Impacto da criptografia da camada de transporte em análises de fluxos com aprendizado de máquina

Resumo. *Com o aumento no uso da criptografia nas comunicações de redes de computadores, é possível que, no futuro, a criptografia dos protocolos da camada de transporte se torne algo comum, o que pode dificultar ou tornar menos eficiente a análise automatizada de fluxos de rede. Este artigo propõe e implementa uma análise sobre o impacto dessa possível criptografia em análises de fluxos com aprendizado de máquina. Os resultados mostraram que a criptografia dessa camada da rede poderia afetar a análise de fluxos de rede. Os conceitos de explicabilidade e interpretabilidade foram utilizados para avaliar a qualidade dos resultados.*

Abstract. *With the increase in the use of cryptography in computer network communications, it is possible that, in the future, encryption of transport layer protocols become commonplace, which may hinder or make automated network flow analysis less efficient. This paper proposes and implements an analysis on the impact of this possible encryption on flow analysis with machine learning. The results showed that the encryption of this network layer could affect the analysis of network flows. The concepts of explainability and interpretability were used to evaluate the quality of the results.*

1. Introdução

Existem diversas formas de análise de tráfego de rede, como a inspeção profunda de pacotes (*Deep Packet Inspection*), que analisa o conteúdo detalhado dos pacotes capturados, e a análise de tráfego de rede baseada em fluxos, que analisa o tráfego de rede a partir de fluxos de tráfego extraídos da captura de pacotes. A DPI é mais eficiente que a análise baseada em fluxos, porém, pode ser computacionalmente mais custosa. A análise baseada em fluxos, por sua vez, tende a ser mais eficiente em termos de custo computacional, porém, pode ser menos eficiente em termos de detecção de anomalias e ataques cibernéticos [Boukhtouta et al. 2016]. A análise de tráfego de rede é uma das principais técnicas utilizadas para detectar atividades maliciosas, como malware, ataques de negação de serviço (DDoS), *phishing* e roubo de dados [Joshi and Hadi 2015].

A captura de pacotes (pcap), tradicionalmente utilizada no contexto da biblioteca *libpcap*, é uma das principais formas de análise passiva de redes, sendo extensivamente utilizada por administradores de rede e por sistemas de detecção de intrusão (*Intrusion Detection System*) [Alias et al. 2013]. A partir da captura de pacotes, é possível extrair informações sobre o tráfego analisado, como o número de *bytes* transferidos, o número de pacotes enviados, o tempo de resposta, entre outros. Essa extração de informações é feita através dos cabeçalhos dos pacotes e do agrupamento dos mesmos em fluxos de tráfego, que são agrupados de acordo com alguma característica, como o endereço IP de origem e destino.

A detecção de anomalias é uma técnica de análise de dados que busca identificar padrões incomuns ou desvios em relação a um comportamento esperado. Em redes de computadores, a detecção de anomalias pode ser utilizada para identificar atividades suspeitas que possam indicar a presença de um ataque cibernético em andamento. Existem diversas técnicas de detecção de anomalias, como *clustering*, redes neurais e árvores de decisão [Usama et al. 2019]. A detecção de anomalias é frequentemente utilizada em conjunto com técnicas de criptografia de dados e aprendizado de máquina para identificar e mitigar ameaças de segurança em tempo real [Bhuyan et al. 2013].

A criptografia de dados é uma técnica essencial para garantir a privacidade e segurança das informações que trafegam em redes de comunicação. Porém, a implementação desse tipo de técnica pode interferir na análise de tráfego de rede, que é crucial para a detecção de anomalias e ataques cibernéticos. Recentemente, cada vez mais protocolos da camada de aplicação têm sido criptografados, como a migração do protocolo HTTP para HTTPS, ou a implementação de criptografia no protocolo FTP. Porém, a criptografia de dados em protocolos da camada de transporte, como o protocolo TCP, é ainda pouco explorada, apesar de já possuir propostas de extensão, como visto em [Bittau et al. 2010], [Bittau et al. 2019b] e [Bittau et al. 2019a], que propõem a aplicação de criptografia oportunística.

Apesar da criptografia de dados na camada de aplicação, existem diversas metodologias para, mesmo assim, realizar a análise automatizada de fluxos de tráfego e derivar informações relevantes sobre o tráfego analisado, como qual protocolo está sendo utilizado ou qual aplicação é responsável pelo tráfego gerado [Karagiannis et al. 2005]. Os autores do *framework* BLINC, por exemplo, afirmam que se os cabeçalhos da camada de transporte fossem criptografados, o tráfego de rede não poderia ser analisado [Boutaba et al. 2018].

Esse artigo associa as perspectivas mencionadas para entender como a criptografia da camada de transporte nas comunicações de rede poderia afetar a análise automatizada de fluxos de tráfego e a detecção de anomalias a partir dos mesmos. Para isso, foram comparados os resultados obtidos de um modelo de aprendizado supervisionado treinado com fluxos de tráfego de rede não criptografados e de outro treinado com os mesmos fluxos de tráfego de rede, porém, sem as estatísticas obtidas a partir dos cabeçalhos da camada de transporte. Foi utilizado o *framework* SHAP [Lundberg and Lee 2017] para apresentar os resultados obtidos de forma mais explicativa.

O artigo está estruturado da seguinte forma. Na Seção 2 são apresentados alguns trabalhos relacionados, na Seção 3 é proposta uma solução para realizar a investigação, que tem sua implementação e seus resultados apresentados na Seção 4. Como conclusão, a Seção 5 expõe as considerações finais do estudo desenvolvido, além de proposições para possíveis melhorias para essa pesquisa e trabalhos futuros.

2. Trabalhos relacionados

Utilizando diversos *datasets*, incluindo o CIC-IDS2017 [Sharafaldin et al. 2018], [Gupta et al. 2022] apresentam algoritmos de *deep learning ensemble* que lidam com o desbalanceamento de classes para o uso em NIDS. Os autores utilizam um sistema multicamadas, com *deep neural networks* (DNN) na primeira, XGBoost na segunda e *random forest* na terceira camada. No CIC-IDS2017, o CSE-IDS proposto pelos autores apresenta

acurácia de 92% e taxa de detecção de ataques (*Attack Detection Rate*) de 98%.

