Classificação de ataques cibernéticos em redes MQTT com base em aprendizado de máquina ⋆

Matheus Figueiredo de Castro * **Renan Landau Paiva de Medeiros** ** **Vicente Ferreira de Lucena Junior** *** **Iury Bessa** **

* *Faculdade de Tecnologia, Universidade Federal do Amazonas, AM, (e-mails: matheusfigueiredo@ufam.edu.br).*
** *Departamento de Eletricidade, Universidade Federal do Amazonas, AM, (e-mails: {renanlandau, iurybessa }@ufam.edu.br).*
*** *Departamento de Eletrônica e Computação, Universidade Federal do Amazonas, AM, (e-mails: vicente@ufam.edu.br).*

Abstract: This article presents an analysis of the performance of classification algorithms Decision Tree Machine Learning, Random Forest, Gradient Boosting, Naive Bayes, Sequential Neural Network and Multilayer Perceptron for Malicious Attack Detection in cyber-physical systems with Internet of Things (IoT) networks with communication-based on Message Queuing Telemetry Transport (MQTT). The dataset used in algorithm training is called MQTTset. The approaches used in this work were based on a study of the communication packages of that protocol, analysis of relevant features in that dataset and pre-processing techniques data. The results show that the algorithms classify well the network traffic that is under attack or not, which favours the security of a cyber-physical system.

Resumo: Este artigo apresenta uma análise do desempenho dos modelos de classificação para a detecção de ataques maliciosos em sistemas ciberfísicos com redes em Internet das Coisas (IoT) e comunicação baseada em Transporte de Telemetria de Enfileiramento de Mensagens (MQTT). O conjunto de dados utilizado nos treinamentos dos algoritmos se chama MQTTset. As abordagens utilizadas neste trabalho se basearam em um estudo sobre os pacotes de comunicação daquele protocolo, análise de características relevantes naquele conjunto de dados e técnicas de pré processamento de dados. Os resultados mostram que os algoritmos são capazes de identificar a ocorrência de ataques favorecendo a segurança de sistemas ciber-físicos.

Keywords: Cybersecurity; artificial intelligence; machine learning; internet of things; MQTT.
Palavras-chaves: Segurança cibernética; inteligência artificial; aprendizado de máquinas; internet das coisas; MQTT.

1. INTRODUÇÃO

Os primeiros sistemas de controle e automação tiveram papel fundamental desde a revolução industrial. Esses sistemas tipicamente utilizavam estratégia de comunicação ponto a ponto, ou seja, seus vários atuadores e sensores eram conectados aos sistemas supervisórios e de controle individualmente, não compartilhando recursos de comunicação (Sánchez et al., 2019). Com o intenso avanço recente das tecnologias de comunicação e informação, os sistemas evoluíram com melhor integração dos seus elementos por meio de redes de conexões de alta velocidade e internet das coisas (IoT), além de proporcionarem uma convergência entre as camadas físicas e virtuais, dando origem aos Sistemas Ciberfísicos (CPSs) (Cordeiro et al., 2019).

Os dispositivos conectados em rede IoT trocam informações cruciais para um sistema, permitindo que indústrias e usuários monitorem e tomem decisões em tempo real e que objetos do sistema elaborem, processem e se comuniquem com outros conectados na própria rede (Gyamfi and Jurcut, 2022). Na área da saúde, por exemplo, os dispositivos IoT implantáveis e vestíveis monitoram e extraem dados biomecânicos para permitir alertas de emergência em tempo real e melhorar a qualidade e expectativa de vida (Neshenko et al., 2019). No setor industrial, pode-se mencionar a adoção da IoT Industrial (Alladi et al., 2019) no contexto da Indústria 4.0. Logo, a troca de dados em sistemas IoT requer protocolos leves que permitam a comunicação em cenários onde a largura de banda limitada está disponível e o consumo de energia seja o mínino possível (Buccafurri et al., 2023). Para estas aplicações IoT, o Transporte de Telemetria de Enfileiramento de Mensagens (MQTT) se tornou o protocolo padrão (Longo et al., 2022).

Sabendo que as redes IoT assumem um papel importante em sistemas críticos envolvendo a segurança de indivíduos e processos, é possível notar que a natureza complexa e, geralmente, em grande escala de CPSs abre vulnerabili-

⋆ O presente trabalho foi realizado com apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico -Brasil (CNPq, processo 140593/2023-3) e da Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM, processos 01.02.016301.02763/2023-23, 01.02.016301.00673/2023-06 e Programa de Apoio à Iniciação Científica do Amazonas).

dades que podem ser exploradas por potenciais agentes e componentes maliciosos (Kanellopoulos and Vamvoudakis, 2020). Ademais, segundo Jaber and Fritsch (2022), a quantidade de ataques a CPSs está aumentando devido ao ambiente de tecnologia da informação estar mais complexo e em rápida evolução. Por conseguinte, o desenvolvimento de CPSs levanta novas preocupações relativas a segurança e proteção dos sistemas de controle, tais como a prevenção e resiliência em relação a ataques cibernéticos (Bessa et al., 2022). No entanto, o reconhecimento de ataques cibernéticos não é considerado uma questão trivial, especialmente quando os adversários engenhosos possuem informações sobre os sistemas (Kanellopoulos and Vamvoudakis, 2020).

