UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS QUIXADÁ

CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

FRANCISCO THIAGO DOS SANTOS GONÇALVES

UMA HEURÍSTICA PARA OFUSCAÇÃO DE TRÁFEGO EM INTERNET DAS COISAS

QUIXADÁ

2022

FRANCISCO THIAGO DOS SANTOS GONÇALVES

UMA HEURÍSTICA PARA OFUSCAÇÃO DE TRÁFEGO EM INTERNET DAS COISAS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Campus Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Jeandro de Mesquita Bezerra

QUIXADÁ

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

G625h Gonçalves, Francisco Thiago Santos.
Uma heurística para ofuscação de tráfego em internet das coisas / Francisco Thiago Santos Gonçalves. – 2022.
48 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Computação, Quixadá, 2022.
Orientação: Prof. Dr. Jeandro de Mesquita Bezerra.

1. Internet das coisas. 2. Segurança computacional.. 3. Aprendizagem profunda. 4. Heurística. I. Título.

CDD 621.39

FRANCISCO THIAGO DOS SANTOS GONÇALVES

UMA HEURÍSTICA PARA OFUSCAÇÃO DE TRÁFEGO EM INTERNET DAS COISAS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Campus Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Aprovada em: ____/____/_____

BANCA EXAMINADORA

Prof. Dr. Jeandro de Mesquita Bezerra (Orientador)
Universidade Federal do Ceará (UFC)

Dr. Antônio Janael Pinheiro
Universidade Federal de Pernambuco (UFPE)

Prof. Dr. Criston Pereira de Souza
Universidade Federal do Ceará (UFC)

“A lógica pode levar de um ponto A a um ponto B. A imaginação pode levar a qualquer lugar.”

(Albert Einstein)

RESUMO

O uso de tecnologias inteligentes está cada vez mais presente no cotidiano da população devido ao avanço da *Internet* das Coisas (IoT). Dispositivos inteligentes captam a interação do usuário com o ambiente, hábitos, localização, saúde, entre outras informações e compartilham esses dados via Internet para serem processados por servidores remotos ou consumidos por diferentes serviços e aplicativos, dessa forma provendo uma experiência personalizada para cada usuário. Pesquisas demonstram que invasores podem utilizar estatísticas presentes em atributos do tráfego, como tamanho dos pacotes, intervalos de envio e quantidade de pacotes enviados para inferir informações sigilosas do usuário, portanto, violando sua privacidade. Na literatura, diversas abordagens de preenchimento foram propostas para proteger a privacidade do usuário contra ataques baseados em análise de tráfego. A revisão desses trabalhos demonstra que, em geral, dados redundantes são inseridos na rede, o que pode causar atrasos e sobrecargas. Nesse contexto, é necessária uma análise de *trade-off* antes de aplicar métodos de ofuscamento em uma rede de dispositivos IoT. Esse trabalho propõe uma heurística probabilística para preenchimento de pacotes com o objetivo de mitigar ataques baseados em análise de tráfego que utilizem o tamanho dos pacotes para inferir informações do usuário. O *trade-off* é calculado entre a redução na taxa de acerto de um invasor monitorando a rede e a quantidade de *bytes* inseridos nos pacotes. Para balancear o *tradeoff* entre privacidade e desempenho, a heurísitica tem o objetivo de minimizar a quantidade de *bytes* adicionados nos pacotes. O método de preenchimento desenvolvido obteve uma melhoria de até 90% no *trade-off* comparada com a abordagem similar.

Palavras-chave: IoT; segurança; aprendizado de máquina; análise de tráfego

ABSTRACT

The use of smart technologies is more present in population daily lives thank to the advance of the Internet of Things (IoT). Smart Devices capture user's iteration with the environment, habits, localization, health status, and other information and share these by internet to be processed by remote servers or consumed by several services or applications, that way providing customized user experience. Researches shows that intruders can use patterns present in traffic attributes, like packets size, send intervals and sent packet number to infer user information, that way invading his privacy. In current literature, many methods were proposed to protect users against attacks based on traffic analysis. The review of those works shows that, in general, redundant data are added to the network, what can cause delays and overloads. In this context, its necessary a trade-off analysis before applying obfuscation methods in a IoT network. This project propose an obfuscation approach based in packet padding to reduce attacks based on traffic analysis that use packet lengths to infer user's information. To balance the trade-off between performance e privacy, the method use a padding heuristic to minimize the number of bytes inserted in the packets. In addition, the results show an improvement up to 90% in the tradeoff between the reduction of attacker success rate and the number of bytes inserted in the packets.

Keywords: IoT, security, machine learning, traffic analysis

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE SÍMBOLOS

Ω	Tamanho dos pacotes enviados em um canal de comunicação
$\underline{X}$	Tamanho dos *frames* utilizados em um canal de comunicação
M	Quantidade de *frames* utilizados.
α_k	Frequência relativa de envio de um pacote de tamanho k

SUMÁRIO

1 INTRODUÇÃO

O avanço da indústria de semicondutores possibilitou embutir dispositivos eletrônicos, como sensores e microcontroladores, em objetos do cotidiano, logo permitindo obter e compartilhar via rede de computadores informações contextuais a respeito do ambiente em que estão inseridos. Dessa forma, esses objetos podem oferecer experiências e funcionalidades personalizadas para cada usuário. Nesse contexto, vale citar o conceito de Casa Inteligente (*Smart Home*), que consiste em residências equipadas com sensores e atuadores em sua estrutura utilizados para captar a interação do residente com o ambiente doméstico a fim de prover serviços ao usuário (BAI *et al.*, 2020).

A transmissão de dados relacionados a hábitos, comportamento, saúde, entre outras informações, expõem riscos em relação à privacidade dos usuários de tecnologias IoT (*Internet Of Things*) . A pesquisa realizada por (TSANTIKIDOU; SKLAVOS, 2021) aponta falhas de segurança causadas por limitações de recursos energéticos e computacionais em dispositivos IoT voltados à saúde, onde protocolos seguros não são utilizados na comunicação desses aparelhos. Ainda nesse contexto, o autor de (LEE *et al.*, 2017) demonstra ser possível obter dados do usuário a partir de falhas de autenticação em pulseiras inteligentes (*Smart Bands*).

Nesse contexto, protocolos seguros como *Security Socket Layer* (SSL) podem ser utilizados para garantir que agentes maliciosos não tenham acesso às informações enviadas na rede. Contudo, mesmo em um tráfego cifrado ainda é possível identificar eventos e interação desses dispositivos analisando estatísticas e padrões presentes nos pacotes enviados. Trabalhos como (SANTOS *et al.*, 2018) e (AMMAR; NOIRIE; TIXEUIL, 2020) demonstram ser possível utilizar métodos de Aprendizado de Máquina para classificar informações relacionadas aos dispositivos na rede analisando dados não criptografados dos pacotes como *flags* de envio, quantidade de pacotes enviados, etc.

Ainda nesse contexto, a pesquisa realizada por (PINHEIRO *et al.*, 2019) demonstra ser possível diferenciar dispositivos em um cenário com 27 dispositivos IoT e 8 não IoT utilizando modelos de classificação que analisam o tamanho dos pacotes, obtendo acurácias acima de 90%. Os modelos gerados pelo autor conseguiram distinguir eletrônicos IoT e não IoT além de identificar eventos enviados na rede. O autor explora apenas ataques baseados no tamanho dos pacotes, entretanto, (SKOWRON; JANICKI; MAZURCZYK, 2020) aponta que aspectos temporais, como intervalos de envio dos pacotes, também podem ser usados para identificar os dispositivos na rede.

Dessa forma, a privacidade do usuário pode ser invadida mesmo que os dispositivos usem métodos de confidencialidade de dados, como criptografia. Logo, um invasor pode inferir informações sobre o usuário analisando padrões presentes nos envios realizados pelos dispositivos presentes na rede. Por exemplo, um invasor pode analisar eventos gerados por dispositivo como lâmpadas inteligentes para identificar a presença do usuário em sua residência.

Diferentes técnicas de ofuscamento foram propostas na literatura para mascarar estatísticas presentes no tráfego. A revisão desses trabalhos mostrou que geralmente esses métodos consistem no preenchimento dos pacotes ou envio de tráfego fictícios na rede. No contexto do Ofuscamento de Tráfego, técnicas de preenchimento ou *padding* visam alterar o tamanho dos pacotes a fim de reduzir a taxa de acertos de modelos utilizados para classificar informações relevantes presentes no tráfego, como dispositivo de origem dos pacotes.

Técnicas de preenchimento consistem em adicionar *bytes* redundantes nos pacotes transmitidos na rede, consequentemente adicionando um custo (*overhead*) que pode ocasionar atrasos ou sobrecargas. Falhas na comunicação de dispositivos Inteligentes podem causar transtornos ou danos ao usuário, principalmente no contexto de Casas Inteligentes de Saúde (*Health Smart Homes*), que podem ser utilizadas para monitorar a saúde de pacientes (MSHALI *et al.*, 2018) e detectar eventos anormais como quedas e desmaios (BAI *et al.*, 2020). Nesse contexto, o *trade-off* é caracterizado pela melhoria na privacidade provida ao usuário em contraste com o *overhead* adicionado no tráfego.

O objetivo geral deste trabalho é desenvolver e aplicar um método de preenchimento de pacotes que proteja a privacidade do usuário contra ataques baseados em análise de tráfego e que balanceie o *tradeoff* entre privacidade e desempenho. Para isso, uma heurística de preenchimento é utilizada para encontrar valores otimizados para a quantidade de *bytes* inseridos nos pacotes. Dessa forma, espera-se aprimorar a segurança de usuários que utilizam tecnologias IoT em seu cotidiano. Os objetivos específicos desse objetivo são listados a seguir.

- Implementar uma heurística baseada na distribuição de probabilidade de envio dos pacotes para minimizar a quantidade de *bytes* inseridos;
- Comparar o desempenho e *overhead* gerado pelo método desenvolvido com abordagens de ofuscamento baseados em preenchimento disponíveis na literatura.

O trabalho está organizado da seguinte forma, a Seção 2 resume os principais conceitos teóricos usados pelo projeto, na Seção 3, trabalhos similares disponíveis na literatura são analisados e comparados com a abordagem proposta, a Seção 4 apresenta o modelo de

otimização para minimizar o *overhead*, a Seção 5 apresenta a metodologia escolhida para implementar e avaliar a abordagem proposta, a Seção 6 apresenta os resultados obtidos pela abordagem de preenchimento e a Seção 7 apresenta as conclusões e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Métodos de aprendizado de máquina podem ser utilizados para explorar padrões característicos dos pacotes enviados pelos dispositivos presentes na rede e inferir informações sobre o usuário. Dessa forma, métodos de ofuscamento alteram atributos do tráfego, como tamanho dos pacotes, a fim de reduzir o desempenho desses modelos. Essa seção resume os conceitos teóricos de aprendizado de máquina, classificação de tráfego e métodos de ofuscamento de tráfego.

2.1 Aprendizado de Máquina

Aprendizado de Máquina é uma área da Inteligência Artificial que permite computadores, a partir de dados prévios, resolverem problemas de predição, regressão e classificação sem a necessidade de explicitamente programar uma solução para cada problema (MENG *et al.*, 2020). Aprendizado de máquina possui três categorias principais, Supervisionado e Não Supervisionado, onde informações relevantes são obtidas a partir de dados rotulados e não rotulados, respectivamente (LIU; LANG, 2019) e Aprendizado por Reforço . No contexto desse trabalho, Aprendizado de Máquina Supervisionado é utilizado para identificar dispositivos presentes em uma rede IoT.

Dessa forma, os algoritmos de Aprendizado de Máquina Supervisionado utilizam valores de entrada e saída predeterminados para prever e classificar atributos relacionados aos dados analisados, nesse contexto, o processo de aprendizado se encerra quando o algoritmo atinge desempenho aceitável (ALLOGHANI *et al.*, 2020). No contexto dos algoritmos de classificação, os dados analisados são separados em um conjunto de treino, utilizado pelo algoritmo para construir os modelos de classificação, e um conjunto de teste, utilizado para validar os modelos gerados (SINGH; THAKUR; SHARMA, 2016).

Além disso, diferentes métodos são utilizados para separar os dados e validar os modelos gerados. Nesse contexto, Cross-Validation consiste em separar os dados em k conjuntos mutuamente exclusivos, onde um dos conjuntos é usado como grupo de teste enquanto os demais como grupo de treino (XU; GOODACRE, 2018). O processo é repetido até que todos os grupos sejam usados como conjunto de teste. Por fim, a média das métricas obtidas em cada iteração é calculada. A seguir são listados alguns dos algoritmos de Aprendizado de Máquina Supervisionado utilizados.

- ***k-Nearest Neighbors* (KNN):** o algoritmo KNN analisa a diferença ou similaridade das entradas baseada na proximidade de seus atributos, calculada pela distância Euclidiana. Dessa forma, uma instância é classificada a partir da classe com maior ocorrência nas k instâncias mais próximas a ela (XIN *et al.*, 2018);
- ***Decision Tree* (DT):** *Decision Tree* ou Árvore de Decisão utiliza estrutura de árvore para classificar as entradas, onde cada nó representa um teste de um dos atributos, os ramos representam os resultados desse teste e as folhas representam a categoria que será atribuída a instância analisada (XIN *et al.*, 2018);
- ***Random Forest* (RF):** consiste em construir diversas Árvores de Decisão e selecionar a classe com mais ocorrências nos resultados individuais (ILIADIS; KAIFAS, 2021);
- ***Support Vector Machine* (SVM):** nesse método de Aprendizado de Máquina, os dados são separados em duas classes distintas por um hiperplano n-dimensional (SARAVANAN; SUJATHA, 2018). A dimensão deste hiperplano varia conforme a quantidade de atributos analisados;

Diversas métricas são utilizadas para calcular o desempenho dos algoritmos de classificação, onde elas variam em função dos valores preditos pelos modelos de classificação e os valores reais. A seguir são listadas algumas métricas utilizadas para analisar o desempenho dos modelos de classificação utilizados.