[Al-Essa and Appice 2022] utilizam o XGBoost com os *datasets* NSL-KDD [Tavallaee et al. 2009] e CIC-IDS2017 para avaliar quais *features* dos conjuntos de dados são mais relevantes para a classificação de tráfego, através das metodologias de *feature selection* e *oversampling*. Os autores relatam que o uso de estratégias de *OneVsRest (OVR)* com o XGBoost apresenta melhores resultados, com acurácia acima de 99% e *f1 score* de 87% no CIC-IDS2017.

Em [Le et al. 2022], é feita uma análise sobre como as *features* dos fluxos de rede afetam o desempenho de algoritmos de aprendizado de máquina para detecção de anomalias, através da explicabilidade. Os autores utilizam os *datasets* IotIDS20 [Ullah and Mahmoud 2020] e NetFlow IoT V2 [Sarhan et al. 2022], que contém fluxos de rede de dispositivos IoT. Os autores utilizam a técnica de *ensemble* com algoritmos de DT e RF e, em seguida, utilizam explicações locais e explicações globais do SHAP para analisar a importância das *features* para a classificação.

3. Análise proposta

A eficiência de sistemas de detecção de intrusão e da segurança de redes dependem da velocidade em que ataques e ameaças são detectados. Por isso, conjuntos de tráfego de rede são extremamente necessários para o melhor desempenho das ferramentas utilizadas [Dijkhuizen and Ham 2018]. Dentre os conjuntos de dados publicamente disponíveis, foi selecionado o CIC-IDS2017 [Sharafaldin et al. 2018] pois, além da captura de pacotes, o *dataset* também possui o fluxo de tráfego da rede, que é o que será utilizado nessa análise. Os fluxos de rede anômalos foram gerados a partir de um conjunto de ataques de intrusão, como *Brute Force*, *Heartbleed*, *Botnet*, *Denial of Service (DoS)*, *Distributed Denial of Service (DDoS)*, *PortScan* e um conjunto de ataques *web*, como *SQL Injection* e *Cross-site scripting (XSS)*. [Rosay et al. 2022] discutem em mais detalhes as *features* e os ataques disponibilizados no *dataset* CIC-IDS2017.

Os fluxos de rede do CIC-IDS2017 possuem informações detalhadas sobre a direção dos pacotes enviados. Portanto, algumas *features* do *dataset* possuem os prefixos *bwd* e *fwd*, que indicam, respectivamente, pacotes que foram enviados de um cliente para um servidor (uma requisição) ou do servidor para o cliente, a resposta para uma requisição. Por exemplo, a *feature fwd_pkt_len_max* indica o tamanho máximo dos pacotes enviados do cliente para o servidor, enquanto a *feature bwd_pkt_len_max* indica o tamanho máximo dos pacotes enviados do servidor para o cliente.

3.1. Tratamento dos dados

Apesar de ser um conjunto de dados bastante completo, o CIC-IDS2017 possui alguns problemas, como dados inválidos ou duplicados e incoerência nos rótulos (*labelling*) de alguns fluxos. Para corrigir esses problemas, [Rosay et al. 2021] criaram o *dataset* LYCOS2017, que é uma versão corrigida do CIC-IDS2017. Dentre as correções feitas, estão a remoção de *features* duplicadas, a correção nos cálculos de algumas *features*, a contagem dos tamanhos dos pacotes e o uso das *flags* TCP.

Após a correção dos dados, o *dataset* LYCOS2017 possui quatorze classificações possíveis para os fluxos, que são: *benign*, *bot*, *ddos*, *dos_goldeneye*, *dos_hulk*,

dos_slowhttptest, *dos_slowloris*, *ftp_patator*, *heartbleed*, *portscan*, *ssh_patator*, *webattack_bruteforce*, *webattack_sql_injection* e *webattack_xss*. Cada fluxo de rede possui 84 atributos, sendo que 83 deles são numéricos. O atributo restante é a classificação do fluxo, que foi utilizado para treinar o modelo de aprendizado de máquina de forma supervisionada nessa análise.

3.2. Criptografia da camada de transporte

Os dados da camada de transporte são encontrados nos campos *source port*, *destination port*, *sequence number*, *acknowledgement number*, *flags*, *window size*, *urgent pointer* e *options*. Os dados relativos a esses campos serão ofuscados através da remoção das estatísticas de fluxo geradas a partir deles.

Para simular a criptografia dos dados relativos à camada de transporte, foi feita uma análise das *features* estatísticas que foram obtidas através desses dados. Os dados e estatísticas removidas do *dataset* original são:

- *src_port* e *dst_port*: os números de portas de origem e destino dos fluxos de rede. Esses números são utilizados para identificar os processos que estão enviando e recebendo os dados;
- *fwd_tcp_init_win_bytes* e *bwd_tcp_init_win_bytes*: o tamanho da janela TCP inicial do fluxo de rede.
- *flag_rst*, *flag_urg*, *flag_ack*, *flag_psh*, *flag_syn*, *flag_fin*, *flag_cwr* e *flag_ece*: a quantidade de pacotes que possuem cada uma das *flags* de controle de fluxo do TCP, independente da direção do fluxo de comunicação.
- *fwd_flag_urg* e *bwd_flag_urg*: a quantidade de pacotes que possuem a *flag* de urgência do TCP, na direção do fluxo de comunicação.
- *fwd_flag_psh* e *bwd_flag_psh*: a quantidade de pacotes que possuem a *flag* de *push* do TCP, na direção do fluxo de comunicação.

Dessa forma, os conjuntos de dados gerados a partir da remoção das estatísticas de fluxo possuem 76 atributos, incluindo o rótulo de classificação.

