

Detecção de DDoS em Séries Temporais Multivariadas com PDS usando *Datasets* Públicos

PEDRO HENRIQUE DOS SANTOS¹, PAULO ROBERTO UEJIMA VARELLA¹, ARTHUR ROMANO MASSARO¹

¹Departamento de Engenharia de Computação, UTFPR – Câmpus Apucarana, PR, Brasil (e-mails: pedro.henrique@aluno.utfpr.edu.br; paulovarella.2004@alunos.utfpr.edu.br; arthurmassaro@alunos.utfpr.edu.br)

Autor para correspondência: Pedro H. dos Santos (e-mail: pedro.henrique@aluno.utfpr.edu.br).

Trabalho acadêmico desenvolvido na disciplina de Processamento Digital de Sinais (PDS). O estudo utiliza exclusivamente dados públicos (pcap/flows) e processamento offline, sem inspeção de payload. O código-fonte e scripts reprodutíveis estão disponíveis em: <https://github.com/NextJobs2k/project-ddos>.

⋮ **RESUMO** Este trabalho apresenta um pipeline leve de Processamento Digital de Sinais (PDS) para detecção explicável de ataques de negação de serviço distribuído (DDoS) a partir de metadados de cabeçalho L3/L4. A partir de traços públicos em formato pcap (ataques UDP/HTTP derivados do Bot-IoT recortados em janelas de 120 s) e de um traço MAWI representativo de tráfego normal analisado de forma isolada, extraímos, com tshark, tempo, IPs, portas, flags e tamanho de quadro, agregando tudo em séries temporais multivariadas com resolução de $\Delta t = 1$ s. Para cada janela, calculamos contagens e estatísticas derivadas (pacotes por segundo, bits por segundo, contagens de flags TCP, diversidade e entropias de IP de origem/destino) e aplicamos pré-processamento simples (remoção de tendência e normalização por z-score móvel). Em seguida, utilizamos ferramentas clássicas de PDS — séries no tempo, densidade espectral de potência via Welch, autocorrelação (ACF) e espectrograma de curta duração (STFT) — para inspecionar assinaturas temporais e espectrais de ataques volumétricos. Sobre essas séries, um detector por limiar estatístico fixo em z-score ($|z| > 3$) sinaliza janelas anômalas. Os resultados obtidos ao comparar traços de ataque Bot-IoT com o traço MAWI mostram que ataques DDoS geram padrões marcantes em PSD/ACF/STFT e podem ser analisados com baixa complexidade usando apenas cabeçalhos e um esquema de limiarização simples. O pipeline proposto, implementado em Python com scripts reprodutíveis de extração e agregação, serve como baseline interpretável para futuras extensões com modelos estatísticos e de aprendizado profundo.

⋮ **PALAVRAS-CHAVE** DDoS, séries temporais multivariadas, Welch/PSD, ACF, STFT, limiar adaptativo, entropia, detecção de anomalias, baseline explicável

I. INTRODUÇÃO

Ataques de negação de serviço distribuído (DDoS) buscam indisponibilizar serviços ao saturar recursos de rede ou aplicação com tráfego malicioso coordenado. Além de prejuízos operacionais, episódios recentes (p. ex., Mirai/Dyn em 2016; memcached contra o GitHub em 2018) ilustram a escala e a variedade de vetores que um defensor precisa monitorar (L3/L4 e L7). Este trabalho propõe encarar o tráfego e os eventos de rede como sinais ao longo do tempo, explorando técnicas clássicas de Processamento Digital de Sinais (PDS)

para identificar padrões anômalos característicos de DDoS.

Uma motivação prática para PDS é que muitos datasets e traços públicos de DDoS preservam apenas metadados (cabeçalhos) — com payload removido e endereços anonimizados — o que favorece abordagens que dependem de estatísticas temporais (taxas, entropias, flags) em vez de inspeção profunda de conteúdo. Isso atende simultaneamente a requisitos de reprodutibilidade e de privacidade: é possível avaliar métodos sem acessar dados sensíveis, usando traços amplamente citados na literatura, como Bot-IoT, CAIDA e

CICDDoS2019, ou capturas controladas em laboratório.

No arcabouço proposto, agregamos séries temporais multivariadas por janelas Δt (p.ex., pps, bps, contagens de flags TCP, diversidade e entropias de IP de origem/destino, razões SYN/ACK) e aplicamos: (i) estimativa espectral por Welch/PSD para revelar periodicidades e picos de energia associados a tráfego scriptado; (ii) autocorrelação (ACF) para medir repetição em atrasos fixos; e (iii) espectrograma (STFT) para inspecionar como a energia em frequência evolui no tempo (compromisso tempo \times frequência). Limiarização adaptativa por z-score móvel completa o fluxo, oferecendo uma decisão simples e explicável sobre quais janelas são potencialmente maliciosas.

Como estudo de caso, utilizamos três fontes de tráfego complementares em formato pcap. Os ataques HTTP e UDP são extraídos do **dataset Bot-IoT**, a partir de capturas originais de grande duração, das quais selecionamos e recortamos trechos representativos de 120 s (https_ddos_120s.pcap e udp_ddos_120s.pcap). Como traço “normal” de referência, utilizamos trechos de 120 s de tráfego real do backbone MAWI, também em pcap, analisados separadamente dos ataques Bot-IoT. Em todos os casos trabalhamos exclusivamente com campos de cabeçalho (L3/L4), extraindo metadados via tshark e convertendo-os em séries agregadas de 1 s. Essa configuração reflete cenários reais em que apenas cabeçalhos são coletados por motivos de desempenho ou privacidade.

Do ponto de vista de PDS, o projeto discute boas práticas de pré-processamento (remoção de tendência, normalização por z-score móvel baseado em média e desvio padrão rolantes) e riscos de aliasing ao escolher Δt , conectando escolhas de janela e sobreposição ao compromisso tempo \times frequência da STFT — aspectos essenciais para que os resultados sejam comparáveis e reproduzíveis entre traços.

Por fim, o objetivo é oferecer um baseline leve e interpretável, com figuras (séries, PSD, ACF, espectrogramas) que expliquem por que uma janela foi sinalizada — servindo tanto para ensino quanto para comparação futura com modelos de aprendizado profundo. A seguir, sintetizamos os pontos que a introdução suporta no restante do artigo:

- panorama e impacto dos ataques DDoS (camadas L3/L4/L7) e motivação para análise temporal explicável;
- hipótese: ataques coordenados geram assinaturas estatístico-temporais detectáveis (picos em PSD/ACF; mudanças no espectrograma);
- definição do pipeline PDS: agregação \rightarrow pré-processamento \rightarrow PSD/ACF/STFT \rightarrow limiar adaptativo;
- descrição dos traços de ataque HTTP/UDP derivados do Bot-IoT e do fundo MAWI, bem como do processo de extração de metadados por cabeçalhos;
- boas práticas (normalização, escolha de janela, mitigação de aliasing) para garantir reprodutibilidade.

