Avaliação de algoritmos de *machine learning* para detecção de *malware* IoT no *dataset* IoT-23

Cristian H. M. Souza[1], **Carlos H. Arima**[1]

[1]Centro Estadual de Educação Tecnológica Paula Souza (CEETEPS)
São Paulo-SP, Brasil

cristianmsbr@gmail.com, carlos.arima@cpspos.sp.gov.br

Abstract. *This article presents an evaluation of different machine learning algorithms for malware detection in IoT devices using the IoT-23 dataset. Models based on Random Forest, SVM, decision tree, and a convolutional neural network were implemented and compared. The results show that the Random Forest algorithm achieved the highest accuracy, while the convolutional neural network and Random Forest obtained the best precision and F1-Score metrics. The data preprocessing methodology and evaluation metrics are detailed, providing a comprehensive overview of the models' effectiveness and guiding future research.*

Resumo. *Este artigo apresenta uma avaliação de diferentes algoritmos de machine learning para detecção de malware em dispositivos IoT utilizando o dataset IoT-23. Modelos baseados nos algoritmos Random Forest, SVM, árvore de decisão e uma rede neural convolucional foram implementados e comparados. Os resultados evidenciam que o algoritmo Random Forest alcançou a maior acurácia, enquanto a rede neural convolucional e também o Random Forest obtiveram as melhores métricas de precisão e F1-Score. A metodologia de pré-processamento de dados e as métricas de avaliação são detalhadas, proporcionando uma visão abrangente da eficácia dos modelos e guiando pesquisas futuras.*

1. Introdução

Falhas de segurança em dispositivos IoT podem ser críticas à privacidade dos usuários e à segurança pessoal. O uso de protocolos inseguros pode expor informações sensíveis dos usuários a invasores por meio da interceptação do tráfego da rede. Ademais, um dispositivo comprometido (e.g., em uma rede industrial) pode ser utilizado para realização de ações no ambiente físico, ocasionando riscos a vidas humanas [Yang et al. 2022].

Malwares continuam sendo um dos principais desafios à segurança de sistemas computacionais. O advento do paradigma IoT foi acompanhado pelo aumento do número de programas maliciosos projetados para as arquiteturas ARM (do inglês *Advanced RISC Machine*) e MIPS (do inglês *Microprocessor without Interlocked Pipeline Stages*) [Darki et al. 2019]. Um exemplo disso é a utilização da *botnet* Mirai [Kumar and Chandavarkar 2023], cujo objetivo é infectar e controlar dispositivos embarcados (como câmeras IP e roteadores) para execução de ataques de negação de serviço distribuídos (DDoS, do inglês *Distributed Denial of Service*) [Kolias et al. 2017].

Essas ameaças são responsáveis por diversos danos, como o comprometimento da integridade dos dados, roubo de informações confidenciais e prejuízos financeiros a usuários e corporações. Recentemente, *malwares* do tipo *ransomware* estão sendo amplamente utilizados por criminosos para impedir o acesso a arquivos por meio da encriptação das informações, exigindo um pagamento para resgate dos dados [Razaulla et al. 2023]. A efetividade dos ataques por *ransomware* deu origem ao mercado criminoso de *Ransomware-as-a-Service* (RaaS), tornando a compra de ameaças avançadas acessível a todos os públicos [Alwashali et al. 2021].

Segundo o *ICS and OT threat predictions for 2024* [Goncharov 2024], publicado pela Kaspersky Lab, empresa referência em segurança cibernética, *ransomwares* continuam sendo uma das principais ameaças aos ambientes industriais. Ademais, o uso generalizado de IoT em conjunto com a tecnologia 5G está gerando diversas discussões a respeito da segurança de tais dispositivos e do ambiente em que operam [Salahdine et al. 2023].

Diversos avanços foram e vêm sendo feitos pela indústria e academia nas áreas de análise e detecção de artefatos maliciosos [Gopinath and Sethuraman 2023]. Ferramentas de detecção devem ser capazes de identificar e conter ameaças desconhecidas [Balogh et al. 2022]. Para isso, múltiplas técnicas de análise devem ser empregadas para aumentar a confiabilidade das soluções, sendo a análise comportamental a mais efetiva na detecção de *malwares* de dia zero [Aboaoja et al. 2022].

Nesse contexto, o uso de *machine learning* tem se mostrado efetivo na detecção genérica de artefatos maliciosos em nível de rede [Gaurav et al. 2023, Tayyab et al. 2022]. Para isso, um modelo é treinado com base em uma grande quantidade de dados a respeito de ameaças conhecidas. Uma vez treinado, ele é capaz de realizar previsões assertivas ao lidar com novas informações. Isso permite a detecção e classificação efetiva das ameaças com base na análise de tráfego, minimizando o risco de comprometimentos [Alqudah and Yaseen 2020].