- **Acurácia:** a razão entre a quantidade de acertos e quantidade de amostras analisadas;
- **Precisão:** a razão entre a quantidade de verdadeiros positivos e a quantidade de amostras classificadas como verdadeiras;
- ***Recall*:** a razão entre os verdadeiros positivos e a quantidade de amostras corretamente classificadas;
- ***F1-Score*:** média harmônica entre as métricas precisão e *recall*.

2.2 Classificação de Tráfego e *Traffic Fingerprint*

O avanço das tecnologias IoT permitiu que objetos do cotidiano proporcionassem uma melhor interação com o usuário e com outros dispositivos, dessa forma ampliando suas funcionalidades. Contudo, o uso dessas tecnologias traz preocupações em relação à privacidade, visto que esses eletrônicos captam e compartilham informações sobre o ambiente e comportamento dos usuários, onde muitos dispositivos IoT não utilizam protocolos seguros em suas requisições. Entretanto, mesmo em uma comunicação cifrada, estatísticas características dos

dispositivos continuam presentes no tráfego e podem ser utilizadas para expor informações do usuário.

Técnicas de Classificação de Tráfego visam categorizar o tráfego de rede em classes apropriadas (REZAEI; LIU, 2019). Classificação de Tráfego é utilizada por diferentes aplicações para prover serviços como sistemas de detectores de intrusão, qualidade de serviço e *firewall* (TAHAEI *et al.*, 2020). Nesse contexto, diferentes métodos, como inspecionamento das portas de origem dos pacotes ou seus conteúdos encapsulados, são utilizados para classificar informações referentes ao tráfego analisado.

Traffic Fingerprint é uma técnica de Classificação de Tráfego que consiste em comparar padrões presentes no tráfego com modelos de classificação já conhecidos a fim de identificar informações como endereços WEB acessados, serviços utilizados e dispositivos presentes no tráfego (SKOWRON; JANICKI; MAZURCZYK, 2020). Algoritmos de Aprendizado de Máquina podem ser utilizados para construir os modelos de classificação de tráfego, nesse contexto, características como tamanho dos pacotes, tempo de resposta, quantidade de envios entre outros atributos podem ser usados como entradas dos algorítimos de classificação.

2.3 Ofuscamento de Tráfego e Preenchimento de Pacotes.

A privacidade de usuários de dispositivos IoT pode ser invadida por ataques baseados em técnicas de Classificação de Tráfego, nesse contexto, técnicas de Ofuscação de Tráfego visam impedir que invasores monitorem o comportamento do usuário a partir de padrões existentes nos envios dos dispositivos presentes na rede (SHEN *et al.*, 2022). Dessa forma, alterar características do tráfego é uma abordagem utilizada para prevenir que invasores relacionem padrões presentes na comunicação com as atividades do usuário (MOHIT; ANSARI; KUMAR, 2021).

Pesquisas no contexto de métodos de ofuscação de tráfego focam em alterar as entradas de modelos de Classificação de Tráfego baseados em Aprendizado de Máquina de forma que reduza a acurácia desses classificadores (PERERA *et al.*, 2022), dessa forma, impedindo invasores infiram dados sobre o usuário. Nesse contexto, as principais técnicas de Ofuscamento de Tráfego consistem em preenchimento de pacotes, modelagem de tráfego e injeção de tráfego (CHEN *et al.*, 2021).

- **Preenchimento de Pacotes:** essa técnica consiste em alterar o tamanho dos pacotes adicionando *bytes* redundantes a fim de ofuscar estatísticas no tráfego (SKOWRON; JANICKI; MAZURCZYK, 2020);

- **Modelagem de Tráfego:** visa fazer com que estatísticas presentes no tráfego não reflitam características reais do usuário (CHEN *et al.*, 2021);
- ***Injeção de Tráfego*:** consiste em adicionar pacotes redundantes na rede de forma que um invasor monitorando o tráfego não seja capaz de distinguir os pacotes reais e os pacotes fictícios (SANTOS *et al.*, 2022);

3 TRABALHOS RELACIONADOS

Nesta seção serão apresentados trabalhos relacionados ao projeto proposto, comparando seus aspectos positivos e quais contribuições podem ser acrescentadas pelo trabalho.

3.1 Adaptive Packet Padding Approach for Smart HomeNetworks: A Tradeoff Between Privacy and Performance

O trabalho desenvolvido por (PINHEIRO *et al.*, 2021) propõe um método de ofuscamento de tráfego baseado em preenchimento, onde dados extras são adicionados no *payload* dos pacotes com o intuito de remover padrões presentes nos tamanhos dos *frames* enviados por diferentes dispositivos IoT. Segundo o autor, soluções similares adicionam quantidades fixas de *bytes* nos pacotes, algo problemático em um cenário de redes IoT. O sistema consiste em uma controladora SDN (*Software Defined Network*) executado por um *home router* que recebe o tráfego IoT, um mecanismo de preenchimento e uma API *Representational State Transfer* (REST). Segundo o autor, o mecanismo de preenchimento poderia ser executado diretamente pelos dispositivos IoT, mas isso causaria complicações devido às limitações energéticas dos dispositivos.

O método possui 4 níveis de preenchimento distintos 100, 500, 700, 900 e ALL. Os níveis mais elevados adicionam uma quantidade maior de dados. O mecanismo analisa o tamanho de cada pacote e baseado no nível atual eleva a quantidade de *bytes* homogenizando o tamanho dos *frames*. Segundo o autor, a maioria do tráfego é ocupado por pacotes de no máximo 300 *bytes*, portanto, o tamanho do *padding* deve ser próximo ao original. A SDN monitora o uso da rede e envia comandos para aumentar ou diminuir o nível de preenchimento. A comunicação entre a SDN e o mecanismo de *padding* é feita pela API REST desenvolvida pelo autor. A API possui comandos simples para economizar uso de banda.

O autor aplicou diferentes algoritmos de aprendizado de máquina em capturas de redes IoT antes e após o método ser aplicado e averiguou como métricas como acurácia, F1-score e recall foram afetadas. O artigo conclui que houve uma redução drástica na taxa de acerto dos modelos após o preenchimento ser aplicado, sendo que a acurácia do *Random Forest* caiu de 96% para 4.96%. O autor analisa o impacto no desempenho do tráfego como a quantidade extra de *bits* adicionados na rede, variação na latência no envio dos pacotes (*jitter*) e a quantidade de *bits* uteis enviados por unidade de tempo (*goodput*). Conclui-se que uma redução da taxa de

acertos dos algoritmos de classificação resulta em um maior *overhead*.

Apesar de obter bons resultados, o trabalho não aborda ataques que exploram outras características dos pacotes como intervalo de envio, IP de origem e destino, *timestamp*, etc. Ademais, o autor usa valores empíricos para modelar o mecanismo de preenchimento o que pode ser problemático caso seja necessário aplicar o sistema em diferentes redes.

3.2 Um Método de Ofuscação para Proteger a Privacidade no Tráfego da Rede IoT

Os autores de (SANTOS *et al.*, 2022) propõem um método de ofuscamento de rede chamado Mitra. O projeto usa inserção de tráfego fictício (*Dummy Traffic*) para mitigar ataques baseados na inspeção de pacotes na rede. A proposta usa o tráfego original para modelar os pacotes fictícios adicionados na rede. O método provê quatro níveis de geração de tráfego, tf-baixo, tf-médio, tf-alto e tf-rand. O último nível mescla aleatoriamente a quantidade de pacotes gerados por tf-baixo e tf-alto. Os pacotes fictícios são gerados com base no tráfego real dos dispositivos, onde o MAC e o IP de destino dos pacotes são preservados, enquanto dados como IP de origem são gerados aleatoriamente.

O tamanho do *frame* gerado é idêntico ao último *frame* real enviado. Para cada dispositivo presente na rede uma quantidade específica de pacotes falsos é gerada. Essa quantidade varia conforme o nível de ofuscamento escolhido. Algoritmos de aprendizado de máquina como XGBoost, CART, Random Forest e Bagging foram aplicados em capturas que totalizam mais de 32GB. Em média, as métricas F1-score e acurácia resultaram aproximadamente 65% e 98% respectivamente.

Os autores aplicaram o método de ofuscamento na mesma captura e compararam o impacto causado nas métricas. Como resultado, o F1-score reduziu para 30% com TF-baixo e 25% com TF-médio gerando um acréscimo de 0.13% e 0.82% de dados, respectivamente. A quantidade de pacotes enviados na rede pelo método é modelada de forma empírica, além de não variar conforme a intensidade de tráfego na rede. Possíveis contribuições seriam métricas mais precisas e dinâmicas para escolher a quantidade de dados inseridos no tráfego.

3.3 Protecting IoT-environments against Traffic Analysis Attacks with Traffic Morphing

Os autores de (HAFEEZ *et al.*, 2019) propõem um sistema de ofuscamento de tráfego que usa tráfego fictício para mascarar estatísticas presentes na rede. Os pacotes gerados pelo

método são enviados em períodos de inatividade dos dispositivos IoT prevenindo que invasores detectem padrões no tráfego, já que os pacotes fictícios diferem pouco dos reais. O projeto proposto conecta um conjunto de dispositivos IoT como *smart cameras* e assistentes virtuais a uma placa Raspbarry-Pi 3 que serve de ponto de acesso. Todos os dispositivos são controlados por aplicativos móveis disponibilizados pelo fabricante.

O autor aplica 2 algoritmos de aprendizado de máquina, KNN e *Random Forest*, em capturas feitas nessa rede. Em média, os algoritmos obtiveram uma acurácia de 87% e 86%, respectivamente, quando utilizados para detectar os dispositivos na rede. Para evitar ataques baseados em estatísticas, o autor propõe enviar constantemente tráfego fictício na rede independente da atividade dos dispositivos, e quando o IoT estiver inativo, tráfego fictício representando atividades também é inserido. O sistema descrito usa duas filas Q_r e Q_d com o tráfego real e fictício, respectivamente, sendo que Q_r possui uma maior prioridade. Para gerar os pacotes fictícios, o autor usa dados do tráfego real e aplica métodos de interpolação para escolher o tamanho dos pacotes e o tempo de envio. Segundo o autor, é garantido que Q_d possui sempre pacotes suficientes para serem enviados.

O resultado obtido foi uma queda da acurácia para 21% e 24% para o RF e KNN, respectivamente. O trabalho identificou que dispositivos com menor frequência de envio apresentaram uma melhor acurácia mesmo após o tráfego fictício ser aplicado. O autor atribui esse fato a baixa diversidade de dispositivos IoT.

O projeto usa o próprio tráfego para modelar os pacotes fictícios mantendo similaridade com os pacotes reais. Apesar do tráfego real possuir uma maior prioridade, o autor não explora nenhum método para escolher a quantidade de tráfego fictício gerado, o que pode acarretar atrasos na comunicação dos dispositivos IoT.

3.4 Comparação

A Tabela 1 compara os trabalhos apresentados com o projeto propostos. Em geral, os trabalhos apresentados definem um conjunto de níveis de ofuscamento, onde níveis que adicionam um maior *overhead* no tráfego reduzem a taxa de acerto dos modelos de classificação utilizados para identificar os dispositivos na rede. A abordagem de ofuscamento proposta por esse projeto utiliza preenchimento para remover estatísticas presentes nos pacotes. Como contribuição, o projeto utiliza modelos de otimização para encontrar o tamanho do preenchimento que reduza a quantidade extra de *bytes* inseridos na rede. A abordagem utiliza 4 níveis de ofuscamento,

onde o tamanho dos pacotes após o preenchimento é definido pela heurística utilizada.

Tabela 1 – Comparação dos trabalhos relacionados

Autor	Abordagem	Metodologia	Algoritmos de Classificação Utilizados	Métricas Analisadas
(PINHEIRO *et al.*, 2021)	Preenchimento de pacotes	Define 4 níveis de preenchimento, onde uma controla SDN escolhe o nível utilizado conforme a intensidade do tráfego.	KNN, RF, DT e SVM	Acurácia, Recall e F1-Score
(SANTOS *et al.*, 2022)	Inserção de Tráfego Fictício	Define 4 níveis de ofuscamento, onde cada nível adiciona uma quantidade maior de pacotes fictícios na rede.	XGBoost, CART, Random Forest e Bagging	Acurácia, Recall, Precisão e F1-Score
(HAFEEZ *et al.*, 2019)	Inserção de Tráfego Fictício	Constantemente envia pacotes fictícios na rede e, em períodos de inatividade, pacotes simulando o tráfego real dos dispositivos presentes na rede.	RF e KNN	Precisão, Recall e F1-Score
Proposto	Preenchimento de pacotes	Define 4 níveis de preenchimento, onde uma heurística é utilizada para definir o tamanho do preenchimento que reduza o overhead.	KNN, RF, DT e SVM	Acurácia, Recall e F1-Score

Fonte: O Autor

4 HEURÍSTICA DE PREENCHIMENTO

A heurística proposta nesse trabalho é baseada no projeto realizado por (VALE; BRANDAO; GRIVET, 2006), que apresenta uma abordagem de preenchimento de pacotes voltado à comunicação de redes móveis. No contexto das *Universal Mobile Telecommunication System* (UTMS), os blocos de informações são preenchidos para que seus tamanhos encaixem em um conjunto de *frames* de tamanhos especificados pelo canal de comunicação. A heurística define um modelo de otimização para encontrar tamanhos para esses *frames* que minimizem a quantidade de *bits* extras adicionados nos pacotes.