Os ataques em redes IoT buscam afetar disponibilidade, integridade e confidencialidade da infraestrutura. Como estes sistemas dependem da internet para trocar mensagens e controlar sinais entre os seus componentes, a vulnerabilidade a ataques lançados no domínio da rede aumenta (Farivar et al., 2020). Neste contexto, existe uma infinidade de ataques no ambiente IoT, como *Buffer Overflow*, *Brute Force*, *DNS Poisoning*, *Replay Attacks*, *SQL Injection*, *Back-Door Exploits*, *False Data Injection*, entre outros.

Segundo Vaccari et al. (2020b), a proteção dos dados trocados em uma rede pode ser feita por sistemas de detecção e mitigação de ataques cibernéticos com o uso de algoritmos de Aprendizado de Máquina (ML) e Inteligência Artificial (IA), os quais já se mostraram capazes de identificar e prever ataques cibernéticos, protegendo o sistema subjacente. Partindo do problema dos poucos conjuntos de dados disponíveis no contexto da IoT, Vaccari et al. (2020b) propuseram um novo conjunto de dados focado em IoT, o *MQTTset*. Este dataset inclui comunicações com base no protocolo MQTT, contendo dados de redes de dispositivos IoT de diferentes naturezas (por exemplo, temperatura, umidade, sensores de movimento, etc.), a fim de simular um sistema inteligente em ambiente de casa, escritório, ou edifício (Vaccari et al., 2020b). Além disso, o *MQTTset* inclui informações legítimas e maliciosas de tráfegos.

Tendo em vista a relevância deste tema, este trabalho aborda a avaliação do desempenho de algoritmos de aprendizado de máquina, quais sejam, *Decision Tree*, *Random Forest*, *Gradient Boosting*, *Naive Bayes*, *Multilayer Perceptron* e *Sequential Neural Network*, para detectar ataques em redes IoT, utilizando, como base de dados, o dataset *MQTTset*. O objetivo principal consiste em detectar ataques na rede de tráfego, utilizando técnicas de redução da dimensionalidade dos dados e algorimos de aprendizado de máquina. Os resultados mostram que os modelos possuem excelentes resultados na classificação binária, isto é, quando há ou não ataque na rede. Na classificação multiclasses, os resultados são inferiores em razão da complexidade de diferenciar alguns tipos de ataque.

O restante deste trabalho é organizado da seguinte forma: a Seção 2 apresenta conceitos básicos sobre MQTT e ataques em sistemas IoT; a Seção 3 apresenta a metodologia proposta; a Seção 4 mostra os resultados da avaliação dos modelos para detecção e classificação de ataques cibernéticos e, enfim, a Seção 5 apresenta as conclusões do trabalho.

2. PRELIMINARES

2.1 Protocolo MQTT

MQTT é um protocolo padrão OASIS para sistemas IoT. Foi projetado para ser um protocolo de troca de mensagens baseado na arquitetura *Publish/Subscribe*, em que os clientes podem ser cadastrados como *publisher* ou *subscriber*.

Quando um cliente está cadastrado como publisher, ele envia dados para o broker em um tópico específico no qual está inserido. Já como subscriber, ele recebe as mensagens no tópico ao qual ela foi enviada. A Figura 1 mostra a arquitetura deste protocolo:

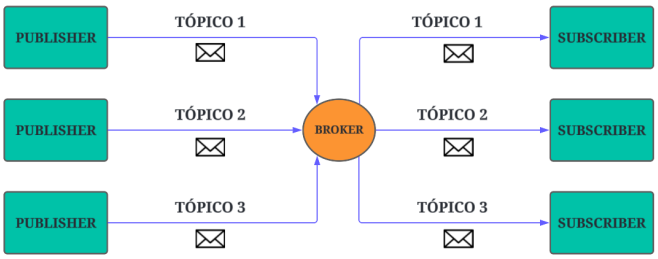

Figura 1. Arquitetura do Protocolo MQTT.

Dessa forma, o protocolo MQTT opera com camadas de aplicação e usa serviços do protocolo TCP/IP para o transporte de mensagens. O broker e os clientes podem trocar mensagens (e.g., *Publish*, *Puback*, and *Pubrel*), e o conteúdo da mensagem não é obrigatório, podendo conter diversos dados que o cliente necessita enviar a todos os inscritos no tópico da mensagem (Živić et al., 2023).