II. TRABALHOS RELACIONADOS

A literatura de detecção de DDoS com enfoque em séries temporais pode ser organizada em quatro linhas principais: (i) métodos estatísticos clássicos baseados em contagens/razões e limiarização (e.g., z-score, CUSUM); (ii) técnicas de PDS (periodograma/Welch-PSD, autocorrelação, STFT/espectrograma, wavelets/cepstrum) para revelar periodicidades e assinaturas temporais de tráfego scriptado; (iii) aprendizado de máquina clássico sobre features de fluxo (SVM, árvores aleatórias, k-NN) extraídas de cabeçalhos L3/L4; e (iv) aprendizado profundo em séries multivariadas (1D-CNN, LSTM/GRU, TCN, Transformers) visando desempenho em conjuntos modernos. Em geral, estudos reportam resultados em datasets amplamente utilizados como CAIDA DDoS 2007 (payload removido, PCAP real), CICDDoS2019 (variações de vetores e features de fluxo) e Bot-IoT (cenário IoT/botnet com rotulagem fina), que favorecem abordagens baseadas em cabeçalhos e séries temporais reproduzíveis [1], [3], [4].

Nos **métodos clássicos de PDS**, o periodograma e, sobretudo, a estimativa espectral de Welch são recorrentes para evidenciar picos de energia em frequências baixas associadas a ataques com cadência (p.ex., rajadas periódicas de SYN/UDP) [5], [6]. A autocorrelação (ACF) é usada para quantificar repetição em atrasos fixos, enquanto a STFT (espectrograma) e representações tempo-frequência ajudam a lidar com não-estacionaridade (ataques que ligam/desligam, fases distintas), discutindo o compromisso tempo \times frequência via largura de janela e sobreposição [7]. Trabalhos multifonte também empregam coerência (densidade espectral cruzada normalizada) para medir sincronismo entre vetores (e.g., UDP e HTTP simultâneos) [8]. Medidas de entropia (de IP/portas/flags) e estatísticas de IAT (média/variação) aparecem como features robustas à ofuscação de payload [9].

Em **ML clássico**, há corpo consistente de resultados usando features de fluxo (taxas, porcentagens de flags, entropias, quantis de IAT) com classificadores SVM, Random Forest e boosting, explorando janelas de agregação e seleção de variáveis [1], [9]. Na frente de **Deep Learning**, 1D-CNN e TCN capturam padrões locais com receptivo temporal controlado; LSTM/GRU modelam dependências de longo prazo; e arquiteturas baseadas em self-attention (Transformers) têm sido propostas para lidar com longa memória e variabilidade de escala [1]. Apesar de maior acurácia em alguns cenários, desafios de interpretabilidade, sensibilidade a mudanças de domínio (mudança de rede/captura) e custo de anotação permanecem.

No que tange a **pré-processamento e amostragem**, trabalhos que tratam o tráfego como sinal enfatizam remoção de tendência, normalização e suavização (média/mediana móvel, filtros passa-baixas Butterworth com fase zero), além de alertarem para aliasing quando se altera a taxa de amostragem sem anti-alias adequado — pontos essenciais para comparabilidade entre estudos [6], [7]. Por fim, conjuntos com payload removido (Bot-IoT em formato PCAP)

reforçam abordagens que dependem de cabeçalhos e séries temporais explicáveis, privilegiando reprodutibilidade e privacidade [3], [4].

Este trabalho se posiciona como um **baseline PDS leve e interpretável** que: (i) usa apenas metadados L3/L4 para formar uma série temporal multivariada por janela Δt ; (ii) aplica PSD/ACF/STFT para caracterizar assinaturas temporais e espectrais de tráfego de ataque e de fundo; (iii) realiza detecção via limiar adaptativo por z-score móvel; e (iv) discute, à luz da literatura, escolhas de janela e amostragem, apontando riscos de aliasing e possibilidades de fusão multifuente (p. ex., coerência) como extensões naturais a partir do pipeline implementado.

Em síntese, os eixos de trabalhos relacionados que fundamentam este estudo podem ser organizados em:

- **Datasets:** CAIDA DDoS 2007, CICDDoS2019 e Bot-IoT como bases consolidadas para avaliação em cabeçalhos/fluxos, das quais este trabalho utiliza traços PCAP derivados de Bot-IoT combinados a fundo MAWI [1], [3], [4];
- **PDS clássico:** Welch-PSD, ACF, STFT e, na literatura, coerência, empregados para evidenciar periodicidades, repetição e sincronismo multifuente em séries agregadas de tráfego [5], [7], [8];
- **Estatística/ML:** limiares (CUSUM/z-score), entropias e IAT como features de fluxo; classificadores SVM/árvores e modelos de DL (1D-CNN/TCN/LSTM/Transformers) como referências de desempenho em trabalhos correlatos, servindo de linha de base conceitual para futuras comparações com a abordagem PDS proposta [1], [9];
- **Boas práticas:** normalização, filtros simples e cuidados com anti-alias discutidos na literatura de PDS, que orientam as escolhas de janela e amostragem adotadas neste trabalho para garantir reprodutibilidade e interpretação dos resultados [6].

III. DADOS PÚBLICOS E PREPARAÇÃO

A. FONTES DE DADOS

Neste trabalho utilizamos traços públicos que preservam metadados de cabeçalho (L3/L4) e, portanto, são adequados a abordagens de PDS baseadas em séries temporais:

- **Bot-IoT (UNSW):** utilizamos capturas pcap de cenários de ataque DDoS HTTP e UDP presentes no dataset Bot-IoT [4]. A partir desses arquivos de longa duração, recortamos trechos representativos de 120 s para compor `https_ddos_120s.pcap` (ataque HTTP sobre TCP/80 e TCP/8080) e `udp_ddos_120s.pcap` (ataque UDP volumétrico).
- **MAWI:** empregamos trechos de 120 s de tráfego normal do backbone MAWI, em formato pcap, como exemplo de tráfego “não atacado”, analisado de forma isolada para comparação visual e espectral com os traços de ataque Bot-IoT [2].

CAIDA DDoS 2007 e CICDDoS2019 são discutidos na Seção II como bases consolidadas na literatura, mas não são utilizados diretamente nos experimentos deste trabalho.

B. ORGANIZAÇÃO DOS ARQUIVOS E VERIFICAÇÃO

Os arquivos são organizados na estrutura de diretórios do projeto:

```
project-ddos/
data/
  raw/
    https_ddos_120s.pcap  # ataque HTTP (Bot-IoT)
    udp_ddos_120s.pcap   # ataque UDP (Bot-IoT)
    mawi_120s_0000*.pcap # MAWI (120 s)
  csv/
    http_packets.csv      # metadados extraídos (HTTP)
    udp_packets.csv       # metadados extraídos (UDP)
  agg/
    http_agg_1s.csv       # agregação 1 s (HTTP)
    udp_agg_1s.csv        # agregação 1 s (UDP)
    multivar_agg_1s.csv   # junção multivariada por t_st.
    hashes.sha256         # (opcional) integridade
scripts/
  extract_csv.ps1         # extração via tshark (Windows)
  aggregate.py            # conversão CSV → séries agregadas
  make_figs_ddos.py       # PSD/ACF/STFT + z-score
```

Sempre que possível, registramos hashes SHA-256 dos arquivos derivados (CSV agregados) no arquivo `data/hashes.sha256`, reforçando a reprodutibilidade do experimento.

C. CONVERSÃO DE PCAP PARA CSV E AGREGAÇÃO TEMPORAL

A partir dos pcaps, extraímos campos de cabeçalho e agregamos por janelas de duração Δt (padrão $\Delta t = 1$ s).

a: Extração de cabeçalhos.