Sendo $\Omega = \{x_1, x_2, \ldots, x_N\}$ o conjunto com os possíveis tamanhos dos pacotes enviados em um canal de comunicação e $\underline{X} = \{X_1, X_2, \ldots, X_M\}$ o vetor contendo os tamanhos dos *frames* utilizados nesse canal, onde $M < N$ e $X_1 < X_2 < \ldots < X_M$, um pacote de tamanho x_k deve ser preenchido para acomodar um *frame* de tamanho X_m se e somente se $X_{m-1} < x_k \leq X_m$. A Tabela 2 demonstra um exemplo de uma sessão *Voice Over IP* (VoIP) onde os pacotes enviados devem ser preenchidos para acomodar um conjunto de 3 *frames* de tamanho $\underline{X} = \{928, 312, 104\}$. Após o preenchimento, todos os pacotes possuem o mesmo tamanho de um dos valores de $\underline{X}$.

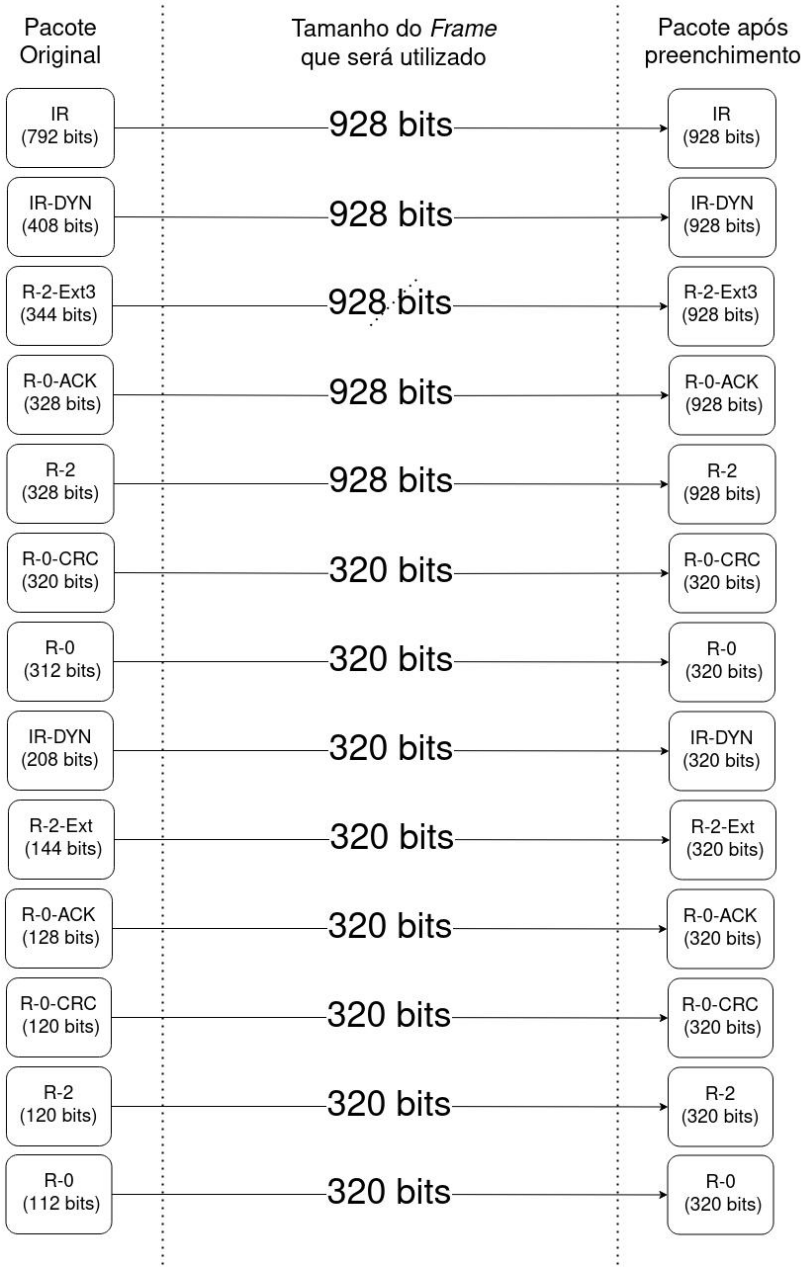
Tabela 2 – Exemplo de preenchimento.

Tipo do pacote	Tamanho original (bits)	Tamanho após preenchimento(bits)	Frequência Relativa(%)	Razão de Preenchimento(%)
IR (speech)	792	928	0.000047	14.66
IR-DYN (speech)	408	928	0.89	56.03
R-0 (speech)	312	312	77.02	0
R-0-CRC (speech)	320	928	3.92	65.52
R-2 (speech)	328	928	3.56	64.66
R-2-Ext3 (speech)	344	928	0.89	62.93
R-0-ACK (speech)	328	928	2.61	64.66
IR-DYN (SID)	208	312	0.11	33.33
R-0 (SID)	112	312	9.63	64.10
R-0-CRC (SID)	120	312	0.49	61.54
R-2 (SID)	128	312	0.44	58.97
R-2-Ext (SID)	144	312	0.11	53.85
R-0-ACK (SID)	128	312	0.33	58.97

Fonte: Adaptado de (VALE; BRANDAO; GRIVET, 2006).

A frequência relativa a_k do pacote de tamanho x_k, resumida na Equação 4.1, é definida como a razão entre a quantidade de envios desse pacote e a quantidade total de pacotes enviados. A razão de preenchimento, definida na Equação 4.2, resulta em 0 caso o tamanho do pacote não seja alterado e cresce a medida que o tamanho do pacote é elevado pelo preenchimento. A Figura 1 ilustra o tamanho dos pacotes na sessão VoIP antes e após o preenchimento ser aplicado no tráfego.

Figura 1 – Ilustração do processo de preenchimento.



Fonte: Adaptado de (VALE; BRANDAO; GRIVET, 2006).

$$a_k = \frac{\text{Quantidade de envios do pacote } x_k}{\text{Quantidade total de pacotes enviados}} \tag{4.1}$$

$$\text{Razão de Preenchimento} = 1 - \frac{\text{Tamanho original do pacote}}{\text{Tamanho do pacote após o preenchimento}} \tag{4.2}$$

Dessa forma, após o preenchimento, o tamanho dos pacotes de tamanho entre 928 e 313 *bits*, 312 e 105 *bits* e menores que 104 *bits* será 928, 312 e 104 respectivamente. A média das razões de preenchimento dos pacotes é cerca de 14.64%, logo, em média, o tamanho do pacote é acrescido em aproximadamente 15% após o preenchimento. A heurística proposta por (VALE; BRANDAO; GRIVET, 2006) visa escolher os valores de $\underline{X}$ de forma que reduza o *overhead* gerado pelo preenchimento. O custo médio de aplicar o preenchimento dos pacotes é definido como:

$$D(\underline{X}) = \sum_{m=1}^{M} X_m \cdot [F(X_m) - F(X_{m-1})] \tag{4.3}$$

onde a função $F(W)$, definida na Equação 4.4, calcula a soma das frequências relativas dos pacotes de tamanhos menores que W, logo o termo $X_m[F(X_m) - F(X_{m-1})]$ é o custo médio de aplicar o preenchimento nos pacotes de tamanho entre X_m e X_{m-1}.

$$F(W) = \sum_{n|x_n \leq W} \alpha_n \tag{4.4}$$

Dessa forma, para encontrar o tamanho dos *frames* que reduzam o custo do preenchimento, basta encontrar os valores de $\underline{X}$ que minimizem a Equação 4.3. A Equação 4.5 resume a heurística como um modelo de otimização que pode ser utilizado por solucionadores para encontrar os valores contidos em $\underline{X}$ que minimizem o *overhead*, dessa forma $D(\underline{X})$ pode ser entendida como a função objetivo da heurística.

$$\begin{aligned} &\min_{X} \quad D = \sum_{m=1}^{M} X_m \cdot [F(X_m) - F(X_{m-1})] \\ &\text{s.t} \quad X_1 < X_2 < \ldots < X_M \end{aligned} \tag{4.5}$$

Aplicando a heurística proposta na sessão VoIP resumida na Tabela 2, os tamanhos obtidos por (VALE; BRANDAO; GRIVET, 2006) a partir do modelo de otimização foram

$\underline{X} = \{128, 328, 792\}$. Utilizando os valores obtidos a partir da heurística proposta, a razão de preenchimento resultante é cerca de 6.52%, representando uma redução média acima de 50% ao aplicar o preenchimento nos pacotes. Após definir os valores de $\underline{X}$, o tamanho do pacote de tamanho k pode ser definido a partir do Algoritmo 1, que percorre os valores contidos em $\underline{X}$ a fim de encontrar o primeiro valor X_i que satisfaça $X_{i-1} < k \leq X_i$.

Algoritmo 1: preenchimento(k,$\underline{X}$).

Require: O Vetor contendo o tamanho dos *frames* ($\underline{X}$) e o tamanho do pacote enviado (k).
Ensure: $\underline{X}$ minimiza o custo médio $D(\underline{X})$
  $M \leftarrow$ Quantidade de *frames* em $\underline{X}$.
  for $i \leftarrow 1, 2, \ldots, M-1$ **do**
    if $X_i < k \leq X_{i+1}$ **then**
      return X_{i+1}
    end if
  end for

Fonte: O Autor.

5 METODOLOGIA

Essa seção apresenta o conjunto de passos necessários para implementar e avaliar a abordagem proposta.

5.1 Obtenção do tráfego IoT

A primeira etapa realizada pelo projeto é a obtenção de capturas de tráfego gerado por dispositivos IoT utilizados para treinar os modelos de classificação e testar a eficácia do método de preenchimento para ofuscar estatísticas presentes nos pacotes. Os dados utilizados são *datasets* preprocessados e disponibilizados por (PINHEIRO *et al.*, 2021), que consistem em capturas de tráfego realizadas por (SIVANATHAN *et al.*, 2019) em uma rede com 28 dispositivos IoT e não IoT como câmeras inteligentes, sensores de fumaça, sensores de pressão arterial, etc.

Os dados utilizados consistem em 20 arquivos *Comma-Separated Values* (CSV) contendo informações como tamanho dos pacotes, tempo de envio e endereço IP de origem e destino. Os endereços *Media Acess Control* (MAC) de origem são utilizados para rotular os dispositivos, onde um valor numérico distinto é atribuído a cada endereço. Esse pré-processamento é necessário pois os algoritmos de classificação recebem apenas entradas numéricas.

O autor (PINHEIRO *et al.*, 2021) analisa apenas o tamanho de pacotes enviados em um período de 1 segundo onde a média, desvio padrão e o total de *bytes* enviados são utilizados para treinar os modelos de classificação K-ésimo Vizinho mais Próximo (KNN), Árvores de Decisão (DT), Floresta Aleatória (RF) e Máquina de vetores de suporte (SVM). Além do *dataset* utilizado, o autor disponibiliza a implementação utilizada para aplicar o método de preenchimento, treinar os modelos de Aprendizado de Máquina e calcular a desempenho dos modelos após o método de preenchimento ser aplicado.

Nesse contexto, a implementação desenvolvida por (PINHEIRO *et al.*, 2021) para treinar e validar os modelos de Aprendizado de Máquina é utilizada para aplicar e avaliar o método de preenchimento proposto pelo projeto, onde a única alteração realizada é o algoritmo de preenchimento aplicado nos pacotes. Dessa forma, os resultados obtidos são comparados com o resultados apresentados por (PINHEIRO *et al.*, 2021) para avaliar a desempenho do método e a quantidade de *bytes* adicional.

5.2 Abordagem de Preenchimento

O projeto utiliza a heurística de preenchimento proposta por (VALE; BRANDAO; GRIVET, 2006), contudo no contexto de ofuscamento de tráfego de dispositivos IoT. Dessa forma, é necessário definir apenas a quantidade de *frames* (M) que deverão acomodar o tamanho dos pacotes. Essa heurística foi escolhida devido à sua simplicidade, já que depende apenas da distribuição de probabilidade do tamanho dos pacotes, e pela semelhança com o método de preenchimento proposto por (PINHEIRO *et al.*, 2021), que homogeniza o tamanho dos pacotes enviados na rede.

Inicialmente é necessário obter os valores de α e Ω para o tráfego analisado. A Unidade Máxima de Transmissão (MTU) se refere ao tamanho máximo de um pacote transmitido na rede, logo Ω é definido como todos os valores menores ou iguais que a MTU no contexto do projeto. A frequência relativa a_k é obtida percorrendo os arquivos CSV e armazenando a quantidade de ocorrências dos pacotes de tamanho k. As Equações 5.1 e 5.2 resumem os vetores Ω e α_k.

$$\Omega = \{1, 2, \ldots, MTU\} \tag{5.1}$$

$$\alpha_k = \frac{\text{Quantidade de pacotes de tamanho k}}{\text{Quantidade total de pacotes nos } \textit{Datasets}} \tag{5.2}$$

Uma vez que os valores de Ω e α foram obtidos, é necessário obter os valores do vetor $\underline{X}$ que minimize a função objetivo $D(\underline{X})$. A heurística foi implementada utilizando a linguagem Python e a biblioteca Scipy, onde o método Powell foi usado para obter os valores de $\underline{X}$ que minimizem o *byte overhead* gerado pelo preenchimento. A partir dos valores $\underline{X}$ é possível obter o tamanho do pacote após o preenchimento a partir do Algoritmo 1.

5.3 Análise de desempenho

A análise de desempenho do mecanismo de preenchimento proposto é feita aplicando os algoritmos de classificação apresentados na Seção 5.1 após o método alterar o tamanho dos pacotes enviados nas capturas e analisando como as métricas utilizadas são afetadas, onde uma redução na taxa de acertos indica melhorias na privacidade fornecida ao usuário. Além disso, é necessário analisar o *byte overhead* gerado pelo método de preenchimento, o que é realizado comparando o tamanho total do tráfego antes e após o preenchimento ser aplicado.

