância e pacote são utilizados de maneira intercambiável ao longo do trabalho.

O objetivo central consiste na identificação de ataques DDoS volumétricos por meio da análise incremental de metadados do tráfego de rede, sem a necessidade de rótulos prévios. Os eventos de mudança de conceito refletem alterações abruptas no comportamento estatístico do tráfego. Em contraste com abordagens tradicionais que empregam tais eventos apenas como mecanismos de detecção de deriva para descarte e retreinamento de modelos, o método proposto incorpora explicitamente a mudança de conceito como variável preditiva. Essa informação é integrada ao processo de detecção, permitindo ao modelo capturar diretamente instabilidades estatísticas associadas aos pacotes monitorados. Além disso, o método opera sob a restrição de passagem única, na qual cada pacote é processado uma única vez e descartado após a atualização do modelo, contribuindo para a redução do consumo de memória e viabilizando a aplicação em cenários de alto volume e elevada taxa de chegada de dados. O método é estruturado em três fases principais, conforme ilustrado na Figura 1: (a) preparação do fluxo de dados e engenharia de atributos; (b) enriquecimento de atributos e monitoramento de mudança de conceito; e (c) detecção de ataques. As subseções seguintes apresentam a descrição detalhada de cada uma dessas fases.

3.1. Preparação do Fluxo de Dados e Engenharia de Atributos

A entrada do método consiste em um conjunto de dados estático contendo instâncias em nível de pacote. A Fase (a) compreende quatro ações principais: $(a.1)$ ordenação temporal dos registros, $(a.2)$ seleção e a codificação de características comportamentais, $(a.3)$ transformação vetorial dos atributos e $(a.4)$ encapsulamento em um fluxo contínuo, etapa final que consolida a transição dos dados em fluxos contínuos para o ambiente de processamento dinâmico.

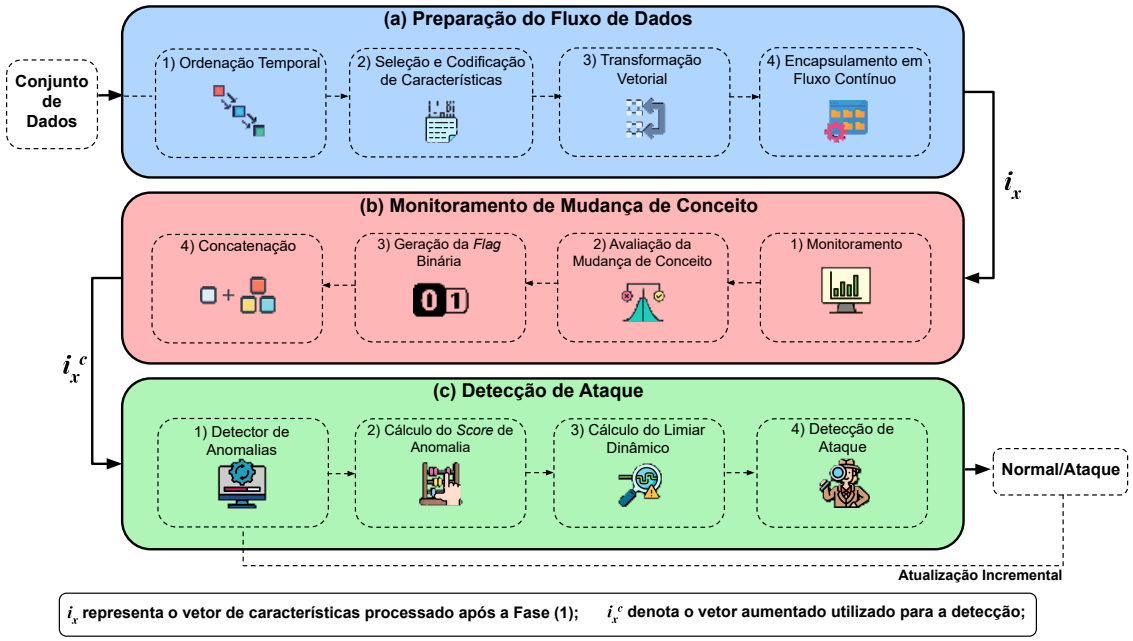

Figura 1. Fases de execução do método proposto

Na ação $(a.1)$, o método assegura a integridade causal dos dados ao ordenar estritamente os pacotes com base na marcação temporal. No contexto de Processamento em Fluxo de Dados, diferentemente de abordagens em lote que reorganizam as amostras de forma arbitrária, essa ordenação é essencial para simular fielmente a chegada sequencial dos pacotes. Esse rigor preserva as dependências temporais e a evolução não estacionária da rede, características típicas de ataques DDoS volumétricos, que operam por variações abruptas e concentradas no tempo.

A ação $(a.2)$ consiste na seleção e a codificação estrutural dos atributos, visando maximizar a capacidade de generalização do modelo. Em consonância com a literatura, identificadores estáticos de topologia, como endereços IP, são deliberadamente descartados. O escopo preditivo abrange apenas características estritamente comportamentais da comunicação: portas de origem e destino (submetidas à imputação de zeros na ausência de valor), tamanho do pacote e protocolo. Essa supressão impede que o modelo memorize a infraestrutura de rede monitorada [Sommer and Paxson 2010]. Em seguida, é realizada a codificação do protocolo, convertendo seu valor categórico original para uma representação numérica discreta. Assim, o trabalho adota a tupla fundamental de cabeçalho composta por portas de origem e destino, tamanho do pacote e protocolo. Embora dimensionalmente reduzida, essa seleção constitui um padrão amplamente empregado para representar propriedades em dados baseados em fluxo [Ring et al. 2019].