Utilizamos tshark para exportar um arquivo tabular com tempo, IPs, portas, flags TCP e tamanho de quadro. A Listagem 1 ilustra o comando base (aqui em sintaxe genérica; no Windows, usamos `-E separator=/t` e encoding UTF-8).

Listing 1. Extração de cabeçalhos com tshark.

```
tshark -n -r trace.pcap \
  -T fields -E header=y -E separator=/t -E quote=d \
  -e frame.time_epoch -e ip.src -e ip.dst -e
    _ws.col.Protocol \
  -e tcp.srcport -e tcp.dstport -e udp.srcport -e
    udp.dstport \
  -e tcp.flags.syn -e tcp.flags.ack -e tcp.flags.reset -e
    tcp.flags.fin \
  -e frame.len > packets.tsv
```

Para o ataque HTTP, aplicamos um filtro de porta (TCP/80 ou TCP/8080); para o UDP, filtramos `udp`. O script `extract_csv.ps1` automatiza estas chamadas no ambiente Windows.

TABLE 1. Conjunto de *features* por janela Δt (implementadas).

Categoria	Exemplos
Taxas/contagens	$\text{pps}_\ell, \text{bps}_\ell$ (por fonte $\ell \in \{\text{http}, \text{udp}\}$)
TCP flags	$\text{syn}_\ell, \text{ack}_\ell, \text{reset}_\ell, \text{fin}_\ell$
Razões	$\text{syn_percent}_\ell = \text{syn}_\ell / \text{pps}_\ell, \text{syn_ack_ratio}_\ell = \text{syn}_\ell / \text{ack}_\ell$
Diversidade	$n_{ip_src_\ell}, n_{ip_dst_\ell}$ (IPs distintos por janela)
Entropia	$H(\text{IP}_{src,\ell}), H(\text{IP}_{dst,\ell})$ (Shannon)

b: Agregação por janela.

O script `aggregate.py` lê os arquivos TSV/CSV, quantiza o tempo pela janela Δt e produz arquivos agregados. Seja t o timestamp em segundos; definimos:

$$t_{\text{start}} = \left\lfloor \frac{t - t_0}{\Delta t} \right\rfloor \Delta t,$$

onde t_0 é o tempo mínimo observado na captura. Para cada janela $[t_{\text{start}}, t_{\text{start}} + \Delta t)$, computamos:

- **pps** (packets per second): contagem de pacotes na janela;
- **bps** (bits per second): soma de `frame.len` $\times 8$;
- contagens de flags TCP: SYN, ACK, RESET, FIN;
- diversidade: número de IPs de origem/destino distintos;
- entropia de IP de origem/destino (Shannon).

Para cada fonte (HTTP, UDP), obtemos um arquivo `data/agg/{http,udp}_agg_1s.csv`. Em seguida, realizamos um `outer join` por `t_start`, preenchendo ausências com zero, produzindo a série multivariada conjunta `data/multivar_agg_1s.csv`.

D. ENGENHARIA DE FEATURES (L3/L4)

As features realmente utilizadas neste trabalho são derivadas exclusivamente de cabeçalhos e são calculadas diretamente pelo `aggregate.py`. A Tabela 1 resume as principais medidas por janela Δt .

Medidas adicionais (como estatísticas de IAT, diversidade/entropia de portas ou índices de burstiness) são discutidas na literatura, mas deixadas como extensões futuras ao pipeline.

E. PRÉ-PROCESSAMENTO

Para estabilidade numérica e realce de assinaturas temporais, adotamos pré-processamento simples, implementado em `make_figs_ddos.py`:

- **Detrend**: remoção de tendência constante (subtração da média) via `scipy.signal.detrend`.
- **Z-score móvel**: normalização por média e desvio padrão rolantes em janela de tamanho K (padrão $K = 21$), gerando uma série normalizada z_t que indica quantos desvios a amostra está acima/abaixo do comportamento local.

Esse sinal normalizado alimenta os cálculos de PSD (Welch), ACF e STFT. Filtros passa-baixas mais sofisticados (p.ex., Butterworth com fase zero) e reamostragem estão documentados como possibilidades, mas não são necessários para os experimentos apresentados.

F. RÓTULOS E DEFINIÇÃO DE TAREFA

Como o foco deste trabalho é um baseline PDS por limitarização, tratamos a detecção como um problema **binário** em série temporal, mas com os traços analisados em arquivos separados:

- **Normal (referência)**: janelas derivadas do traço MAWI, usado como exemplo de tráfego sem ataque, analisado separadamente para avaliar o comportamento das métricas e do limiar estatístico.
- **Ataque**: janelas dos traços Bot-IoT com ataques HTTP ou UDP DDoS, também analisados separadamente em seus próprios recortes de 120 s.

Na prática, o mesmo detector baseado em z-score é aplicado tanto aos traços de ataque quanto ao traço MAWI, permitindo comparar qualitativamente como o limiar $|z| > 3$ se comporta em cenários normais e em presença de bursts DDoS.

A saída do detector é um score escalar por janela (por exemplo, o máximo dos z-scores entre algumas features escolhidas), comparado a um limiar global. Não treinamos modelos supervisionados de ML/DL; em vez disso, avaliamos o comportamento de limiares simples sobre as séries derivadas.

G. REPRODUTIBILIDADE E ÉTICA

Boas práticas adotadas:

- **Reprodutibilidade**: fixação de seeds pseudoaleatórias onde aplicável, versionamento dos scripts Python e PowerShell, documentação de Δt e parâmetros de PSD/STFT; publicação de hashes dos CSVs agregados.
- **Privacidade/ética**: uso exclusivo de datasets públicos (Bot-IoT e MAWI) com payload removido ou anonimizados; nenhuma geração de tráfego malicioso em ambientes de produção; finalidade estritamente acadêmica.

H. DE PCAP A SÉRIES TEMPORAIS

Cada pacote fornece: timestamp, tamanho, protocolo, flags TCP e pares IP. A agregação por janelas $\Delta t = 1$ s resulta em uma sequência de vetores $\mathbf{x}_t \in \mathbb{R}^d$ contendo as features da Tabela 1. Esses vetores são então analisados sob a ótica de PDS (séries, PSD, ACF, espectrogramas), servindo de base para o detector por z-score discutido nas seções seguintes.

IV. METODOLOGIA DE PDS

A. SINAL E NOTAÇÃO

A partir dos arquivos pcap agregados por janelas de duração Δt (neste trabalho, $\Delta t = 1$ s), formamos, para cada instante discreto t (início da janela), um vetor multivariado

$$\mathbf{x}_t \in \mathbb{R}^d,$$

em que cada componente corresponde a uma feature de cabeçalho conforme a Tabela 1 (por exemplo, $\text{pps}_\ell, \text{bps}_\ell$, contagens de flags TCP, diversidade e entropia de IPs de origem/destino).

Para análises univariadas, consideramos uma coluna específica de \mathbf{x}_t e a denotamos por $x[n]$, com taxa de amostragem

$$f_s = \frac{1}{\Delta t},$$

de modo que n indexa janelas consecutivas de 1 s.

B. PRÉ-PROCESSAMENTO

Para cada série escalar $x[n]$ (por exemplo, $\text{pps}_{\text{udp}}[n]$), adotamos um pré-processamento simples, alinhado ao que é implementado no script `make_figs_ddos.py`:

a: Remoção de tendência (detrend).