As métricas analisadas são Acurácia, *Recall* e *F1-Score*. Em todas elas, valores próximos a 1 indicam que os modelos de Aprendizado de Máquina são capazes de identificar com uma boa taxa de acerto os dispositivos IoT presentes na rede analisando o tamanho dos pacotes. Logo, o objetivo do método de preenchimento é reduzir as métricas para valores próximos a 0.

Ademais, os dois cenários propostos por (PINHEIRO *et al.*, 2021) são analisados pelo projeto. No primeiro cenário, invasor externo, os algoritmos de Aprendizado de Máquina são treinados antes do método de preenchimento ser aplicado. Dessa forma, é simulado um invasor que possui acesso a dispositivos IoT e pretende utilizar os modelos gerados para identificar os dispositivos presentes na rede do usuário.

No segundo cenário, invasor interno, os modelos são treinados com o tráfego após o tamanho dos pacotes serem alterados pelo preenchimento. Este cenário simula um invasor que, além dos dispositivos IoT, possui acesso ao método de preenchimento utilizado, como um Provedor de Acesso à Internet que analisa a atividade do usuário em sua residência. No cenário de um invasor interno, o autor (PINHEIRO *et al.*, 2021) utiliza a técnica de *cross-validation* para treinar e validar e os modelos de classificação de tráfego.

O custo de aplicar os métodos de ofuscação no tráfego é analisado comparando o tamanho original do tráfego com o tamanho após o preenchimento ser aplicado. O *byte overhead* é calculado analisando o tamanho dos pacotes nos arquivos CSV antes e após o algoritmo de preenchimento ser aplicado.

$$\textit{Byte Overhead} = \frac{\text{Tamanho após preenchimento}}{\text{Tamanho original}} \tag{5.3}$$

5.4 Trade-off

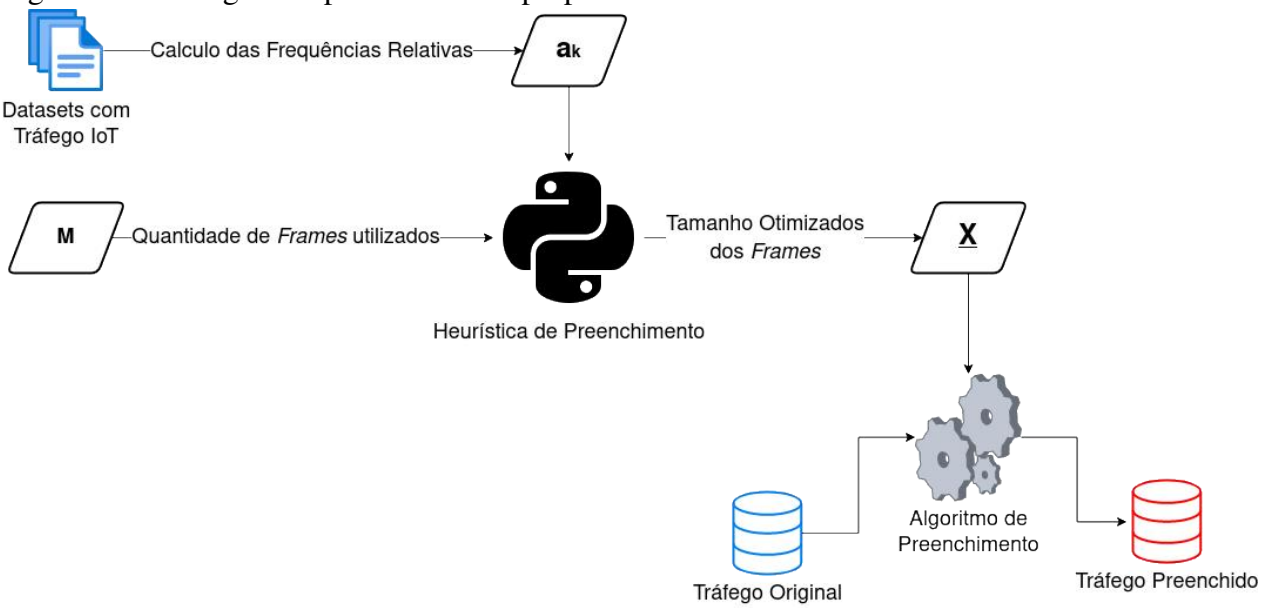
Reduzir a quantidade de *bytes* inserido nos pacotes pode favorecer um invasor monitorando o tráfego de dispositivos IoT aumentando a acurácia dos modelos de classificação utilizados. Nesse contexto, analisar a acurácia e o *byte overhead* separadamente não reflete totalmente a eficácia da abordagem de preenchimento analisada. Dessa forma, o *trade-off* entre privacidade e desempenho pode ser quantificado em função da quantidade de *bytes* inseridos e da taxa de acerto dos algoritmos utilizados.

$$\textit{trade-off} = \frac{1 - \text{Acurácia}}{\textit{Byte Overhead}} \tag{5.4}$$

A Equação 5.4 resume a métrica utilizada por esse projeto para calcular o *trade-off* aplicado pela abordagem de preenchimento. A equação se aproxima de 1 conforme a acurácia e o *overhead* são reduzidos, indicando um bom balanceamento entre privacidade e desempenho. Em contrapartida, um resultado próximo a 0 indica que a privacidade fornecida ao usuário, medida pela acurácia, é pequena quando comparada com o *overhead* aplicado no tráfego. Dessa forma, mesmo que o método de preenchimento reduza o exito de um invasor monitorando o tráfego, um alto *byte overhead* reduzirá o valor da métrica, da mesma forma que um baixo *byte overhead* e uma alta acurácia fará o resultado tender a 0.

A Figura 2 resume a abordagem de preenchimento proposta pelo projeto. Inicialmente *datasets* contendo capturas de tráfego IoT são utilizados para obter a frequência relativa de um pacote em função do seu tamanho, onde esses dados são utilizados pela heurística demonstrada na Seção 4 para encontrar os valores de $\underline{X}$. Por fim, os valores de $\underline{X}$ são usados pelo algoritmo de preenchimento para alterar o tamanho dos pacotes, consequentemente, afetando o desempenho de modelos de classificação aplicados no tráfego.

Figura 2: Abordagem de preenchimento proposta.

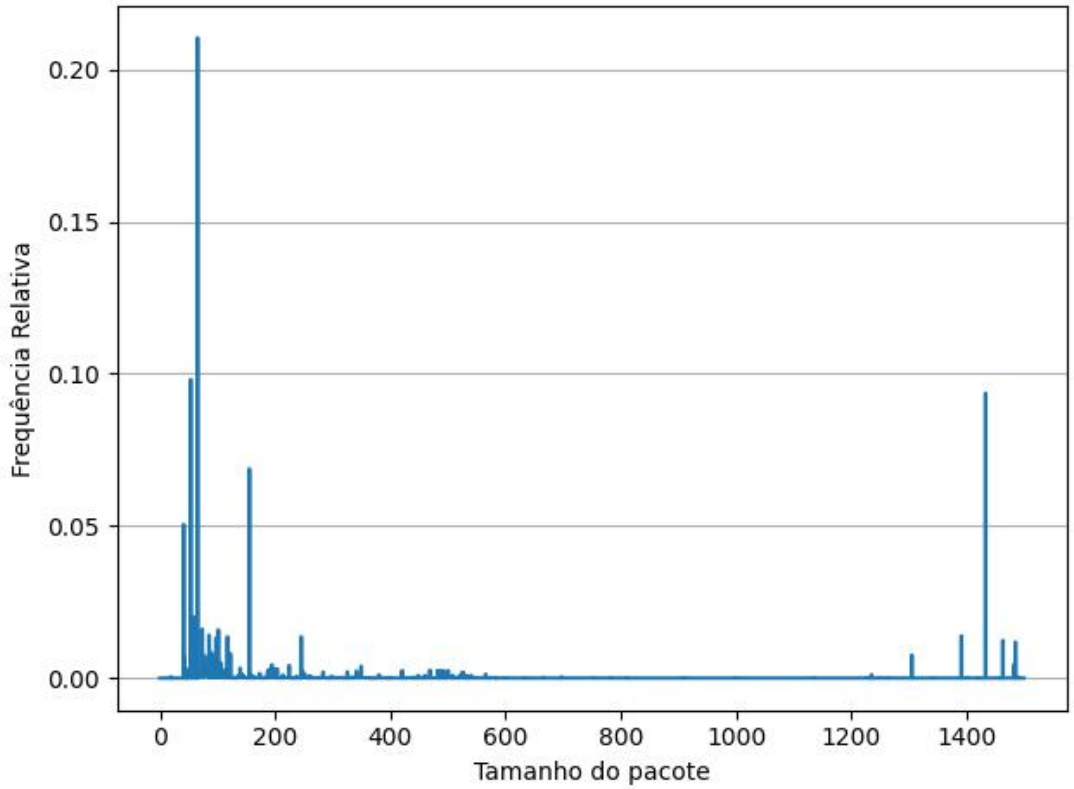


Fonte: O Autor.

6 RESULTADOS

A Figura 3 ilustra a frequência relativa em função do tamanho do pacote enviado em *bytes*. Como o autor de (PINHEIRO *et al.*, 2021) cita em seu trabalho, os pacotes com tamanho menor que 300 *bytes* possuem uma frequência de envio maior que os demais pacotes. As frequências relativas calculadas são utilizadas pela heurística de preenchimento apresentada na Seção 4 para obter o tamanho do preenchimento em 4 cenários onde 5,4,3 e 2 *frames* são utilizados para acomodar os pacotes. A Tabela 3 resume o tamanho dos *frames* obtidos a partir da heurística de preenchimento.

Figura 3: Probabilidade de envio dos pacotes com diferentes tamanhos



Fonte: O Autor.

Tabela 3: Tamanho dos *frames* obtidos

Quantidade de frames utilizado	Tamanho dos frames
5	66,123,235,543,1500
4	66,196,542,1500
3	124,541,1500
2	156,1500

Fonte: O Autor.

Similar a (PINHEIRO *et al.*, 2021), que propõe 4 níveis de preenchimento, *Level 100*, *Level 500*, *Level 700* e *Level 900*, o método de preenchimento desenvolvido propõe 4 níveis de preenchimento *frames 5*, *frames 4*, *frames 3* e *frames 2*, onde o Algoritmo 1 é utilizado com 5, 4, 3 e 2 *frames*, respectivamente. Para cada um dos níveis de preenchimento, a Tabela 4 relaciona o tamanho original dos pacotes com o seu tamanho após o preenchimento utilizando os valores da Tabela 3.

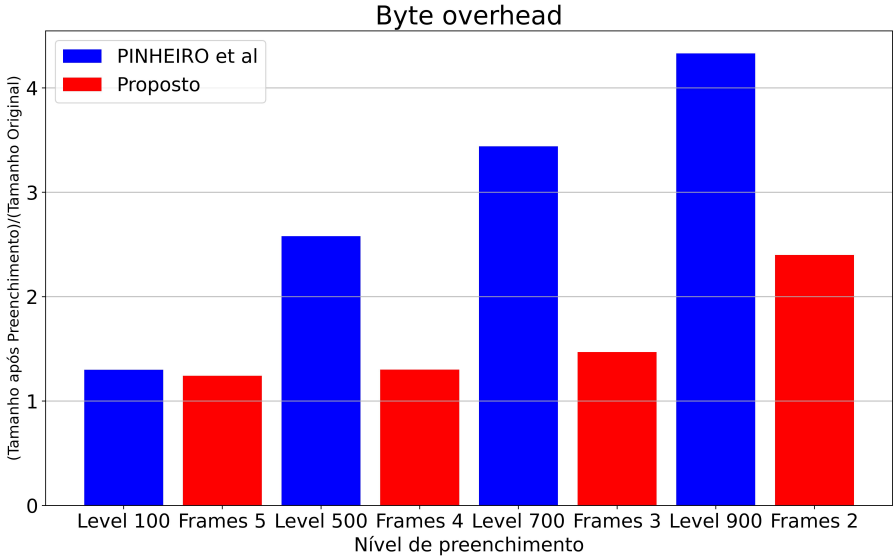
Tabela 4: Preenchimento para os 4 níveis propostos.

Nível de preenchimento	Tamanho original (k)	Tamanho após preenchimento
Frames 5	$1 < k \leq 66$	66
	$66 < k \leq 123$	123
	$123 < k \leq 235$	235
	$235 < k \leq 543$	543
	$543 < k \leq 1500$	1500
Frames 4	$1 < k \leq 66$	66
	$66 < k \leq 196$	196
	$196 < k \leq 543$	543
	$543 < k \leq 1500$	1500
Frames 3	$1 < k \leq 124$	124
	$124 < k \leq 541$	541
	$541 < k \leq 1500$	1500
Frames 2	$1 < k \leq 156$	156
	$156 < k \leq 1500$	1500

Fonte: O Autor.