Portanto, este trabalho propõe uma avaliação de algoritmos de *machine learning* para classificação de *malware* IoT com base no *dataset* IoT-23 [Garcia et al. 2020], visando auxiliar pesquisadores de segurança na escolha e implementação de modelos para detecção de artefatos maliciosos em tais ambientes. Foram implementados os algoritmos de *Random Forest*, SVM e uma árvore de decisão, além de uma rede neural convolucional. Os modelos são comparados com base nas métricas de acurácia, precisão, *recall* e F1-Score.

O restante deste trabalho está organizado da seguinte maneira: a Seção 2 apresenta a metodologia utilizada para treinamento dos modelos, incluindo detalhes sobre o *dataset* utilizado; a Seção 3 expõe as implementações e resultados; e, por fim, a Seção 4 apresenta as considerações finais e sugestões para trabalhos futuros.

2. Metodologia

O *dataset* utilizado para treinamento dos modelos propostos neste estudo é o IoT-23 [Garcia et al. 2020], criado pelo Avast AIC Laboratory. Esta base contém 20 capturas de *malwares* coletadas de diversos dispositivos IoT, além de 3 capturas de tráfego benigno. Seu objetivo é fornecer aos pesquisadores um grande conjunto de dados reais e

rotulados de infecções e tráfego legítimo, visando auxiliar o desenvolvimento de algoritmos de *machine learning*.

O motivo da escolha desse *dataset* é a sua ampla utilização em estudos anteriores [Souza and Arima 2024, Jeelani et al. 2022, Oha et al. 2021]. Em números totais, o *dataset* possui 325.307.990 registros, sendo 294.449.255 deles maliciosos. Os tipos de ameaças presentes na base são descritos na Tabela 1 e suas colunas são detalhadas na Tabela 2.

Tipo de ameaça	**Descrição**
Attack	Anomalias que não puderam ser identificadas e classificadas.
Benign	Tráfego benigno.
C&C	Tráfego gerado pela comunicação entre um dispositivo infectado e uma estação de comando e controle.
DDoS	Tráfego gerado por ataques de negação de serviço distribuídos.
FileDownload	Tráfego gerado pela transferência de arquivos maliciosos.
HeartBeat	Tráfego gerado pela estação de C&C para verificar a conexão com o alvo.
Mirai	Tráfego que possui características da *botnet* Mirai.
Okiru	Tráfego que possui características da *botnet* Okiru.
PartOfAHorizontalPortScan	Tráfego gerado por *scanners* de rede para coleta de informações.
Torii	Tráfego que possui características da *botnet* Torii.

Tabela 1. Tipos de ameaças presentes no *dataset*

Coluna	**Tipo**	**Descrição**
`ts`	`int`	Horário da captura no formato Unix Timestamp.
`uid`	`str`	ID da captura.
`id.orig_h`	`str`	Endereço IPv4 ou IPv6 do dispositivo que originou o ataque.
`id.orig_p`	`int`	Porta utilizada pelo dispositivo que originou o ataque.
`id.resp_h`	`str`	Endereço IPv4 ou IPv6 do dispositivo onde a captura foi realizada.
`id.resp_p`	`int`	Porta utilizada pelo dispositivo alvo onde a captura foi realizada.
`proto`	`str`	Protocolo de rede utilizado.
`service`	`str`	Protocolo de aplicação utilizado.
`duration`	`float`	Duração da troca de dados entre o dispositivo infectado e o atacante.
`orig_bytes`	`int`	Quantidade de dados enviados ao dispositivo.
`resp_bytes`	`int`	Quantidade de dados enviados pelo dispositivo.
`conn_state`	`str`	Estado da conexão.
`local_orig`	`bool`	Identifica se a conexão foi originada localmente.
`local_resp`	`bool`	Identifica se a resposta foi originada localmente.
`missed_bytes`	`int`	Número de *bytes* faltantes.
`history`	`str`	Histórico do estado da conexão.
`orig_pkts`	`int`	Número de pacotes sendo enviados ao dispositivo.
`orig_ip_bytes`	`int`	Número de *bytes* sendo enviados ao dispositivo.
`resp_pkts`	`int`	Número de pacotes sendo enviados pelo dispositivo.
`resp_ip_bytes`	`int`	Número de *bytes* sendo enviados dispositivo.
`tunnel_parents`	`str`	Identificador da conexão, se ela for tunelada.
`label`	`str`	Tipo de captura (maliciosa ou benigna).
`detailed_label`	`str`	Detalhes sobre a captura, caso ela seja maliciosa.