Embora os dados relativos à camada de transporte sejam criptografados, camadas inferiores do modelo OSI também possuem informações sobre o tamanho do pacote, portanto, não é possível remover estatísticas como a quantidade de *bytes* e de pacotes em um fluxo de rede. Vale ressaltar que, até a versão responsável por gerar o CIC-IDS2017, o CICFlowMeter não gerava estatísticas com os dados de camadas superiores à de transporte, como a camada de aplicação, por exemplo.

3.3. Definição do modelo

Para definir qual o melhor algoritmo e quais os melhores parâmetros para ele, foi utilizado o *framework* H2O [H2O.ai 2023]. Executando a ferramenta *AutoML* do H2O, foi obtida uma tabela com os *scores* de cada algoritmo testado. O modelo a utilizado foi o que obteve o melhor *score* na métrica *mean_per_class_error* (erro médio por classe), que é a métrica utilizada para comparar o desempenho de modelos de classificação multinomial [H2O.ai 2023].

Após obter o melhor algoritmo e parâmetros para os conjuntos de dados de treinamento e validação, foi possível obter os *scores* de cada modelo a partir dos dados de teste, que foram utilizados para comparação com os resultados obtidos pelos autores do *dataset* LYCOS2017.

3.4. Comparação de resultados

Para fins de comparação, tanto os autores do *dataset* LYCOS2017 quanto este trabalho usam as matrizes de confusão global dos modelos gerados. A partir dessas matrizes, diversas métricas estatísticas podem ser obtidas. Dentre elas, estão a taxa de verdadeiros positivos (TPR), a taxa de falsos positivos (FPR), a taxa de verdadeiros negativos (TNR), a taxa de falsos negativos (FNR), a acurácia (*accuracy*), a precisão (*precision*), a revocação (*recall*), a medida F1 (*f1 score*) e o coeficiente de correlação de Matthews (MCC) [H2O.ai 2023].

Através da comparação dos *scores* dos modelos criados nesse trabalho, é possível verificar se a criptografia dos dados da camada de transporte afeta, de forma significativa, o desempenho do modelo na detecção de atividades anômalas de rede.

4. Implementação e avaliação

Os conjuntos de dados do LYCOS2017 já haviam sido separados previamente em conjuntos de treinamento, validação e teste por [Rosay et al. 2021], que os separou em porções de 50%, 25% e 25%, respectivamente. Após o tratamento e criação dos dados criptografados, conforme descrito nas seções 3.1 e 3.2, foi utilizado o *framework* H2O para identificar os melhores modelos de aprendizado de máquina para o *dataset* analisado, assim como seus respectivos *scores*. Ordenando o desempenho pela métrica *mean_per_class_error*, o modelo de XGBoost obteve o melhor desempenho, seguido do modelo de GBM, ambos são modelos de *ensemble* de árvores de decisão. Nessa análise, o modelo de XGBoost será considerado o principal modelo a ser utilizado, enquanto que o modelo de GBM será utilizado apenas para efeitos de comparação.

Os parâmetros usados com o *AutoML* do H2O são apresentados abaixo:

- *max_runtime_secs=600*: limita a execução do *AutoML* a 10 minutos.
- *max_models=3*: limita o número de modelos a serem treinados para focar em desenvolver modelos melhores.
- *sort_metric=mean_per_class_error*: ordena os modelos de acordo com a métrica de avaliação.
- *seed=6898*: define uma semente para a geração de números aleatórios para garantir a reprodutibilidade.

Os demais parâmetros foram definidos com os valores padrão do *AutoML*. Maiores detalhes sobre os parâmetros do *AutoML* podem ser encontrados na documentação oficial do H2O [H2O.ai 2023].

Apesar de obter resultados satisfatórios com o H2O, foi identificado posteriormente que o *framework* não possui a ferramenta de explicabilidade SHAP para classificação multinomial. Por esse motivo, os modelos de XGBoost foram traduzidos para o *framework* scikit-learn, que possui maior suporte para a ferramenta SHAP. Após a conversão, o desempenho dos modelos foi comparado e os resultados obtidos foram equivalentes.

4.1. Resultados obtidos

Ao aplicar os modelos de aprendizado de máquina nos seus respectivos *datasets* de teste, foi possível obter as matrizes de confusão resumidas, que são apresentadas na tabela 1 e na tabela 2.

É importante ressaltar que, em [Rosay et al. 2021], uma classe considerada positiva foi definida como uma classe que corresponde a um ataque ou tráfego anômalo na rede, enquanto que uma classe considerada negativa foi definida como uma classe que corresponde ao tráfego benigno na rede. Os mesmos critérios foram utilizados para definir as classes positivas e negativas para os modelos apresentados neste trabalho.

		Classe predita		
		Positivo	Negativo	Total
Classe real	Positivo	110034	13	110047
	Negativo	16	110142	110158
	Total	110050	110155	

Tabela 1. Matriz de confusão do modelo de XGBoost com os dados completos

		Classe predita		
		Positivo	Negativo	Total
Classe real	Positivo	109896	30	109926
	Negativo	80	110078	110158
	Total	109976	110108	

Tabela 2. Matriz de confusão do modelo de XGBoost com os dados criptografados

Com base nas matrizes de confusão, foi possível obter os resultados estatísticos apresentados na tabela 3.

Algoritmo	***Dataset***	***Accuracy (%)***	***Precision (%)***	***Recall (%)***	***f1 score (%)***	***False Positive Rate (%)***	***MCC***
XGBoost	Completo	99,9868	99,9855	99,9882	99,9861	0,00145	0.999737
XGBoost	Criptografado	99,9500	99,9273	99,9727	99,9500	0,00726	0.999000
GBM	Completo	99,9764	99,9764	99,9764	99,9764	0,00236	0.999528
GBM	Criptografado	99,9382	99,9127	99,9636	99,9382	0,00871	0.998764

Tabela 3. Resultados obtidos dos modelos

Com esses resultados, é possível observar que, em ambos os algoritmos, os modelos de GBM e XGBoost obtiveram resultados ligeiramente divergentes entre os *datasets* completos e criptografados. No entanto, os resultados obtidos de todos os modelos foram satisfatórios, com acurácia, precisão, *recall* e *f1 score* superiores a 99,9%, e MCC superior a 99,8%.