6.1 *Byte Overhead*

Figura 4: *Byte overhead* dos métodos de preenchimento



Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

A Figura 4 compara o *byte overhead* gerado pelos níveis de preenchimento propostos por (PINHEIRO *et al.*, 2021) e pela heurística utilizada neste projeto. É possível verificar que

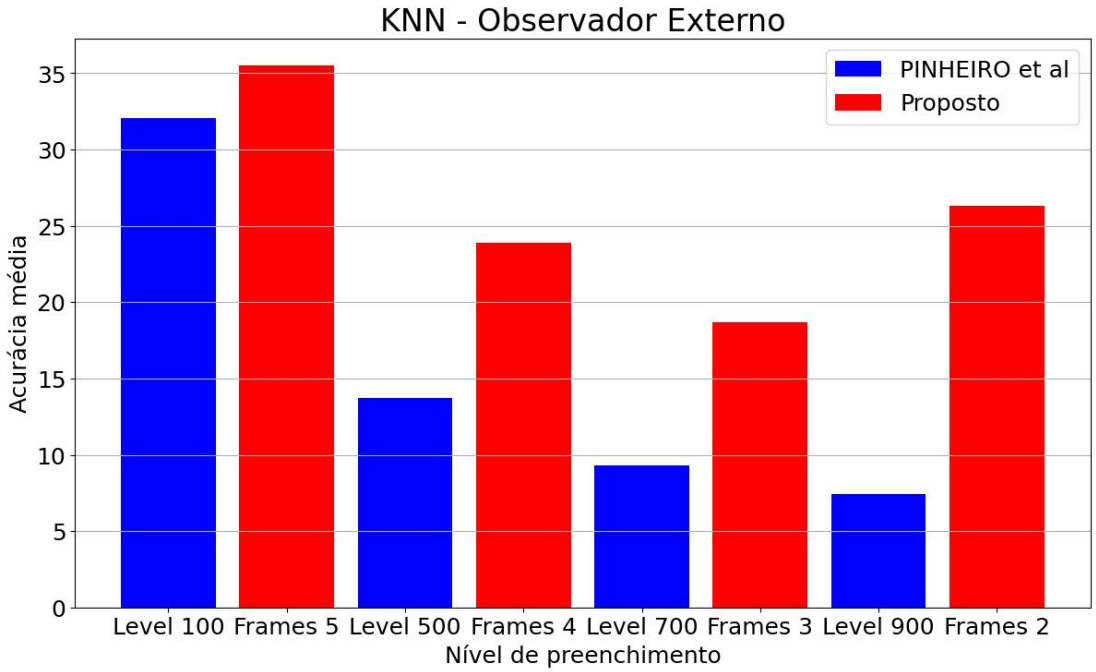
a quantidade de *bytes* adicionados no tráfego aumenta conforme menos *frames* são utilizados. Ademais, a abordagem de preenchimento proposta apresenta um custo de preenchimento menor que (PINHEIRO *et al.*, 2021), onde, no máximo, a quantidade de *bytes* após o preenchimento é 2.4 vezes maior a quantidade original.

6.2 Observador externo

Inicialmente, é analisado como a abordagem de preenchimento afeta os modelos de classificação utilizados por um observador externo a fim de inferir informações sobre o usuário. Dessa forma, o invasor utiliza o tráfego real de dispositivos IoT para treinar os modelos e aplicá los na rede interna do usuário. As Figuras 5, 6, 7 e 8 comparam as acurácias obtidas pelos modelos de classificação após as abordagens de preenchimento serem aplicadas. Além disso, as Figuras comparam os resultados obtidos por (PINHEIRO *et al.*, 2021).

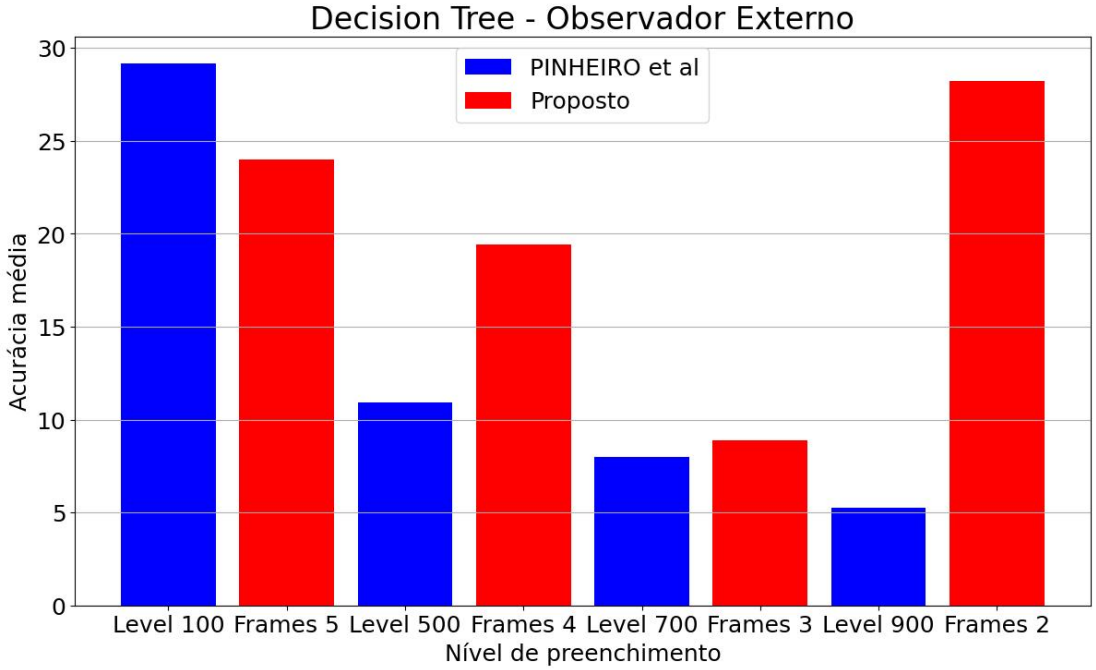
Nos algoritmos KNN, DT e RF, a acurácia diminui conforme o *overhead* aumenta, com exceção do nível de preenchimento *frames 2*, onde a acurácia apresentou um acréscimo mesmo sendo o nível de preenchimento que adiciona a maior quantidade de *bytes* extras. Por outro lado, o nível *frames 2* apresentou a menor acurácia quando o algoritmo SVM foi utilizado. Em todos os casos, o método de preenchimento foi capaz de reduzir a acurácia dos modelos de classificação para valores abaixo de 50%.

Figura 5: Acurácia média KNN



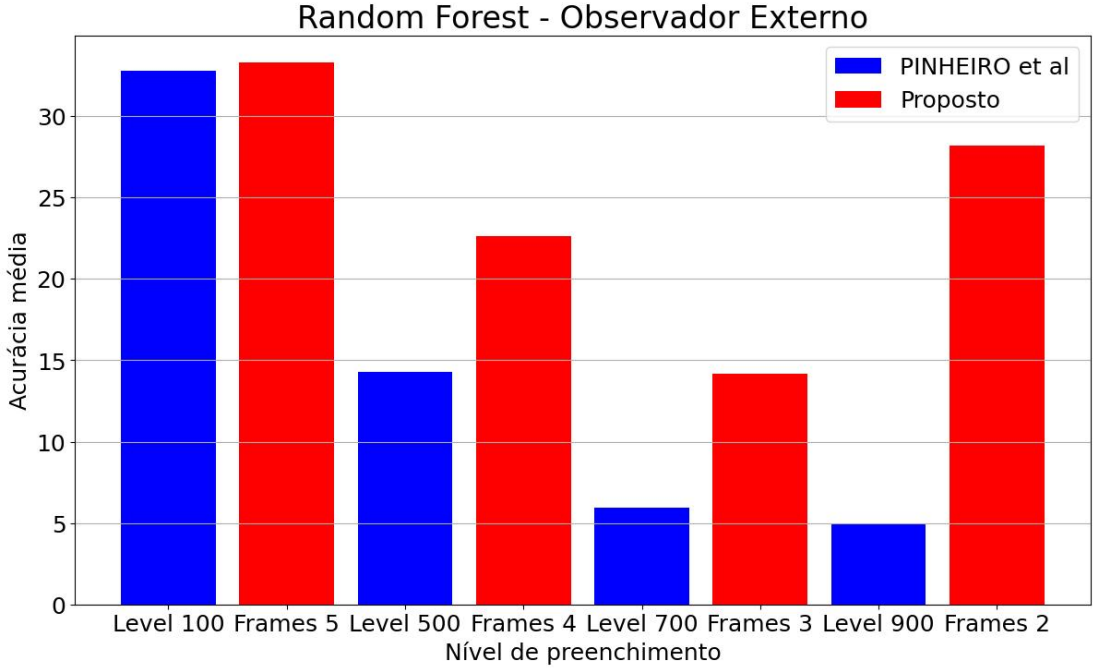
Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

Figura 6: Acurácia média *Decison Tree*



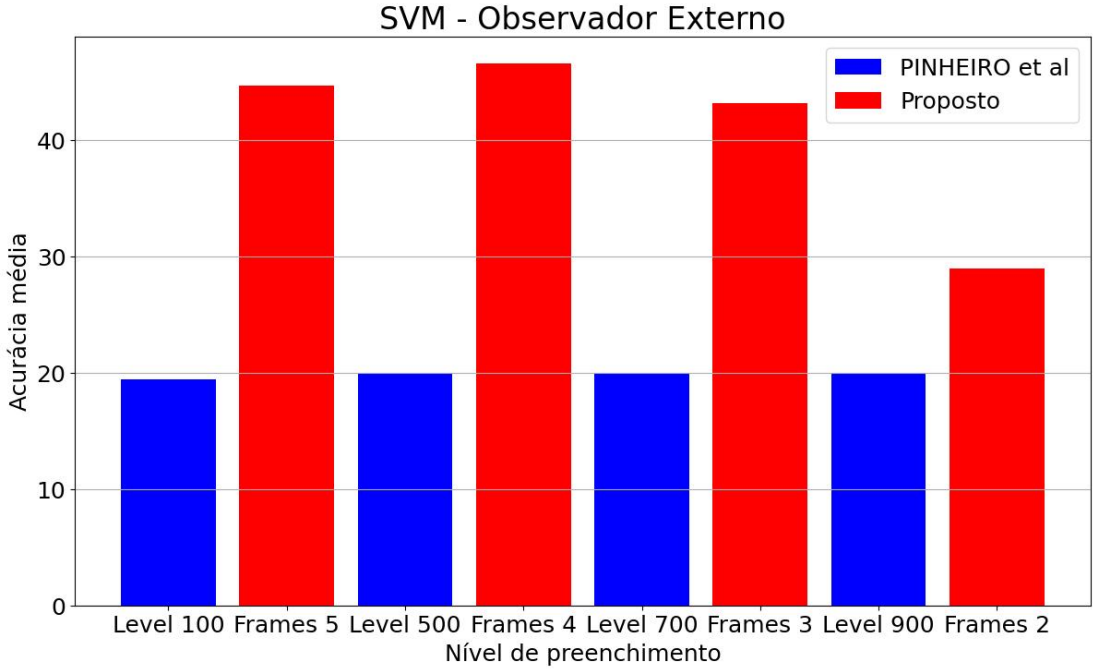
Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

Figura 7: Acurácia média *Random Forest*



Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

Figura 8: Acurácia média Support *Vector Machine*



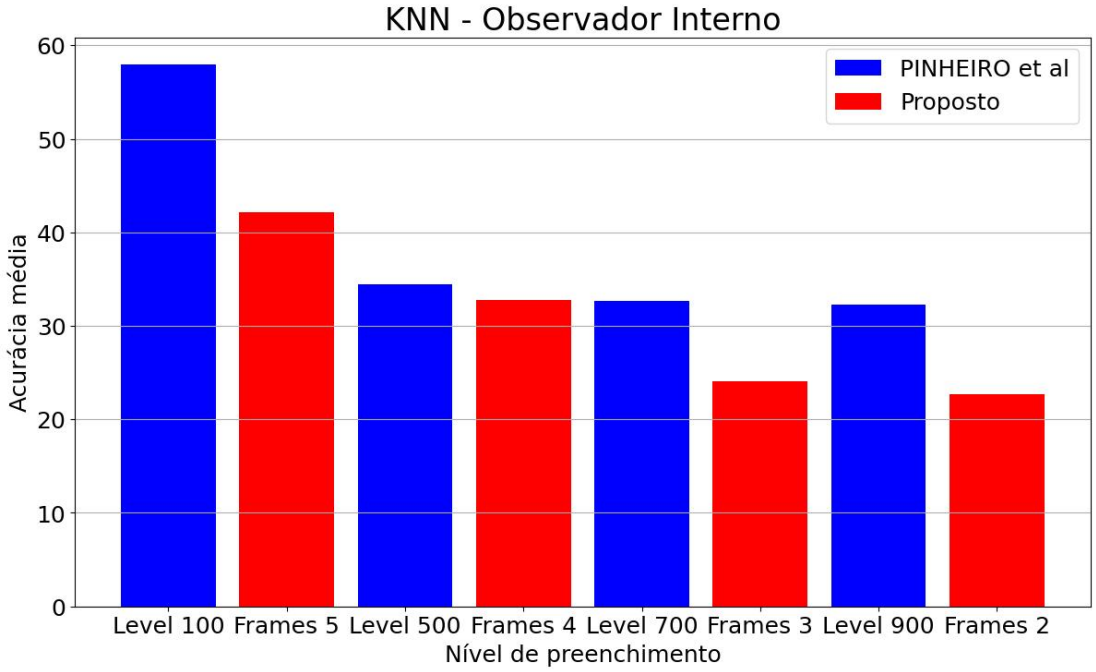
Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

6.3 Observador interno

No caso do observador interno, o invasor possui acesso ao algoritmo de preenchimento utilizado, dessa forma os modelos de classificação são treinados após o algoritmo de preenchimento ser aplicado nos pacotes. Novamente, as Figuras 9, 10, 11 e 12 ilustram as acurácias obtidas pelos modelos de classificação após aplicar as abordagens de ofuscamento, onde os resultados indicam que um invasor obtém maior exito treinando os modelos de classificação utilizando o tráfego já preenchido.

