

Detecção de Botnets em Tráfego IoT na Borda: Uma Avaliação de Modelos Leves sob Protocolo Leave-One-Scenario-Out

Ramon Mayor Martins e Diego da Silva de Medeiros

Resumo— Este trabalho avalia o compromisso entre desempenho e custo computacional na detecção binária de botnets em tráfego IoT, visando dispositivos de borda. Oito modelos de aprendizado de máquina são comparados sobre o dataset IoT-23 sob dois protocolos: divisão i.i.d. e Leave-One-Scenario-Out (LOSO) com seis famílias de malware. Sob LOSO, uma Árvore de Decisão de profundidade 10 atinge F1-macro de 0,851 ocupando 7,7 KB, desempenho equivalente ao Random Forest de 669 KB (87× maior). Mostra-se ainda que a divisão i.i.d. superestima o desempenho real em cerca de 14 pontos percentuais, motivando o LOSO como protocolo padrão na área.

Palavras-Chave — Detecção de botnets, IoT, IoT-23, aprendizado de máquina, edge computing, Leave-One-Scenario-Out.

Abstract — This work evaluates the trade-off between detection performance and computational cost for binary botnet detection in IoT traffic, targeting edge deployment. Eight machine learning models are compared on the IoT-23 dataset under two protocols: stratified i.i.d. split and Leave-One-Scenario-Out (LOSO) over six malware families. Under LOSO, a depth-10 Decision Tree reaches macro-F1 of 0.851 in only 7.7 KB, statistically equivalent to a 669 KB Random Forest (87× larger). Furthermore, the i.i.d. split overestimates real-world performance by roughly 14 percentage points, motivating LOSO adoption as a standard evaluation protocol in the field.

Keywords— Botnet detection, IoT, IoT-23, machine learning, edge computing, Leave-One-Scenario-Out.

I. INTRODUÇÃO

Botnets baseadas em dispositivos de Internet das Coisas (IoT), como Mirai, Gafgyt e suas variantes, representam uma das ameaças mais relevantes para a infraestrutura de redes contemporâneas, tendo sido responsáveis por ataques de negação de serviço distribuído (DDoS) com capacidades de tráfego sem precedentes [1]. A heterogeneidade dos dispositivos IoT, somada à severa restrição de recursos computacionais em gateways de borda, motiva a busca por detectores precisos, leves e com baixa latência de inferência. O dataset IoT-23 [1], composto por capturas reais de tráfego malicioso e benigno, tornou-se referência para a avaliação de tais detectores. Entretanto, grande parte da literatura reporta desempenhos próximos da saturação ($F1 > 0,99$) sob protocolos de divisão aleatória estratificada (i.i.d.), o que tende a superestimar a capacidade real de generalização dos modelos diante de famílias de malware desconhecidas [2], [3]. Alguns

trabalhos, como Stoian [2], aplicaram algoritmos clássicos de aprendizado de máquina ao IoT-23, obtendo acurácias elevadas sob divisão aleatória. Hussain et al. [3] propuseram um conjunto universal de atributos para detecção cross-dataset de botnets IoT. Nenhum desses trabalhos, contudo, avaliou o desempenho sob cenários distintos do mesmo dataset. Hegde et al. [4] investigaram classificadores supervisionados e de aprendizado profundo para identificação de atividade de botnet em redes IoT domésticas, sem reportar custos computacionais relevantes para implantação na borda. Dessa forma, este trabalho aborda essa lacuna por meio de uma comparação sistemática de oito modelos de aprendizado de máquina sob dois protocolos: (A) divisão i.i.d. tradicional e (B) Leave-One-Scenario-Out (LOSO), no qual cada família de malware é avaliada sem ter sido vista no treinamento. As contribuições são: (i) evidência quantitativa do viés otimista da avaliação i.i.d. (queda média de ~14 pontos percentuais em F1-macro); (ii) análise da fronteira de Pareto entre desempenho e custo (tamanho do modelo, latência, throughput); e (iii) recomendação prática de uma Árvore de Decisão de 7,7 KB com desempenho equivalente a um Random Forest 87× maior, viabilizando implantação em gateways IoT.