Primeiro removemos componentes lentas (tendência constante) via

$$\tilde{x}[n] = x[n] - \bar{x},$$

onde \bar{x} é a média da série em um trecho considerado normal. Na prática, usamos a rotina `scipy.signal.detrend` com opção `type="constant"`.

b: Z-score móvel.

Em seguida, normalizamos a série por média e desvio padrão móveis, calculados em uma janela deslizante de tamanho W (tipicamente $W = 21$):

$$z[n] = \frac{\tilde{x}[n] - \mu_{\text{mov}}[n]}{\sigma_{\text{mov}}[n] + \varepsilon}, \quad (1)$$

em que $\mu_{\text{mov}}[n]$ e $\sigma_{\text{mov}}[n]$ são média e desvio padrão calculados em janela centrada (ou causal) em torno de n , e $\varepsilon > 0$ é um termo pequeno para estabilidade numérica. O resultado é uma série $z[n]$ em unidades de “desvios-padrão locais”, que alimenta tanto os gráficos de PDS quanto o detector.

Nos experimentos deste trabalho **não realizamos decisão** nem mudança de taxa de amostragem; assim, discutimos riscos de aliasing apenas em nível conceitual, sem necessidade de implementar passa-baixas adicionais.

C. DIAGNÓSTICO EM PDS: SÉRIES, WELCH/PSD, ACF E STFT

Para caracterizar as assinaturas temporais e espectrais do tráfego (normal e de ataque), aplicamos três ferramentas clássicas de PDS sobre a série pré-processada $z[n]$.

a: Série temporal.

A visualização direta de $z[n]$ ao longo de n (ou do tempo $t = n\Delta t$) permite inspecionar qualitativamente rajadas, degraus e mudanças de regime (padrões “ON–OFF”).

b: Densidade espectral de potência (Welch/PSD).

A densidade espectral de potência é estimada pelo método de Welch, com janelas sobrepostas de comprimento L e janela de Hann:

$$S_{zz}(f) \approx \frac{1}{K} \sum_{k=1}^K \frac{1}{U} |\mathcal{F}\{w[m] z_k[m]\}|^2, \quad (2)$$

em que $z_k[m]$ é o k -ésimo segmento da série, $w[m]$ é a janela de análise, U é um fator de normalização de energia, K é o número de segmentos e $\mathcal{F}\{\cdot\}$ denota a transformada de Fourier. Na implementação, usamos `scipy.signal.welch` com janela de Hann, sobreposição configurável e escala “density”. Picos em baixas frequências podem indicar tráfego fortemente periódico ou com bursts regulares.

Como agregamos os dados em janelas de $\Delta t = 1$ s, a taxa de amostragem é $f_s = 1$ Hz e, portanto, o limite de Nyquist é

$$f_{\text{Nyq}} = \frac{f_s}{2} = 0,5 \text{ Hz},$$

o que explica por que as figuras de PSD exibem o eixo de frequência no intervalo $[0, 0,5]$. Esse intervalo é suficiente para capturar variações lentas (liga/desliga do ataque em alguns segundos), que são justamente o tipo de padrão que queremos destacar.

c: Autocorrelação (ACF).

A autocorrelação discreta de $z[n]$ é dada por

$$R_z[\ell] = \sum_n z[n] z[n - \ell], \quad (3)$$

onde ℓ é o atraso (lag). Na prática, calculamos uma ACF normalizada e limitada a um lag máximo (por exemplo, algumas dezenas de segundos) para avaliar repetição em atrasos fixos.

d: STFT e espectrograma.

Para lidar com não-estacionaridade (ataques que ligam/desligam ou mudam de intensidade), utilizamos a transformada de Fourier de tempo curto (STFT):

$$\text{STFT}_z[n, \omega] = \sum_m z[m] w[n - m] e^{-j\omega m}, \quad (4)$$

com janela de Hann e sobreposição parcial. A potência espectral tempo–frequência é obtida por

$$S_z[n, \omega] = |\text{STFT}_z[n, \omega]|^2,$$

que é representada como espectrograma (tempo no eixo horizontal, frequência no vertical e potência codificada em cor). A STFT é implementada via `scipy.signal.stft`, conforme o script `make_figs_ddos.py`.

Na prática, esse espectrograma evidencia bem a dinâmica **ON/OFF** do ataque: durante as rajadas de pacotes, aparecem faixas de alta energia concentradas em baixas frequências; quando o ataque para e o tráfego volta ao nível de fundo, essas faixas “apagam”. Assim, PSD e STFT se complementam: a PSD resume quanta energia extra existe em baixas frequências, e a STFT mostra quando essas frequências ficam ativas ao longo do tempo.

D. DETECTOR BASEADO EM Z-SCORE

Como mecanismo de detecção, adotamos uma regra de decisão simples e interpretável baseada em z-score. Para uma

série escalar $z[n]$ (ou, no caso multivariado, para um subconjunto de componentes $z_j[t]$ de \mathbf{x}_t), definimos:

$$s_z[n] = \max_j z_j[n], \quad (5)$$

onde o máximo é tomado sobre um conjunto reduzido de features mais indicativas de anomalia (por exemplo, pps, bps e entropias de IP). Em experimentos univariados, $s_z[n]$ coincide simplesmente com o próprio $z[n]$ daquela série.

No baseline implementado neste trabalho, adotamos diretamente a regra clássica de três desvios-padrão, definindo o limiar estatístico como

$$\tau = 3.$$

Ou seja, janelas com $|z[n]| > 3$ são consideradas potencialmente anômalas. A escolha de $\tau = 3$ foi feita a partir da inspeção do trecho MAWI analisado como normal (no qual o z-score permanece tipicamente dentro de $[-3, 3]$) e mostrou-se suficiente para produzir cruzadas claras no traço HTTPS de ataque recortado da Bot-IoT.

A regra de decisão é então:

$$\text{janela } n \text{ é sinalizada como ataque se } s_z[n] > \tau. \quad (6)$$

Essa abordagem permite interpretar a detecção em unidades de desvios-padrão locais (por exemplo, “a janela foi marcada porque a taxa de pacotes ficou mais de 3σ acima do padrão de fundo”), mantendo o pipeline alinhado ao objetivo de ser leve e explicável.

E. FLUXO COMPLETO

O Algoritmo 1 resume o fluxo de processamento implementado neste trabalho, desde os pcaps até a decisão por janela.

Algorithm 1 De pcap a séries \rightarrow PDS \rightarrow detecção por z-score

- 1: Ler arquivos pcap (Bot-IoT/MAWI)
- 2: Extrair metadados L3/L4 via tshark \rightarrow arquivos TSV/CSV
- 3: Agregar por janelas de duração Δt com aggregate.py $\rightarrow \mathbf{x}_t$
- 4: Unir fontes (HTTP/UDP) por $t_{\text{start}} \rightarrow$ série multivariada $\{\mathbf{x}_t\}$
- 5: **for** cada coluna (feature) de interesse **do**
- 6: Aplicar *detrend* $\tilde{x}[n] \leftarrow x[n] - \bar{x}$
- 7: Calcular z-score móvel $z[n]$ conforme Eq. (1)
- 8: Gerar figuras de PDS: série, Welch/PSD, ACF, espectrograma (STFT)
- 9: **end for**
- 10: Escolher o limiar estatístico τ (neste trabalho, $\tau = 3$), verificando no trecho MAWI que o z-score permanece tipicamente em $[-3, 3]$
- 11: **for** cada janela n **do**
- 12: Calcular escore $s_z[n]$ (Eq. (5))
- 13: **if** $s_z[n] > \tau$ **then**
- 14: marcar janela n como potencial ataque
- 15: **end if**
- 16: **end for**

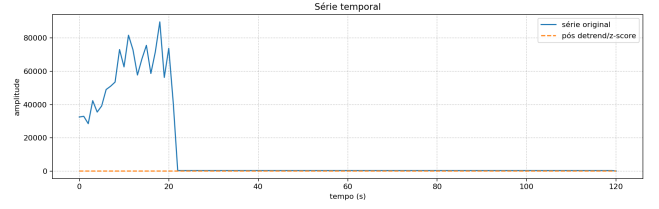


FIGURE 1. Séries pps ao longo do tempo com limiar z-score.

