

Detecção de Ameaças Zero-Day em Tráfego Criptografado TLS 1.3 com LSTM-MAE e Janelas Deslizantes

Ezekiel Netto¹, Mikael Fontoura², Gabriel Stiegemeier², Marcia Henke¹

¹Colégio Técnico Industrial de Santa Maria
Universidade Federal de Santa Maria (UFSM) – Santa Maria – RS – Brasil

²Centro de Tecnologia – Universidade Federal de Santa Maria (UFSM)
Santa Maria – RS – Brasil

{ezeziel.netto, henke}@redes.ufsm.br,
{mikael.fontoura, gabriel.stiegemeier}@acad.ufsm.br

Abstract. *This study investigates the detection of zero-day attacks in Transport Layer Security (TLS) 1.3 encrypted network traffic using NetFlow statistical metadata from the NF-CSE-CIC-IDS2018-v2 dataset. A Recurrent Masked Autoencoder (LSTM-MAE) extracts temporal patterns from sliding windows and feeds an XGBoost classifier for the final decision. By isolating the Infiltration class to emulate a novel, stealthy attack, experiments revealed unsatisfactory performance for both standalone reconstruction thresholds and the hybrid model, with the XGBoost classifier yielding zero sensitivity (recall). The results suggest that, within the limited NetFlow feature space, infiltration is not detectable as a deviation from benign traffic (reconstruction error ROC-AUC of 0.51); this indicates a potential limitation of reconstruction-error-based models and points to contrastive learning as a promising direction.*

Resumo. *Este trabalho investiga a detecção de ataques zero-day em tráfego de rede sob criptografia Transport Layer Security (TLS) 1.3, utilizando metadados estatísticos NetFlow do conjunto NF-CSE-CIC-IDS2018-v2. Um Masked Autoencoder recorrente (LSTM-MAE) extrai padrões temporais de janelas deslizantes e alimenta um classificador XGBoost na decisão final. Ao isolar a classe Infiltration para emular um ataque inédito e furtivo, os experimentos indicaram desempenho insatisfatório tanto do limiar de reconstrução isolado quanto do modelo híbrido, com sensibilidade (recall) nula do XGBoost. Os resultados sugerem que, no espaço reduzido de atributos do NetFlow, a infiltração não é detectável como desvio em relação ao benigno (ROC-AUC do erro de reconstrução de 0,51), o que aponta uma possível limitação dos modelos baseados em erro de reconstrução e o aprendizado contrastivo como direção promissora.*

1. Introdução

A criptografia *end-to-end* via Transport Layer Security (TLS) 1.3 tornou-se o padrão de fato na Internet corporativa, elevando a privacidade, mas criando um paradoxo fundamental para a segurança de perímetro: ferramentas de Inspeção Profunda de Pacotes (DPI) tornaram-se pouco eficazes diante do ciframento do conteúdo (*payload*) e da ocultação inicial de certificados. Como resultado, a detecção de ameaças passa a depender quase exclusivamente de metadados estatísticos de fluxo extraídos em formatos como *NetFlow* v9 e *IPFIX* (*IP Flow Information Export*).

Paralelamente, ataques do tipo *zero-day* continuam a representar um desafio crítico em segurança, uma vez que as técnicas tradicionais de aprendizado supervisionado tendem a apresentar desempenho reduzido quando confrontadas com ameaças inéditas. Nesse cenário, o Aprendizado Auto-Supervisionado (*Self-Supervised Learning* – SSL) desponta como uma direção promissora, pois dispensa rótulos do ataque-alvo, permitindo que o modelo aprenda o padrão estatístico do tráfego conhecido e trate desvios como anomalias. Neste trabalho, *ameaça* designa o comportamento malicioso em sentido amplo, e *ataque*, sua ocorrência concreta; o ataque inédito (*zero-day*) é emulado aqui pela classe *Infiltration*.