Trabalhos recentes reforçam a importância de detectores viáveis para a borda. Wei et al. [5] propuseram um NIDS leve em duas etapas (autoencoder + CNN sobre imagens de comprimento de pacotes), implantado em Raspberry Pi, demonstrando que é possível atingir alta acurácia com recursos computacionais mínimos. Contudo, esse e outros trabalhos recentes [2], [4] mantêm a avaliação sob divisão i.i.d. tradicional, o que, conforme evidenciado pela presente pesquisa, superestima sistematicamente o desempenho quando famílias de malware inéditas chegam à rede. Esses resultados sugerem que o protocolo Leave-One-Scenario-Out oferece uma avaliação mais realista da generalização entre famílias de malware e pode servir como base para diretrizes metodológicas futuras na área.

Este artigo está organizado da seguinte forma. A Seção II apresenta a metodologia, incluindo dataset, extração de atributos e protocolos de avaliação. A Seção III descreve os modelos considerados. A Seção IV discute os resultados e suas implicações para implantação na borda. A Seção V apresenta as conclusões e trabalhos futuros.

II. METODOLOGIA

Na avaliação i.i.d., as amostras são separadas aleatoriamente entre os conjuntos de treinamento e teste, de modo que ambos podem conter fluxos provenientes dos

mesmos cenários, dispositivos ou capturas. Embora esse procedimento seja comum, ele pode produzir estimativas excessivamente otimistas, pois padrões específicos de um cenário podem estar presentes tanto no treinamento quanto no teste. Para avaliar de forma mais realista a capacidade de generalização dos modelos, o protocolo LOSO é utilizado, no qual todos os dados de um cenário são completamente mantidos fora do treinamento e utilizados exclusivamente para teste, enquanto os demais cenários são empregados no treinamento. Esse processo é repetido até que cada cenário tenha sido utilizado uma vez como conjunto de teste, permitindo verificar o desempenho do modelo diante de condições e padrões de tráfego não observados durante o treinamento.

Utilizou-se o dataset IoT-23 [1], composto por capturas reais de tráfego de dispositivos IoT (formato Zeek conn.log). Para o protocolo i.i.d. empregou-se a versão pré-processada com 6.046.623 fluxos e sete atributos numéricos. Para o protocolo LOSO foram baixados diretamente do servidor da CTU seis cenários de malware (Hide-and-Seek, Muhstik, Hakai, Torii, Mirai e Gafgyt) e dois cenários honeypot benignos (Philips HUE e Amazon Echo). Os seis cenários: Hide-and-Seek (1.008.748 fluxos brutos), Muhstik (156.103), Hakai (10.403), Torii (3.209), Mirai (23.145) e Gafgyt (3.581.028), foram submetidos a uma limitação de 15.000 fluxos por cenário para evitar dominação do Gafgyt no treinamento, resultando em 75.438 fluxos no conjunto final. A disparidade nos volumes originais reflete a diversidade real das campanhas de botnet capturadas pelo IoT-23.

Os atributos compreendem dez variáveis numéricas (duration, orig_bytes, resp_bytes, orig_pkts, resp_pkts, orig_ip_bytes, resp_ip_bytes, missed_bytes, id.orig_p, id.resp_p) e três categóricas (proto, conn_state, service) codificadas via one-hot, totalizando 33 atributos.

Foram avaliados oito modelos: Regressão Logística (LogReg), Árvore de Decisão com profundidade 5 e 10 (DT-d5, DT-d10), Random Forest com 10 e 50 árvores (RF-10, RF-50), XGBoost com 20 e 100 estimadores (XGB-20, XGB-100) e MLP com camada oculta de 32 neurônios (MLP-32) [6]–[8]. Todos os modelos empregaram balanceamento de classes (class_weight='balanced'; scale_pos_weight no XGBoost).

O protocolo A consiste em divisão estratificada 80/20 (linha de base otimista). O protocolo B (LOSO) treina sobre cinco cenários de malware mais 70% dos fluxos benignos e testa no cenário restante mais 30% dos benignos, simulando a chegada de uma família desconhecida ao gateway. Foram registradas métricas de qualidade (F1-macro, F1 da classe benigna, ROC-AUC) e de custo de inferência (tamanho do modelo serializado em KB, latência média por fluxo em μ s e throughput em fluxos/s), em CPU x86 padrão.