Neste trabalho, o foco no protocolo MQTT se concentra nas características dos pacotes de comunicação trocados nas redes. Estes, detalham informações importantes sobre a comunicação dos dispositivos conectados, como, por exemplo, *Flags TCP*, *Mensagem MQTT*, *Nível de QoS*, entre outros. A Tabela 1 mostra a estrutura geral de um pacote de comunicação MQTT.

Tabela 1. Estrutura genal do pacote de comunicação MQTT.

BIT	**7 6 5 4**	**3**	**2 1**	**0**
BIT 1	Tipo de mensagem	DUP	Nível de QoS	Retenção
BIT 2	Comprimento restante			
...	...			
BIT 5	Comprimento restante			

2.2 Metodologia proposta

A metodologia deste trabalho consiste em etapas fundamentais para o desenvolvimento da pesquisa. A primeira delas é a análise geral do *MQTTset*, em que serão verificados o tamanho dos dados e número de amostras. A segunda, é a redução da dimensão do conjunto de dados utilizando técnicas de pré-processamento com *sklearn*. A terceira etapa consiste no treinamento dos algoritmos de ML e, por fim, a última etapa apresenta as métricas para validação dos resultados. A Figura 2 mostra os passos que foram seguidos nesta pesquisa:

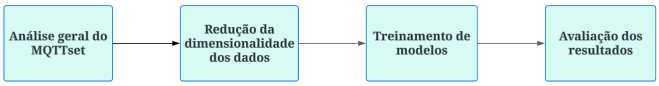

Figura 2. Diagrama da metodologia.

3. TRATAMENTO DE DADOS DO MQTTSET

3.1 MQTTset

O *MQTTset* é um conjunto de dados focado em redes IoT com comunicação MQTT. Estes dados foram criados utilizando o *IoT-Flock* (Ghazanfar et al., 2020), uma ferramenta geradora de tráfego de rede capaz de emular dispositivos IoT com base nos protocolos MQTT e CoAP (Constrained Application Protocol). IoT-Flock oferece a capacidade de configurar o cenário da rede em termos de nós (por exemplo, tipo de sensor, endereços IP, portas de escuta, etc.) e comunicações (por exemplo, intervalo de tempo usado para comunicações entre os sensores e o corretor). Além disso, a ferramenta implementa diferentes ameaças cibernéticas contra o MQTT e o CoAP: inundação de publicações, ataques de criação de pacotes, ataque de falha de segmentação contra CoAP (fazendo uso de um Uri-path nulo) e ataques de vazamento de memória contra CoAP (usando opções CoAP inválidas durante a falsificação de pacotes) (Vaccari et al., 2020b).

Dessa forma, Vaccari et al. (2020b) criaram um cenário que pode ser equiparado a um ambiente doméstico inteligente, onde sensores, identificados exclusivamente por um endereço IP, recuperam informações como temperatura, intensidade da luz, umidade, gás CO, movimento, fumaça, abertura/fechamento de porta e status do ventilador em diferentes intervalos temporais. Além disso, o *MQTTset* inclui ambos os tráfegos, legítimo e malicioso. As features do *MQTTset* são resumidas na Tabela 2.

3.2 Ataques em redes IoT presentes no MQTTset

De acordo com (Vaccari et al., 2020b), foram incluídos no *MQTTset* os seguintes ataques em redes IoT:

- *Flooding Denial of Service*: são executados para impedir que o serviço atenda clientes legítimos (Wood and Stankovic, 2002). Para executar essas ameaças no conjunto de dados, Vaccari et al. (2020b) utilizaram a ferramenta *MQTT-malaria tool*.
- *MQTT Publish Flood*: um dispositivo IoT malicioso envia periodicamente uma grande quantidade de dados MQTT maliciosos, para aproveitar todos os recursos do servidor, em termos de slots de conexão, redes ou outros recursos alocados em quantidade limitada. Para gerar este ataque, Vaccari et al. (2020b) utilizaram um gerador disponibilizado pelo *IoT-Flock*.
- *SlowITe*: é outra ameaça de negação de serviço direcionada ao protocolo de aplicação MQTT (Vaccari et al., 2020a). Este ataque requer largura de banda e recursos mínimos para atacar um serviço MQTT (Vaccari et al., 2020b).
- *Malformed Data*: visa gerar e enviar ao broker pacotes malformados, tentando levantar exceções no serviço de destino (Bhagat Patil and Thakur, 2018). Para gerar esse ataque, Vaccari et al. (2020b) adotaram a ferramenta *MQTTSA*, enviando uma sequência de pacotes *Connect* malformados ou *Publish* para a vítima, a fim de gerar exceções no broker MQTT.
- *Brute Force Authentication*: consiste em executar possíveis tentativas de recuperar credenciais de usuários utilizadas por MQTT (Vaccari et al., 2020b). O objetivo do invasor é quebrar as credenciais dos usuários, como o nome de usuário e senha, adotadas durante a fase de autenticação. Neste caso, Vaccari et al. (2020b) utilizaram a ferramenta *MQTTSA*.