Como o tráfego de rede é naturalmente uma série temporal, adota-se uma arquitetura baseada em *Long Short-Term Memory* (LSTM) acoplada a um *Masked Autoencoder* (MAE). A premissa é mascarar parte dos atributos durante o treinamento, forçando o modelo a reconstruir a entrada com base no contexto remanescente, o que busca ampliar a sensibilidade do erro de reconstrução a desvios sutis de comportamento (anomalias).

Neste trabalho, implementa-se e avalia-se uma arquitetura híbrida de detecção de ataques *zero-day* aplicada exclusivamente sobre telemetria estatística de tráfego TLS 1.3. A base utilizada é o NF-CSE-CIC-IDS2018-v2, sendo a classe *Infiltration* isolada integralmente para o teste, de modo a emular um ataque inédito (*proxy* de cenário *zero-day*). Para aprimorar a detecção baseada apenas no limiar de erro de reconstrução, incorpora-se um classificador *XGBoost* (*eXtreme Gradient Boosting*) como camada de decisão final sobre os dados do *autoencoder*. Os resultados indicam que nem o limiar de reconstrução nem a camada supervisionada detectaram a classe isolada de forma aceitável.

2. Trabalhos Relacionados

A revisão sistemática de [Deldar and Abadi 2023] categoriza a detecção de ataques *zero-day* em modelos não supervisionados, *few-shot* e híbridos, apontando as limitações das abordagens puramente supervisionadas. A partir dessa premissa, a literatura converge para duas linhas principais: a detecção por desvio estatístico e o aprendizado contrastivo no espaço latente.

Fundamentando o desvio estatístico, [Elsayed et al. 2020] propuseram um *LSTM-Autoencoder* treinado exclusivamente sobre tráfego normal. Os autores demonstraram que a representação latente alimentando um classificador *One-Class Support Vector Machine* (OC-SVM) elevou o *F1-score* de 82,50% (limiar simples) para 93,00% (modelo híbrido). Esse padrão inspirou o fluxo arquitetural deste trabalho, com a substituição do OC-SVM por *XGBoost* devido ao desempenho frequentemente competitivo de classificadores baseados em *Gradient Boosting* em dados tabulares [Grinsztajn et al. 2022]. Diferenciando-se da modelagem por reconstrução, [Wilkie et al. 2026] exploram a função de perda contrastiva CLAD (*Contrastive Learning for Anomaly Detection*), utilizando ataques conhecidos como âncoras negativas para distanciar anomalias da normalidade.

No pré-processamento sob criptografia, [Barut et al. 2023] classificaram tráfego malicioso descartando IPs e portas, estratégia seguida neste trabalho. Paralelamente, [Sarhan et al. 2021] padronizaram *datasets* clássicos para o formato *NetFlow*, argumentando que metadados agregados dispensam o uso de DPI; essa base foi posteriormente refinada por [Sarhan et al. 2022], desenvolvendo a versão NF-CSE-CIC-IDS2018-v2 utilizada neste trabalho.

Por fim, a literatura evidencia a profunda furtividade da classe *Infiltration* (“infiltração”), servindo como baliza crítica para nossos resultados. Em regime supervisionado, [Boudaine et al. 2025] reportaram um *recall* de apenas 0,01 para esta categoria, marcando o pior desempenho dentre as 14 classes avaliadas. Similarmente, [Ma et al. 2025], mesmo empregando ajuste fino (*fine-tuning*) de LLM (*Large Language Model*) para aumentar os dados, estagnaram em um F1-score de $\approx 0,47$ para infiltração no conjunto NF-v2. Somado às recomendações de arquiteturas sequenciais e métricas granulares para classes minoritárias de [Xu et al. 2025], este cenário indica que a dificuldade em detectar infiltrações persiste, inclusive sob abordagens complexas e supervisionadas.

3. Metodologia

3.1. Conjunto de Dados e Engenharia de Atributos

O conjunto de dados utilizado é o NF-CSE-CIC-IDS2018-v2, composto por atributos agregados *NetFlow* [Sarhan et al. 2022]. Cada fluxo possui dois rótulos: um binário (0 = benigno; 1 = ataque) e a categoria do ataque. O rótulo binário define o tráfego benigno e os ataques conhecidos usados no treinamento; a categoria é usada apenas para separar a classe *Infiltration*, que, embora rotulada como ataque, foi removida do treinamento e incluída somente no teste. Optou-se pela versão v2 para fins de comparabilidade com trabalhos anteriores que a utilizam, como [Ma et al. 2025].