Dado o forte desbalanceamento da classe benigna no protocolo B (apenas 2,4% dos fluxos), o uso de class_weight='balanced' (e scale_pos_weight para o XGBoost) é decisivo: simulações preliminares sem balanceamento produziram modelos que classificavam quase todo o tráfego como malicioso, com F1 da classe benigna próxima de zero apesar de F1-macro aparentemente alta. O balanceamento força o classificador a tratar erros sobre fluxos benignos como significativamente mais custosos, evitando o colapso degenerado em direção à classe majoritária. Essa observação

reforça a importância de reportar separadamente o F1 da classe benigna, métrica frequentemente omitida em estudos que se limitam à acurácia global [2], [4].

Especificamente, o protocolo B (LOSO) executa seis iterações: em cada uma, cinco cenários de malware compõem o conjunto de treinamento juntamente com 70% dos fluxos benignos (selecionados por amostragem estratificada), e o cenário restante compõe o conjunto de teste juntamente com os 30% complementares de fluxos benignos. Dessa forma, o modelo nunca vê a família de malware de teste durante o treinamento, simulando fielmente a situação operacional de um gateway IoT confrontado por uma botnet inédita. A média e o desvio-padrão das métricas entre as seis iterações são reportados para evidenciar a variabilidade da generalização inter-família.

III. RESULTADOS E DISCUSSÃO

A Tabela I sintetiza os resultados sob ambos os protocolos. Sob a divisão i.i.d. (coluna F1 (A)), modelos baseados em árvores e ensembles atingem desempenho próximo da saturação (F1-macro > 0,98), confirmando a observação recorrente na literatura de que o IoT-23 é trivial neste regime [2].[4]. Modelos lineares (LogReg) e a MLP rasa apresentam desempenho insatisfatório (F1 < 0,55), revelando fronteiras de decisão genuinamente não-lineares no problema.

Sob o protocolo LOSO (coluna F1 (B)), o panorama muda substancialmente: a queda média de aproximadamente 14 pontos percentuais entre os protocolos (RF-50: 1,000 $\rightarrow$ 0,855; DT-d10: 0,981 $\rightarrow$ 0,851) demonstra que a divisão i.i.d. superestima sistematicamente a capacidade de generalização inter-família. Esse efeito é compatível com a hipótese de que modelos treinados i.i.d. memorizam padrões específicos de cada captura (timestamps, portas, duração de fluxos) que não se transferem para malwares novos.

TABELA I. RESULTADOS SOB OS PROTOCOLOS I.I.D. (A) E LOSO (B), ORDENADOS POR F1-MACRO DO PROTOCOLO B. TAM.: TAMANHO DO MODELO SERIALIZADO; LAT.: LATÊNCIA MÉDIA DE INFERÊNCIA POR FLUXO.

Modelo	F1 (A)	F1 (B)	ROC-AUC (B)	Tam. (KB)	Lat. (μ s)
RF-50	1,000	0,855	0,998	669,5	2,44
RF-10	1,000	0,852	0,999	112,9	1,37
DT-d10	0,981	0,851	0,944	7,7	0,07
XGB-100	0,998	0,822	0,990	140,8	0,56
MLP-32	0,541	0,813	0,934	34,4	0,58
DT-d5	0,761	0,804	0,953	4,1	0,08
XGB-20	0,992	0,790	0,980	29,2	0,15
LogReg	0,470	0,690	0,914	1,0	0,07

TABELA II. COMPARAÇÃO DIRETA DE F1-MACRO ENTRE OS PROTOCOLOS A (I.I.D.) E B (LOSO). $\Delta = F1(B) - F1(A)$.

Modelo	F1 (A, i.i.d.)	F1 (B, LOSO)	Δ
RF-50	1,000	0,855	-0,145
RF-10	1,000	0,852	-0,148
DT-d10	0,981	0,851	-0,130
XGB-100	0,998	0,822	-0,177
MLP-32	0,541	0,813	+0,272
DT-d5	0,761	0,804	+0,043
XGB-20	0,992	0,790	-0,202
LogReg	0,470	0,690	+0,220

A Tabela II quantifica a magnitude da queda entre os protocolos. Modelos que saturam sob i.i.d. (RF-50: $F1=1,000$; XGB-100: $F1=0,998$) sofrem quedas acentuadas sob LOSO (-0,145 e -0,177 respectivamente), confirmando a memorização de padrões específicos de cada captura. Notavelmente, MLP-32 e LogReg apresentam Δ positivos (+0,272 e +0,220), indicando que, sob LOSO com balanceamento adequado, esses modelos generalizam melhor do que sugerem seus resultados i.i.d., embora permaneçam abaixo dos modelos baseados em árvores em termos absolutos. A queda média de ~14 pontos percentuais de F1-macro entre os protocolos constitui a principal evidência empírica do artigo em favor da adoção do LOSO como padrão na comunidade.