A ação $(a.3)$ converte as instâncias de rede em vetores numéricos, descartando os metadados tabulares para viabilizar o processamento algébrico. Possuindo os atributos selecionados e codificados na etapa anterior, cada pacote assume a forma de um vetor cujas dimensões abrigam estritamente as portas de origem e destino, o comprimento do pacote e o protocolo codificado. Essa abstração consolida o formato matemático nativo exigido para o cálculo de distâncias e partições espaciais pelos algoritmos subsequentes.

Por fim, a ação $(a.4)$ estabelece a transição do paradigma de armazenamento estático para a simulação de operação em fluxo contínuo, isto é, para o processamento *online*. Os vetores numéricos resultantes da etapa anterior são encapsulados em um objeto iterável sequencial, o que revoga o acesso global à matriz de dados e impõe uma restrição estrita de processamento unidirecional. A partir desse ponto, o método passa a emular restrições de memória e latência, fornecendo apenas uma instância isolada por ciclo computacional, denotado por i_x. Esse encapsulamento, no contexto do método proposto, é fundamental para garantir que os algoritmos de detecção e os monitores de mudança de conceito operem em passagem única, processando cada instância apenas uma vez, sem acesso aleatório à memória ou visibilidade de instâncias futuras.

3.2. Monitoramento de Mudança de Conceito

A Fase (b) é responsável por monitorar e ampliar o espaço de atributos do fluxo de dados com informações relativas à estabilidade estatística do tráfego. Esta fase recebe como entrada o vetor de características gerado anteriormente, com foco específico na métrica de tamanho do pacote, e contempla quatro ações sequenciais: $(b.1)$ monitoramento estatístico adaptativo, $(b.2)$ avaliação da mudança de conceito, $(b.3)$ geração da *flag* binária e $(b.4)$ concatenação de atributos.

Na ação $(b.1)$, o monitoramento estatístico adaptativo é instanciado por meio da adoção do algoritmo ADWIN (*Adaptive Windowing*) [Bifet and Gavaldà 2007]. A escolha por janelas de tamanho variável, em detrimento de janelas fixas ou de acumulação total, é justificada pela necessidade de o método equilibrar a capacidade de reagir rapidamente a novos padrões com a resistência a ruídos momentâneos. O funcionamento do ADWIN se baseia na manutenção de janelas de observação cujo tamanho diminui quando a variância dos dados aumenta e cresce quando os dados permanecem estáveis. Nessa ação, o mecanismo é configurado para monitorar, de forma univariada, a métrica de tamanho do pacote.

A adoção deste atributo se baseia na assinatura observada em certas categorias de ataques DDoS volumétricos, especialmente *UDP floods* e os ataques de amplificação, nos quais o tráfego malicioso apresenta distribuições de tamanho de pacote anômalas ou estreitamente concentradas [Rossow 2014]. Em ataques orientados à maximização da quantidade de pacotes processados por segundo, independentemente do volume de dados, predominam os pacotes de tamanho mínimo. Já nos ataques de amplificação, o tamanho fixo é determinado pela resposta do protocolo explorado. Essa concentração anômala pode deslocar a distribuição do comprimento de pacote observada na rede, diferenciando-a do tráfego legítimo. A técnica mantém uma janela de observação de tamanho variável e avalia, a cada nova instância, a existência de subjanelas adjacentes cujas médias diferem de forma estatisticamente significativa. Quando essa divergência excede um limiar de confiança estatística, o sistema infere que os dados recentes provêm de uma distribuição

distinta da observada anteriormente, acionando um alerta interno. Essa abordagem viabiliza a identificação de mudanças abruptas no comportamento do tráfego e de padrões anômalos associados a variações volumétricas, sem depender de limiares estáticos pré-definidos.

A ação $(b.2)$ realiza a avaliação dessa divergência comportamental por meio de um teste estatístico fundamentado no limite de Hoeffding, mecanismo decisório central do algoritmo ADWIN [Bifet and Gavaldà 2007], responsável por avaliar continuamente a validade da hipótese de que o valor esperado do tamanho dos pacotes permaneceu constante ao longo da janela de observação W. A cada nova instância processada, o algoritmo itera sobre os possíveis pontos de corte da janela W, avaliando múltiplas partições em duas subjanelas adjacentes: uma porção mais antiga, denotada por W_0, de tamanho n_0, e uma porção mais recente, denotada por W_1, de tamanho n_1. A avaliação atesta a ocorrência de uma mudança de conceito ao rejeitar a hipótese nula se, para qualquer uma dessas partições, a diferença absoluta entre as médias observadas nas subjanelas $\hat{\mu}_{W_0}$ e $\hat{\mu}_{W_1}$ superar o limite de corte ϵ_{cut}, formalmente definido pela equação:

$$\epsilon_{cut} = \sqrt{\frac{1}{2m} \ln\left(\frac{4n}{\delta}\right)} \tag{1}$$