Tabela 2. Colunas presentes no *dataset*

Visto que as informações do *dataset* são distribuídas em diretórios, é necessário realizar o tratamento dos dados. Para isso, as informações de cada diretório foram combinadas em um único arquivo CSV, que é utilizado para treinar os modelos.

2.1. Pré-processamento dos dados

A etapa de pré-processamento dos dados é fundamental para preparar e limpar os dados brutos antes de sua utilização no treinamento dos modelos, eliminando inconsistências que podem afetar a qualidade dos resultados. As seguintes operações foram realizadas após a leitura do arquivo CSV unificado:

1. **Remoção de colunas não importantes:** as colunas `ts`, `uid` e `tunnel_parents` foram removidas por não representarem dados relevantes para treinamento dos modelos. Já as colunas `local_orig` e `local_resp` foram eliminadas por não serem únicas.
2. ***Label encoding*:** esta técnica é utilizada para converter dados categóricos em valores numéricos, e foi aplicada para as colunas `id.orig_h`, `id.resp_h`, `proto`, `service`, `conn_state` e `history`.
3. **Substituição de valores ausentes:** linhas sem dados nas colunas `duration`, `orig_bytes` ou `resp_bytes` tiveram os valores ausentes configurados como a média da respectiva coluna.
4. ***Feature scaling*:** esta técnica foi utilizada para melhorar o desempenho dos modelos e reduzir o impacto de diferenças nas escalas das variáveis consideradas.
5. **Separação do conjunto de treinamento e de teste:** o *dataset* foi dividido em conjuntos de treinamento e de teste na proporção de 7:3.

3. Implementações e resultados

Foram escolhidos quatro algoritmos diferentes para serem avaliados: *Random Forest*, SVM, Árvore de Decisão e uma Rede Neural Convolucional (CNN). A escolha desses algoritmos se deu pela sua relevância e popularidade na literatura para problemas de classificação, especialmente na área de cibersegurança. Foi utilizada a linguagem de programação Python, com o apoio das bibliotecas *scikit-learn*[1] e TensorFlow[2].

Para a rede neural convolucional, é adicionada uma camada de *max pooling* 1D com `pool_size=2`, o que reduz a dimensionalidade dos recursos. Uma camada *flatten* é adicionada à rede, transformando os recursos 2D resultantes da camada de *max pooling* em um vetor 1D, que é utilizado como entrada para as camadas totalmente conectadas. São adicionados 500 neurônios na camada totalmente conectada. O otimizador Adam é escolhido para ajustar os pesos da rede.

O modelo baseado em árvores de decisão é configurado com as opções padrão da biblioteca *scikit-learn*. Já o algoritmo *Random Forest* é iniciado com dois *jobs* paralelos, `random_state=0` e o número de árvores é configurado para 100. Isso permite que os resultados sejam facilmente reproduzidos a partir do mesmo conjunto de dados.

Por fim, o modelo baseado no algoritmo SVM é configurado com parâmetro de regularização 1, visando evitar classificações incorretas. O valor de *cache* do *kernel* foi definido como 700MB.

As métricas utilizadas para determinar a efetividade dos modelos foram: acurácia, precisão, *recall* e F1-Score. A acurácia indica a performance geral do modelo (quantidade de classificações corretas); a precisão mede a proporção de exemplos classificados como positivos pelo modelo que são realmente positivos; o *recall* consiste na proporção de exemplos positivos que foram corretamente classificados pelo modelo; e o F1-Score representa a média harmônica entre a precisão e o *recall*. A Tabela 3 apresenta os resultados obtidos após o treinamento dos modelos.

Como exposto na Tabela 3, o modelo de melhor acurácia foi o treinado com o algoritmo *Random Forest*, atingindo a marca de 99.33%. Os algoritmos com precisões

[1]https://scikit-learn.org/

[2]https://www.tensorflow.org/

Algoritmo	**Acurácia**	**Precisão**	***Recall***	**F1-Score**
CNN	92.83%	0.97	0.99	0.98
Decision Tree	97.33	0.93	0.99	0.96
Random Forest	99.33%	0.97	0.99	0.98
SVM	94%	0.91	0.93	0.92

Tabela 3. Métricas dos modelos treinados

mais elevadas foram o *Random Forest* e a CNN, ambos com 0.97. O *recall* de todos os modelos atingiu um valor satisfatório. Por fim, os algoritmos de melhor F1-Score também foram o *Random Forest* e a rede neural, ambos com 0.98.