Para garantir a anonimização e forçar o aprendizado de comportamentos estatísticos em vez de assinaturas específicas, os identificadores diretos de rede (endereços IP e portas de camada 4) foram descartados na ingestão dos dados, seguindo a estratégia adotada por [Barut et al. 2023]. O conjunto totaliza 17.129.715 fluxos, dos quais 88,16% são benignos e apenas 0,67% (115.513 fluxos) pertencem à classe *Infiltration*, desbalanceamento consistente com o relatado por [Sarhan et al. 2022]. Diante disso, aplicou-se amostragem estratificada (10% do benigno e 20% de cada ataque conhecido), preservando 100% da *Infiltration*, conforme detalhado na Seção 3.2, o que resultou em 401.598 janelas.

Como a classe *Infiltration* é conhecida na literatura, ela não é um ataque *zero-day* em sentido estrito; o cenário é emulado por retenção total de classe (*holdout*), alinhado a [Wilkie et al. 2026]: 100% das amostras foram isoladas para o teste e não participaram do treinamento nem da calibração do limiar. O termo *zero-day* é usado, portanto, em sentido operacional.

A divisão treino/validação/teste (70%/10%/20%) foi sequencial, para preservar a ordem cronológica exigida pela continuidade temporal do LSTM-MAE, e segue a proporção de [Wei et al. 2023], que calibra o limiar de reconstrução em uma partição de validação benigna.

Aplicou-se a transformação logarítmica ($\log I_p$) às variáveis volumétricas com alta assimetria (ex.: *bytes*, *pacotes*, *duração*), suavizando valores extremos que distorceriam a escala e preservando os fluxos de valor nulo, comuns em atributos direcionais do *NetFlow*, pois $\log(1 + 0) = 0$. Em seguida, removeram-se os atributos de variância quase nula e, entre pares com alta correlação linear (Pearson), manteve-se apenas um representante, reduzindo os 39 atributos remanescentes (após o descarte dos quatro identificadores de rede) a 22 variáveis. Essa sequência, escalonamento logarítmico seguido de

filtragem por variância e por correlação, segue a de [Corsini and Yang 2023] sobre fluxos do CSE-CIC-IDS2018 em cenário de identificação de ataques não vistos em treinamento. A normalização foi realizada via *RobustScaler*, ajustado estritamente sobre a porção de treinamento para evitar o vazamento de dados (*data leakage*).

3.2. Estruturação Temporal

O tráfego foi estruturado em janelas deslizantes (*sliding windows*) de 15 fluxos consecutivos, com passo 5. Cada janela contém apenas fluxos contíguos da mesma classe, em ordem cronológica original, convertendo os dados tabulares em tensores tridimensionais para análise sequencial [Elsayed et al. 2020]. Como o passo é inferior ao tamanho da janela, há sobreposição parcial entre janelas consecutivas, funcionando como um aumento de dados temporal implícito, conforme evidenciado por [Fang et al. 2022]. A amostragem estratificada (10% do benigno e 20% de cada ataque conhecido) é aplicada às janelas já construídas, e não aos fluxos individuais, para não quebrar a continuidade temporal dentro de cada janela; a *Infiltration* é mantida integralmente e usada apenas no teste. Como cada janela pertence a uma única classe, a mistura de classes típica de capturas reais não é modelada. A janela de 15 fluxos e o passo 5 foram fixados a partir de uma exploração preliminar entre janelas de 5, 10 e 15 fluxos, buscando contexto temporal local com custo computacional moderado.