A Tabela III ilustra a heterogeneidade entre famílias: Gafgyt e Mirai são relativamente fáceis ($F1 > 0,84$ para os melhores modelos), enquanto Hide-and-Seek (H&S) representa o caso mais desafiador, com $F1$ abaixo de 0,46 para todos os classificadores. Esse comportamento sugere que padrões de tráfego de bots furtivos (covert channels) permanecem difíceis de detectar a partir apenas de estatísticas de fluxo. Como limitações do estudo, registra-se que apenas dois cenários benignos (Philips HUE e Amazon Echo) estavam disponíveis no servidor da CTU, totalizando 1.826 fluxos benignos e restringindo a representatividade dessa classe; a latência foi medida em CPU x86 padrão, devendo ser revalidada em hardware embarcado real (Raspberry Pi, ESP32, gateways comerciais); e a binarização ignora a riqueza dos detailed-labels do IoT-23 (PartOfAHorizontalPortScan, C&C, DDoS, Okiru), que motivam trabalhos multi-classe futuros.

Contudo, merece destaque o cenário Hide-and-Seek (H&S), cuja família de malware emprega protocolos P2P descentralizados e técnicas de persistência em disco, diferenciando-se substancialmente das famílias baseadas em Telnet (Mirai, Gafgyt) ou em exploração de serviços (Muhstik, Hakai). Todos os modelos obtiveram F1-macro abaixo de 0,46 neste cenário, inclusive RF-50, que atinge 0,999 em Gafgyt. Esse resultado sugere que as estatísticas de fluxo Zeek

convencionais (duração, bytes transferidos, contagem de pacotes) são insuficientes para capturar o comportamento furtivo do H&S, cuja comunicação C&C mimetiza tráfego legítimo em duração e volume. Para superar essa limitação, trabalhos futuros deverão investigar atributos derivados de análise temporal (e.g., periodicidade de conexões, entropia do destino IP) ou representações aprendidas via autoencoders [5], capazes de modelar padrões anômalos não capturáveis por estatísticas agregadas.

A Figura 1 evidencia visualmente o achado central do estudo. A fronteira de Pareto é definida por LogReg (1,0 KB, $F1=0,690$), DT-d5 (4,1 KB, $F1=0,804$), DT-d10 (7,7 KB, $F1=0,851$) e RF-50 (669,5 KB, $F1=0,855$). Acima de DT-d10, o ganho marginal em F1-macro é negligenciável (apenas +0,4 ponto percentual para um modelo 87× maior), enquanto MLP-32, XGB-20, XGB-100 e RF-10 são estritamente dominados, atingem desempenho inferior a DT-d10 com custo de armazenamento maior. Para implantação em gateways IoT, roteadores OpenWRT e dispositivos similares com tipicamente 16–512 MB de armazenamento flash, essa diferença é decisiva: DT-d10 consome 0,002% do orçamento de memória de um roteador comercial mediano, enquanto RF-50 consome 0,13%, viabilizando a coexistência com outros serviços críticos. Trabalhos correlatos sobre detecção leve na borda [5] reforçam que árvores e ensembles compactos tipicamente superam redes neurais densas quando o orçamento computacional é o critério dominante, corroborando a recomendação prática deste estudo.

TABELA III. F1-MACRO POR MODELO EM QUATRO CENÁRIOS REPRESENTATIVOS SOB O PROTOCOLO LOSO. H&S: HIDE-AND-SEEK.

Modelo	Gafgyt	H&S	Mirai	Torii
RF-50	0,999	0,452	0,992	0,869
RF-10	0,998	0,452	0,988	0,854
DT-d10	0,996	0,451	0,992	0,855
XGB-100	1,000	0,452	0,845	0,816
MLP-32	0,979	0,442	0,720	0,832
DT-d5	0,976	0,451	0,821	0,771
XGB-20	0,999	0,450	0,717	0,761
LogReg	0,987	0,675	0,279	0,753

