Análise de Trafego de Rede com Machine Learning para Identificação de Ameaças IoT

Matheus Magalhães de Paula Paiva, Felipe Augusto Pereira de Figueiredo

***Resumo* – Com o avanço da tecnologia, dispositivos IoT estão cada vez mais presentes no dia-a-dia das pessoas, tanto em contextos particulares quanto públicos. Dessa forma, a segurança desses dispositivos é um ponto que deve ser tratado com atenção. Este trabalho tem como proposta uma abordagem para detecção de ameaças baseada em análise de tráfego de rede, realizada através de modelos de aprendizado de máquina. Após fazer a experimentação e avaliação, foi possível produzir um modelo rapidamente treinável e altamente confiável, comprovando a eficácia da proposta.**

***Palavras-Chave*— Internet das Coisas, cibersegurança, aprendizado de máquina, ataques, detecção.**

***Abstract* – With the advancement of technology, IoT devices are increasingly present in people's daily lives, both in private and public contexts. In this way, the security of these devices is a point that must be treated with attention. This work proposes an approach for threat detection based on network traffic analysis, performed through machine learning models. After experimenting and evaluating, it was possible to produce quickly trainable and highly reliable model, proving the effectiveness of the proposal.**

***Keywords*— Internet of Things, cybersecurity, machine learning, attacks, detection.**

I.INTRODUÇÃO

A medida que a tecnologia avança no mundo, é crescente o número de dispositivos IoT (*Internet of Things*) conectados à rede desempenhando diversos papéis, desde monitoramento residencial à ambientes críticos. Como um resultado da eliminação da necessidade de operadores humanos, os dispositivos IoT. podem processar mais dados do que nunca e de uma forma mais rápida e eficiente. Sendo assim, a segurança desses dispositivos surge como um tema de grande importância. Para tratar este tema, pode-se empregar soluções tanto a nível de *hardware* quando de *software* (como *firewalls*, restringindo a comunicação entre os dispositivos com base em suas regras). Uma abordagem inovadora, que vem chamando atenção da comunidade científica é o emprego de modelos de aprendizado de máquina para analisar o tráfego de rede e, assim, identificar possíveis ameaças a dispositivos IoT.

Utilizando o conjunto de dados IoT-23[1], este sendo apresentado mais detalhadamente na seção Metodologia, Stoian[2] comparou modelos de aprendizado, utilizando os algoritmos *Random Forest*(RF), *Naive Bayes* (NB), *Multi Layer Perceptron*(MLP), uma variante da classe de algoritmos de Rede Neural Artificial, *Support Vector Machine* (SVM) e *Adaboost* (ADA), obtendo resultados promissores, sendo o melhor deles com acurácia de 99,5%.

No mesmo ano, utilizando o conjunto de dados Bot-IoT[3], os autores de Shafiq et al.[4] propuseram uma nova métrica para seleção de características denominada CorrAUC, baseada na combinação de outras duas métricas: CAE(*Correlation Attribute Evaluation*) e AUC (*Area Under the Curve*). Foram utilizados 4 modelos de aprendizado, sendo eles: árvore de decisão C4.5, *Naive Bayes*, *Random Forest* e *Support Vector Machine*. A proposta também apresentou resultados promissores, obtendo uma precisão na detecção de tráfego malicioso acima de 96%.

Uma vez comprovada a relevância do tema, este trabalho tem como objetivo escolher, treinar, otimizar e avaliar modelos de aprendizado de máquina aplicados à identificação de ameaças a dispositivos IoT.

O trabalho está organizado da seguinte forma: na seção I o problema está sendo introduzido, e também são visitadas algumas abordagens da literatura para resolvê-lo. Na seção II está sendo detalhada a metodologia da proposta. Na seção III é apresentado os resultados experimentais, enquanto na seção IV se encontra a conclusão do trabalho.

II.METODOLOGIA

De uma forma resumida, este trabalho visa treinar e otimizar modelos de aprendizado de máquina, utilizando amostras de tráfego de rede,

obtendo como resultado um mecanismo capaz de identificar possíveis ameaças a dispositivos conectados em ambientes reais.

O diagrama apresentado na figura I ilustra as etapas do modelo, que são detalhadas nas subseções posteriores.