Tabela 2. *Features* do *MQTTset*.

Feature	Descrição	Relevante?
tcp.flags	Flag TCP	Sim
tcp.time_delta	Tempo de fluxo	Sim
tcp.len	Tamanho TCP	Sim
mqtt.conack.flags	Flag de reconhecimento	Sim
mqtt.conflag.qos	Nível de QoS	Sim
mqtt.conack.val	Retorno de recebimento	Sim
mqtt.conflag.cleansess	Limpar sessão	Sim
mqtt.conflag.passwd	Flag de senha	Sim
mqtt.conflag.uname	Flag de nome	Sim
mqtt.conflags	Flag de conexão	Sim
mqtt.dupflag	Flag DUP	Sim
mqtt.hdrflags	Cabeçalho MQTT	Sim
mqtt.kalive	Keep Alive MQTT	Sim
mqtt.len	Tam. da mensagem MQTT	Sim
mqtt.msg	Mensagem MQTT	Sim
mqtt.msgid	Id. de mensagem MQTT	Sim
mqtt.msgtype	Tipo de mensagem MQTT	Sim
mqtt.proto_len	Tam. da camada do protocolo	Sim
mqtt.protoname	Camada do protocolo	Sim
mqtt.qos	Nível de QoS	Sim
mqtt.ver	Versão	Sim
mqtt.conack.flags.reserved	Reservado	Sim
mqtt.conack.flags.sp	Session Present	Não
mqtt.conflag.qos	Nível de QoS	Não
mqtt.conflag.reserved	Reservado	Não
mqtt.conflag.retain	Will Retain	Não
mqtt.conflag.willflag	Will Flags	Não
mqtt.retain	Retenção	Não
mqtt.sub.qos	Requisição de QoS	Não
mqtt.suback.qos	Conceder QoS	Não
mqtt.willmsg	Will Message	Não
mqtt.willmsg_len	Tam. Will Message	Não
mqtt.willtopic	Will Topic	Não
mqtt.willtopic_len	Tam. Will Topic	Não

3.3 Análise geral do conjunto de dados MQTTset

Antes de realizar o treinamento de modelos, é necesário analisar os dados do *MQTTset* e entender como são organizados. De forma geral, esse conjunto apresenta 7 milhões de amostras de dados legítimos e 1,2 milhões para cada tipo de ataque: *Denial of Service*, *Brute Force*, *Flood*, *SlowITe* e *Malformed*.

Visando classificar tipos de tráfego, os dados foram balanceados de acordo com dois objetivos: classificação binária (quando se deseja predizer apenas se há ou não ataque na rede) e multiclasses (detectar o tipo de tráfego). Em relação à classificação binária, os registros de ataque foram somados de modo que a quantidade de tráfego malicioso seja igual a de tráfego legítimo. Na classificação multiclasses, o tráfego legítimo foi diminuído para ficar exatamente igual à quantidade do registro de cada um dos ataques (1.200.000 amostras).