Nessa formulação, n representa o número total de instâncias na janela, dado por $n = n_0 + n_1$, enquanto m denota a média harmônica dos comprimentos das partições avaliadas, calculada como $m = (1/n_0 + 1/n_1)^{-1}$. O parâmetro δ corresponde ao nível de confiança estatística. No método proposto, é adotado $\delta = 0.001$. Essa configuração eleva o limiar ϵ_{cut}, exigindo evidências probabilísticas robustas antes de avaliar uma ruptura estatística, o que reduz a probabilidade de que flutuações benignas sejam erroneamente interpretadas como anomalias. Uma vez inferido o estado probabilístico da janela por meio da regra de decisão $|\hat{\mu}_{W_0} - \hat{\mu}_{W_1}| \geq \epsilon_{cut}$, ocorre a conversão desse estado em atributo. A Figura 2 ilustra o funcionamento do ADWIN.

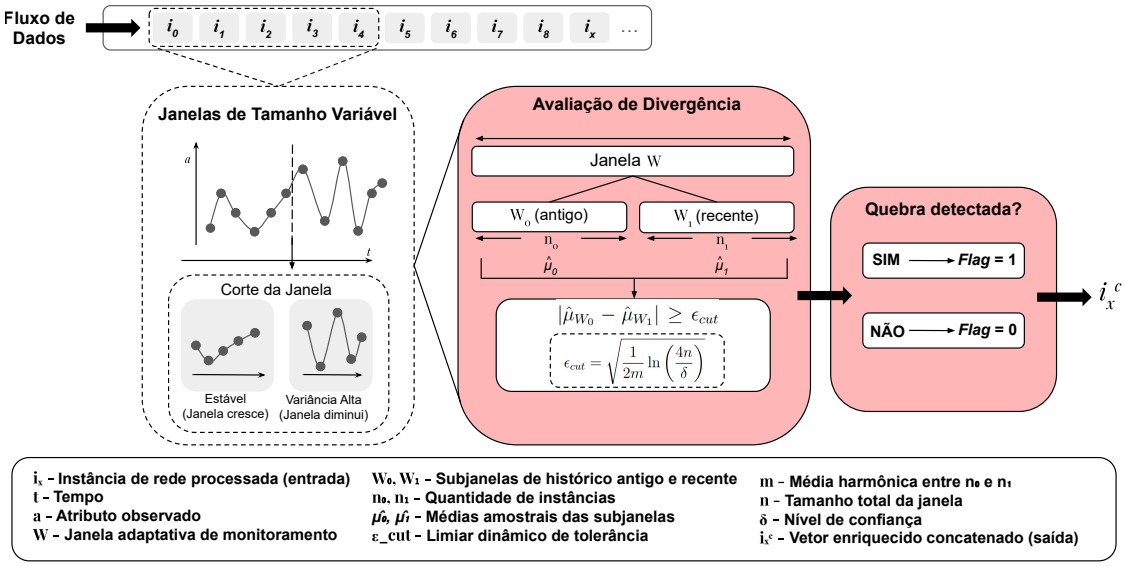

Figura 2. Monitoramento de mudança de conceito

A ação $(b.3)$ consiste na tradução do estado do monitor estatístico em uma variável binária interpretável. O processo gera uma *flag* de mudança de conceito que assume valor "1" quando uma quebra de estacionariedade é detectada no instante atual e "0" caso contrário. Essa binarização converte um evento estatístico em um sinal compatível com o vetor de características. Por fim, a ação $(b.4)$ realiza a concatenação dos atributos para a formação do vetor aumentado i_x^c. O sinal binário de mudança de conceito é anexado ao vetor original proveniente da Fase (a). Essa operação constitui o núcleo da contribuição proposta, pois transforma a detecção de mudança de conceito, tradicionalmente utilizada apenas como gatilho para a adaptação de modelos, em uma variável preditiva que informa diretamente o detector sobre a instabilidade do ambiente.

3.3. Detecção de Ataque

A Fase (c) é responsável pela detecção não supervisionada de ataques DDoS volumétricos, encerrando o ciclo de processamento. Esta fase recebe como entrada o vetor de características aumentado i_x^c, que consolida os atributos comportamentais e o sinal de instabilidade estatística. Sua execução contempla quatro ações sequenciais e um laço de atualização incremental: $(c.1)$ detector de anomalias, $(c.2)$ cálculo do *score* de anomalia, $(c.3)$ cálculo do limiar dinâmico e $(c.4)$ detecção de ataque.

A ação $(c.1)$ inicia com a submissão do vetor aumentado i_x^c ao algoritmo de detecção de anomalias HST *(Half-Space Trees)* [Tan et al. 2017]. A escolha desse algoritmo decorre de sua eficiência computacional em fluxos de alta velocidade e de sua capacidade de adaptação contínua. O detector constrói um conjunto de estruturas hierárquicas que particionam recursivamente o espaço vetorial em sub-regiões. Diferentemente de árvores de decisão tradicionais guiadas por ganho de informação, o HST particiona o espaço recursivamente na construção de cada árvore e reatualiza essa estrutura de forma iterativa a cada nova janela do fluxo, escolhendo atributos e valores de corte de forma estritamente aleatória, mitigando o custo computacional no fluxo de dados. Nesse contexto, a *flag* de mudança de conceito introduzida na Fase (b) atua como uma dimensão discriminante, favorecendo a separação de instâncias associadas às instabilidades do tráfego.

