Caracterização do tráfego na Darknet utilizando árvores de decisão

Mateus Coutinho Marim[1], **Paulo Vitor Barbosa Ramos**[1]

[1]Departamento de Ciência da Computação
Universidade Federal de Juiz de Fora (UFJF)
Rua José Kelmer - Campus Universitário, Bairro São Pedro, 36036-330
Juiz de Fora – MG – Brasil

mateus.marim@ice.ufjf.br, paulo.barbosa@estudante.ufjf.br

Abstract. *The Darknet Traffic Classification Problem, related to the encrypted traffic, has been the main origin classification and application categorization models object. Through a literature revision, it is possible to discover the use of statically inference techniques and the uses of deep learning models. With a feature analysis and dataset complementation, it is possible to apply Decision Tree and Random Forest models in a way to reach accuracy beyond 95% in classes related to the origin and application classification and categorization in the encrypted traffic. Obtained results with correction related to the normalization, features selection and dataset feature expansion.*

Resumo. *O problema de Darknet Traffic Classification, relacionado ao tráfego encriptado, vêm sendo objeto para os modelos de classificação da origem do tráfego e categorização da aplicação. Por meio de uma revisão da literatura é possível descobrir o uso de técnicas de inferência estatística e aplicações de modelos usando redes de aprendizado profundo. Com uma análise de atributos e complementação do dataset é possível aplicar modelos de Decision Tree e Random Forest de forma a alcançar acurácias acima de 95% na classificação e categorização para classes relacionadas a origem e aplicação usada no tráfego encriptado. Resultado alcançado por meio de correções relacionadas à normalização e expansão dos atributos já inseridos na base de dados.*

1. Introdução

A *Internet* é um diversificado nicho de camadas de segurança e disponibilização de dados, sendo possível verificar conjuntos separados pelo nível de anonimato do serviço prestado pelas partes interessadas. Aplicações de redes sociais, plataformas de hospedagem e tantas outras ferramentas amplamente divulgadas, são serviços que representam uma ínfima parcela da real rede mundial de computadores. Denominada *Surface Web*, essa reduzida parte é aquela amplamente disponibilizada pelos indexadores e comumente utilizado pelos usuários que procuram serviços e aplicações ditas comuns [Rudesill et al. 2015]. A *Deep Web*, conjunto mais encriptado e almejado por aqueles que buscam serviços peculiares, disponibiliza os resultados não indexados pelas ferramentas de buscas convencionais. Além de ter um subconjunto denominado *Darknet*, ou *Dark web*, a totalidade do conjutno possui seus princípios fundados na contínua alteração de hospedagem e estabelecimento de conexões em pares seguros, *peer-to-peer* (P2P).

Esse subconjunto da *Deep Web*, possui o mais alto nível de técnicas de segurança, afim de preservar o anonimato desses grupos e prestadores de serviços, preservando a identidade dos sujeitos ativos e passivos nas relações, por exemplo, de venda de produtos no mercado negro, negociações de serviços e a troca de informações. Embora essas sejam algumas das atividades que estão no lado da ilegalidade, a *Darknet* demonstra uma diversificada rede estabelecida em princípios fundados na segurança da informação e tecnicidade de desenvolvimento. Pelas características de natureza anônima, como o uso de criptomoedas, conexões encriptadas e mercados virtuais, a *Darkweb* torna-se um repositório seguro para que qualquer indivíduo possa estabelecer atividades independente de sua natureza, possuindo a vantagem da dificultosa rastreabilidade [Mirea et al. 2019].

Classificar e categorizar o tipo de aplicação e origem nessas situações de criptografia é um dos objetivos almejados pelo estudo do problema de *Traffic Classification*. O objetivo consiste em determinar certas classes simplesmente pela análise do tráfego de dados, verificando padrões de duração de conexão, informações sobre origem e destino desses dados, portas conectadas e o tipo de aplicação relacionada. As abordagens mais comuns para a solução do problema são relacionadas à análise de portas ou inferência estatística dos pacotes enviados e recebidos [Valenti et al. 2013]. Essas abordagens possibilitam o aperfeiçoamento do *Quality of Service*, sendo possível a correta modulação de conexão e previnindo a baixa qualidade do serviço prestado em rede [Parchekani et al. 2020].