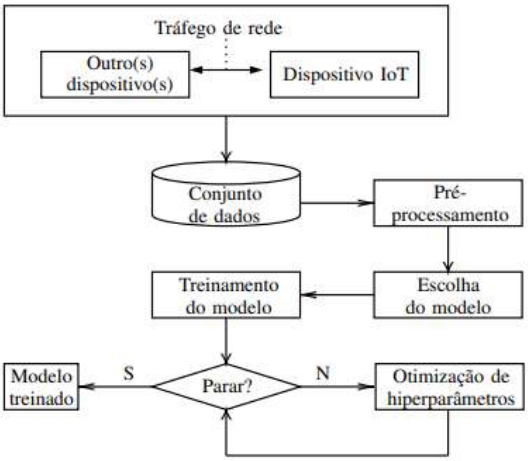

Fig 1: Diagrama de blocos do modelo experimental

A. Conjunto de dados

O conjunto de dados utilizado nesse trabalho foi o IoT-23, encontrado no laboratório *Stratosphere*. Este conjunto de dados é um conjunto de tráfego de rede de dispositivos IoT, publicado pela primeira vez em janeiro de 2020, com capturas variando de 2018 a 2019. Tem como objetivo oferecer um grande conjunto de dados de infecções reais e rotuladas de *malware* de IoT e tráfego benigno de IoT para que os pesquisadores possam desenvolver algoritmos de aprendizado de máquina. Possui 20 capturas de *malware* executadas em dispositivos IoT e 3 capturas para tráfego de dispositivos IoT benignos.

É possível obter 2 versões do conjunto de dados, sendo uma versão reduzida (8,8*GBytes*), contendo apenas as amostras rotuladas (tipo de dado previamente identificado), e também a versão completa (21 *GBytes*), incluindo amostras sem o rótulo associado. Para este trabalho foi utilizada a primeira versão, tanto pelo fato de se desejar classificar os dados, quanto por limitações de poder computacional.

B. Pré-processamento dos dados

Apesar de estar utilizando o conjunto de dados reduzido, o volume de dados é grande e demanda tratamento. Sendo assim, foram realizadas diversas etapas de pré-processamento, visando simultaneamente minimizar a quantidade de dados e maximizar a quantidade de informação. Os processos a seguir foram realizados utilizando as ferramentas *scikit-learn*[5], *pandas*[6] e *scipy*[7] :

1) Concatenação dos 23 conjuntos de dados, com o objetivo de produzir um único arquivo, facilitando o processamento nas etapas posteriores;
2) Padronização dos rótulos das classes.
3) Remoção das amostras que não correspondem aos cenários mais relevantes, considerando os quatro cenários mais frequentes no conjunto de dados, sendo eles:
   - Tráfego benigno: 48,59% (≈4,3M amostras);
   - DDoS: 24,76% (≈2,2M amostras);
   - *Horizontal Port Scan*: 26,48% (≈2,3M amostras);
   - Okiru: 0,17% (≈15,2k amostras).
4) Preenchimento dos valores ausentes com caracteres indicadores padronizados;
5) Remoção de amostras duplicadas;
6) Codificação de características categóricas usando *one hot encoding* (ex.: a característica categórica proto, referente ao protocolo utilizado (“TCP” ou “UDP”) foi codificada em duas características numéricas (proto_tcp e proto_udp), podendo assumir valor 0 ou 1;
7) Definição dos tipos de dados de cada característica, assegurando que não seja demandada mais memória do que o necessário.
8) Separação dos dados em subconjunto de treinamento (80% das amostras) e subconjunto de testes (20% das amostras), usados para treinamento e avaliação dos modelos de aprendizado de máquina respectivamente;
9) Seleção das características, mantendo aquelas que apresentam alta variância, por apresentarem maior potencial preditivo;
10) Escalonamento dos valores numéricos para a faixa [0,1]
11) Codificação dos rótulos das amostras usando números ordinais, agilizando na identificação dos dados pertencentes a cada cenário.

C. Identificação dos modelos de aprendizado de máquina

Uma vez que os dados estejam processados, deve-se identificar os modelos de aprendizado de máquina para realizar a tarefa de classificação. Optou-se por esta abordagem, pois a capacidade de aprender por meio de exemplos permitem que os modelos se adaptem à distribuição subjacente dos dados e realizem o processo de inferência rapidamente. Estes modelos também podem ser configurados para indicar *outliers* e/ou anomalias nos dados.

Os modelos considerados neste trabalho são: árvores de decisão, *Adaboost*, *Random Forest*,

modelo linear (*Perceptron*) e modelo baseado em distância entre amostras (*Nearest Centroid*).

Mesmo concebidos a partir de fundamentos teóricos distintos, e apresentando algumas particularidades de treinamento e operação, os modelos servem ao mesmo propósito da classificação.

D. *Treinamento dos modelos e otimização de hiperparâmetros*

Primeiramente deve-se "treinar" os modelos, utilizando uma fração do conjunto de dados para que apresentem um resultado satisfatório na classificação dos dados que não foram utilizados no treinamento.

O sucesso ou fracasso desse processo, realizado por meio da minimização iterativa de uma função de erro, está associado à configuração inicial dos hiperparâmetros. Dessa forma, deve-se avaliar diversas combinações de valores de modo que o modelo possa ter seu desempenho otimizado, maximizando a acurácia na classificação. Para isso foi utilizada a biblioteca optuna[8], e os espaços de hiperparâmetros foram especificados manualmente, considerando suas particularidades ao se determinar as faixas e/ou conjuntos de valores aceitáveis de cada hiperparâmetro. Cada modelo foi submetido a uma rodada de otimização realizada pela biblioteca optuna.