A ação $(c.2)$ consiste em quantificar o desvio comportamental do tráfego por meio do cálculo do *score* de anomalia. O vetor i_x^c percorre as partições hierárquicas de cada árvore do *ensemble* até ser confinado em um nó folha. O algoritmo HST computa, então, um *score* contínuo $S_t \in [0, 1]$ para a instância, fundamentado na massa do nó em que ela foi isolada. As instâncias pertencentes a ataques DDoS volumétricos tendem a ser projetadas em regiões esparsas do espaço dimensional. Por consequência, sofrem isolamento geométrico e recebem *scores* de anomalia elevados.

A ação $(c.3)$ realiza a comparação entre o *score* de anomalia calculado na ação anterior e o comportamento recente do fluxo de dados, utilizando um limiar dinâmico adaptativo τ_t baseado no MAD (*Median Absolute Deviation*) [Iglewicz and Hoaglin 1993]. Para que o cálculo estatístico possua validade, é imposta uma exigência populacional mínima local N_{min}. Durante a coleta dessas primeiras instâncias, o modelo utiliza um piso de ruído estático τ_{min}.

$$\tau_t = \max\Big(\tau_{min},\ \text{mediana}(\mathcal{S}) + \alpha \cdot \text{MAD}\Big) \tag{2}$$

Nesta formulação, o α é o fator de sensibilidade, definido como hiperparâmetro fixo do modelo. O MAD é calculado pela mediana das diferenças absolutas em relação à mediana central:

$$\text{MAD} = \text{mediana}(|S_i - \text{mediana}(\mathcal{S})|) \quad \forall S_i \in \mathcal{S} \tag{3}$$

A escolha do MAD decorre de sua resistência a valores atípicos, tornando o recálculo do limiar robusto a distorções causadas por *scores* anômalos [Romo-Chavero et al. 2024]. A inclusão do parâmetro limitador τ_{min} previne o colapso do limiar para zero durante períodos de tráfego estritamente homogêneo ou silêncio na rede. Sem esse piso de ruído, a variância da janela despencaria, tornando o sistema sensível e gerando cascatas de falsos positivos frente a flutuações milimétricas e benignas. Essa formulação estabelece que o modelo opere sob o paradigma de detecção por exclusão. Ao retroalimentar a janela de cálculo $\mathcal{S}$ exclusivamente com instâncias benignas, se constrói um perfil contínuo da normalidade da rede. O modelo busca reconhecer qualquer instância interceptada cujo *score* estocástico ultrapasse a fronteira superior de variabilidade traçada por τ_t. Nesse caso, o evento é interpretado como uma ruptura da normalidade rotineira, sendo classificado sumariamente como uma anomalia.

A ação $(c.4)$ consolida a etapa preditiva e gerencia o aprendizado contínuo do modelo fundamentado na abordagem *Test-Then-Train*, na qual cada nova instância interceptada é avaliada pelo HST no exato instante de sua chegada para, apenas subsequentemente, ser habilitada para a etapa de treinamento e atualização estrutural. Entretanto, antes de habilitar essa regra de decisão, é imposto um período de aquecimento B, fixado em 50.000 instâncias. Esse valor foi estabelecido de forma a garantir que o *ensemble* de árvores tenha processado volume amostral suficiente de tráfego exclusivamente benigno antes de iniciar a emissão de predições, assegurando que a geometria interna das partições reflita o perfil legítimo da rede. Durante essa fase inicial, o algoritmo suspende a emissão de predições e processa os dados de forma incondicional, alocando todos os pacotes recebidos diretamente na etapa de treinamento. Essa fase preparatória cumpre o papel de mapeamento topológico e garante o preenchimento de massa probabilística nos nós folha do *ensemble* de árvores, fornecendo o volume amostral necessário para a estabilização assintótica da mediana na janela $\mathcal{S}$ inicial.

Encerrado o aquecimento estrutural, o método transita definitivamente para a abordagem *Test-Then-Train*. A etapa de teste opera comparando o *score* calculado com o limiar dinâmico vigente. Se o *score* avaliado ultrapassar essa fronteira de tolerância $S_t > \tau_{t-1}$, a instância recebe o rótulo preditivo $\hat{y}_t = 1$ (ataque). Caso contrário, recebe o rótulo $\hat{y}_t = 0$ (tráfego normal). Após a emissão do rótulo, é executada a etapa condicional de treinamento. As instâncias classificadas como ataque são descartadas do ciclo de atualização, tendo o seu processamento interrompido. Esse mecanismo de atualização seletiva auxilia o HST a não assimilar gradualmente as assinaturas estatísticas do ataque como parte do perfil normal da rede, elevando o limiar dinâmico e tornando o detector progressivamente cego à ameaça, fenômeno conhecido como envenenamento do modelo. Ao descartar vetores maliciosos da fila de atualização, o mecanismo impede que as partições espaciais das árvores e a janela histórica de cálculo do MAD absorvam essas assinaturas, preservando a fronteira de decisão construída sobre o tráfego legítimo. Por fim, apenas os pacotes avaliados como benignos concluem o ciclo *Test-Then-Train*, retroalimentando as

estruturas internas e assegurando que a evolução do modelo seja pautada pela legitimidade do tráfego de rede. O Algoritmo 1 sintetiza o fluxo proposto na Fase (c).