Sendo essas algumas das abordagens clássicas, constatam-se soluções envolvendo aplicações de redes neurais profundas, como *Deep Image Learning*, recentemente usada por [Gurdip Kaur 2020] na demonstração de seus resultados na análise do tráfego na *Darknet*. Além de demonstrar seus experimentos, contribui com a publicação dos dados coletados, possibilitando a comparabilidade de resultados usando outros modelos e inferências no mesmo conjunto abordado.

A comparabilidade de resultados no mesmo conjunto permite revisar e expandir os dados analisados, fornecendo uma nova solução e uma base de dados revisada para futuros interessados. Tais evoluções são alcançados pelo presente trabalho, no qual traz a público as inovações na base *CIC-Darknet2020*, a revisão da literatura sobre os métodos empregados em outros trabalhos e os resultados obtidos com a manipulação da base com a implementação dos modelos de *Decision Tree* e *Random Forest*, aplicados até então em conjuntos diferentes da *Darknet*.

Desta forma, o presente trabalho aborda o problema de classificação de tráfego pelo *CIC-Darknet2020*, que contém registros de tráfego real à partir da *Internet* comum, denominado tráfego benigno, e da *Darknet*. É feita uma análise dos atributos existentes para a criação de novos campos, expandindo a gama de dados existentes no *dataset*, sendo parâmetros para os modelos de classificação aplicados para a determinação da origem do tráfego, *Darknet* ou benigno, e categorização do tráfego advindo da *Darknet* pelo seu serviço.

O trabalho está dividido da seguinte forma: a Seção 2 traz alguns trabalhos relacionados ao tema, fundamentando a base estudada e o tema relacionado. A Seção 3, além de descrever as tratativas feitas, descreve seus principais atributos e as abordagens que podem ser realizados com seus dados. A Seção 4 demonstrará os resultados obti-

dos pela implementação dos modelos e a Seção 6 traz as considerações finais sobre as interpretações do trabalho, da base de dados e da análise feita, disponibilizando sugestões para trabalhos futuros a serem realizados usando os mesmos dados estudados.

2. Trabalhos Relacionados

Antes de demonstrar diferentes soluções para o problema de *Traffic Classification*, é necessário entender a definição do problema relacionado, que consiste em usar os dados de tráfego entre remetente e destinatário para classificar e categorizar a aplicação usada. Um dos principais desafios é realizar essa tarefa usando dados encriptados, abordagem feita em duas bases disponibilizadas pela *University of New Brunswick*, a *ISCXVPN2016* [Draper-Gil et al. 2016] e *ISCXTor2016* [Lashkari et al. 2017], que, respectivamente, fornecem o tráfego em redes usando VPN e Tor.

Recentemente, em um trabalho publicado por [Gurdip Kaur 2020], houve a disponibilização de uma base de dados que é a união das outras duas bases supra-referenciadas, a chamada *CIC-Darknet2020*. Tal trabalho realiza a classificação das aplicações provenientes da *Sufarce Web* e da *Darknet*, respectivamente, sendo definidas como origem benigna e *Darknet*, tendo o tráfego encriptado pelo uso da VPN e Tor. Além da publicação da base de dados, o autor apresentou acurácia de 92% para identificação da origem do tráfego e 86% para a categorização do tráfego em seu modelo de classificação usando redes neurais profundas, utilizando uma técnica chamada de *Deep Image Learning*.

[Draper-Gil et al. 2016] aborda a classificação da comunicação via VPN, usando redes neurais e a base de dados *ISCXVPN2016* para a classificação do tráfego em dois estágios. O primeiro, usando *Multi-Layer Perceptron* como função de ativação para o segundo estágio, uma *Recorrent Neural Network* para identificar as 6 classes empregadas pelo *dataset*. Estabelecendo dois cenários de categorização da aplicação, define modelos de *K-Nearest Neighbors* (KNN) e C4.5 *Decicion Tree* para a tarefa, demonstra-se um resultado com acurácia acima de 80%, tendo destaque para o C4.5 que teve resultados de medidas de precisão melhores.

[Lotfollahi et al. 2017], usando a mesma base, demonstra o desenvolvimento do *framework Deep Packet* para a solução do problema. O *framework* compreende em dois métodos de rede de aprendizado profundo, uma rede neural convolucional e um auto-encoder, ambos para a tarefa de classificação e caracterização. Tendo os resultados de acurácia e precisão acima de 90%, o trabalho teve um importante papel em fundamentar outros métodos além daquele que foi empregado para a solução do problema. Essas comparações envolvem a classificação por meio das portas, não sugerido pela baixa porcentagem de classificação [Moore and Papagiannaki 2005], usando a inspeção de *payload* e o uso da abordagem estatística, tendo a Função de Densidade de Probabilidade (PDF) demonstrado acurácias em torno de 91% para a classificação de protocolos HTTP, POP3 e SMTP no trabalho de [Crotti et al. 2007] e 87% para FTP, IMAP, SSH e TELNET em [Wang and Parish 2010].