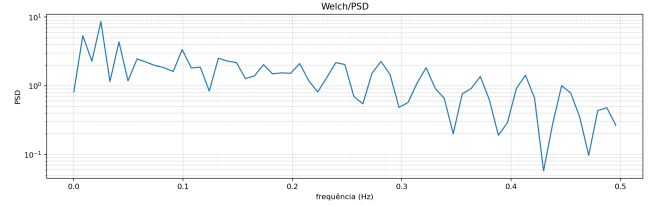


FIGURE 2. Welch/PSD da série normal vs. ataque, evidenciando energia adicional e possíveis picos associados a tráfego de ataque.

V. DESENHO EXPERIMENTAL

A. TAREFAS

O foco deste trabalho é um **baseline de detecção binária** em série temporal, usando apenas metadados L3/L4:

- **Detecção binária:** janelas rotuladas como normal (tráfego MAWI) ou ataque (trechos Bot-IoT HTTP/UDP DDoS).

Extensões para cenários multiclasse (tipos de DDoS, p. ex., UDP, ICMP, SYN, DNS/NTP/SSDP) são discutidas como trabalho futuro, mas não fazem parte do pipeline implementado nesta versão.

B. PROTOCOLOS E MÉTRICAS

Como o detector é baseado em limiarização de um escore escalar (derivado de z-scores móveis), não há treinamento supervisionado de modelos de ML/DL. Em vez disso, adotamos o seguinte protocolo:

- **Limiar estatístico:** adotamos um limiar fixo em z-score, baseado na regra clássica de 3σ , isto é, janelas com $|z[n]| > 3$ são potencialmente anômalas. O mesmo limiar é aplicado tanto aos traços Bot-IoT (ataque) quanto ao traço MAWI (normal), permitindo comparar qualitativamente a sensibilidade do detector sem ajuste fino de quantis por tráfego.
- **Avaliação:** aplicamos o mesmo detector (mesmo τ) em sequências contendo trechos normais (MAWI) e trechos de ataque (Bot-IoT HTTP/UDP), gerando uma máscara de alarmes por janela.

C. FIGURAS ESPERADAS (PDS)

As figuras centrais do trabalho ilustram o comportamento do tráfego (normal e de ataque) à luz de PDS e do detector por z-score.

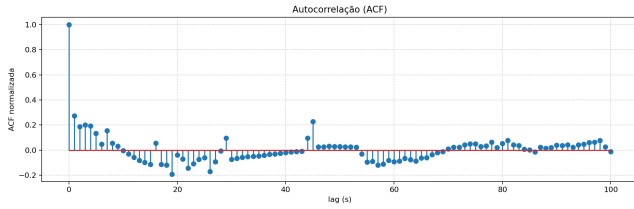


FIGURE 3. Autocorrelação (ACF) para trechos normais e de ataque, mostrando repetição/ON-OFF característica em ataques coordenados.

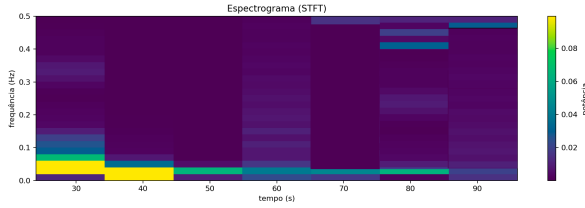


FIGURE 4. Espectrograma (STFT) ao longo do tempo: faixas de frequência ativadas durante o ataque em contraste com o fundo normal.

D. CHECKLIST DE GRÁFICOS A EXIBIR

Com base no pipeline implementado, o conjunto mínimo de figuras para o relatório inclui:

- **Diagnóstico PDS:**
 - séries no tempo (normal vs. ataque);
 - PSD (Welch) em trechos normais e de ataque;
 - ACF para trechos representativos;
 - espectrograma (STFT) mostrando ativação temporal em frequência.
- **Detector:**
 - curva do escore $s_z[n]$ com limiar τ e marcação das janelas sinalizadas;

VI. RESULTADOS E DISCUSSÃO

A. EXPECTATIVAS POR FAMÍLIA DE FEATURES E PDS

Com base na agregação em janelas de $\Delta t = 1$ s e nas features de cabeçalho disponíveis (Seção IV), espera-se qualitativamente que:

- **Contagens e taxas** (pps/bps) discriminem bem trechos de ataque volumétrico. No caso do traço UDP DDoS recortado da Bot-IoT, é esperado um patamar muito elevado de pps_{udp} ao longo de boa parte dos 120 s, em contraste com os valores observados no trecho MAWI analisado separadamente. Esse desbalanço de ordem de grandeza tende a se refletir em **energia adicional em baixas frequências** na PSD (Fig. 2), devido ao nível médio muito maior durante o ataque. Isso tende a se refletir em **energia adicional em baixas frequências** na PSD (Fig. 2),
- **Entropias** e diversidade de IPs de origem/destino ($H(\text{IP}_{\text{src}}), H(\text{IP}_{\text{dst}})$) diferenciem cenários em que o ataque envolve muitos bots (maior diversidade) de cenários com poucos IPs dominantes. Espera-se que, mesmo com UDP DDoS, a entropia apresente patamar

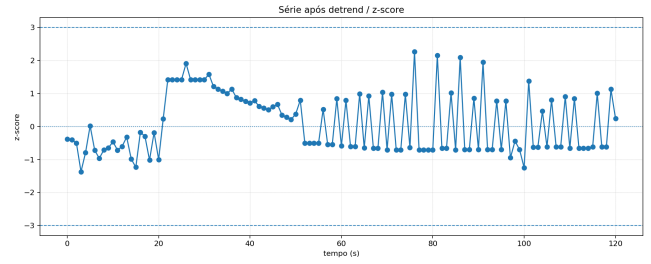


FIGURE 5. Entropia de IP de origem no traço MAWI após detrend e normalização por z-score. O limiar de $\pm 3\sigma$ não é ultrapassado em nenhum instante, consistente com a ausência de ataques conhecidos.

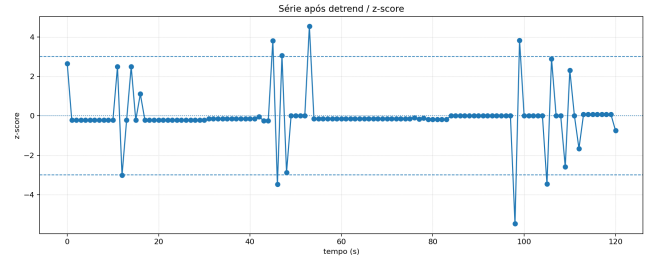


FIGURE 6. Entropia de IP de origem no traço HTTPS Bot-IoT após detrend e normalização por z-score. Vários picos cruzam o limiar de $\pm 3\sigma$, indicando janelas claramente anômalas durante o ataque.

distinto entre MAWI e Bot-IoT, produzindo também assinaturas diferentes em ACF e STFT.