Algorithm 1 Fluxo de Procedimentos do Método

Require: Fluxo contínuo $\mathcal{F}$, parâmetros do HST, do MAD e do ADWIN, período de aquecimento B
Ensure: Rótulo predito $\hat{y}_t \in \{0 = \text{benigno},\ 1 = \text{ataque}\}$ para cada instância
Inicializar HST, ADWIN, janela $\mathcal{S} \leftarrow \emptyset$, limiar $\tau_t \leftarrow \tau_{min}$, contador $t \leftarrow 0$
while $\mathcal{F}$ possui instâncias **do**
    Obter próxima instância i_x de $\mathcal{F}$
    Calcular sinal de mudança de conceito via ADWIN sobre i_x
    Construir vetor aumentado $i_x^c \leftarrow [i_x \parallel \textit{sinal\_mudança}]$
    Calcular pontuação de anomalia $S_t \leftarrow \text{HST}(i_x^c)$
    if $|\mathcal{S}| \geq N_{min}$ **then** ▷ Atualizar limiar dinâmico via MAD
        $\tau_t \leftarrow \max\big(\tau_{min},\ \text{mediana}(\mathcal{S}) + \alpha \cdot \text{MAD}(\mathcal{S})\big)$
    end if
    if $t < B$ **then** ▷ Fase de aquecimento: treinar incondicionalmente
        $\hat{y}_t \leftarrow 0$
        Treinar HST com i_x^c; Inserir S_t em $\mathcal{S}$ e remover o elemento mais antigo se $|\mathcal{S}| > W_{max}$
    else if $S_t > \tau_t$ **then** ▷ Anomalia detectada
        $\hat{y}_t \leftarrow 1$
    else ▷ Tráfego normal: atualizar modelo e janela
        $\hat{y}_t \leftarrow 0$
        Treinar HST com i_x^c; adicionar S_t à janela $\mathcal{S}$
    end if
    $t \leftarrow t + 1$
end while

4. Avaliação

Esta seção descreve a estrutura de avaliação adotada para verificar a eficácia do método MODUS na detecção *online* e não supervisionada de ataques DDoS volumétricos em fluxos contínuos de tráfego de rede. O objetivo central consiste em investigar se o monitoramento de mudança de conceito contribui para a adaptação do modelo ao longo do tempo na detecção de ataques. A avaliação foi conduzida respeitando as restrições inerentes ao aprendizado em fluxos de dados, que incluem o processamento incremental, a passagem única sobre os dados e a preservação da ordem temporal das instâncias.

A avaliação do método foi estruturada com base em três conjuntos de dados: CTU-13 (cenário 11), Edge-IIoT e ASEADOS-SDN-IoT. O CTU-13 é um conjunto de dados de tráfego de rede rotulado, capturado em ambiente real na Universidade CTU (*Czech Technical University*), na República Tcheca, em 2011, composto por treze cenários distintos. O cenário 11 corresponde ao botnet Rbot, caracterizado pela realização de ataques DDoS e pela varredura de portas via protocolo IRC. O Edge-IIoT é um conjunto de dados realista de cibersegurança voltado à IoT e à IIoT. Abrange catorze ataques categorizados em cinco classes de ameaças, incluindo ataques DoS/DDoS por inundação nos protocolos HTTP, ICMP, TCP e UDP, tornando-o representativo da diversidade de vetores de ataque em ambientes IoT modernos. O ASEADOS-SDN-IoT é um conjunto de dados público e

recentemente proposto para avaliação em ambientes que integram SDN e IoT, cobrindo quatro categorias de ataque: DoS, DDoS, probe e botnet.

Os conjuntos CTU-13 e Edge-IIoT foram selecionados por sua ampla adoção na literatura, com a inclusão do recentemente proposto ASEADOS-SDN-IoT para contemplar cenários de ataques DDoS contemporâneos. Vale destacar que o Edge-IIoT contém múltiplos tipos de ataque; portanto, foi realizada uma filtragem para reter apenas o tráfego benigno e os ataques DDoS. O mesmo procedimento foi aplicado ao ASEADOS-SDN-IoT, que inclui, além do DDoS, categorias como DoS, bot e probe. A Tabela 1 apresenta a composição dos conjuntos de dados, detalhando os tipos de tráfego, suas respectivas proporções e as partições utilizadas na avaliação.

O ambiente de avaliação emula um fluxo contínuo de dados, processando cada instância individualmente sob o paradigma de passagem única e preservando a sequência temporal dos eventos. Independentemente de os conjuntos de dados estarem organizados em nível de pacote ou de fluxo, a Fase (a) converte os pacotes em fluxos de dados compatíveis com o CapyMOA, biblioteca voltada ao aprendizado de máquina em fluxos de dados com suporte nativo ao DSP. A partir dessa representação unificada, o CapyMOA orquestra o método, submetendo cada instância às fases de detecção e preservando as características temporais do tráfego de rede. As Fases (b) e (c) do método são executadas integralmente na biblioteca, desenvolvida em *Python*, da qual fazem parte também os algoritmos empregados nessas etapas: o ADWIN (*Adaptive Windowing*), utilizado na detecção de mudança de conceito, e o HST *(Half-Space Trees)*, utilizado na detecção de anomalias.