Utilizando esses dados e os resultados obtidos pelos autores do LYCOS2017, foi possível obter uma comparação entre os resultados obtidos pelos dois *datasets*, conforme apresentado na figura 1.

Apesar da alta performance dos modelos de GBM e XGBoost, é importante observar que o *dataset* de base utilizado por [Rosay et al. 2021] é, propositalmente, um *dataset* com um enorme desbalanceamento entre as classes, principalmente entre o tráfego benigno e o tráfego anômalo, como forma de simular uma rede de computadores real [Ring et al. 2019]. Por esse motivo, pequenas variações nas taxas de detecção de tráfego anômalo podem representar um impacto significativo no desempenho real do modelo.

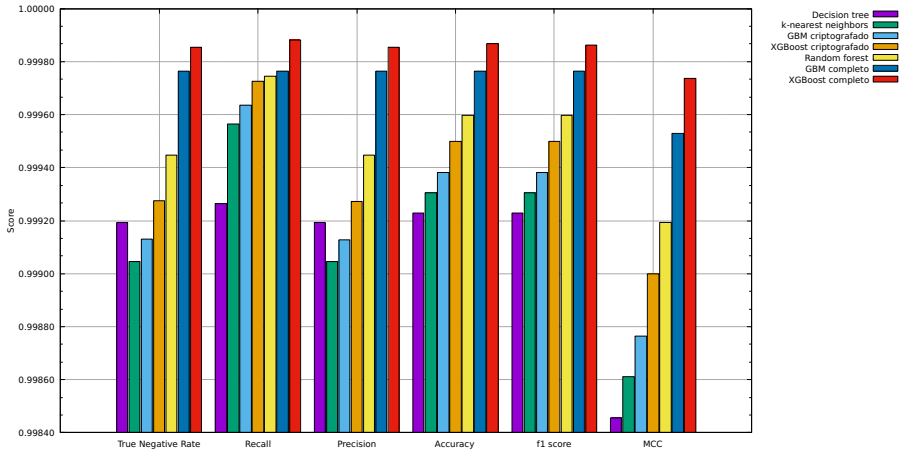

Figura 1. Comparação entre os resultados obtidos pelos modelos dos autores do LYCOS2017 e os modelos usados nesse artigo

4.2. Análise e explicação dos resultados

Através do SHAP, foi possível obter uma análise mais detalhada dos resultados dos modelos de XGBoost. As figuras 2 e 3 apresentam os gráficos de sumário SHAP, que distribuem a importância de cada *feature*, por classe, na predição do modelo.

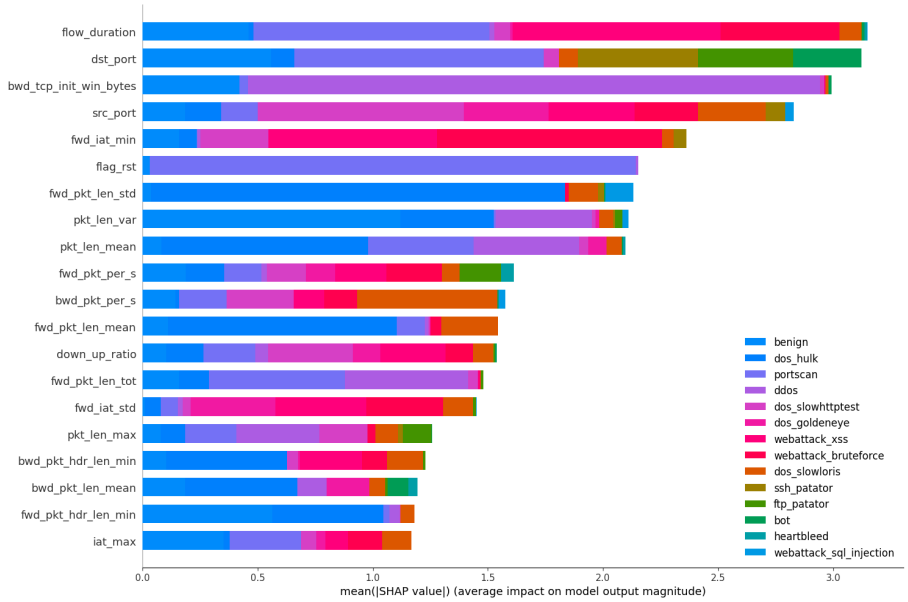

Figura 2. Importância das features por classe - dataset completo

No gráfico de sumário SHAP do modelo treinado com o *dataset* completo, é possível observar que as *features* relacionadas aos dados da camada de transporte, como porta de destino (*dst_port*) e a porta de origem (*src_port*), são bastante presentes entre as *features* mais importantes. Porém, *features* de metadados dos fluxos, como duração do fluxo e o intervalo entre pacotes são importantes para a detecção de alguns ataques, como DDoS e de ataques web.

Já no gráfico de sumário SHAP do modelo treinado com o *dataset* criptografado, podemos notar que a *feature* de duração do fluxo se mantém como a mais importante,