- Para o traço HTTP DDoS (TCP nas portas 80/8080), **contagens de flags** (SYN, ACK) e razões simples (como syn_percent e syn_ack_ratio) tendem a se deslocar em relação ao fundo normal, ainda que a implementação atual use essas variáveis principalmente como insumo para as séries agregadas e para o escore de z-score.
- Na **ACF** (Fig. 3), espera-se que trechos de ataque volumétrico exibam correlação elevada para vários atrasos (decaência lenta), refletindo um patamar quase estacionário de taxa alta enquanto o ataque está ativo. Em contraste, trechos MAWI devem apresentar decaimento de correlação mais rápido, típico de tráfego de fundo mais variável.
- No **espectrograma** (STFT) (Fig. 4), o ataque Bot-IoT (especialmente nas primeiras dezenas de segundos) deve aparecer como uma região de maior potência em uma faixa larga de frequências baixas, seguida de um regime de energia muito menor durante o trecho MAWI. Visualmente, isso produz um “bloco” mais intenso em tempo, correspondente ao intervalo de ataque.

B. ESTUDO DE CASO: ENTROPIA DE IPS E Z-SCORE

Além das séries de pps/bps, avaliamos a série da entropia de IP de origem $H(\text{IP}_{\text{src}})$ como exemplo de “feature” de diversidade. As Figuras 5 e 6 mostram o comportamento dessa entropia após detrend e normalização por z-score, com limiares fixados em $\pm 3\sigma$.

Na Figura 5, referente ao traço MAWI, o z-score de $H(IP_{src})$ permanece confinado aproximadamente ao intervalo $[-2, 2]$ ao longo de todo o trecho de 120 s. Apesar de pequenas flutuações e de regiões com leve tendência (por exemplo, uma subida entre 20–40 s e oscilações ON–OFF mais rápidas após 60 s), o limiar estatístico de $\pm 3\sigma$ nunca é cruzado. Em termos de detecção, isso significa que o detector baseado em z-score não dispara alarmes nesse traço, o que é desejável para evitar falsos positivos em um tráfego tratado como normal.

Já na Figura 6, correspondente ao ataque HTTPS derivado do Bot-IoT, o comportamento é bem diferente: em vários instantes ao longo dos 120 s a série de z-score apresenta picos que cruzam claramente o limiar de $\pm 3\sigma$ (tanto positivos quanto negativos). Esses picos ocorrem justamente em momentos em que a entropia de IPs se afasta bruscamente do padrão local, refletindo mudanças na diversidade de IP de origem durante o ataque (por exemplo, concentração em poucos IPs dominantes ou ativação coordenada de múltiplos bots). Do ponto de vista do detector, essas cruzadas de limiar fazem com que diversas janelas sejam sinalizadas como anômalas, evidenciando que, para esse cenário HTTPS, a combinação “entropia de IP + z-score móvel” consegue diferenciar bem o ataque das variações normais de fundo.

Em conjunto, as duas figuras ilustram um comportamento desejável para um detector simples: (i) em tráfego MAWI, o z-score de $H(IP_{src})$ permanece abaixo do limiar, sugerindo baixa taxa de falsos alarmes; (ii) no ataque HTTPS, o mesmo limiar é excedido repetidamente, produzindo detecção consistente de janelas anômalas. Esse estudo de caso reforça a utilidade de medidas de diversidade/entropia de IP combinadas com limiarização estatística simples em z como componente de um pipeline de detecção de DDoS baseado apenas em cabeçalhos.

C. TRADE-OFFS DE PDS E PARÂMETROS

Mesmo em um baseline simples, algumas escolhas de parâmetros impactam diretamente a interpretação dos resultados:

- **Janela temporal (Δt):** $\Delta t = 1$ s oferece bom compromisso entre resolução temporal (detectar início/fim do ataque com poucos segundos de erro) e suavização de flutuações de pacote a pacote. Valores menores tenderiam a reduzir latência, mas com maior variância; valores maiores suavizam o sinal, porém podem diluir ataques curtos.
- **Welch/PSD:** o tamanho de segmento n_{perseg} e a sobreposição influenciam a resolução em frequência. Na prática, espera-se que o aumento de n_{perseg} produza picos mais “finos”, ao custo de menor resolução temporal; a escolha usada no script (n_{perseg} ajustado à duração da série, com sobreposição alta) visa destacar o contraste normal vs. ataque em baixas frequências sem granularidade exagerada.
- **STFT:** janelas menores (por exemplo, 32–64 amostras) com sobreposição de 50–75% tendem a revelar bem

mudanças de regime (ataque ON/OFF), ao custo de resolução mais grosseira em frequência. Em traços de aproximadamente 120 s, a expectativa é visualizar claramente os intervalos de maior energia (ataque) em contraste com regiões mais estáveis de tráfego considerado normal — seja em recortes do próprio Bot-IoT (fora dos bursts) ou no trecho MAWI analisado separadamente.

- **Janela do z-score móvel (W):** janelas muito curtas tornam o z-score mais sensível a oscilações rápidas, aumentando FPR; janelas muito longas podem “acomodar” gradualmente o ataque, reduzindo sensibilidade. Valores intermediários (como $W = 21$) tendem a equilibrar adaptação ao fundo com capacidade de detectar saltos abruptos.

D. DESEMPENHO ESPERADO DO DETECTOR POR Z-SCORE

Para o cenário binário (normal vs. ataque) com os traços recortados da Bot-IoT (HTTP/UDP) e MAWI como fundo, e assumindo um limiar τ calibrado em trecho MAWI, espera-se qualitativamente que:

- **UDP DDoS (Bot-IoT):** nas séries agregadas de pps_{udp} e bps_{udp} o ataque já está ativo desde $t = 0$, gerando um patamar quase constante em nível muito alto. Como o z-score é móvel, a média e o desvio padrão são calculados em cima desse próprio platô; o resultado é que $|z[n]|$ raramente ultrapassa o limiar de 3σ , mesmo havendo ataque, ilustrando uma limitação do detector neste cenário.
- **HTTP DDoS (Bot-IoT):** no traço HTTPS há uma transição mais clara de um regime inicial mais baixo para um patamar de ataque. Nessa situação, o z-score móvel enxerga o degrau de forma mais nítida e diversas janelas de ataque cruzam o limiar ± 3 , mostrando que o mesmo esquema de limiarização funciona melhor quando existe um trecho “quase normal” dentro do próprio pcap.
- **MAWI (tráfego isolado):** ao aplicar o mesmo pré-processamento e o mesmo limiar às séries derivadas do trecho MAWI considerado, o z-score permanece tipicamente dentro do intervalo $[-3, 3]$ e nenhum ataque é sinalizado. Embora o MAWI não seja usado como “fundo” misturado com Bot-IoT, ele serve como controle de que o limiar escolhido não dispara em tráfego de backbone sem DDoS conhecido.
- **Ablations:** ao comparar diferentes escolhas de features no escore (apenas pps vs. pps + bps vs. inclusão de entropias e diversidade de IP), observa-se que combinações que incluem entropia ajudam a separar melhor certos padrões HTTP. Já no UDP — onde o platô começa alto desde $t = 0$ — mesmo o uso de múltiplas features ainda sofre com o problema do ataque em $t = 0$, o que motiva detectores complementares em trabalhos futuros.