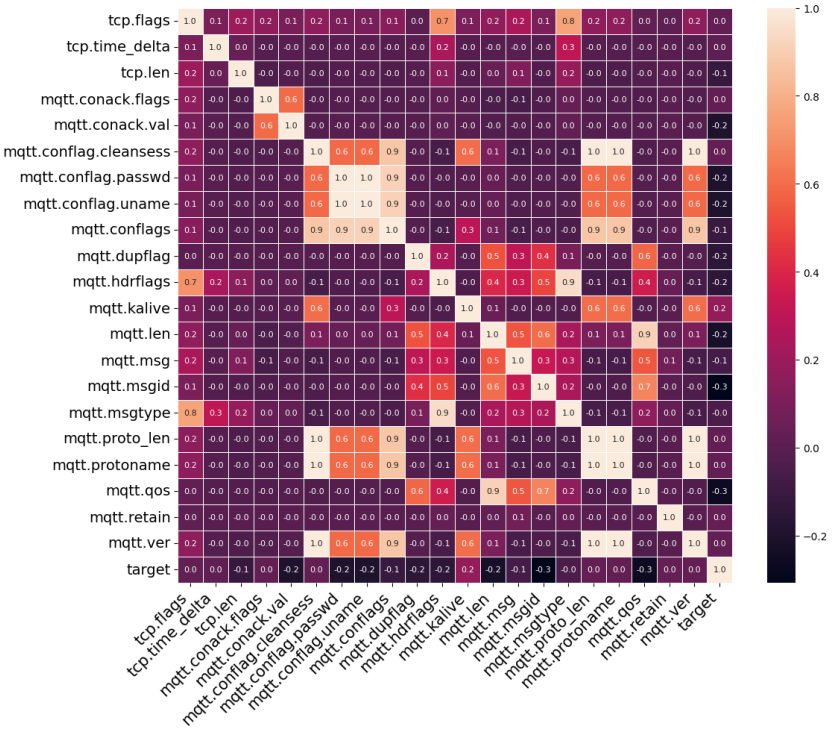

Figura 3. Heatmap das variáveis do dataset MQTTset.

3.4 Preparação do data frame

Redução da dimensionalidade. Nesta etapa, foi verificado o tamanho dos dados do *MQTTset*. Primeiramente, foram analisadas as *features* com dados irrelevantes para o treinamento dos dados. Para isso, as amostras de cada coluna foram verificadas, e todas as *features* que possuem amostras iguais a 0 foram consideradas irrelevantes para o treinamento. A tabela 2 mostra as *features* do *MQTTset* de forma geral, indicando as relevantes para os treinamentos e as não relevantes. Ademais, a figura 3 apresenta o *Heatmap* destas mesmas *features*, ilustrando a correlação entre as mesmas e indicando a existência de informações redundantes no dataset.

Tratamento de dados. A próxima etapa consistiu na análise e tratamento de *features* categóricas no conjunto de dados. Um pacote de mensagem do protocolo *MQTT* não consiste apenas em dados numéricos, possuindo, também, variáveis categóricas, por exemplo: *Flags TCP*, *MQTT Message* e *MQTT Header Flags*. Para ajustar as variáveis categóricas, foi utilizado o *Label Enconder*.

4. RESULTADOS EXPERIMENTAIS

Esta seção apresenta os resultados e as métricas obtidas para a aplicação de modelos de ML na predição de ataques em redes MQTT. Em particular, comparou-se os seguintes modelos de ML: *Decision Tree*, *Random Forest*, *Gradient Boosting*, *Naive Bayes*, *Multilayer Perceptron* (MLP) e *Sequential Neural Network*. Mais detalhes sobre esses algoritmos podem ser encontrados em (Li, 2024). A implementação e treinamento foi feita na linguagem de programação *Python* com o uso da ferramenta de *Sklearn*.

Nos treinamentos de classificação binária e multiclasses, o algoritmo *Decision Tree* foi implementado com os parâmetros *gini criterium*, *best splitter*, *42 random state* e *Maximun Depth*, enquanto os *Random Forest*, *Grandient Boosting* e *Naive Bayes* tiveram os parâmetros da configuração padrão. Sobre as redes neurais, o algoritmo *MPL Classifier* teve seus parâmetros configurados com *adam solver*, *True shuffle*, *0 random state* e *0.0000001 alpha*. Por fim, a Rede Neural Sequencial foi implementada com *Kers* com a primeira camada oculta de 50 nós, a segunda de 30, a terceira de 20 e, em relação à última camada, foram definidas 2 para a classificação binária e 6 para a classificação multiclasses. A camada oculta foi configurada com *relu activation function* e *normal kernel*, com exceção da última camada, configurada com *softmax activation*. Por conseguinte, o desempenho dos modelos foi avaliado pela técnica *k-Fold Cross Validation* com k=5.

Para avaliar os resultados dos treinamentos, foram considerados parâmetros de desempenho comumente utilizados:

- *Verdadeiro Positivo (TP)*: o modelo de ML previu realmente o fluxo de ataque como um ataque.
- *Verdadeiro negativo (TF)*: O modelo ML previu realmente o tráfego legítimo como legítimo.
- *Falso Positivo (PF)*: o modelo de ML previu o fluxo de ataque como um legítimo.
- *Falso Negativo (NF)*: o modelo de ML previu o fluxo legítimo como um ataque.
- *Precision*: capacidade do sistema de detectar corretamente o ataque à ocorrência da violação de segurança. Descreve a razão entre o previsto corretamente (ou seja, TP) e os resultados reais (ou seja, TP + FP) (Hussain et al., 2021):

$$Precision = \frac{TP}{TP + FP} \cdot 100 \qquad (1)$$

Tabela 3. Tabela de métricas da classificação binária.