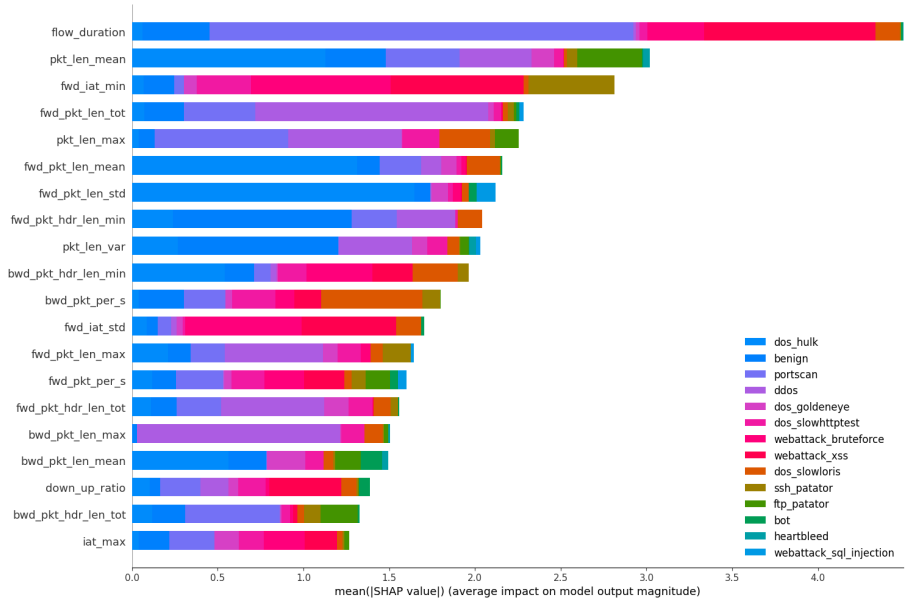

Figura 3. Importância das features por classe - dataset criptografado

enquanto que as *features* relacionadas à variação de tamanho de pacotes e cabeçalhos dos pacotes dominam na detecção de todos os tipos de tráfego. Um detalhe importante é que *features* que tinham pouca importância para o modelo com dados completos, como o tamanho mínimo dos headers dos pacotes da origem do fluxo (*fwd_pkt_hdr_len_min*) e o tamanho máximo dos pacotes no fluxo (*pkt_len_max*), passam a ser importantes para o modelo com dados criptografados. Essa mudança pode indicar uma dependência entre as *features*, que pode ser explorada para melhorar a performance dos modelos.

Nas subseções a seguir, serão apresentados os gráficos de sumário SHAP para cada classe, com as *features* mais importantes para a classificação de cada fluxo como tráfego de determinado ataque ou tráfego benigno. Nesses gráficos, as *features* são ordenadas de acordo com a importância para a classificação de cada fluxo, e as cores indicam o valor de cada *feature* para o fluxo em questão. As cores mais azuis indicam valores mais baixos, enquanto as cores mais avermelhadas indicam valores mais altos. Além disso, a posição de cada ponto no gráfico indica o valor SHAP da *feature* para aquele fluxo, sendo que valores mais altos indicam que a *feature* teve um impacto positivo na sua classificação, enquanto valores mais baixos indicam que a *feature* teve um impacto negativo na mesma classificação.

4.2.1. Detecção de tráfego benigno

Na detecção de tráfego benigno, as figuras 4 e 5 apresentam quais *features* impactam mais na classificação de um fluxo como tráfego benigno.

No *dataset* completo, podemos notar como que algumas *features* da camada de transporte impactam na classificação de um tráfego como benigno. Baixas quantidades da *flag* de sincronização (*flag_SYN*) na comunicação fornecem um *score* mais alto para a predição de tráfego benigno, assim como a presença de *src_port* mais altas. É interessante notar que a simples presença de algumas *features*, independente do seu valor, impactam

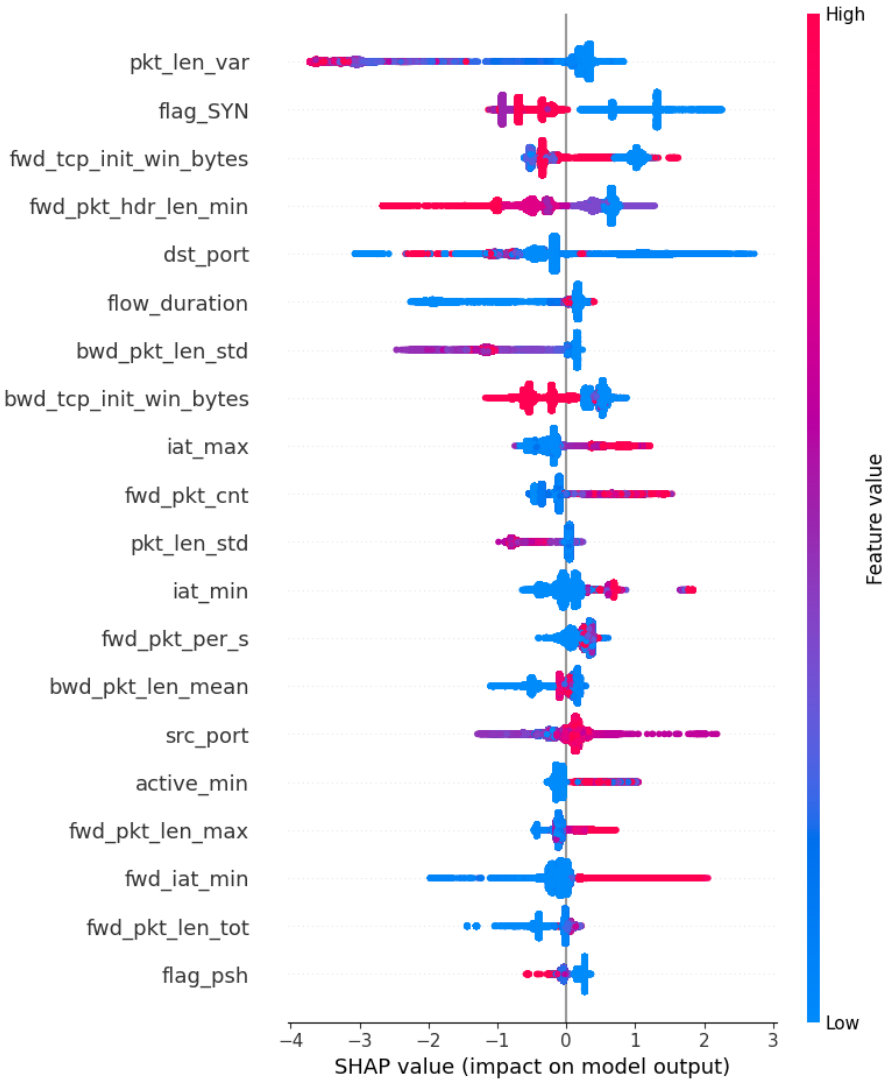

Figura 4. Impacto das features na detecção de tráfego benigno - dataset completo

positivamente na classificação de um fluxo como benigno, como a quantidade de *bytes* na janela de iniciação da comunicação TCP da origem do fluxo (*fwd_tcp_init_win_bytes*).