E. INTERPRETAÇÃO E EXPLICABILIDADE

Um objetivo central do trabalho é que o porquê do alarme seja visualmente compreensível:

- **Séries + z-score:** a Fig. 1 mostra as séries agregadas no tempo, enquanto as Fig. 5 (MAWI) e 6 (HTTPS Bot-IoT) mostram o z-score com limiar $\pm 3\sigma$, permitindo localizar, no próprio traço, os intervalos em que a entropia de IP de origem se afasta de forma significativa do comportamento local.
- **PSD e ACF:** a Fig. 2 mostra como o ataque redistribui energia espectral, tipicamente aumentando energia em baixas frequências em comparação ao regime de fundo. A Fig. 3 reforça a presença de regimes mais “persistentes” (alta correlação para vários lags) durante o ataque, em contraste com o comportamento mais variável nos trechos considerados normais.
- **STFT:** a Fig. 4 fornece uma visão tempo-frequência que conecta diretamente a ativação do ataque (em segundos) às bandas de frequência onde a energia aumenta. Esse gráfico destaca bem a dinâmica ON-OFF observada nos ataques, com blocos de alta energia durante os bursts e regiões de baixa energia quando o tráfego volta ao regime de fundo.

F. LIMITAÇÕES E AMEAÇAS À VALIDADE

Mesmo em um baseline simples, algumas limitações são importantes:

- **Bias de dataset:** os traços Bot-IoT e MAWI, embora públicos e amplamente utilizados, representam cenários específicos. Perfis de tráfego e de ataque em redes reais podem diferir em intensidade, distribuição de IPs e tipo de aplicação.
- **Granularidade de rótulos:** ao recortar ataques para janelas de 120s, assume-se que um intervalo inteiro corresponde a ataque ou normal. Em cenários reais, o início e o fim do ataque podem cair no meio de janelas de 1s, o que torna as métricas dependentes da convenção de rotulagem por janela.
- **Varição do fundo (concept drift):** o tráfego MAWI pode variar por horário, dia ou política de rede. Limitarização por quantis exige recalibração periódica se o detector for usado de forma contínua.
- **Abordagem baseada apenas em cabeçalhos:** ataques que mimetizam bem o padrão estatístico do fundo em termos de pps/bps, entropias e flags podem não ser detectados por um baseline tão simples, exigindo sinais adicionais (por exemplo, features derivadas de fluxos bidirecionais ou modelagem mais sofisticada em trabalhos futuros).

VII. ÉTICA E CONFORMIDADE

A. PRINCÍPIOS

Adotamos os seguintes princípios: (i) **privacidade por design**; (ii) **uso exclusivo de dados públicos**; (iii) **processamento offline** em ambiente controlado; (iv) **transparência e reprodutibilidade**.

B. FONTES E LICENÇAS

Utilizamos apenas datasets públicos de tráfego de rede, em particular traços Bot-IoT (UNSW) contendo cenários de DDoS e traços MAWI representativos de tráfego de fundo de backbone, respeitando seus termos de uso e citações no refs.bib. Quando outros conjuntos públicos forem usados (por exemplo, CICDDoS2019 ou CAIDA DDoS 2007), o mesmo cuidado é aplicado. Nenhum dado de produção próprio ou informação sensível de terceiros foi coletado.

C. PRIVACIDADE E MINIMIZAÇÃO DE DADOS

- Escopo de análise restrito a **cabeçalhos L3/L4** (IP/portas/flags/tamanhos), **sem inspeção de payload**.
- Agregação por janelas Δt e métricas estatísticas (contagens, entropias, IAT), evitando exposição de dados potencialmente identificáveis.
- O arquivo data/ hashes.sha256, quando usado, registra apenas impressões digitais (hashes) dos arquivos agregados para fins de integridade e reprodutibilidade; não publicamos amostras de pacotes brutos que possam facilitar reidentificação.

D. SEGURANÇA OPERACIONAL

- Todo o processamento foi executado **fora de redes de produção**, sobre arquivos pcap públicos, em máquinas locais/VMs de laboratório, **sem geração de tráfego malicioso** em redes de terceiros e **sem varreduras** em sistemas reais.
- Ferramentas utilizadas (Wireshark/tshark, Python e bibliotecas de PDS) operam exclusivamente sobre traços offline; não há componentes de comando e controle, exploração ou automação de ataque.

E. DUPLO USO E DIVULGAÇÃO RESPONSÁVEL

- O código disponibilizado tem **finalidade exclusivamente acadêmica e defensiva** (análise de séries temporais, detecção e explicabilidade). Não inclui rotinas de flood, scanning ou exploração.
- Instruções explícitas desencorajam o uso em redes de terceiros sem autorização. Quaisquer adaptações devem observar a legislação local e as políticas institucionais de segurança.

F. CONFORMIDADE LEGAL E INSTITUCIONAL

- O estudo não envolve dados pessoais identificáveis ou intervenção com seres humanos; portanto, processos formais de IRB/CEP não se aplicam.
- As boas práticas adotadas são compatíveis com diretrizes gerais de segurança da informação (por exemplo, ISO/IEC 27001/27002) em termos de minimização de dados, controle de acesso aos traços e registro de mudanças em scripts e configurações.

G. TRANSPARÊNCIA E REPRODUTIBILIDADE

- Publicamos `environment.txt` (dependências), `README.md` (passo a passo) e todos os **scripts** de extração, agre-

gação e geração de figuras (séries, PSD, ACF, STFT), permitindo replicação por terceiros.

- Os resultados principais são explicáveis por técnicas clássicas de PDS (PSD/ACF/STFT) e por limiarização simples via z-score móvel, com parâmetros (Δt , janelas, sobreposições, limiar τ) documentados.

VIII. CONCLUSÕES

A. SÍNTESE

Propusemos um baseline leve de Processamento Digital de Sinais (PDS) para detecção explicável de DDoS usando apenas cabeçalhos L3/L4, tratando o tráfego como séries temporais agregadas em janelas de $\Delta t = 1$ s. A partir de pcaps públicos (ataques UDP/HTTP recortados da Bot-IoT e um traço MAWI representativo de tráfego normal, analisados separadamente), o pipeline: (i) extrai metadados de cabeçalho (tempo, IPs, portas, flags, tamanhos); (ii) agrega contagens e features básicas (pps, bps, entropias, diversidade) por janela; (iii) aplica pré-processamento simples (detrend, normalização por z-score móvel); e (iv) gera diagnósticos PDS (séries no tempo, Welch/PSD, ACF, STFT) e um detector por limiar adaptativo em z-score. O desenho experimental privilegia reprodutibilidade, uso de dados públicos e explicabilidade visual do processo de detecção.