Decision Tree

Tráfego	Precision	Recall	F1-score
Legítimo	0.94	0.98	0.96
Malicioso	0.98	0.94	0.96
Accuracy		0.96	

Random Forest

Tráfego	Precision	Recall	F1-score
Legítimo	0.94	0.98	0.96
Malicioso	0.98	0.94	0.96
Accuracy		0.96	

Gradient Boosting

Tráfego	Precision	Recall	F1-score
Legítimo	0.95	0.96	0.96
Malicioso	0.96	0.95	0.95
Accuracy		0.95	

Naive Bayes

Tráfego	Precision	Recall	F1-score
Legítimo	0.61	1.00	0.75
Malicioso	1.00	0.35	0.52
Accuracy		0.67	

Multilayer Perceptron

Tráfego	Precision	Recall	F1-score
Legítimo	0.87	0.99	0.93
Malicioso	0.98	0.86	0.92
Accuracy		0.92	

Sequential Neural Network

Tráfego	Precision	Recall	F1-score
Legítimo	0.87	0.99	0.93
Malicioso	0.99	0.86	0.92
Accuracy		0.92	

Tabela 4. Tabela de métricas da classificação multiclasses.

Decision Tree

Tráfego	Precision	Recall	F1-score
Bruteforce	0.78	0.88	0.83
Dos	0.91	0.89	0.90
Flood	0.87	0.88	0.88
Legítimo	0.85	0.90	0.88
Malformed	0.89	0.72	1.00
Slowite	1.00	1.00	1.00
Accuracy		0.88	

Random Forest

Tráfego	Precision	Recall	F1-score
Bruteforce	0.78	0.88	0.83
Dos	0.91	0.89	0.90
Flood	0.87	0.88	0.88
Legítimo	0.85	0.90	0.88
Malformed	0.89	0.72	1.00
Slowite	1.00	1.00	1.00
Accuracy		0.88	

Gradient Boosting

Tráfego	Precision	Recall	F1-score
Bruteforce	0.72	0.88	0.80
Dos	0.90	0.89	0.90
Flood	0.87	0.88	0.88
Legítimo	0.85	0.90	0.88
Malformed	0.87	0.64	0.74
Slowite	1.00	1.00	1.00
Accuracy		0.86	

Naive Bayes

Tráfego	Precision	Recall	F1-score
Bruteforce	0.28	1.00	0.44
Dos	0.96	0.72	0.82
Flood	0.98	0.49	0.65
Legítimo	0.76	0.62	0.68
Malformed	1.00	0.15	0.26
Slowite	0.98	0.21	0.34
Accuracy		0.53	

Multilayer Perceptron

Tráfego	Precision	Recall	F1-score
Bruteforce	0.67	0.92	0.77
Dos	1.00	0.72	0.84
Flood	0.58	0.99	0.73
Legítimo	0.99	0.63	0.77
Malformed	0.94	0.54	0.69
Slowite	1.00	1.00	1.00
Accuracy		0.80	

Sequential Neural Network

Tráfego	Precision	Recall	F1-score
Bruteforce	0.63	0.91	0.74
Dos	1.00	0.71	0.83
Flood	0.61	0.97	0.75
Legítimo	0.78	0.63	0.70
Malformed	0.87	0.40	0.54
Slowite	1.00	1.00	1.00
Accuracy		0.77	

Tabela 5. Tabela de matrizes de confusão da classificação multiclasses.

		Predito					
		Bruteforce	Dos	Flood	Legitimate	Malformed	Slowite
Verdadeiro	Bruteforce	88.3%	4,30%	0,19%	0,07%	7,13%	0%
	DoS	1,30%	88,90%	3,43%	4,63%	1,74%	0%
	Flood	0,20%	0,80%	88,20%	10,60%	0,20%	0%
	Legítimo	0,01%	1,42%	8,37%	90,09%	0,09%	0%
	Malformed	23,57%	2,66%	0,94%	0,34%	72,49%	0%
	Slowite	0%	0%	0%	0%	0%	100%

Decision Tree

		Predito					
		Bruteforce	Dos	Flood	Legitimate	Malformed	Slowite
Verdadeiro	Bruteforce	88.3%	4,30%	0,19%	0,07%	7,13%	0%
	DoS	1,30%	88,90%	3,43%	4,63%	1,74%	0%
	Flood	0,20%	0,80%	88,20%	10,60%	0,20%	0%
	Legítimo	0,01%	1,42%	8,37%	90,09%	0,09%	0%
	Malformed	23,57%	2,66%	0,94%	0,34%	72,49%	0%
	Slowite	0%	0%	0%	0%	0%	100%

Random Forest

		Predito					
		Bruteforce	Dos	Flood	Legitimate	Malformed	Slowite
Verdadeiro	Bruteforce	100%	0%	0%	0%	0%	0%
	DoS	28,25%	71,68%	0%	0,04%	0,03%	0%
	Flood	50,60%	0%	48,8%	0,6%	0%	0%
	Legítimo	37,06%	0%	0,8%	90,09%	0,09%	0%
	Malformed	73,21%	2,88%	2,88%	0,34%	72,49%	0%
	Slowite	69%	0%	0%	10,2%	0%	20,54%