No *dataset* criptografado, os valores estatísticos das *features* passam a ser mais importantes para a classificação de um fluxo como benigno. Por exemplo, fluxos com *fwd_pkt_hdr_len_min* mais altas e mais baixas impactam de forma positiva e negativa, respectivamente, na classificação de um fluxo como benigno. Além disso, valores altos das *features* do tipo de protocolo IP (*ip_prot*), o intervalo mínimo entre a chegada de pacotes no destino (*fwd_iat_min*) e o *pkt_len_min* impactam positivamente.

4.2.2. Detecção de ataques DDoS

Para a detecção de DDoS, as figuras 6 e 7 apresentam quais *features* impactam mais na classificação de um fluxo como DDoS. Nesse tipo de ataque, é possível notar que o

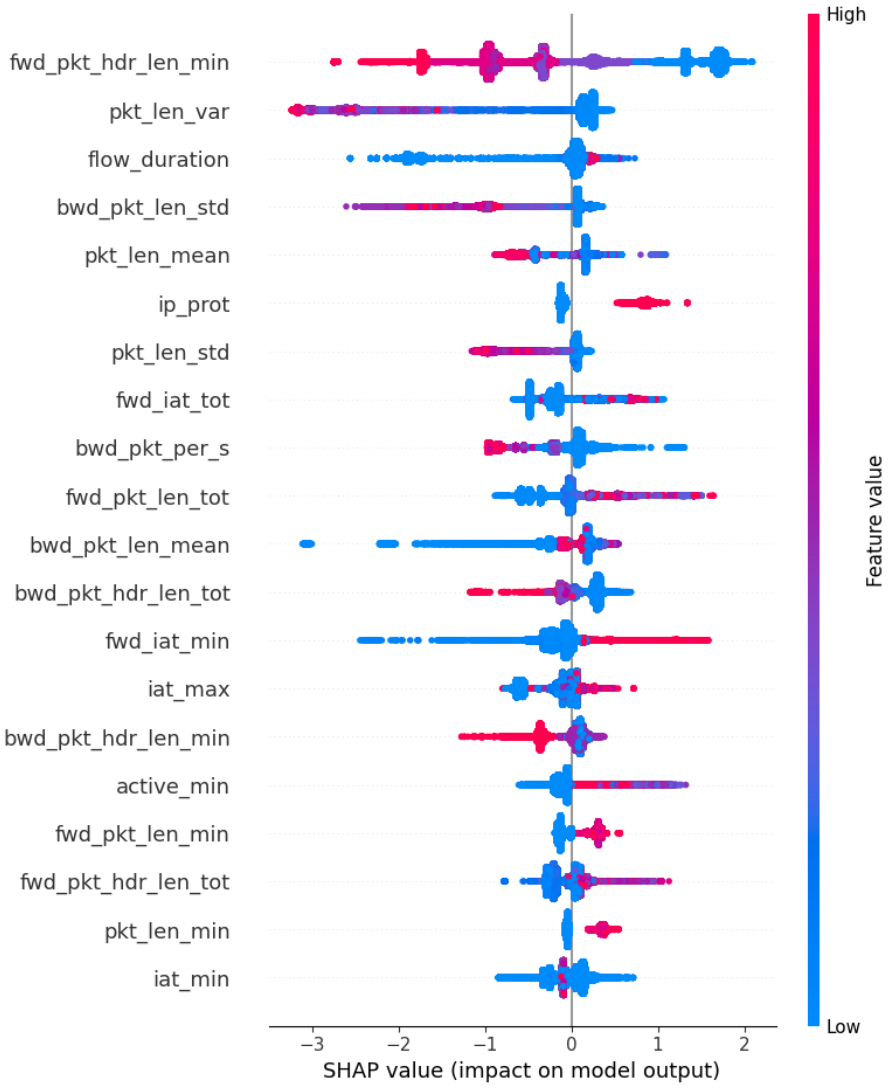

Figura 5. Impacto das features na detecção de tráfego benigno - dataset criptografado

tamanho dos pacotes é a *feature* mais importante para a classificação de um fluxo como DDoS, tanto no *dataset* completo quanto no *dataset* criptografado.

Com o *dataset* completo, notamos que fluxos com *bwd_pkt_len_max* mais altos impactam positivamente na classificação de um fluxo como DDoS. No entanto, o valores mais altos do tamanho máximo dos pacotes na direção oposta (*fwd_pkt_len_max*) impacta negativamente na classificação de um fluxo como DDoS. Uma *feature* que chama atenção é a *bwd_tcp_init_win_bytes*, que fica nos extremos dos valores SHAP, indicando que valores altos da *feature* impactam positiva e negativamente, apesar de valores baixos da *feature* impactarem somente de forma negativa. Essa variação de impacto reforça a ideia de que há uma relação de dependência entre as *features*, dependendo dos seus valores.