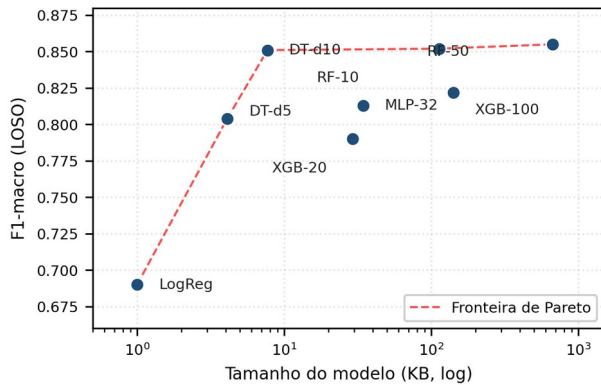


Fig. 1. Fronteira de Pareto entre F1-macro (LOSO) e tamanho do modelo serializado. DT-d10 ocupa o vértice da fronteira: 0,851 de F1 em apenas 7,7 KB.

IV. CONCLUSÕES

Este trabalho comparou oito modelos de aprendizado de máquina para detecção binária de botnets em tráfego IoT, sob protocolos i.i.d. e Leave-One-Scenario-Out, com foco em viabilidade de implantação na borda. Os resultados mostram que a divisão i.i.d. tradicional superestima o desempenho real em aproximadamente 14 pontos percentuais de F1-macro, o que justifica a adoção do LOSO como protocolo padrão na área. Sob avaliação realista, uma Árvore de Decisão de profundidade 10, ocupando apenas 7,7 KB e com latência de 0,07 μ s por fluxo, atinge desempenho equivalente ao de um Random Forest de 669 KB (87 $\times$ maior) e 33 $\times$ mais lento. Como trabalhos futuros, destacam-se a implantação física do detector em hardware embarcado (Raspberry Pi e ESP32) com medição de consumo energético, a quantização int8 dos modelos via TensorFlow Lite Micro e a extensão para classificação multi-rótulo aproveitando os detailed-labels do IoT-23 [1].

Adicionalmente, estudos futuros deverão investigar a inclusão de cenários benignos mais variados, o presente trabalho contou com apenas dois cenários honeypot (Philips HUE e Amazon Echo), totalizando 1.826 fluxos benignos, o que limita a representatividade dessa classe. A incorporação de datasets complementares, como BoT-IoT e CIC-IoT-2023, permitiria validar a transferência dos achados para outros ecossistemas de rede. Por fim, a aplicação de técnicas de explicabilidade (SHAP, LIME) sobre a DT-d10 poderá revelar quais atributos do fluxo Zeek são mais discriminativos para cada família de malware, oferecendo subsídios para a engenharia de regras em firewalls de borda e sistemas de prevenção de intrusões baseados em assinatura.

AGRADECIMENTOS

Os autores agradecem ao Instituto Federal de Santa Catarina (IFSC) – Campus São José pelo apoio institucional à pesquisa.

REFERÊNCIAS

- [1] S. García, A. Parmisano, and M. J. Erquiaga, IoT-23: A Labeled Dataset with Malicious and Benign IoT Network Traffic, version 1.0.0, dataset. Prague, Czech Republic: Stratosphere Laboratory, CTU University, Jan. 2020. doi: 10.5281/zenodo.4743746.
- [2] N. Stoian, *Machine Learning for Anomaly Detection in IoT Networks: Malware Analysis on the IoT-23 Dataset*, 2020.
- [3] F. Hussain et al. “Towards a Universal Features Set for IoT Botnet Attacks Detection”. 2020 IEEE 23rd International Multitopic Conference (INMIC), p. 1–6, 5 Nov. 2020.
- [4] M. Hedge et al. “Identification of Botnet Activity in IoT Network Traffic Using Machine Learning”. 2020 International Conference on Intelligent Data Science Technologies and Applications (IDSTA), 19 Oct. 2020.
- [5] C. Wei; G. Xie; Z. Diao. “A Lightweight Deep Learning Framework for Botnet Detecting at the IoT Edge”. *Computers & Security*, v. 129, p. 103195–103195, 1 June 2023.
- [6] L. Breiman. “Random Forests”. *Machine Learning*, v. 45, n. 1, p. 5–32, Oct. 2001.
- [7] T. Chen; C. Guestrin. XGBoost: “A Scalable Tree Boosting System”. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, v. 1, n. 1, p. 785–794, 13 Aug. 2016.
- [8] F. Pedregosa et al. Scikit-learn: *Machine Learning in Python*. *J. Mach. Learn. Res.*, v. 12, 2011.