Naive Bayes

		Predito					
		Bruteforce	Dos	Flood	Legitimate	Malformed	Slowite
Verdadeiro	Bruteforce	88,47%	4,37%	0,31%	0,07%	6,78%	0%
	DoS	1,64%	88,76%	3,19%	4,63%	1,78%	0%
	Flood	0,2%	0,8%	87,8%	10,6%	0,6%	0%
	Legítimo	0,01%	1,43%	8,44%	90,02%	0,1%	0%
	Malformed	32,055	2,96%	0,84%	0,34%	63,81%	0%
	Slowite	0%	0%	0%	0%	0%	100%

Gradient Boosting

		Predito					
		Bruteforce	Dos	Flood	Legitimate	Malformed	Slowite
Verdadeiro	Bruteforce	91,78%	0%	4,82%	0%	3,40%	0%
	DoS	3,10%	71,31%	25,58%	0%	0,01%	0%
	Flood	0,08%	0%	98,60%	0,4%	0,2%	0%
	Legítimo	0,06%	0%	36,98%	62,93%	0,03%	0%
	Malformed	41,32%	0%	4,4%	0%	54,28%	0%
	Slowite	0%	0%	0%	0%	0%	100%

Multilayer Perceptron

		Predito					
		Bruteforce	Dos	Flood	Legitimate	Malformed	Slowite
Verdadeiro	Bruteforce	91,04%	0%	3,82%	0%	5,14%	0%
	DoS	9,98%	70,99%	19,26%	0,7%	0,07%	0%
	Flood	1,4%	0%	97,2%	0,8%	0,6%	0%
	Legítimo	0,54%	0%	36,52%	62,94%	0%	0%
	Malformed	42,96%	0%	1,45%	16%	39,59%	0%
	Slowite	0%	0%	0%	0%	0%	100%

Sequential Network

- *Recall*: capacidade do sistema de detectar corretamente o ataque após a ocorrência do mesmo na rede:

$$Recall = \frac{TN}{TN + FN} \cdot 100 \tag{2}$$

- *Accuracy:* capacidade do sistema de classificar corretamente o pacote de ataque como um "pacote de ataque" e um pacote normal como um "pacote normal". Ele descreve a proporção de previsões corretas em relação a todas as amostras:

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \cdot 100 \tag{3}$$

- *F1-score*: é a média harmônica de precisão e recuperação. Ele descreve a proporção de previsões corretas no conjunto de testes para tráfego normal e de ataque:

$$F1score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

A Tabela 3 apresenta as métricas dos resultados obtidos na classificação binária em termos de *Precision*, *Recall*, *F1-score* e *Accuracy test*, indicando o desempenho dos modelos selecionados no problema de detecção de ataques. Em seguida, a Tabela 4 apresenta as métricas obtidas nos treinamentos de classificação multicasses em termos de *Precision*, *Recall*, *F1-score* e *Accuracy test*. Para analisar a

classificação multiclasses, a Tabela 5 apresenta as matrizes de confusão do desempenho dos algoritmos treinados.

Ao analisar os resultados, é possível observar que os algoritmos *Decision Tree*, *Random Forest* e *Gradient Boosting* classificam bem quando há ou não ataque na rede. As redes neurais também alcançam um bom resultado na classificação binária, no entanto, a precisão para o tráfego legítimo diminui em relação aos algoritmos *Ensemble* e *Tree*. Na classificação multiclasses, o desempenho dos algoritmos diminui devido à dificuldade em classificar ataques cibernéticos. Pode-se notar, por exemplo, que as redes neurais conseguem classificar excelentemente o ataque *DoS*, mas na detecção do ataque *Bruteforce*, a precisão decai bastante. Percebe-se, também, que os algoritmos *Naive Bayes*, *Gradient Boosting*, *Multilayer Perceptron* e *Sequential Network* confundem significativamente o ataque *Malformed* com o *Bruteforce*, errando na classificação.

Em relação ao algoritmo *Naive Bayes*, nota-se que este não obteve bom desempenho nas classificações binária e multiclasses. Sobre os algoritmos *Decison Tree* e *Random Forest*, estes alcançam extamente os mesmos resultados nas duas classificações.

Em suma, observa-se que os algoritmos, com exceção do modelo *Naive Bayes*, conseguem prever quando há um ataque na rede. Entretanto, para classificar qual o tipo de ataque, o desempenho dos treinamentos cai e os algoritmos não classificam bem um ataque em específico.