[Lotfollahi et al. 2017] e [Draper-Gil et al. 2016], embora tenham contribuído para a categorização da aplicação pelo tráfego, não focaram nos dados provenientes da *Darknet*. [Gurdip Kaur 2020] aborda a categorização em duas camadas, a primeira relacionada a classificação da origem e a segunda em relação ao tráfego proveniente da *Dark-*

net, verificando os atributos do *dataset* mais importantes para a classificação. Mesmo com uma acurácia de 86% para o problema, os autores não realizam qualquer tipo de comparação com modelos de classificação mais simples.

Diferentemente dos trabalhos supracitados, utilizamos os modelos *Decision Tree* e *Random Forest* com o objetivo de realizar a comparação entre resultados dos modelos de classificação da origem do tráfego e a categorização da aplicação dos dados provenientes da Darknet. Dessa forma, a base de dados disponiblizada por [Gurdip Kaur 2020] é a escolhida para alcançar os objetivos propostos de classificação, tendo conseguido expandi-lá com novas características para os registros, inserindo informação dos endereços IP de origem e destino, realizando a divisão *n-grams* e manipulações nos registros originais. Essas alterações e comparações possibilitam a evolução tanto da base quanto do tema em destaque.

3. Dataset

Como fundamentado na seção anterior, *CIC-Darknet2020* é a base de dados analisada para o desenvolvimento de modelos de classificação. [Gurdip Kaur 2020] propõe duas tarefas de *Traffic Classification*, sendo a primeira classificar o tráfego proveniente da *Internet* benigna, não utilizando Tor ou VPN, e da *Darknet*. A Figura 1 demonstra a divisão dessas duas classes relacionadas à origem do tráfego. A segunda tarefa diz respeito à categorização da aplicação correspondente ao tráfego, tendo as seguintes classes envolvidas: *browsing*, *email*, *chat*, *audio-streaming*, *video-streaming*, *File-Transfer*, VOIP e P2P.

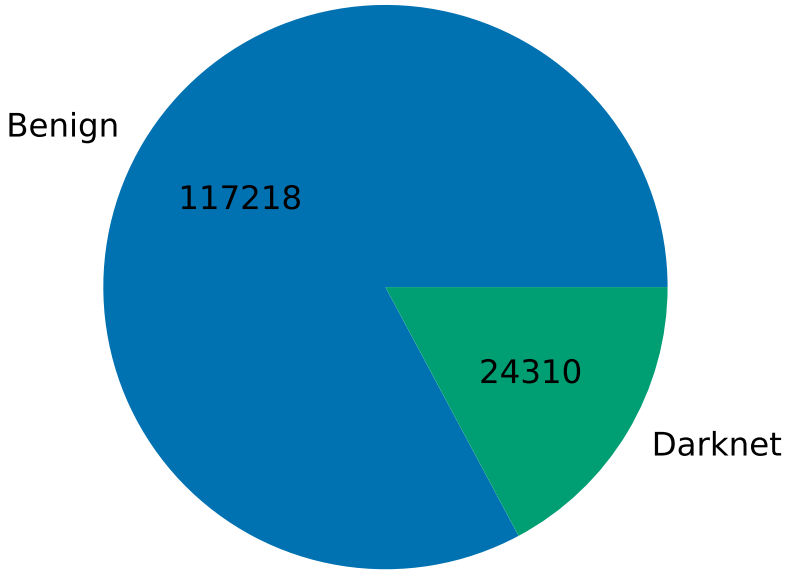


Figura 1. Distribuição pela origem dos dados

O *CIC-Darknet2020* possui uma variedade de campos para análise, trazendo informações sobre IP, porta, classificador, duração do tráfego de pacotes e outras medidas relacionadas. No total, são 141.528 registros dos quais, 24.310 provenientes da *Darknet* e 117.218 da rede benigna.

A segunda classe da base de dados diz respeito a aplicação envolvida. A Figura 2 possibilita verificar as aplicações relacionadas à sua origem, sendo possível inferir que classes relacionadas à *Streaming* de áudio e *Chat* são mais comuns para os dados provenientes da *Darknet*, enquanto para a rede benigna são as aplicações minoritárias.