Tabela 1. Tipos de tráfego dos conjuntos de dados empregados nas avaliações

Conjunto de Dados	Tipo de Tráfego	Porcentagem (%)	Utilizado
CTU-13 (C11)	Benigno	32,03	✓
	DDoS	67,97	✓
Edge-IIoT	Tráfego Normal	93,157	✓
	DDoS	0,948	✓
	Ataques de senha	4,360	–
	Injeção	0,212	–
	Upload malicioso	0,156	–
	Scanner de vuln.	1,100	–
	Ataques XSS	0,067	–
	Outros	0,000	–
ASEADOS-SDN-IoT	Benigno	57,06	✓
	DoS	27,96	–
	DDoS	11,26	✓
	Bot	1,99	–
	Probe	1,73	–

Para quantificar o desempenho do método proposto no contexto de fluxos contínuos, a avaliação utiliza as métricas de Acurácia, Precisão, Revocação (*Recall*) e *F1-Score*, extraídas de forma incremental a partir da matriz de confusão, que acumula os Verdadeiros Positivos (VP), Verdadeiros Negativos (VN), Falsos Positivos (FP) e Falsos Negativos (FN) durante a etapa de teste da abordagem *Test-Then-Train*. A

acurácia (Equação 4) expressa a proporção global de classificações corretas. A precisão (Equação 5) indica a confiabilidade dos alertas emitidos, ou seja, a fração das instâncias sinalizadas como ataque que de fato o eram. A revocação (Equação 6), por sua vez, mede a capacidade do detector em identificar os ataques reais, refletindo quantos deles foram efetivamente capturados. Como em cenários de detecção esses dois aspectos frequentemente se opõem, é adotado o *F1-Score* (Equação 7), que os combina por meio da média harmônica em uma única métrica robusta ao desbalanceamento entre as classes.

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \quad (4) \qquad \text{Precisão} = \frac{VP}{VP + FP} \quad (5)$$

$$\text{Recall} = \frac{VP}{VP + FN} \quad (6) \qquad \text{F1-Score} = 2 \times \frac{\text{Prec.} \times \text{Recall}}{\text{Prec.} + \text{Recall}} \quad (7)$$

4.1. Experimentos

Os experimentos foram conduzidos de forma distinta para cada conjunto de dados, respeitando o nível de granularidade em que cada um se encontrava originalmente. O CTU-13 (cenário 11) e o Edge-IIoT estavam organizados em nível de pacote, exigindo a execução da Fase (a) do método para o encapsulamento em fluxo antes do processamento. O ASEADOS-SDN-IoT, por sua vez, já se encontrava em nível de fluxo, tornando essa etapa dispensável. Essa distinção não implica qualquer alteração no método proposto; ao contrário, evidencia sua flexibilidade para operar sobre diferentes granularidades de dados de rede. Para o CTU-13 e o Edge-IIoT, as características extraídas após o encapsulamento foram: porta de origem, porta de destino, tamanho do pacote e protocolo codificado. Para o ASEADOS-SDN-IoT, as características utilizadas em nível de fluxo foram: porta de origem, porta de destino, protocolo e tamanho médio dos pacotes do fluxo. No caso do ASEADOS-SDN-IoT, as características foram normalizadas via Min-Max para o intervalo $[0, 1]$, técnica que reescala cada atributo proporcionalmente aos seus valores mínimo e máximo, adequada à geometria esperada pelo HST.

Em todos os experimentos, os dados foram processados sob o paradigma de passagem única, preservando a ordem temporal das instâncias. O período de aquecimento foi fixado em 50.000 instâncias em todos os conjuntos, durante o qual o modelo é treinado exclusivamente com tráfego benigno, sem emissão de predições, permitindo a estabilização da geometria interna do HST antes da fase operacional. O detector HST foi configurado com parâmetros específicos para cada conjunto de dados. Para o CTU-13 e o Edge-IIoT, foram adotadas a janela de 100.000 instâncias, 80 árvores e profundidade máxima de 15. Para o ASEADOS-SDN-IoT, a janela de 2.500 instâncias, 150 árvores e profundidade máxima de 12. Em todos os casos, o detector ADWIN foi configurado com $\delta = 0.001$ e monitorou a métrica de comprimento de pacote/fluxo, por ser a característica mais sensível a variações volumétricas típicas de ataques DDoS.

O limiar de detecção foi calculado dinamicamente em todos os experimentos por meio do estimador MAD, atualizado a cada instância classificada como normal, sobre uma janela deslizante de 5.000 pontuações. Para o CTU-13 e o Edge-IIoT, o limiar foi definido como $\tau_t = \text{mediana} + 6.0 \times \text{MAD}$, com piso mínimo de 0.40. Para o ASEADOS-SDN-IoT, é adotado $k = 8.0$, piso de 0.75 e teto de 0.95, refletindo a maior concentração de *scores* no intervalo superior decorrente da normalização Min-Max aplicada. A atualização do

modelo, em que apenas instâncias classificadas como normais retroalimentam o treinamento, foi aplicada em todos os experimentos como proteção contra o envenenamento do modelo por instâncias maliciosas.