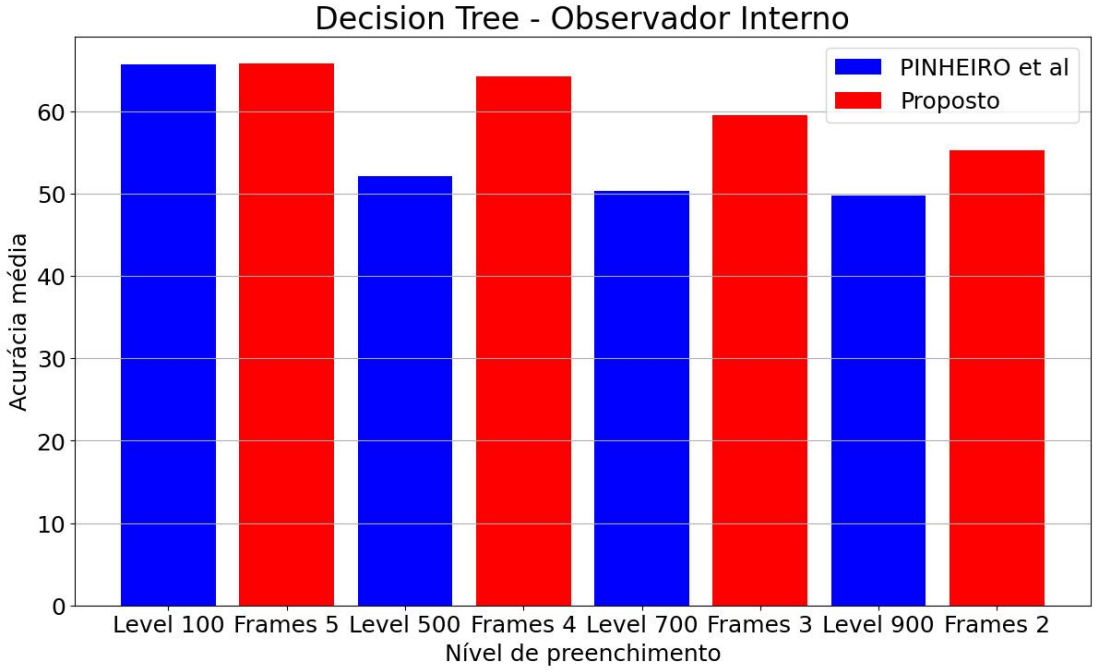
Diferente do observador externo, a acurácia dos algoritmos KNN, DT, RF e SVM tendem a decrescer conforme o tamanho do preenchimento aumenta em todos os níveis propostos. Todos os algoritmos utilizados obtiveram uma acurácia média inferior a 70%, onde a abordagem proposta apresentou melhor desempenho que o trabalho comparado quando o algoritmo KNN foi utilizado.

Figura 9: Acurácia média KNN



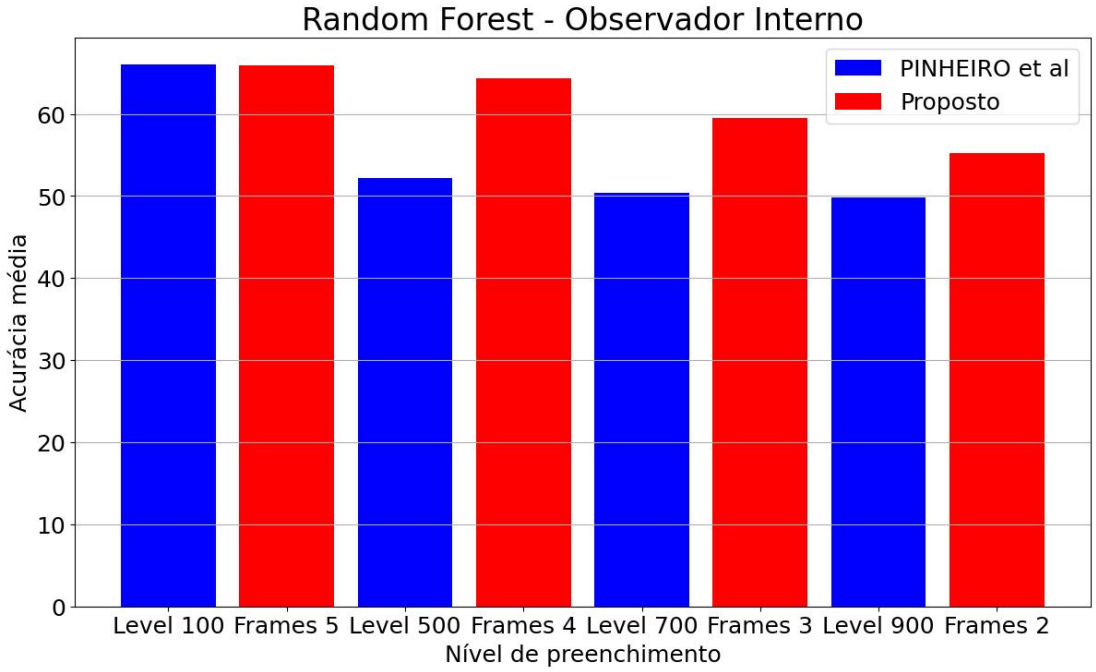
Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

Figura 10: Acurácia média *Decison Tree*



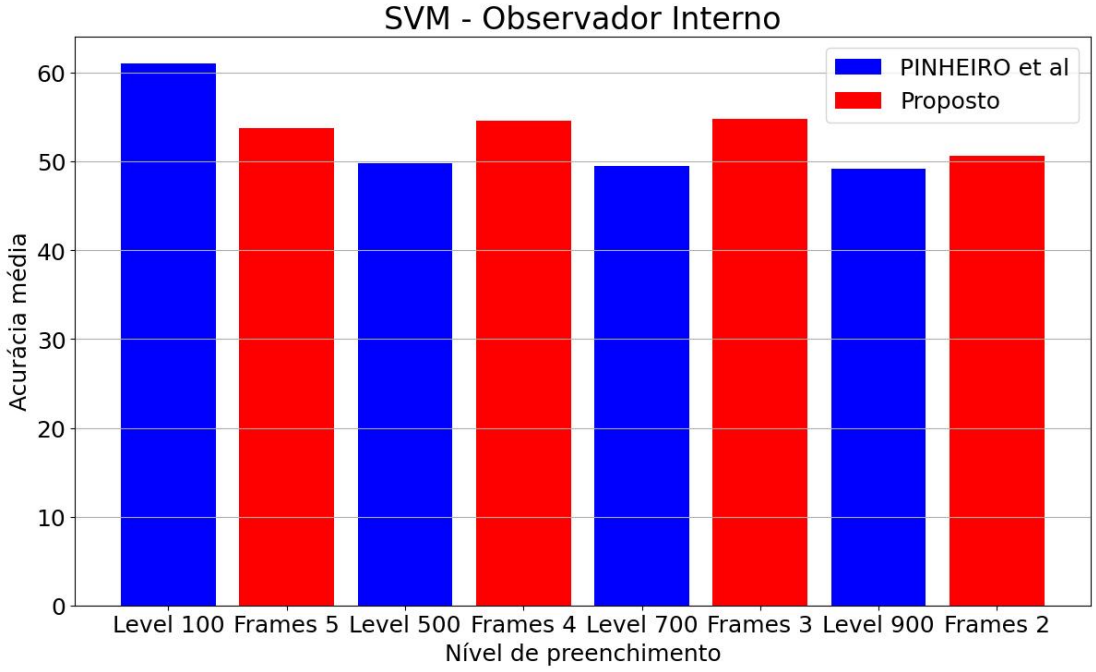
Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

Figura 11: Acurácia média *Random Forest*



Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

Figura 12: Acurácia média Suport *Vector Machine*



Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

6.4 *Trade-off*

Os resultados apresentados nas Seções 6.2 e 6.3 demonstram que, em geral, o método proposto por (PINHEIRO *et al.*, 2021) é mais eficiente em reduzir a taxa de acertos dos modelos de classificação de tráfego, contudo, os resultados apresentados na seção 6.1 indicam que a abordagem desenvolvida adiciona uma quantidade menor de *bytes* extras a rede. Dessa forma, essa seção analisa o *trade-off* entre privacidade, medida através da acurácia, e o desempenho, medido através do *byte overhead*, de ambos os projetos.

Nesse contexto, a Equação 5.4 é utilizada para quantizar o *trade-off*, onde valores próximos a 1 indicam que o preenchimento conseguiu reduzir o exito de um invasor monitorando o tráfego e minimizou a quantidade de *bytes* inseridos nos pacotes. Por outro lado, resultados próximos de zero indicam que os modelos de classificação obtiveram uma alta acurácia ou o preenchimento adicionou um alto *overhead* no tráfego.

6.4.1 Observador Externo

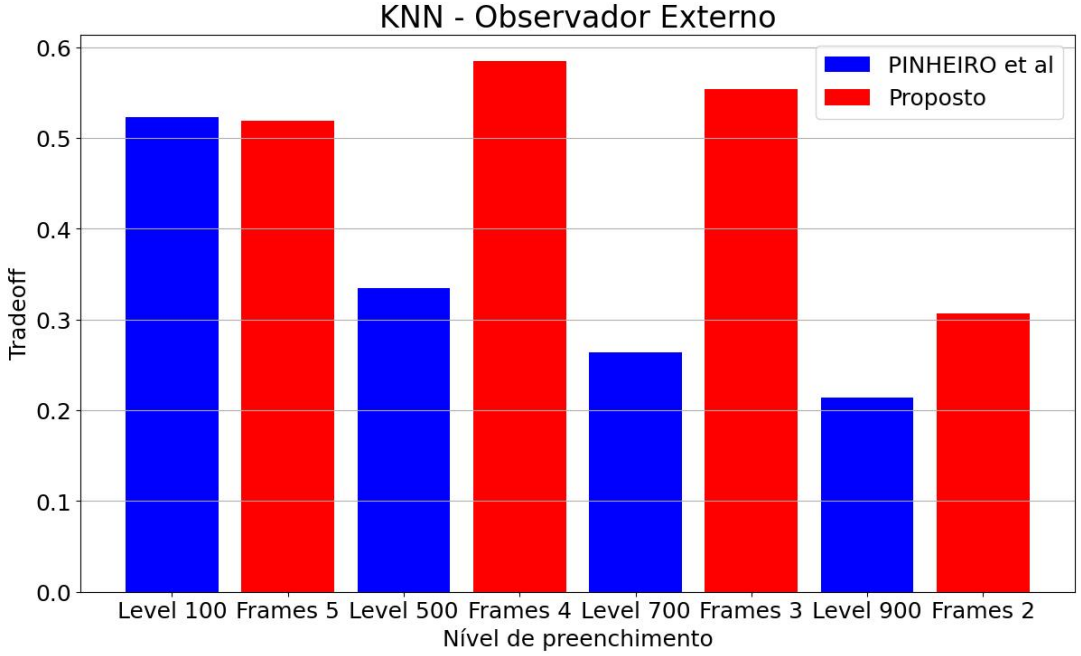
As Figuras 13, 14, 15 e 16 ilustram a métrica apresentada na Equação 5.4 aplicada no cenário de um observador externo utilizando os algoritmos KNN, DT, RF e SVM respectivamente. Quando o *byte overhead* é considerado, a abordagem de preenchimento desenvolvida apresenta melhores resultados em relação ao trabalho comparado nos algoritmos KNN, DT e RF, contudo *Level 100* apresenta um *tradeoff* maior que *Frames 5* quando o algoritmo SVM é utilizado. A Tabela 5 compara a média dos *trade-offs* obtidos pelos níveis de preenchimento analisados, onde a abordagem de preenchimento desenvolvida apresenta uma melhoria de até 55% quando o algoritmo DT é utilizado.

Tabela 5: *Trade-off* médio - Observador externo

Algoritmo	(PINHEIRO *et al.*, 2021)	Abordagem desenvolvida
KNN	0.33	0.49
DT	0.34	0.53
RF	0.33	0.50
SVM	0.33	0.38

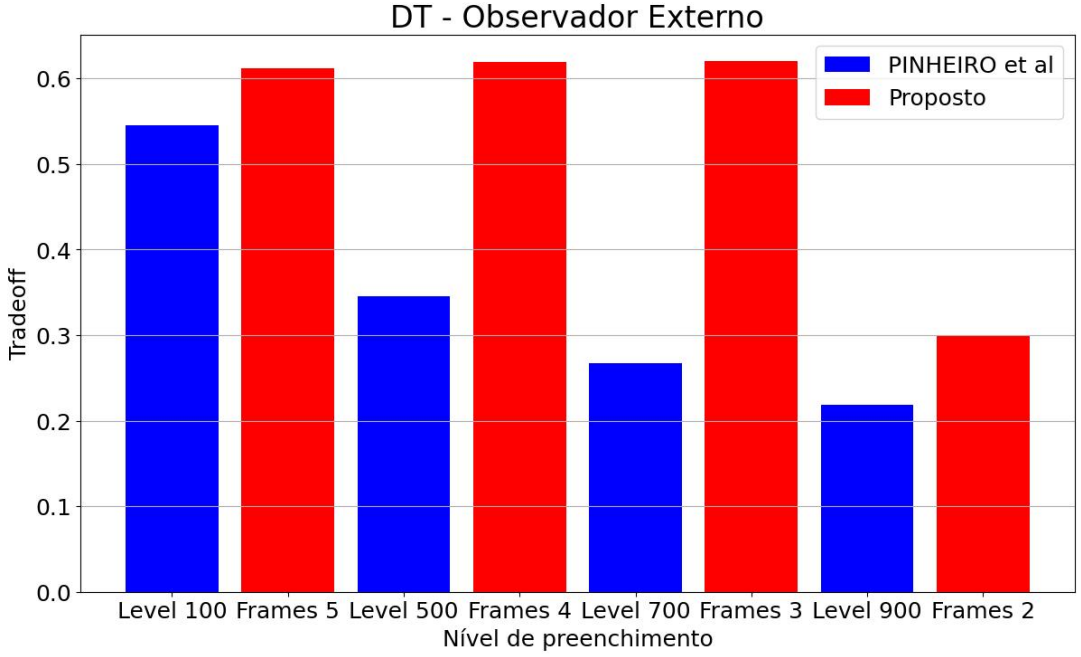
Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