3.3. Arquitetura Híbrida: LSTM-MAE e XGBoost

A modelagem foi desenvolvida na biblioteca PyTorch, baseada em um *Masked Autoencoder* com blocos recorrentes (LSTM-MAE). A arquitetura é composta por um *encoder* que comprime a sequência mascarada (fração de 0,15 ocultada no treino) em uma dimensão latente, e um *decoder* simétrico focado na reconstrução estatística. O treinamento ocorreu via Erro Quadrático Médio (MSE), utilizando janelas de tráfego benigno e de ataques conhecidos, mantendo a classe de infiltração oculta. O limiar de anomalia corresponde ao percentil 90 do erro de reconstrução das janelas benignas de validação.

Para a etapa de decisão, adotou-se um classificador *XGBoost*. O vetor de entrada do modelo em árvore foi construído pela concatenação das janelas temporais originais achatadas (*flatten*) com o MSE gerado pelo LSTM-MAE, adaptando a fundação arquitetural proposta por [Elsayed et al. 2020]. O classificador foi treinado com janelas benignas (classe 0) e de ataques conhecidos (classe 1); a *Infiltration* é considerada detectada quando classificada como 1.

3.4. Métricas de Avaliação

Dado o severo desbalanceamento do teste, a acurácia global é enganosa: um classificador trivial que sempre prevê a classe negativa (não *Infiltration*) obtém 0,7662, acima do LSTM-MAE isolado. Avalia-se, então, por classe, com o *recall*, $VP/(VP+FN)$, como métrica principal, pois um falso negativo (FN) é um ataque inédito não detectado, complementado por precisão, *F1-score* [Xu et al. 2025], matrizes de confusão (VP, FP e VN: verdadeiros positivos, falsos positivos e verdadeiros negativos) e, para avaliar o erro de reconstrução sem depender do limiar, a área sob a curva ROC (ROC-AUC).

4. Resultados e Discussão

Na configuração adotada ($WINDOW_SIZE=15$, passo=5, 30 épocas, limiar no percentil 90 do erro de reconstrução benigno de validação, calibrado em 5,7904), o LSTM-MAE

isolado obteve *recall* de 0,1166, precisão de 0,2944 e F1-score de 0,1670 para *Infiltration*, com acurácia global de 0,7281, inferior à de um classificador trivial que sempre previsse a classe negativa (0,7662). Pela matriz de confusão (Tabela 1; 98.795 janelas de teste), o elevado volume de FN mostra que o erro de reconstrução falha em discriminar a classe-alvo, e o volume expressivo de FP (6.455, dos quais cerca de 6.113 são janelas benignas) indica que o mesmo limiar sinaliza também o tráfego benigno, evidência de forte sobreposição estatística entre as distribuições de erro no espaço reduzido de atributos *NetFlow*.

Tabela 1. Comparativo de Desempenho (*Infiltration* = positivo; FP inclui ataques conhecidos sinalizados).

Modelo	Classe	VP	FN	FP	VN
LSTM-MAE	Infiltration	2.693	20.407	6.455	69.240
LSTM-MAE + XGBoost	Infiltration	0	23.100	15.289	60.406

A incorporação do *XGBoost* como decisor final reduziu a acurácia global para 0,6114 e zerou o *recall* sobre *Infiltration*: das 23.100 janelas dessa classe, todas recaíram em Falso Negativo (VP=0, FN=23.100, FP=15.289, VN=60.406). Ao contrário do ganho de F1 de 0,825 para 0,93 relatado por [Elsayed et al. 2020] com o OC-SVM, aqui o classificador não extraiu sinal aproveitável, não por limitação própria, mas porque o sinal parece já ser indistinguível do benigno: apenas 11,66% das janelas de *Infiltration* ultrapassaram o limiar de anomalia, contra 10,12% do benigno de teste, valor compatível com os 10% esperados pelo percentil 90.

Essa sobreposição independe do limiar: a ROC-AUC do erro de reconstrução entre *Infiltration* e benigno foi de 0,5099, praticamente equivalente ao acaso. Já os ataques conhecidos ultrapassaram esse limiar em apenas 2,24% das janelas, como esperado por terem participado do treinamento. Esse achado é consistente com a furtividade da classe documentada por [Boudaine et al. 2025] e sugere uma possível limitação de abordagens baseadas só em reconstrução sobre metadados *NetFlow* agregados, o que aproxima os resultados da direção contrastiva de [Wilkie et al. 2026].