4.2. Resultados e Discussão

Os resultados obtidos nos três conjuntos de dados são apresentados na Tabela 2, que consolida as métricas de acurácia, precisão, revocação, *F1-Score* e tempo total de processamento para cada experimento. As matrizes de confusão detalhadas se encontram na Tabela 3. O ASEADOS-SDN-IoT apresentou o melhor resultado do estudo, atingindo acurácia de 99,87%, revocação de 100% e *F1-Score* de 99,62%, sem incidência de falsos negativos. Esse resultado reflete a combinação entre o pré-processamento e a parametrização do modelo, na qual a normalização Min-Max confinou o espaço de características ao intervalo $[0, 1]$, favorecendo um particionamento mais granular pelo HST, configurado com 150 árvores de profundidade 12. Nesse contexto, as instâncias de DDoS, caracterizadas por valores extremos no comprimento médio do fluxo, tenderam a ser isoladas em partições de baixa massa, gerando *scores* de anomalia consistentemente elevados frente ao limiar ajustado $k = 8.0$. Vale destacar que o ASEADOS-SDN-IoT já se encontrava organizado em nível de fluxo, o que eliminou a necessidade da Fase (a) de encapsulamento.

Tabela 2. Resultados dos experimentos por conjunto de dados

Métrica	CTU-13(C11)	Edge-IIoT	ASEADOS-SDN-IoT
Instâncias	6.336.394	30.919.961	312.262
Tempo (s)	2895.26s	14893.37s	189.02s
Acurácia (%)	87,25	90,87	99,87
Precisão (%)	99,67	91,44	99,25
Revocação (%)	72,28	73,16	100,00
F1-Score (%)	83,79	81,24	99,62

O CTU-13 (cenário 11) apresentou precisão de 99,67%, porém revocação de 72,28%, resultando em *F1-Score* de 83,79%. A incidência de 800.954 falsos negativos decorre de dois fatores combinados. O primeiro se relaciona à característica do próprio conjunto: o CTU-13 contém uma terceira classe massiva denominada *Background*, correspondente ao ruído real da rede universitária capturado durante os experimentos. Esse tráfego contém escaneamentos de porta, anomalias não rotuladas, artefatos de rede e, conforme documentado pelos autores do conjunto, pode conter tráfego de ambos os tipos [García et al. 2014]. Ao retroalimentar a janela histórica do MAD durante o aquecimento, esse ruído infla o limiar dinâmico e reduz a separabilidade entre as classes na fronteira de decisão do HST. O segundo fator é relacionado ao ataque documentado no cenário 11: diferentemente de ataques volumétricos, o Rbot é uma botnet que, além de executar ataques DDoS, mantém canais de Comando e Controle, mecanismo pelo qual o atacante envia instruções remotas aos dispositivos infectados utilizando o protocolo IRC, originalmente projetado para comunicação em texto entre usuários [García et al. 2014]. Esse canal de controle gera tráfego estatisticamente próximo ao tráfego benigno, o que não produz deslocamento suficiente no comprimento de pacote para acionar o ADWIN de forma consistente, dificultando a detecção desse subconjunto do ataque. Ainda assim,

a precisão próxima de 100% indica que os alertas emitidos foram confiáveis, com apenas 6.796 falsos positivos em mais de 6 milhões de instâncias processadas.

Tabela 3. Matrizes de confusão por conjunto de dados

Dataset	VP	VN	FP	FN
CTU-13(C11)	2.088.551	3.440.093	6.796	800.954
Edge-IIoT	6.130.307	21.966.507	574.072	2.249.075
ASEADOS-SDN-IoT	51.465	260.406	391	0

O Edge-IIoT, o maior conjunto avaliado com mais de 30 milhões de instâncias, apresentou acurácia de 90,87%, precisão de 91,44%, revocação de 73,16% e *F1-Score* de 81,24%. O volume de falsos negativos e falsos positivos é esperado dada a escala e a forte heterogeneidade do tráfego IoT. Esse conjunto abrange mais de dez tipos de dispositivos com protocolos heterogêneos, o que dispersa a variabilidade do tráfego benigno no espaço de características e provoca oscilações esporádicas no *score* de anomalia, dificultando a separação precisa entre as classes. O processamento de mais de 30 milhões de instâncias corresponde a uma taxa superior a 2.000 instâncias por segundo, valor consistente também com o CTU-13, que processou 6.336.394 instâncias em 2.895 segundos sob as mesmas condições. Esse desempenho ratifica a viabilidade do método frente às restrições de baixa latência impostas por ambientes de processamento de fluxos de dados. Os resultados evidenciam que o método opera em fluxos contínuos de dados sem supervisão, com adaptação ao longo do tempo por meio do monitoramento de mudança de conceito via ADWIN. O desempenho superior no ASEADOS-SDN-IoT reforça sua adequação a cenários contemporâneos de ambientes SDN-IoT. Nos conjuntos CTU-13 e Edge-IIoT, a precisão consistentemente elevada indica que os alertas emitidos foram confiáveis, enquanto a revocação reduzida aponta que o ruído de *Background* e a heterogeneidade de dispositivos introduzem desafios à separabilidade entre as classes.