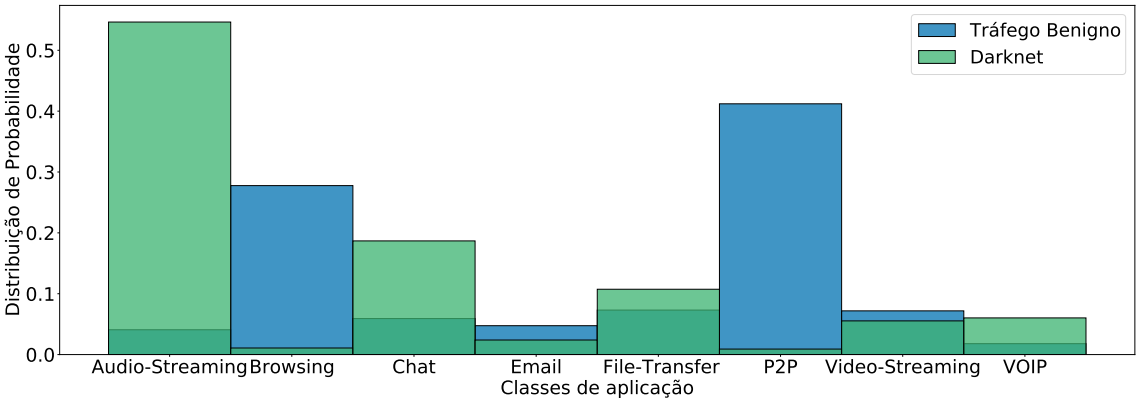


Figura 2. Probabilidade de ocorrência do tráfego de um tipo de serviço

Alguns dados da base não são úteis e outros precisaram ser complementados. Uma dessas complementações esta relacionada ao IP de origem e destino, que não está vinculada a qualquer informação sobre localização ou organização registrado. Para suprir essa lacuna, foi usada a biblioteca *IpInfo*, do *Python*, tendo a entrada desses IP's como parâmetros.

3.1. Pré-processamento do *dataset*

A primeira correção consiste na normalização das duas classes disponibilizadas pela base de dados, a de origem dos dados e do tipo de aplicação do tráfego. Através da inspeção das classes do *dataset* é possível verificar falta de padronização na nomeclatura das classes, assim como a sua redundância. Para corrigir esse problema, realizamos a padronização dos nomes dessas classes.

A base nos fornece IP's de origem e destino, o que possibilita extrair mais atributos relacionados aos endereços de rede. Uma das possibilidades é o uso de *One Hot Encoder*, mas como visto em [Baiardi et al. 2017] o uso de *n-grams* pode ajudar na diminuição dos erros do modelo de classificação gerado, consequentemente, reduzindo a percentagem de previsões falsas positivas. Dessa forma, a base foi expandida usando os IP's fragmentando-os em *n-grams* (Unigram, Bigram e Trigram), além de trazer as informações de hospedagem, geolocalização, *bogons* (endereços falsos), entre outros.

Com essas divisões em *n-grams* dos endereços de origem e destino, usamos a técnica de *Hashing Encoding* para criação de 100 novos atributos nomeados de col_i onde i representa o *ID* do novo atributo pois, como visto nos resultados de [Weinberger et al. 2009], possibilita a compressão dos atributos em relação ao, por exemplo, *One Hot Encoding* na qual poderia gerar milhares de novos atributos dependendo do número de categorias únicas de um único processado, sendo bastante útil para registros grandes e que são aplicados em modelos de aprendizado de máquina, mas tendo a desvantagem de que os novos atributos criados perdem no quesito de interpretabilidade. Além disso, como a pesquisa dos endereços de rede trouxe o país de origem do IP, criamos um novo atributo com essa informação e convertemos os países em números ordinais.

Uma outra característica que foi extraída foi a hora em que ocorreu a captura dos dados pelo campo *TimeStamp* do *dataset*. A Figura 3 demonstra a relação entre as horas de captura para as duas classes de origem do tráfego. É possível verificar dois diferentes padrões, um para cada classe, sendo possível inferir a provável importância do resgistro na classificação do tráfego, consequentemente, na sua utilização no modelo.

Além de ser possível a distinção dos horários, essa relação de tempo permite dizer quando há uma maior probabilidade de utilização. Para a rede benigna, constate-se um volume alto do tráfego contido entre às 7 horas e 12 horas, tendo alguns picos durante a madrugada. Para a rede *Darknet*, a distribuição é mais esparsa, não possuindo picos de utilização além da normal, contida entre às 24 horas e 7 horas. Essa relação demonstra uma disjunção exclusiva, ou seja, há uma probabilidade considerável de não haver tráfego da *Darknet* e benigna concomitantemente.