Figura 13: *Trade-off KNN*



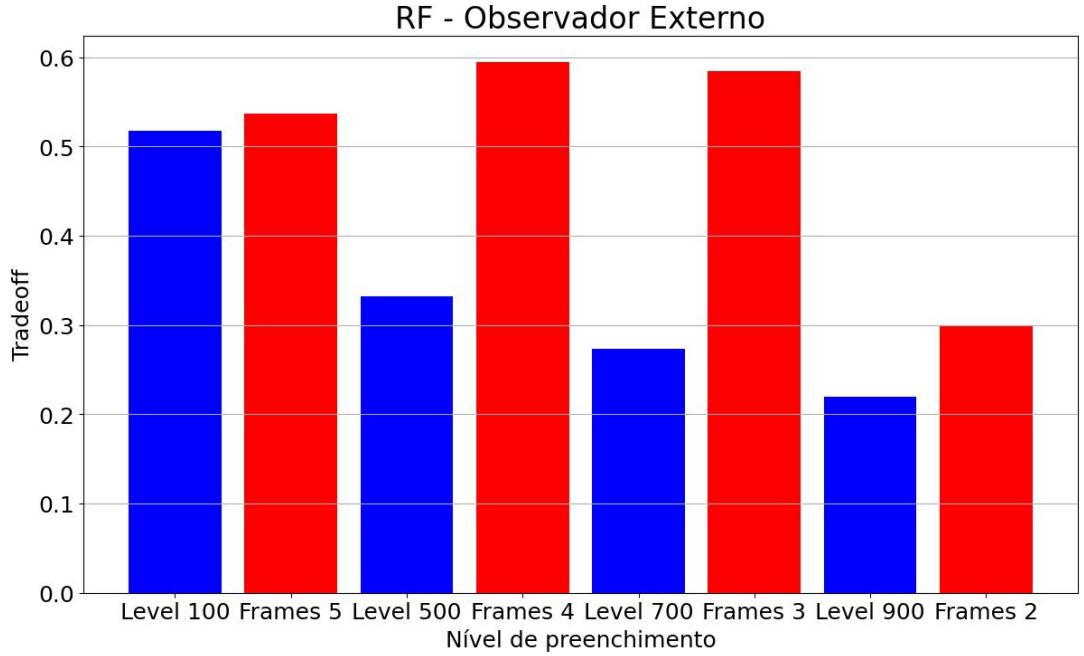
Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

Figura 14: *Trade-off DT*



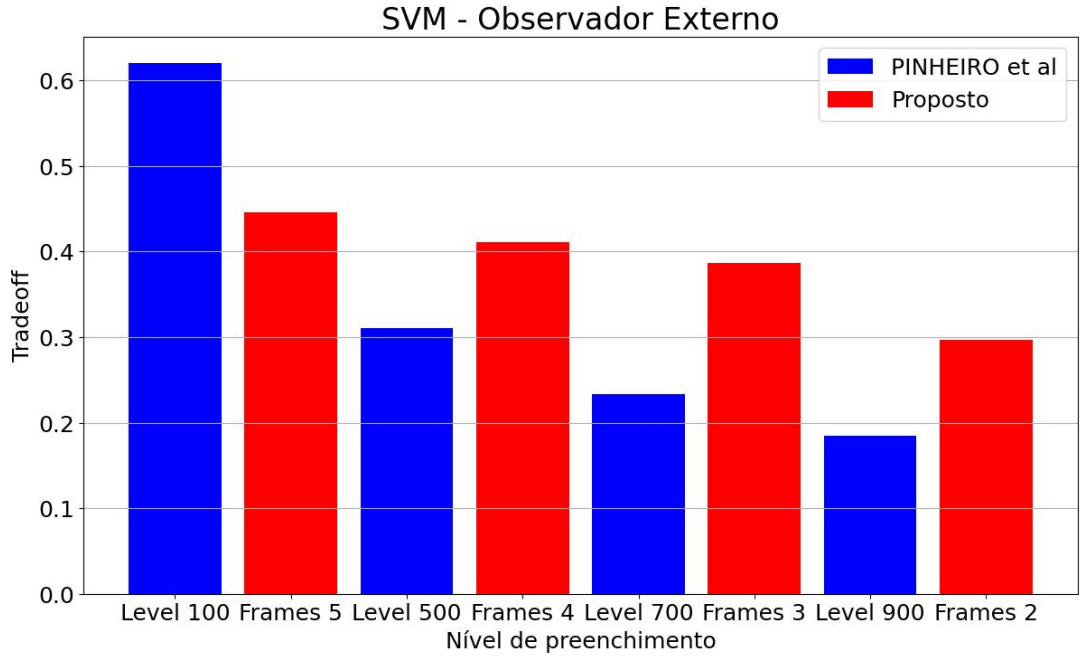
Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

Figura 15: *Trade-off RF*



Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

Figura 16: *Trade-off SVM*



Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

6.4.2 Observador Inteno

As Figuras 17, 18, 19 e 20 ilustram o *trade-off* em um cenário de observador interno nos algoritmos KNN, DT, RF e SVM respectivamente. Em geral, a abordagem de preenchimento apresenta um melhor *trade-off*, onde o nível de preenchimento *frames 2* apresentou o menor valor em comparação com os níveis *frames 5*, *frames 4* e *frames 3*. A média dos *trade-offs*

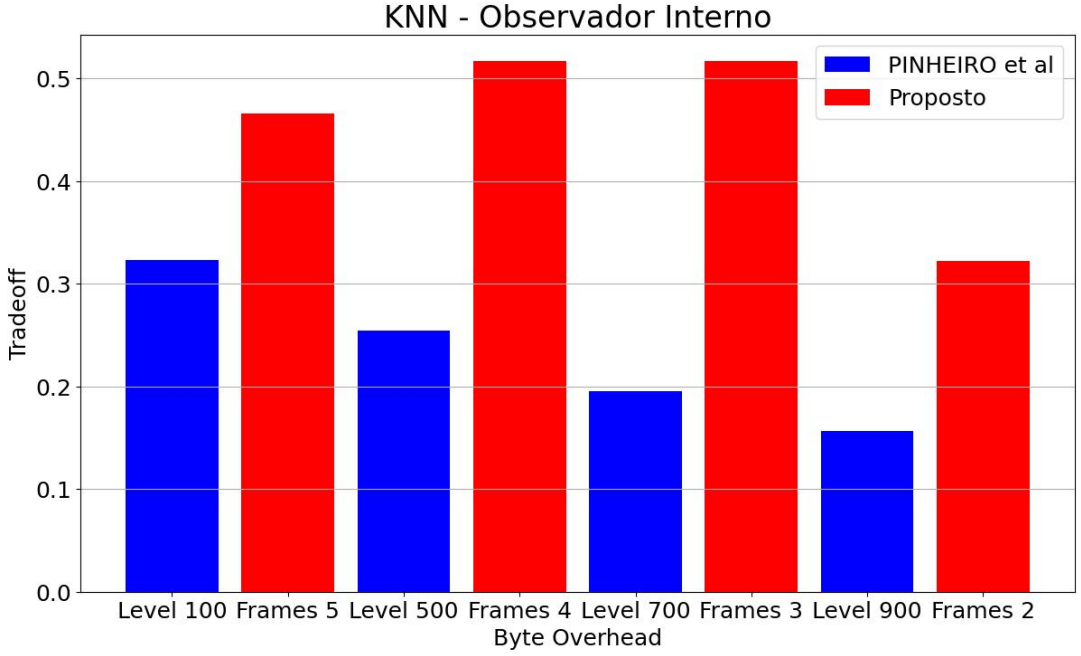
de cada nível de preenchimento é calculada e apresentada na Tabela 6, onde a abordagem de preenchimento desenvolvida obteve melhores resultados quando o algoritmo KNN é utilizado, resultando em uma melhoria de até 90% em relação ao trabalho comparado.

Tabela 6: Média dos *tradeoffs* dos níveis de preenchimento - Observador Interno.

Algoritmo	(PINHEIRO *et al.*, 2021)	Abordagem desenvolvida
KNN	0.23	0.45
DT	0.17	0.25
RF	0.17	0.25
SVM	0.18	0.30

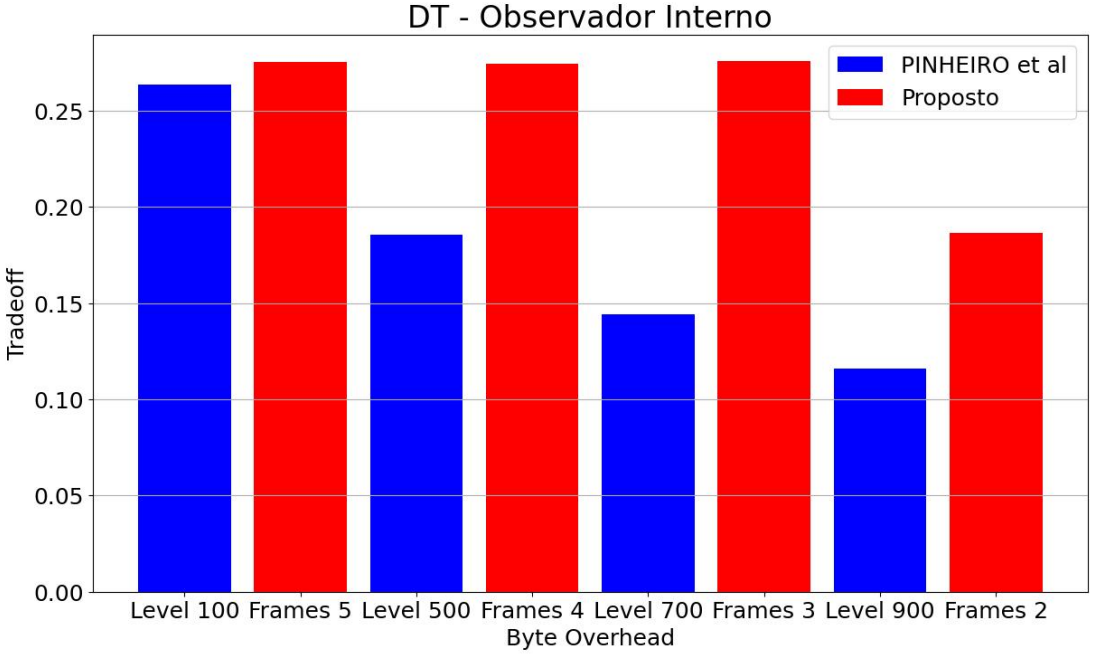
Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

Figura 17: *Trade-off KNN*



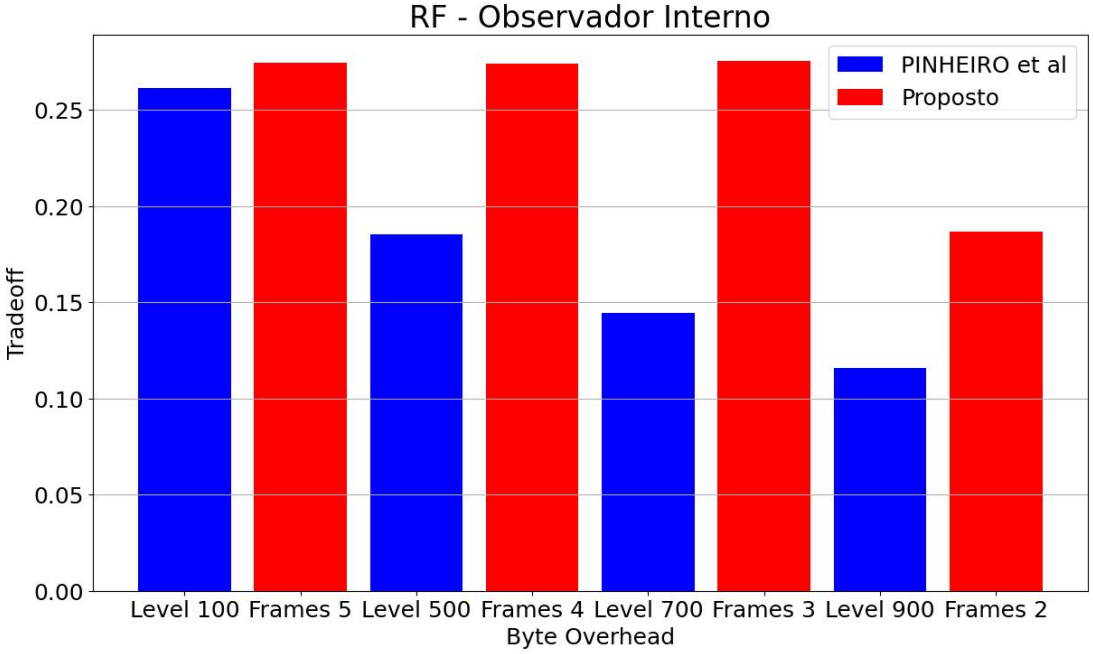
Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