Sem os dados da camada de transporte, as *features* direcionais dominam na classificação de um fluxo como DDoS. Por exemplo, fluxos com *fwd_pkt_len_max* mais altos impactam negativamente na classificação de um fluxo como DDoS, enquanto que

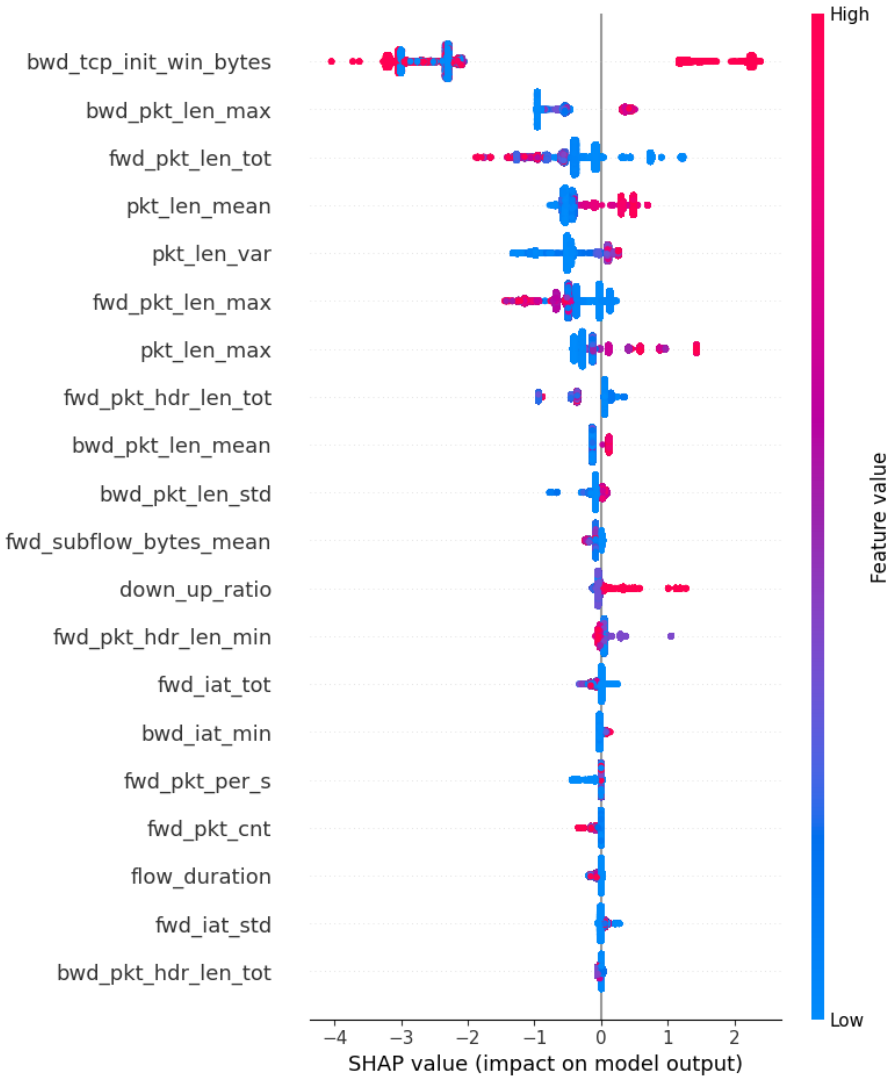

Figura 6. Impacto das features na detecção de DDoS - dataset completo

fluxos com *bwd_pkt_len_max* mais altos impactam positivamente.

Em ambos os casos, altos valores da *feature* da razão entre download e upload de pacotes (*down_up_ratio*) indicam fortemente que um fluxo é do tipo DDoS.

5. Conclusão e trabalhos futuros

Neste trabalho, foi proposta e implementada uma forma de avaliar a eficácia de modelos de aprendizado de máquina para a detecção de ataques em redes de computadores em situações onde os dados relativos a camada de transporte, como as portas de origem e destino e as *flags* TCP, não estariam disponíveis. Dentre os modelos avaliados, foi possível perceber que a perda de informações da camada de transporte afetou o desempenho dos algoritmos em todas as métricas avaliadas, ainda que em pequena escala. Os resultados foram comparados com os obtidos por [Rosay et al. 2021], que propuseram uma correção para o *dataset* CIC-IDS2017.

Através da comparação das métricas de *accuracy*, *precision*, *recall*, *f1 score*