4. Conclusão e trabalhos futuros

Este trabalho apresenta uma avaliação de diferentes algoritmos de *machine learning* para a detecção de *malware* em dispositivos IoT, utilizando o *dataset* IoT-23. Os resultados evidenciam que o algoritmo *Random Forest* obteve a melhor performance geral, seguido pela CNN. A alta acurácia e os valores robustos de precisão e *recall* indicam que esses modelos são altamente eficazes na classificação de tráfego malicioso em redes IoT.

Como trabalhos futuros, pretende-se avaliar o desempenho dos algoritmos contra artefatos maliciosos não presentes no *dataset* IoT-23, com o objetivo de mensurar a acurácia dos modelos em cenários reais. Ademais, *malwares* podem utilizar diversas técnicas de evasão para não serem detectados. Logo, avaliar o comportamento dos modelos considerando artefatos especializados na evasão de defesas pode fornecer dados importantes para aprimorar a capacidade de detecção dos algoritmos.

Referências

Aboaoja, F. A., Zainal, A., Ghaleb, F. A., Al-rimy, B. A. S., Eisa, T. A. E., and Elnour, A. A. H. (2022). Malware detection issues, challenges, and future directions: A survey. *Applied Sciences*, 12(17):8482.

Alqudah, N. and Yaseen, Q. (2020). Machine learning for traffic analysis: a review. *Procedia Computer Science*, 170:911–916.

Alwashali, A. A. M. A., Abd Rahman, N. A., and Ismail, N. (2021). A survey of ransomware as a service (raas) and methods to mitigate the attack. In *2021 14th International Conference on Developments in eSystems Engineering (DeSE)*, pages 92–96. IEEE.

Balogh, Š., Mojžiš, J., and Krammer, P. (2022). Evaluation of system features used for malware detection. In *Proceedings of the Future Technologies Conference (FTC) 2021, Volume 3*, pages 46–59. Springer.

Darki, A., Faloutsos, M., Abu-Ghazaleh, N., Sridharan, M., et al. (2019). {IDAPro} for {IoT} malware analysis? In *12th USENIX Workshop on Cyber Security Experimentation and Test (CSET 19)*.

Garcia, S., Parmisano, A., and Erquiaga, M. J. (2020). IoT-23: A labeled dataset with malicious and benign IoT network traffic. More details here https://www.stratosphereips.org /datasets-iot23.

Gaurav, A., Gupta, B. B., and Panigrahi, P. K. (2023). A comprehensive survey on machine learning approaches for malware detection in iot-based enterprise information system. *Enterprise Information Systems*, 17(3):2023764.

Goncharov, E. (2024). Ics and ot threat predictions for 2024.

Gopinath, M. and Sethuraman, S. C. (2023). A comprehensive survey on deep learning based malware detection techniques. *Computer Science Review*, 47:100529.

Jeelani, F., Rai, D. S., Maithani, A., and Gupta, S. (2022). The detection of iot botnet using machine learning on iot-23 dataset. In *2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM)*, volume 2, pages 634–639. IEEE.

Kolias, C., Kambourakis, G., Stavrou, A., and Voas, J. (2017). Ddos in the iot: Mirai and other botnets. *Computer*, 50(7):80–84.

Kumar, S. and Chandavarkar, B. (2023). Analysis of mirai malware and its components. In *Machine Learning, Image Processing, Network Security and Data Sciences: Select Proceedings of 3rd International Conference on MIND 2021*, pages 851–861. Springer.

Oha, C. V., Farouk, F. S., Patel, P. P., Meka, P., Nekkanti, S., Nayini, B., Carvalho, S. X., Desai, N., Patel, M., and Butakov, S. (2021). Machine learning models for malicious traffic detection in iot networks/iot-23 dataset. In *International Conference on Machine Learning for Networking*, pages 69–84. Springer.

Razaulla, S., Fachkha, C., Markarian, C., Gawanmeh, A., Mansoor, W., Fung, B. C., and Assi, C. (2023). The age of ransomware: A survey on the evolution, taxonomy, and research directions. *IEEE Access*.

Salahdine, F., Han, T., and Zhang, N. (2023). Security in 5g and beyond recent advances and future challenges. *Security and Privacy*, 6(1):e271.

Souza, C. H. and Arima, C. H. (2024). A hybrid approach for malware detection in sdn-enabled iot scenarios. *Internet Technology Letters*, page e534.

Tayyab, U.-e.-H., Khan, F. B., Durad, M. H., Khan, A., and Lee, Y. S. (2022). A survey of the recent trends in deep learning based malware detection. *Journal of Cybersecurity and Privacy*, 2(4):800–829.

Yang, X., Shu, L., Liu, Y., Hancke, G. P., Ferrag, M. A., and Huang, K. (2022). Physical security and safety of iot equipment: A survey of recent advances and opportunities. *IEEE Transactions on Industrial Informatics*, 18(7):4319–4330.