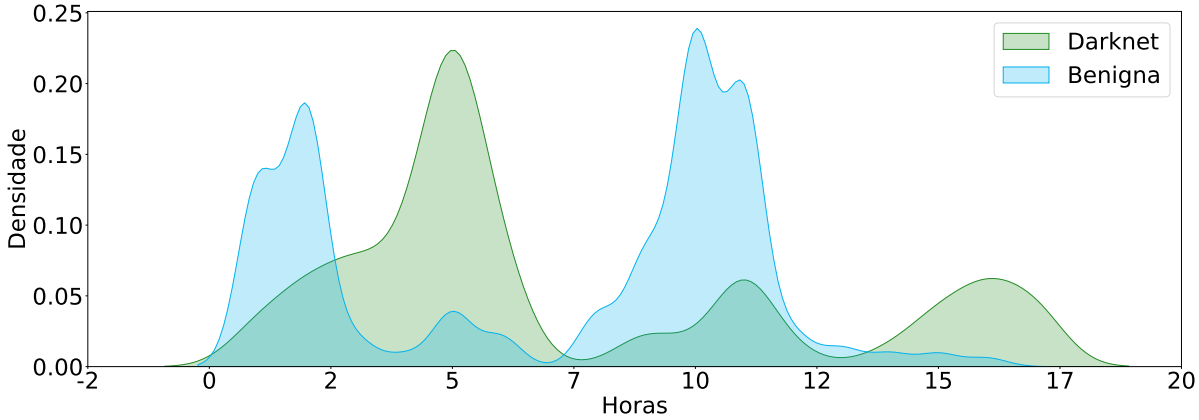


Figura 3. Densidade do tráfego em uma dada hora

Uma das transformações mais importantes a serem aplicadas nos dados é o escalonamento dos atributos, sendo representados por números reais através da padronização dos seus intervalos. Geralmente, poucos algoritmos de aprendizagem de máquina tem um bom desempenho com atributos de escalas muito diferentes [Géron 2019].

4. Experimentos

Nesse trabalho abordamos as duas tarefas de classificação propostas por [Gurdip Kaur 2020], a da predição da origem do tráfego da rede, com classificações entre *Benign* e *Darknet*, e caracterização dos serviços do tráfego na Darknet. Os modelos escolhidos são baseados em árvores de decisão e foram selecionados devido a sua simplicidade e facilidade na interpretação, além de, em conjunto com uma seleção de características, ser possível estimar a importância dos atributos de acordo com a sua influência na classificação. Abaixo são brevemente descritos os modelos selecionados [Géron 2019].

- *Decision Trees* (DT): modelos de aprendizado supervisionado não paramétrico que podem ser usados para classificação e regressão. Funciona pelo aprendizado de regras de decisão simples inferidas dos dados para a predição da variável alvo. São modelos simples de entender e interpretar e as árvores geradas podem ser visualizadas. A implementação do *sklearn* é uma CART otimizada.

- *Random Forest* (RF): é um modelo *ensemble* que usa DTs como classificadores fracos com o objetivo de gerar um classificador forte, a RF treina cada uma das DTs com a técnica de *Bagging* com o objetivo de gerar um classificador com uma performance melhor que a de seus componentes individuais.

4.1. Metodologia

Cada modelo treinado usando o *dataset* em apreço, os parâmetros pré-definidos pelo *sklearn* foram utilizados, tendo o devido cuidado em separar a base de dados em conjuntos de treinamento e teste, na proporção de 33% para esse e 67% para aquele. Essa divisão permite que o modelo analise dados que não foram usados durante o treinamento, garantindo a correta mensuração da acurácia na aplicação do conjunto de teste como parâmetro para o classificador.