5. Conclusão

Este trabalho avaliou a detecção de ataques *zero-day* emulados em tráfego criptografado *TLS* 1.3, operando exclusivamente sobre metadados estatísticos *NetFlow*, com um *LSTM Masked Autoencoder* (LSTM-MAE) e um classificador *XGBoost*, sob retenção estrita da classe *Infiltration*. Nem o limiar simples sobre o erro de reconstrução (*recall* de 0,1166) nem a camada adicional com *XGBoost* (*recall* nulo) detectaram a classe de forma minimamente aceitável.

No espaço de 22 atributos, a *Infiltration* não se mostrou separável do tráfego benigno, o que é coerente com a dificuldade relatada por abordagens supervisionadas [Boudaine et al. 2025, Ma et al. 2025] e aponta uma possível limitação dos modelos baseados em erro de reconstrução para ataques furtivos.

Como trabalhos futuros, pretende-se explorar funções de perda contrastivas, como CLAD, incorporar atributos temporais adicionais entre fluxos e validar a arquitetura em outras composições de tráfego e classes de ataque furtivo.

Referências

- Barut, O., Luo, Y., Li, P., and Zhang, T. (2023). R1dit: Privacy-preserving malware traffic classification with attention-based neural networks. *IEEE Transactions on Network and Service Management*, 20:2071–2085.
- Bouidaine, A. B., Moussaoui, D., Hadjila, M., Ferhi, W., and Hachemi, M. H. (2025). Deep learning-based anomaly and intrusion detection using the cse-cic-ids2018 dataset. *Engineering, Technology & Applied Science Research*, 15:24782–24787.
- Corsini, A. and Yang, S. J. (2023). Are existing out-of-distribution techniques suitable for network intrusion detection? In *2023 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9. IEEE.
- Deldar, F. and Abadi, M. (2023). Deep learning for zero-day malware detection and classification: A survey. *ACM Comput. Surv.*, 56(2).
- Elsayed, M. S., Le-Khac, N.-A., Dev, S., and Jurcut, A. D. (2020). Network anomaly detection using lstm based autoencoder. In *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pages 37–45. ACM.
- Fang, C., Mi, W., Han, P., and Zhai, L. (2022). A method of network traffic anomaly detection based on packet window transformer. In *2022 7th IEEE International Conference on Data Science in Cyberspace (DSC)*, pages 199–205.
- Grinsztajn, L., Oyallon, E., and Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 507–520.
- Ma, H., Zhang, W., Zhang, D., and Chen, B. (2025). An iot intrusion detection framework based on feature selection and large language models fine-tuning. *Scientific Reports*, 15:21158.
- Sarhan, M., Layeghy, S., Moustafa, N., and Portmann, M. (2021). *NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems*, page 117–135. Springer International Publishing.
- Sarhan, M., Layeghy, S., and Portmann, M. (2022). Towards a standard feature set for network intrusion detection system datasets. *Mobile Networks and Applications*, 27(1):357–370.
- Wei, Y., Jang-Jaccard, J., Sabrina, F., Xu, W., Camtepe, S., and Dunmore, A. (2023). Reconstruction-based LSTM-autoencoder for anomaly-based DDoS attack detection over multivariate time-series data. *Preprint*, arXiv:2305.09475.
- Wilkie, J., Hindy, H., Michie, C., Tachtatzis, C., Irvine, J., and Atkinson, R. (2026). A novel contrastive loss for zero-day network intrusion detection. *IEEE Transactions on Network and Service Management*, 23:2064–2076.
- Xu, C., Zhan, Y., Chen, G., Wang, Z., Liu, S., and Hu, W. (2025). Elevated few-shot network intrusion detection via self-attention mechanisms and iterative refinement. *PLOS ONE*, 20:e0317713.