5. Conclusão

Este trabalho aborda o desafio da detecção de ataques DDoS volumétricos em ambientes de rede modernos, onde a natureza dinâmica do tráfego e o volume crescente de dados tornam os classificadores estáticos tradicionais insuficientes. Para avançar a literatura, este artigo propõe o MODUS, um método *online* e não supervisionado que integra três componentes principais: o encapsulamento incremental de pacotes em fluxos, a detecção de anomalias por meio do HST *(Half-Space Trees)* com limiar dinâmico baseado no estimador MAD, e o monitoramento de mudança de conceito via ADWIN. A eficácia do método foi avaliada por experimentos em três conjuntos de dados de características distintas: CTU-13 (cenário 11), Edge-IIoT e ASEADOS-SDN-IoT, abrangendo desde capturas históricas de botnets reais até cenários contemporâneos de ambientes SDN-IoT. Os resultados demonstram que o método detecta ataques DDoS com eficácia em todos os cenários avaliados, atingindo no ASEADOS-SDN-IoT acurácia de 99,87%, revocação de 100% e F1-Score de 99,62%. Nos conjuntos CTU-13 e Edge-IIoT, a precisão superior a 91% indica alertas confiáveis, embora a revocação reduzida aponte limitações frente a conjuntos com ruído de *Background* e alta heterogeneidade de dispositivos IoT. Como trabalho futuro, a expansão para detecção multiclasse representa o próximo passo natural, visando a identificação simultânea de diferentes categorias de ataque em fluxos contínuos.

Referências

Bhatia, S., Behal, S., and Ahmed, I. (2018). Distributed denial of service attacks and defense mechanisms: current landscape and future directions. *Versatile Cybersecurity*, pages 55–97.

Bifet, A., Gavaldà, R., Holmes, G., and Pfahringer, B. (2018). *Machine Learning for Data Streams with Practical Examples in MOA*. MIT Press. `https://moa.cms.waikato.ac.nz/book/`.

Bifet, A. and Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. In *Proceedings of the 7th SIAM International Conference on Data Mining*, pages 1–7.

Cardellini, V., Lo Presti, F., Nardelli, M., and Russo, G. R. (2022). Runtime adaptation of data stream processing systems: The state of the art. *ACM Comput. Surv.*, 54(11s):1–36.

Cloudflare (2026). DDoS threat report for 2025 Q4. Online. Acessado em: 3 maio 2026.

Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., and Janicke, H. (2022). Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning. *IEEE Access*, 10.

García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., and Herrera, F. (2016). Big data preprocessing: methods and prospects. *Big Data Analytics*, 1(1):9.

García, S., Grill, M., Stiborek, J., and Zunino, A. (2014). An empirical comparison of botnet detection methods. *Computers & Security*, 45:100–123.

Garg, U., Kaur, M., Kaushik, M., and Gupta, N. (2021). Detection of DDoS attacks using semi-supervised based machine learning approaches. In *International Conference on Computational Methods in Science & Technology*, pages 112–117.

Hindy, H., Brosset, D., Bayne, E., Seeam, A., Tachtatzis, C., and Bellekens, X. (2020). A taxonomy of network threats and the effect of current datasets on intrusion detection systems. *IEEE Access*, PP:1–28.

Iglewicz, B. and Hoaglin, D. (1993). *How to Detect and Handle Outliers*. ASQC basic references in quality control. ASQC Quality Press.

Khraisat, A., Gondal, I., Vamplew, P., and Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2:1–22.

Lakshan Yasarathna, T. and Le-Khac, N.-A. (2026). Aseados-sdn-iot: A novel sdn-iot network intrusion detection dataset and framework. *Internet of Things*, 36:1–30.

NETSCOUT Systems, Inc. (2026). Brazil - latest cyber threat intelligence report. Online. Acessado em: 3 maio 2026.

Ring, M., Wunderlich, S., Scheuring, D., Landes, D., and Hotho, A. (2019). A survey of network-based intrusion detection data sets. *Comput. Secur.*, 86(C):147—-167.

Romo-Chavero, M. A., Cantoral-Ceballos, J. A., Pérez-Díaz, J. A., and Martinez-Cagnazzo, C. (2024). Median absolute deviation for bgp anomaly detection. *Future Internet*, 16(5):5–18.

Rossow, C. (2014). Amplification hell: Revisiting network protocols for ddos abuse. In *2014 Network and Distributed System Security Symposium*, pages 1–15.

Scaranti, G. F., Carvalho, L. F., Barbon, S., Lloret, J., and Proença, M. L. (2022). Unsupervised online anomaly detection in software defined network environments. *Expert Systems with Applications*, 191:1–13.

Selvam, S. and Maheswari Balasubramanian, U. (2024). UASDAC: An unsupervised adaptive scalable DDoS attack classification in large-scale IoT network under concept drift. *IEEE Access*, 12:64701–64716.

Shi, Z., Li, J., and Wu, C. (2019). Deepddos: Online DDoS attack detection. In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–6.

Sommer, R. and Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE Symposium on Security and Privacy*, pages 305–316, USA. IEEE Computer Society.

Tan, S. C., Ting, K. M., and Liu, T. F. (2017). Fast anomaly detection for streaming data. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, volume 106, pages 1469–1495.

Yang, S., Zheng, X., Li, J., Xu, J., Zhang, X., and Ngai, E. C. H. (2025). Self-supervised adaptation method to concept drift for network intrusion detection. *IEEE Transactions on Dependable and Secure Computing*, 22(6):7632–7646.

Zhou, P. (2025). A survey of streaming data anomaly detection in network security. *PeerJ Computer Science*, pages 21–22. DOI: 10.7717/peerj-cs.3066.