Seja *TP* o número predições corretas, *FP* as predições classificadas erroneamente como corretas e *FN* a quantidade de predições classificadas corretamente como erradas. São utilizadas como métricas, em ambas tarefas de classificação, a performance do modelo estimada através da validação cruzada estratificada com *10-fold*, uma validação final da eficiência do modelo com o conjunto de teste separado, e por fim as métricas definidas a seguir:

- Precisão: acurácia das predições positivas ou corretas.

$$precisao = \frac{TP}{TP + FP}$$

- *Recall*: a precisão é geralmente encontrada com a medida *recall* que representa taxa de positivos verdadeiros.

$$recall = \frac{TP}{TP + FN}$$

- *F-score*: o *F-score* combina a precisão e o *recall* com a média harmônica dos dois, enquanto a média comum trata todos valores igualmente a média harmônica coloca mais peso em valores mais baixos. O *F-score* assume valores próximos de 1 quando ambas a precisão e o *recall* estão altos.

$$F\text{-}score = \frac{TP}{TP + \frac{FN+TP}{2}}$$

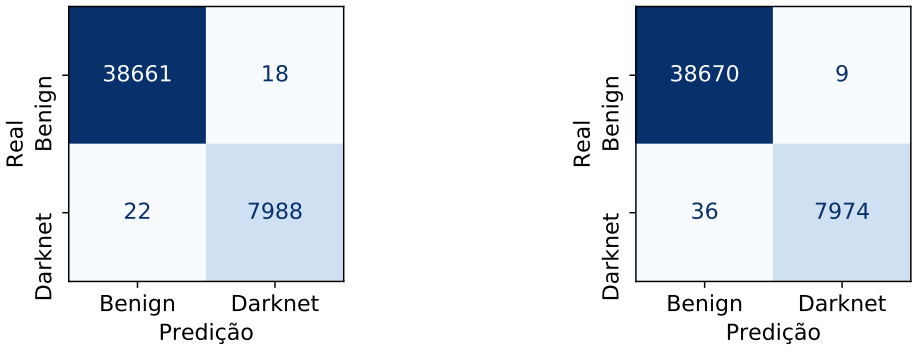
Todos experimentos foram feitos com a biblioteca *sklearn* [Pedregosa et al. 2011] do Python em um computador com processador *Intel Core i5-7200U* com 4 núcleos de $2.5GHz$, $20GB$ de RAM e sistema operacional *Ubuntu* 20.04. Além disso, foi usada a semente de valor 42 nos algoritmos com alguma aleatoriedade, afim de permitir a reprodutibilidade dos resultados.

4.2. Detecção do tráfego da *Darknet*

A Tabela 1 sumariza os valores das métricas de cada modelo na classificação entre os rótulos *Benign* e *Darknet*. Além disso, o modelo conseguiu uma acurácia de 99.89% no *10-fold* e uma boa capacidade de generalização, qualidade percebida pelo resultado similar a acurácia de teste de cada rótulo.

		Precisão	Recall	F-score	Acc. de teste
Decision tree	**Benign**	0.9994	0.9992	0.9993	99.94%
	Darknet	0.9964	0.9971	0.9967	99.78%
Random forest	**Benign**	0.9989	0.9997	0.9993	99.91%
	Darknet	0.9987	0.9947	0.9967	99.89%

Tabela 1. Sumário das métricas de avaliação dos modelos



(a) Decision tree **(b) Random forest**

Figura 4. Matrizes de confusão da detecção de tráfego da *Darknet*

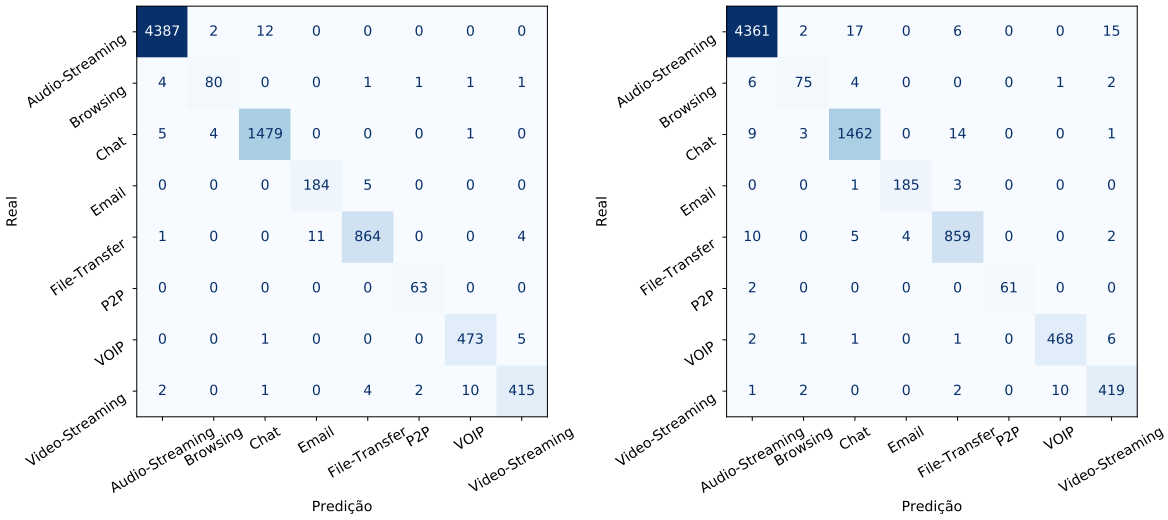
Nas matrizes de confusão das Figuras 4a e 4b, correspondentes aos modelos de *Decision tree* e *Random Forest*, é possível observar que, mesmo que um dos modelos esteja com padrão atênuado, verifica-se uma tendência de classificar erroneamente um tráfego da *Darknet* como benigno. Esse erro, provavelmente, tenha sido causado pelo insuficiente número de exemplos de tráfego da *Darknet* contidos no *dataset*. Apesar disso, a ocorrência desse erro é insignificante quando comparado ao desempenho geral do modelo até então treinado.