Figura 18: *Trade-off DT*



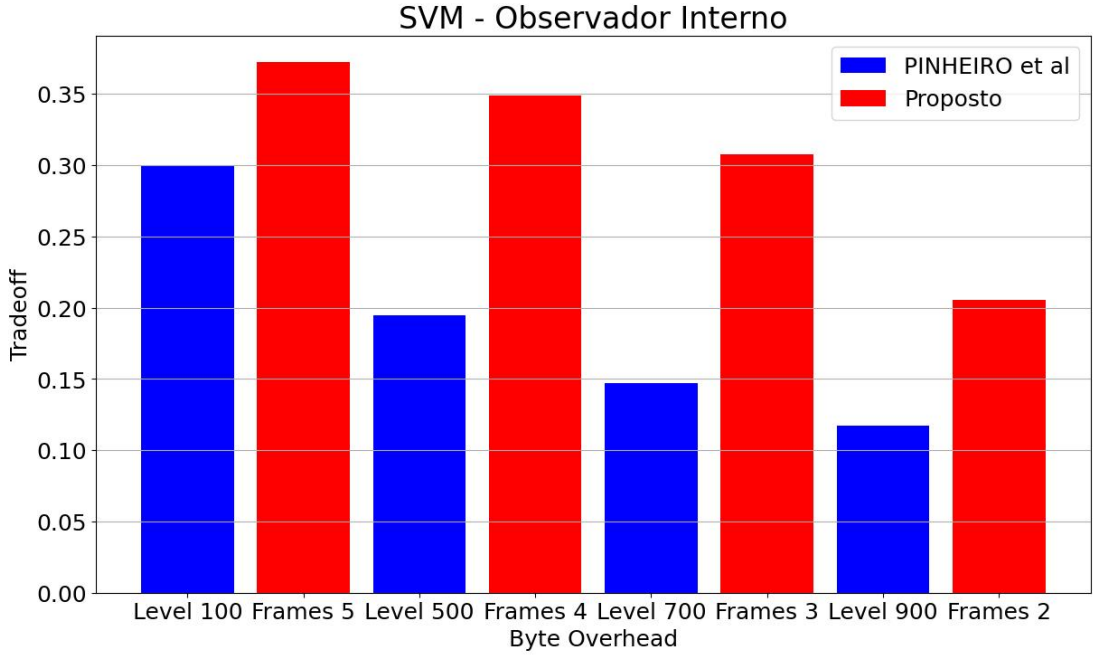
Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

Figura 19: *Trade-off RF*



Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

Figura 20: *Trade-off SVM*



Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

6.5 Considerações Finais

Os resultados apresentados demonstram que a abordagem desenvolvida conseguiu reduzir o desempenho de métodos de aprendizado de máquina utilizados para identificar os dispositivos presentes na rede. Em comparação com os níveis de preenchimento propostos pelo projeto, *Frames 3* apresentou a melhor eficácia em reduzir a acurácia dos algoritmos de classificação utilizados no cenário de um observador externo, já no cenário de um observador interno, *Frames 2* apresentou os melhores resultados. Além disso, a heurística de preenchimento utilizada reduziu o *byte overhead* em relação à abordagem de preenchimento utilizada para comparar os resultados. As Tabelas 7 e 8 resumem os resultados obtidos por um observador interno e externo quando o preenchimento é aplicado.

Ademais, quando a métrica apresentada na Seção 6.4 é utilizada para qauntificar o *trade-off* entre privacidade e desempenho, a abordagem de preenchimento desenvolvida apresenta melhores resultados em relação ao trabalho comparado. No cenário de um observador externo, a abordagem de preenchimento apresentou uma melhoria de 55% quando o algoritmo DT é utilizado, enquanto, no cenário de um observador interno, foi possível obter uma melhoria de até 90% quando o algoritmo KNN foi utilizado. Dessa forma, os resultados indicam que a heurística utilizada equilibra a troca entre desempenho e a privacidade fornecida ao usuário.

Tabela 7: Resultados obtidos no cenário de um observador externo

Nível de Preenchimento	Algoritmo	Acurácia	Recall	F1-score	*Trade-off*
Frames 5	KNN	0.30	0.30	0.30	0.51
	DT	0.25	0.25	0.25	0.61
	RF	0.27	0.27	0.27	0.53
	SVM	0.49	0.49	0.49	0.44
Frames 4	KNN	0.23	0.23	0.23	0.58
	DT	0.16	0.16	0.16	0.61
	RF	0.19	0.19	0.19	0.59
	SVM	0.48	0.48	0.48	0.41
Frames 3	KNN	0.32	0.32	0.32	0.55
	DT	0.29	0.29	0.29	0.62
	RF	0.35	0.35	0.35	0.58
	SVM	0.38	0.38	0.38	0.38
Frames 2	KNN	0.25	0.25	0.25	0.30
	DT	0.27	0.27	0.27	0.29
	RF	0.29	0.29	0.29	0.29
	SVM	0.34	0.34	0.34	0.29

Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

Tabela 8: Resultados obtidos no cenário de um observador interno

Nível de Preenchimento	Algoritmo	Acurácia	Recall	F1-score	*Trade-off*
Frames 5	KNN	0.42	0.42	0.42	0.46
	DT	0.65	0.65	0.65	0.27
	RF	0.65	0.65	0.65	0.27
	SVM	0.53	0.53	0.53	0.37
Frames 4	KNN	0.37	0.37	0.37	0.51
	DT	0.65	0.65	0.65	0.27
	RF	0.66	0.66	0.66	0.27
	SVM	0.57	0.57	0.57	0.34
Frames 3	KNN	0.35	0.35	0.35	0.51
	DT	0.60	0.60	0.60	0.27
	RF	0.60	0.60	0.60	0.27
	SVM	0.53	0.53	0.53	0.30
Frames 2	KNN	0.17	0.17	0.17	0.32
	DT	0.50	0.50	0.50	0.18
	RF	0.50	0.50	0.50	0.18
	SVM	0.47	0.47	0.47	0.20

Fonte: Adaptado de (PINHEIRO *et al.*, 2021).

7 CONCLUSÕES E TRABALHOS FUTUROS

O projeto apresenta uma abordagem de ofuscamento de tráfego baseada em preenchimento de pacotes que utiliza uma heurística para balancear o *tradeof* entre privacidade e desempenho, dessa forma protegendo a privacidade do usuário contra invasores que utilizam análise de tráfego a fim de inferir informações sobre o usuário. Os resultados demonstram que o algoritmo de preenchimento utilizado apresenta uma redução no *byte overhead* quando comparado com outros métodos de preenchimento similares. A abordagem de preenchimento proposta foi capaz de reduzir a acurácia de modelos de Aprendizado de Máquinas utilizados para detectar os dispositivos baseados apenas na quantidade de *bytes* contidos nos pacotes, onde, após o preenchimento ser aplicado, as taxas de acerto obtidas por esses modelos são abaixo de 50% quando os modelos são treinados com o tráfego original e abaixo de 70% quando são treinados com o tráfego já preenchido.

A solução apresentada depende da frequência relativa dos tamanhos dos pacotes enviados na rede para encontrar o tamanho do preenchimento que reduza o tamanho *overhead*. Esses valores são obtidos a partir de capturas feitas em um período de 3 semanas, dessa forma os resultados obtidos são otimizados para as capturas utilizadas para aplicar a heurística de preenchimento, logo a solução apresenta um caráter estático. Contudo, as redes IoT apresentam grande dinamicidade, logo as frequências de envio podem ser alteradas conforme novos dispositivos se conectam na rede, recebem atualizações, entre outros fatores. Nesse contexto, a abordagem apresentada pode não apresentar resultados otimizados em um cenário real.

Em trabalhos futuros, esse problema pode ser resolvido alterando a abordagem para reaplicar a heurística quando a frequência de envio é alterada. Dessa forma, o preenchimento aplicado nos pacotes é alterado à medida que as frequências relativas de envio são atualizadas. Por fim, espera-se que abordagem apresentada contribua no desenvolvimento de projetos no contexto de ofuscamento de tráfego que utilizam preenchimento de pacotes ou que alterem outras características dos envios realizados pelos dispositivos inteligentes.

REFERÊNCIAS

ALLOGHANI, M. *et al.* A systematic review on supervised and unsupervised machine learning algorithms for data science. **Supervised and unsupervised learning for data science**, Springer, p. 3–21, 2020.

AMMAR, N.; NOIRIE, L.; TIXEUIL, S. Autonomous identification of iot device types based on a supervised classification. In: **ICC 2020 - 2020 IEEE International Conference on Communications (ICC)**. [S.l.: s.n.], 2020. p. 1–6.

BAI, L. *et al.* A low cost indoor positioning system using bluetooth low energy. **IEEE Access**, v. 8, p. 136858–136871, 2020. ISSN 2169-3536.

CHEN, B. *et al.* A survey on smart home privacy data protection technology. In: **2021 IEEE Sixth International Conference on Data Science in Cyberspace (DSC)**. [S.l.: s.n.], 2021. p. 583–590.

HAFEEZ *et al.* Protecting iot-environments against traffic analysis attacks with traffic morphing. In: IEEE. **2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)**. [S.l.], 2019. p. 196–201.

ILIADIS, L. A.; KAIFAS, T. Darknet traffic classification using machine learning techniques. In: **2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST)**. [S.l.: s.n.], 2021. p. 1–4.

LEE, M. *et al.* Security threat on wearable services: Empirical study using a commercial smartband. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2017. ISBN 9781509027439.

LIU, H.; LANG, B. Machine learning and deep learning methods for intrusion detection systems: A survey. **Applied Sciences**, v. 9, n. 20, 2019. ISSN 2076-3417. Disponível em: https://www.mdpi.com/2076-3417/9/20/4396. Acesso em: 1 maio. 2022.

MENG, T. *et al.* A survey on machine learning for data fusion. **Information Fusion**, v. 57, p. 115–129, 2020. ISSN 1566-2535. Disponível em: https://www.sciencedirect.com/science/article/pii/S1566253519303902. Acesso em: 15 set. 2022.

MOHIT; ANSARI, S.; KUMAR, A. Traffic privacy study on internet of things - smart home applications. In: **2021 9th International Conference on Cyber and IT Service Management (CITSM)**. [S.l.: s.n.], 2021. p. 1–6.

MSHALI, H. *et al.* A survey on health monitoring systems for health smart homes. **International Journal of Industrial Ergonomics**, v. 66, p. 26–56, 2018. ISSN 0169-8141. Disponível em: https://www.sciencedirect.com/science/article/pii/S0169814117300082. Acesso em: 1 out. 2022.

PERERA, Y. *et al.* Iot traffic obfuscation: Will it guarantee the privacy of your smart home? In: **ICC 2022 - IEEE International Conference on Communications**. [S.l.: s.n.], 2022. p. 2954–2959. ISSN 1938-1883.

PINHEIRO, A. J. *et al.* Adaptive packet padding approach for smart home networks: A tradeoff between privacy and performance. **IEEE Internet of Things Journal**, Institute of Electrical and Electronics Engineers Inc., v. 8, p. 3930–3938, 3 2021. ISSN 23274662.

PINHEIRO, A. J. *et al.* Identifying iot devices and events based on packet length from encrypted traffic. **Computer Communications**, Elsevier B.V., v. 144, p. 8–17, 8 2019. ISSN 1873703X.

REZAEI, S.; LIU, X. Deep learning for encrypted traffic classification: An overview. **IEEE Communications Magazine**, v. 57, n. 5, p. 76–81, May 2019. ISSN 1558-1896.

SANTOS, B. *et al.* Um método de ofuscação para proteger a privacidade no tráfego da rede iot. In: **Anais do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**. Porto Alegre, RS, Brasil: SBC, 2022. p. 126–139. ISSN 2177-9384. Disponível em: https://sol.sbc.org.br/index.php/sbrc/article/view/21166. Acesso em: 15 out. 2022.

SANTOS, M. R. P. *et al.* An efficient approach for device identification and traffic classification in iot ecosystems. In: **2018 IEEE Symposium on Computers and Communications (ISCC)**. [S.l.: s.n.], 2018. p. 00304–00309.

SARAVANAN, R.; SUJATHA, P. A state of art techniques on machine learning algorithms: A perspective of supervised learning approaches in data classification. In: **2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)**. [S.l.: s.n.], 2018. p. 945–949.

SHEN, F. *et al.* A survey of traffic obfuscation technology for smart home. In: **2022 International Wireless Communications and Mobile Computing (IWCMC)**. [S.l.: s.n.], 2022. p. 997–1002. ISSN 2376-6506.

SINGH, A.; THAKUR, N.; SHARMA, A. A review of supervised machine learning algorithms. In: **2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)**. [S.l.: s.n.], 2016. p. 1310–1315.

SIVANATHAN, A. *et al.* Classifying iot devices in smart environments using network traffic characteristics. **IEEE Transactions on Mobile Computing**, v. 18, n. 8, p. 1745–1759, 2019.

SKOWRON, M.; JANICKI, A.; MAZURCZYK, W. Traffic fingerprinting attacks on internet of things using machine learning. **IEEE Access**, Institute of Electrical and Electronics Engineers Inc., v. 8, p. 20386–20400, 2020. ISSN 21693536.

TAHAEI, H. *et al.* The rise of traffic classification in iot networks: A survey. **Journal of Network and Computer Applications**, v. 154, p. 102538, 2020. ISSN 1084-8045. Disponível em: https://www.sciencedirect.com/science/article/pii/S1084804520300126. Acesso em: 1 nov. 2023.

TSANTIKIDOU, K.; SKLAVOS, N. Vulnerabilities of internet of things, for healthcare devices and applications. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2021. p. 498–503. ISBN 9781665410014.

VALE, E. R.; BRANDAO, J. C. B.; GRIVET, M. An algorithm for umts padding optimization. In: **2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications**. [S.l.: s.n.], 2006. p. 1–5. ISSN 2166-9589.

XIN, Y. *et al.* Machine learning and deep learning methods for cybersecurity. **IEEE Access**, v. 6, p. 35365–35381, 2018. ISSN 2169-3536.

XU, Y.; GOODACRE, R. On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. **Journal of Analysis and Testing**, v. 2, n. 3, p. 249–262, Jul 2018. ISSN 2509-4696. Disponível em: https://doi.org/10.1007/s41664-018-0068-2. Acesso em: 7 nov. 2022.