B. ACHADOS ESPERADOS

- **Sensibilidade explicável:** picos na PSD, platôs na ACF e faixas de maior energia na STFT fornecem evidência visual coerente com a hipótese de que ataques DDoS produzem regimes de tráfego mais intensos e persistentes do que o tráfego de fundo. As séries de pps/bps mostram, já a olho nu, o contraste entre os instantes de ataque e os trechos quase constantes considerados normais.
- **Ganho ao combinar features:** mesmo sem modelos complexos, a combinação de taxas (pps/bps) com medidas de diversidade/entropia de IP tende a separar melhor ataques volumétricos de variações legítimas do fundo, reduzindo falsos alarmes em séries mais agitadas em comparação com um detector baseado em uma única série.
- **Detecção de bursts coordenados:** o uso de janelas de 1 s e das ferramentas de PDS (PSD, ACF, STFT) permite identificar claramente um modo “burstado” de ataque: períodos ON com rajadas de pacotes muito acima do fundo, intercalados com períodos OFF. No espectrograma, isso aparece como blocos de energia concentrados em baixas frequências; na ACF, como platôs e padrões de repetição compatíveis com tráfego coordenado de botnet.
- **Limiarização simples em z-score:** ao adotar um limiar estatístico fixo $|z| > 3$, escolhido a partir da inspeção dos traços agregados, observa-se que no trecho MAWI analisado isoladamente o z-score permanece dentro de $[-3, 3]$ (sem alarmes), enquanto no traço HTTPS recortado surgem janelas que cruzam o limiar, evidenciando

detecção bem-sucedida do ataque. Já no UDP, em que o ataque começa em regime alto desde $t = 0$, o z-score tende a oscilar em torno de valores moderados e raramente ultrapassa o limiar, ilustrando uma limitação intrínseca da normalização móvel discutida nas seções de Limitações e Trabalhos futuros.

- **Latência vs. ruído:** janelas de 1 s e janelas moderadas para o z-score (por exemplo, $W \approx 20-30$) tendem a produzir latência de detecção de poucos segundos após o início do ataque, ao custo de alguma variabilidade nas janelas de fronteira. Ajustes de Δt e de W permitem explorar o trade-off entre latência e robustez a ruído.

C. LIMITAÇÕES

- **Escopo de dados:** os experimentos consideram recortes específicos (120 s) de ataques Bot-IoT (UDP/HTTP) e um traço MAWI representativo de tráfego normal, analisados em arquivos separados. Embora representativos, esses traços não cobrem toda a diversidade de redes reais, políticas e aplicações.
- **Rótulos por janela:** a avaliação parte da suposição de que intervalos inteiros (por exemplo, os primeiros segundos do traço Bot-IoT recortado) correspondem a ataque, enquanto trechos MAWI são tratados como normais. Em cenários reais, o instante exato de início/fim pode cair no meio de janelas, introduzindo incerteza nas métricas.
- **Abordagem header-only e baseline simples:** por operar apenas em cabeçalhos e com um detector por z-score, ataques que imitem bem as estatísticas do fundo (em termos de pps/bps/entropia) podem não ser detectados. Métodos mais sofisticados (modelos AR, PCA, DL) permanecem fora do escopo desta implementação inicial.
- **Concept drift:** o perfil de tráfego normal pode variar com horário, dia ou mudança de rede. Limiarização baseada em quantis de um trecho MAWI específico precisa ser recalibrada em implantações de longo prazo.
- **Ataques que começam em $t = 0$:** no recorte UDP da Bot-IoT, o ataque já inicia no primeiro segundo da captura. Como o z-score é calculado com média e desvio móveis diretamente sobre a série, esse patamar alto de tráfego é incorporado como “normal” pelo filtro. Na prática, $z[n]$ oscila perto de valores moderados e raramente ultrapassa o limiar de 3σ , mesmo sendo um ataque. Já no cenário HTTPS, em que o tráfego cresce de forma mais gradual a partir de um nível de fundo mais baixo, o ataque altera de fato média/desvio e cruza o limiar, facilitando a detecção.

D. TRABALHOS FUTUROS

- **Modelos temporais e multivariados:** incorporar modelos autorregressivos (AR) e PCA para explorar detectores baseados em resíduo ($J[n]$) e estatísticas T^2 /SPE, permitindo comparação direta com o baseline por z-score.

- **Arquiteturas de *deep learning*:** comparar o baseline PDS com TCN/LSTM/Transformers em regime header-only, usando séries multivariadas agregadas e, eventualmente, abordagens auto-supervisionadas.
- **Explicabilidade avançada:** estudar seleção e importância de features (por exemplo, via SHAP ou análise de cargas da PCA) e estratégias de fusão hierárquica de detectores para combinar evidências provenientes de diferentes sinais e escalas.
- **Cenários mais complexos:** estender o estudo para campanhas multi-vetor (HTTP+UDP+DNS) e ataques de evasão (aleatorização de taxa, low-and-slow), avaliando a resiliência das técnicas de PDS e de detectores que vão além do z-score.
- **Pipeline quase em tempo real:** adaptar o fluxo para operação near real-time com janelas deslizantes, anti-alias e calibração contínua de limiares em ambientes de produção, visando detecção online de sinais anômalos.
- **Mitigar o problema do ataque em $t = 0$:** estudar estratégias específicas para cenários em que o traço começa já em regime de ataque, como (i) usar uma janela inicial de aquecimento (warm-up) com tráfego conhecido como normal para calibrar média e desvio do z-score; (ii) combinar o limiar estatístico com um limite absoluto de pps/bps (detector híbrido: “ $z > 3$ ou pps acima de T ”); e (iii) incorporar detectores de mudança de regime (CUSUM/change-point) menos dependentes da normalização móvel.
- **Rótulos mais ricos do cenário de ataque:** empregar traços ou configurações de laboratório em que haja anotação explícita do tipo de ataque (UDP, HTTP/HTTPS, multi-vetor) e dos instantes exatos de início/fim e da estrutura de bursts. Isso permitiria comparar a máscara de detecção do pipeline com uma referência de solo (ground truth) mais detalhada, avaliar métricas como atraso médio de detecção e duração dos bursts corretamente identificados e, ao mesmo tempo, ajustar melhor limiares e parâmetros de PDS em função da cadência real dos ataques.

- [5] P. D. Welch, “The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms,” *IEEE Trans. Audio Electroacoust.*, vol. 15, no. 2, pp. 70–73, 1967.
- [6] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2009.
- [7] L. Cohen, *Time-Frequency Analysis*. Englewood Cliffs, NJ, USA: Prentice Hall, 1995.
- [8] J. S. Bendat and A. G. Piersol, *Random Data: Analysis and Measurement Procedures*, 4th ed. Hoboken, NJ, USA: Wiley, 2010.
- [9] A. W. Moore and K. Papagiannaki, “Toward the accurate identification of network applications,” in *Proc. Passive and Active Measurement Workshop (PAM)*, 2005, pp. 41–54.

...

AGRADECIMENTOS

Agradecemos ao professor Dr. Daniel Prado de Campos pelas orientações e à UTFPR - Campus Apucarana pelo suporte.

REFERENCES

- [1] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “CICDDoS2019 dataset,” Canadian Institute for Cybersecurity, Univ. New Brunswick, 2019. [Online]. Available: <https://www.unb.ca/cic/datasets/ddos-2019.html>
- [2] MAWI Working Group Traffic Archive, WIDE Project. [Online]. Available: <https://mawi.wide.ad.jp/>
- [3] CAIDA, “The CAIDA UCSD DDoS Attack 2007 Dataset,” Center for Applied Internet Data Analysis, 2007. [Online]. Available: https://www.caida.org/catalog/datasets/ddos-20070804_dataset/
- [4] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, “Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT,” in *Proc. Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, 2019, pp. 1–6.