4.3. Caracterização do tráfego da *Darknet*

Nas matrizes de confusão das Figuras 5a e 5b, respectivamente correspondentes a *Decision Tree* e *Random Forest*, fica evidente que os erros comuns estão relacionados ao tráfego com rótulos de *Chat* e *Audio-Streaming*, podendo indicar que existe alguma similaridade nos rótulos, o quê pode causar certa confusão nos modelos.

As Figuras 6a e 6b relacionam, em coordenadas polares, os valores das métricas de precisão, *recall* e *F-score* para os modelos de *Decision Tree* e *Random Forest* respectivamente. Vale ressaltar que, como os resultados das métricas estão bem próximas de 1 para todas consideradas, colocamos no gráfico a diferença entre o valor máximo das métricas e o valor obtido no modelo em questão para haver destaque nas diferenças dos resultados. É possível verificar que apenas a classe *Browsing* ficou com uma distância discrepante do valor máximo das métricas.

A Figura 6c mostra os erros obtidos pelos modelos na classificação de cada tipo de serviço associado aos tráfegos, fica evidente que as classes que obtiveram maiores erros de classificação são aquelas menos representadas no *dataset* e que a *Random forest* foi capaz de atenuar os erros dessas classes, sendo mais adequada para a classificação do tráfego



(a) Decision tree **(b) Random forest**

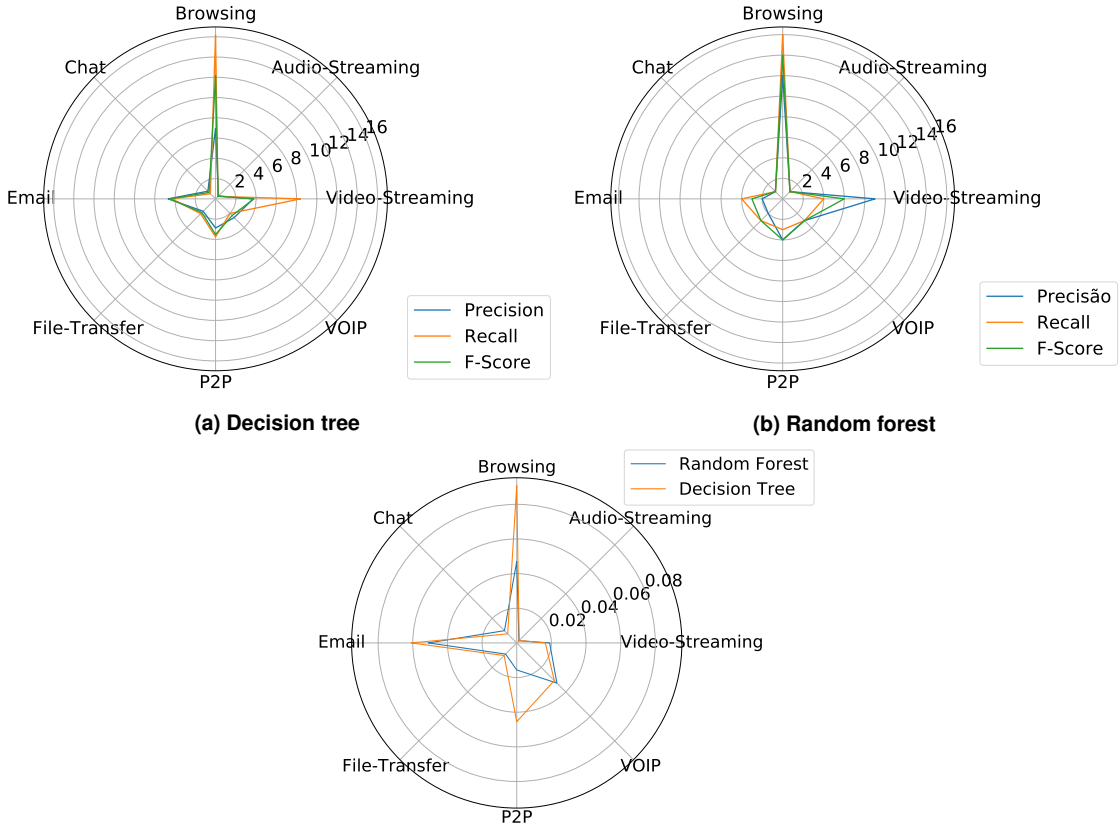
Figura 5. Matrizes de confusão da detecção de tráfego da *Darknet*