5. CONCLUSÕES

Este trabalho apresenta uma avaliação comparativa de diversas técnicas de aprendizado de máquina para o problema de detecção de ataques maliciosos em redes IoT com protocolo MQTT. Esse problema é crucial no cenário atual onde os sistemas IoT se tornam cada vez mais populares. No entanto, a detecção dessas ameaças é desafiadora, o que motiva o emprego de inteligência artificial para aprender relações dos dados de tráfego em redes MQTT com eventuais ocorrências de ataques. Os resultados mostraram que os algoritmos de ML conseguem detectar se o tráfego é legítimo ou malicioso, além de identificar o tipo de ataque que está ocorrendo. Isto indica a viabilidade de se implementar esta tecnologia em sistemas ciberfísicos a fim de obter proteção contra ameaças cibernéticas.

Em trabalhos futuros, estratégias defensivas e reconfiguração de redes serão investigadas, de forma a permitir a exclusão dos ataques da rede a partir da detecção fornecida por algoritmos de inteligência artificial.

REFERÊNCIAS

Alladi, T., Chamola, V., Parizi, R.M., and Choo, K.K.R. (2019). Blockchain applications for industry 4.0 and industrial iot: A review. *IEEE Access*, 7, 176935–176951.

Bessa, I., Trapiello, C., Puig, V., and Palhares, R.M. (2022). Dual-rate control framework with safe watermarking against deception attacks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52, 7494–7506.

Bhagat Patil, A.R. and Thakur, N.V. (2018). *Mitigation Against Denial-of-Service Flooding and Malformed Packet Attacks*, 335–342. Springer Singapore.

Buccafurri, F., de Angelis, V., and Lazzaro, S. (2023). Mqtt-a: A broker-bridging p2p architecture to achieve anonymity in mqtt. *IEEE Internet of Things Journal*, 10, 15443–15463.

Cordeiro, L.C., Lima Filho, E.B., and Bessa, I.V. (2019). Survey on automated symbolic verification and its application for synthesising cyber-physical systems. *IET Cyber-Physical Systems: Theory & Applications*, 5(1), 1–24.

Farivar, F., Haghighi, M.S., Jolfaei, A., and Alazab, M. (2020). Artificial intelligence for detection, estimation, and compensation of malicious attacks in nonlinear cyber-physical systems and industrial iot. *IEEE Transactions on Industrial Informatics*, 16, 2716–2725.

Ghazanfar, S., Hussain, F., Rehman, A.U., Fayyaz, U.U., Shahzad, F., and Shah, G.A. (2020). Iot-flock: An open-source framework for iot traffic generation. In *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*. IEEE.

Gyamfi, E. and Jurcut, A. (2022). Intrusion detection in internet of things systems: A review on design approaches leveraging multi-access edge computing, machine learning, and datasets. *Sensors*, 22, 3744.

Hussain, F., Abbas, S.G., Shah, G.A., Pires, I.M., Fayyaz, U.U., Shahzad, F., Garcia, N.M., and Zdravevski, E. (2021). A framework for malicious traffic detection in iot healthcare environment. *Sensors*, 21, 3025.

Jaber, A. and Fritsch, L. (2022). *Towards AI-powered Cybersecurity Attack Modeling with Simulation Tools: Review of Attack Simulators*, 249–257. Springer International Publishing.

Kanellopoulos, A. and Vamvoudakis, K.G. (2020). A moving target defense control framework for cyber-physical systems. *IEEE Transactions on Automatic Control*, 65, 1029–1043.

Li, H. (2024). *Machine Learning Methods*. Springer Nature Singapore.

Longo, E., Redondi, A.E.C., Cesana, M., and Manzoni, P. (2022). Border: A benchmarking framework for distributed mqtt brokers. *IEEE Internet of Things Journal*, 9, 17728–17740.

Neshenko, N., Bou-Harb, E., Crichigno, J., Kaddoum, G., and Ghani, N. (2019). Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations. *IEEE Communications Surveys & Tutorials*, 21, 2702–2733.

Sánchez, H.S., Rotondo, D., Escobet, T., Puig, V., and Quevedo, J. (2019). Bibliographical review on cyber attacks from a control oriented perspective. *Annual Reviews in Control*, 48, 103–128.

Vaccari, I., Aiello, M., and Cambiaso, E. (2020a). Slowite, a novel denial of service attack affecting mqtt. *Sensors*, 20, 2932.

Vaccari, I., Chiola, G., Aiello, M., Mongelli, M., and Cambiaso, E. (2020b). Mqttset, a new dataset for machine learning techniques on mqtt. *Sensors*, 20, 6578.

Živić, M., Nemec, D., and Bojović, v. (2023). Mqtt protocol in iot environment: Comparison with coap and zeromq protocols. In *2023 31st Telecommunications Forum (TELFOR)*. IEEE.

Wood, A. and Stankovic, J. (2002). Denial of service in sensor networks. *Computer*, 35, 54–62.