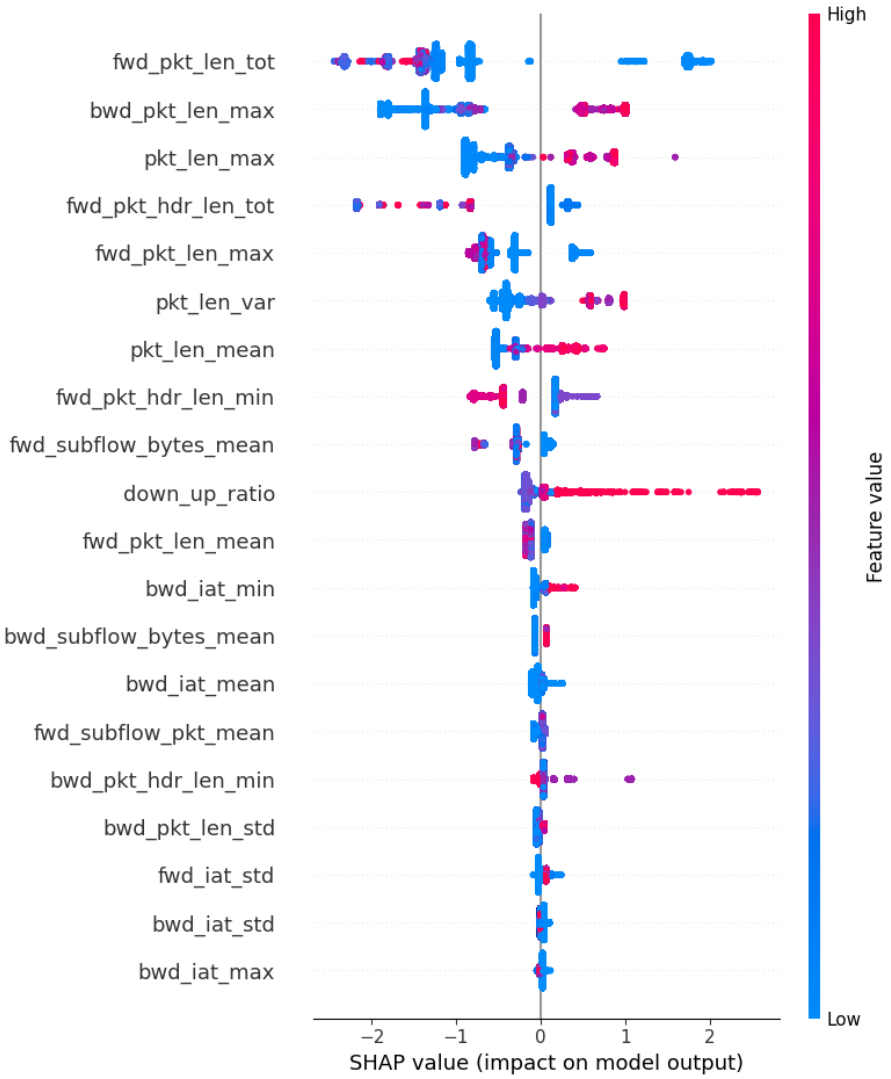

Figura 7. Impacto das features na detecção de DDoS - dataset criptografado

e MCC, foi possível perceber que os algoritmos, quando usando o conjunto de dados original, obtiveram resultados melhores que todos os modelos avaliados por [Rosay et al. 2021], porém, quando usando o conjunto de dados criptografado, os resultados foram piores que os obtidos pelo melhor modelo gerado pelos autores. Através da explicabilidade dos modelos, foi possível identificar quais *features* impactam mais na classificação de um fluxo de rede como cada uma das diferentes classes de ataque, o que pode ser útil para a criação de novas técnicas de detecção de ataques e, também, para a criação de técnicas de ataque que evitem a detecção por sistemas de detecção de intrusão baseados em *machine learning*.

Para trabalhos futuros, é possível avaliar a eficácia de outros algoritmos de aprendizado de máquina, assim como diferentes *datasets* para a detecção de ataques em redes de computadores. Além disso, seria interessante avaliar como a criptografia dos dados da camada de transporte afetaria outros protocolos, como o SCTP ou o QUIC.

Referências

Al-Essa, M. and Appice, A. (2022). Dealing with imbalanced data in multi-class network intrusion detection systems using xgboost. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases: International Workshops of ECML PKDD 2021, Virtual Event, September 13-17, 2021, Proceedings, Part II*, pages 5–21. Springer.

Alias, S. B., Manickam, S., and Kadhum, M. M. (2013). A study on packet capture mechanisms in real time network traffic. In *2013 International Conference on Advanced Computer Science Applications and Technologies*, pages 456–460.

Bhuyan, M. H., Bhattacharyya, D. K., and Kalita, J. K. (2013). Network anomaly detection: methods, systems and tools. *Ieee communications surveys & tutorials*, 16(1):303–336.

Bittau, A., Giffin, D. B., Handley, M. J., Mazieres, D., Slack, Q., and Smith, E. W. (2019a). Cryptographic Protection of TCP Streams (tcpcrypt). RFC 8548.

Bittau, A., Giffin, D. B., Handley, M. J., Mazieres, D., and Smith, E. W. (2019b). TCP-ENO: Encryption Negotiation Option. RFC 8547.

Bittau, A., Hamburg, M., Handley, M., Mazieres, D., and Boneh, D. (2010). The case for ubiquitous transport-level encryption. In *2010 USENIX Annual Technical Conference (USENIX ATC 10)*. USENIX Association.

Boukhtouta, A., Mokhov, S. A., Lakhdari, N.-E., Debbabi, M., and Paquet, J. (2016). Network malware classification comparison using dpi and flow packet headers. *Journal of Computer Virology and Hacking Techniques*, 12:69–100.

Boutaba, R., Salahuddin, M. A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., and Caicedo, O. M. (2018). A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications*, 9(1):1–99.

Dijkhuizen, N. V. and Ham, J. V. D. (2018). A survey of network traffic anonymisation techniques and implementations. *ACM Computing Surveys (CSUR)*, 51(3):1–27.

Gupta, N., Jindal, V., and Bedi, P. (2022). Cse-ids: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems. *Computers & Security*, 112:102499.

H2O.ai (2023). *H2O*. 3.40.0.2.

Joshi, M. and Hadi, T. H. (2015). A review of network traffic analysis and prediction techniques. *arXiv preprint arXiv:1507.05722*.

Karagiannis, T., Papagiannaki, K., and Faloutsos, M. (2005). Blinc: multilevel traffic classification in the dark. In *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 229–240.

Le, T.-T.-H., Kim, H., Kang, H., and Kim, H. (2022). Classification and explanation for intrusion detection system based on ensemble trees and shap method. *Sensors*, 22(3):1154.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.

Ring, M., Wunderlich, S., Scheuring, D., Landes, D., and Hotho, A. (2019). A survey of network-based intrusion detection data sets. *Computers & Security*, 86:147–167.

Rosay, A., Carlier, F., Cheval, E., and Leroux, P. (2021). From cic-ids2017 to lycos-ids2017: A corrected dataset for better performance. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 570–575.

Rosay, A., Riou, K., Carlier, F., and Leroux, P. (2022). Multi-layer perceptron for network intrusion detection: From a study on two recent data sets to deployment on automotive processor. *Annals of Telecommunications*, 77(5-6):371–394.

Sarhan, M., Layeghy, S., and Portmann, M. (2022). Towards a standard feature set for network intrusion detection system datasets. *Mobile networks and applications*, pages 1–14.

Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, 1:108–116.

Tavallaee, M., Bagheri, E., Lu, W., and Ghorbani, A. A. (2009). A detailed analysis of the kdd cup 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pages 1–6.

Ullah, I. and Mahmoud, Q. H. (2020). A scheme for generating a dataset for anomalous activity detection in iot networks. In *Advances in Artificial Intelligence: 33rd Canadian Conference on Artificial Intelligence, Canadian AI 2020, Ottawa, ON, Canada, May 13–15, 2020, Proceedings 33*, pages 508–520. Springer.

Usama, M., Qadir, J., Raza, A., Arif, H., Yau, K.-l. A., Elkhatib, Y., Hussain, A., and Al-Fuqaha, A. (2019). Unsupervised machine learning for networking: Techniques, applications and research challenges. *IEEE Access*, 7:65579–65615.