III. Experimentos e discussão

Como primeiro resultado esperado, deseja-se obter um classificador com a maior precisão de discernimento entre os tipos de ameaças aos dispositivos. Para que esse objetivo fosse atingido, foi utilizada a métrica "acurácia", responsável por contabilizar exatamente quantas previsões feitas pelo classificador coincidem com o tipo de ameaça em questão. Na tabela abaixo foi elaborado um ranking dos classificadores, dispostos em ordem decrescente de acurácia.

TABELA I: *Ranking* de acurácia dos classificadores

Posição	Classificador	Acurácia (%)
1	*Decision Tree*	99,86
	Random Forest	99,86
2	*Adaboost*	93,46
3	*Perceptron*	73,40
4	*Nearest Centroid*	59,51

Além de obter uma alta acurácia, também é desejável que o classificador possa ser (re)treinado rapidamente. Isso ocorre devido ao fato de que em cenários dinâmicos, as características do ataque podem mudar, assim demandando uma rápida adaptação, ou mesmo uma recriação do modelo, para que continue sendo útil a tarefa de detecção de ameaças. Sendo assim, está disposto na tabela II um ranking dos classificadores, em ordem decrescente de "quantidade de treinamentos realizados em uma hora com, no mínimo uma convergência".

TABELA II: *Ranking* de treinamento dos classificadores

Posição	Classificador	Treinamentos
1	*Decision Tree*	100
	Nearest Centroid	100
2	*Perceptron*	83
3	*Random Forest*	18
4	*Adaboost*	10

Analisando os resultados das tabelas I e II acima, é possível fazer as seguintes afirmações:

- Modelos simples são treinados mais rapidamente, atingindo convergência múltiplas vezes dentro da janela de tempo estabelecida.
- Árvores de decisão e *ensembles* produziram os melhores resultados, tanto em termo de acurácia quanto em tempo de treinamento. Outra vantagem desses modelos é que é possível compreender o que foi aprendido pelo modelo.

IV. Conclusão

Com o aumento de dispositivos IoT no mercado, tendo uma grande participação no dia-a-dia das pessoas, as questões de segurança dos dispositivos vêm demandando muita atenção.

O trabalho propôs uma metodologia de treinamento, avaliação e aprimoramento de modelos de aprendizado de máquina, sugerindo e detalhando desde o pré-processamento dos dados até a otimização de hiperparâmetros.

Diante da análise dos resultados experimentais, foi possível comprovar a eficácia da metodologia proposta, pois foi capaz de produzir um modelo cuja capacidade de detecção se aproximou de 99,9% de acurácia.

As técnicas de pré-processamento empregadas também são muito importantes, pois elas aumentam a relevância dos dados e reduzem o consumo computacional do treinamento.

Como sugestões para trabalhos futuros, pode-se sugerir o emprego de técnicas mais sofisticadas de pré-processamento, utilização de outros conjuntos de dados, utilização de outros modelos de aprendizado de máquina e também a

utilização de um modelo de aprendizado não supervisionado capaz de armazenar novos dados e fazer um treinamento para um possível novo ataque que venha a ser criado no futuro.

REFERÊNCIAS

[1] Sebastian Garcia, Agustin Parmisano, & Maria Jose Erquiaga. (2020). IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.4743746

[2] N.A. Stoian. Machine Learning for Anomaly Detection in IoT networks: Malware analysis on the IoT-23 Data set. 2020.

[3] N. Moustafa. The Bot-IoT dataset. 2019. DOI: 10 . 21227/r7v2-x988.

[4] M. Shafiq, Z. Tian, A.K. Bashir, X. Du e M. Gui zani. "CorrAUC: A Malicious Bot-IoT Traffic Detec tion Method in IoT Network Using Machine-Learning Techniques". Em: IEEE Internet of Things Journal 8.5 (2021), pp. 3242–3254. ISSN: 23274662. DOI: 10 . 1109/JIOT.2020.3002255.

[5] https://scikit-learn.org/stable/user_guide.html. Acesso em 06/12/2022

[6] https://pandas.pydata.org/docs/user_guide/index.html#user-guide. Acesso em 06/12/2022

[7] https://docs.scipy.org/doc/scipy/ Acesso em 06/12/2022

[8] T. Akiba, S. Sano, T. Yanase, T. Ohta e M. Koyama. "Optuna: A Next-generation Hyperparameter Optimi zation Framework". Em: Proceedings of the ACM SIGKDD International Conference on Knowledge Dis covery and Data Mining. 2019, pp. 2623–2631. ISBN: 9781450362016. DOI: 10 . 1145 / 3292500 . 3330701.