supondo que a adição de novos exemplos siga a mesma distribuição de probabilidade do conjunto de treinamento. Obtivemos um erro de classificação menor, até mesmo nas classes com menor representação, que o modelo proposto por [Gurdip Kaur 2020] com *deep learning*, podendo indicar que o problema da caracterização do tráfego da *darknet* é melhor resolvido com modelos mais simples e mais fáceis de interpretar sem a necessidade de recorrer para modelos mais complexos e que demandam mais recursos computacionais.

5. Seleção de atributos

É desejável em qualquer tarefa de classificação que os atributos contidos no *dataset* contenham o máximo de informações sobre o problema no menor número de atributos possível. A existência de atributos irrelevantes ou redundantes podem prejudicar a eficiência do classificador e, além disso, a remoção dos mesmos pode diminuir os efeitos da maldição da dimensionalidade. A seleção de atributos tem a tarefa de reduzir a dimensionalidade do *dataset* através da seleção do subconjunto de atributos mais relevantes para o problema segundo algum critério mantendo os mesmos, ou quase os mesmos, resultados [Villela et al. 2011].

Neste trabalho utilizamos o método *Recursive Feature Elimination* (RFE) que funciona através da remoção recursiva de um número fixo de atributos e do retreinamento do modelo. Para avaliar a qualidade dos subconjuntos gerados pelo RFE é feita uma validação cruzada estratificada com 10-fold e, no final da execução, é selecionado o subconjunto com maior acurácia e menor número de atributos. Devido aos resultados obtidos anteriormente, decidimos utilizar a *random forest* como classificador interno do RFE. Outro motivo para a utilização da RF é que ela também permite saber a importância dos atributos, chamada de importância de Gini na RF, após o treinamento do modelo. Dessa forma, após a seleção de características também fazemos uma análise dos atributos mais importantes considerando os novos inseridos.



(a) Decision tree

(b) Random forest

(c) Erro no conjunto de teste

Figura 6. Resultados das métricas para detecção de tráfego da *Darknet*

Devido a natureza aleatória do RFE, cada execução pode gerar um subconjunto de atributos diferentes como resultado, assim o tamanho final dos subconjuntos deve ser considerado como uma aproximação do subconjunto ótimo de atributos. Na Tabela 2 resumimos os resultados, podemos ver que houve uma redução significativa para o número de atributos do *dataset* sem que houvesse perda na acurácia da classificação em ambas tarefas, na primeira obtivemos uma redução de 83% do total de atributos e na caracterização do tráfego o número de atributos foi reduzido em 72%.

	# atributos	**Acurácia de teste**	**10-fold**
Benign vs. Darknet	30	99.89%	99.88%
Caracterização do tráfego da Darknet	50	98.62%	98.72%

Tabela 2. Sumário dos resultados da seleção de atributos

As Tabelas 3 e 4 mostram os 22 atributos mais importantes nos conjuntos selecionados pelo RFE, fica evidente que em ambos conjuntos os atributos inseridos pelo pré-processamento dos dados estão nos primeiros lugares em relação a sua importância para a classificação do modelo. Isso indica que os atributos inseridos são relevantes e que é mais vantajoso fazer o processamento dos atributos do que removê-los quando não parecem ser relevantes, como feito em [Gurdip Kaur 2020].

Atributo	Importância
col_91	0.7628
col_49	0.1205
Bwd Init Win Bytes	0.0418
col_24	0.0408
Idle Min	0.0141
col_96	0.0034
Idle Std	0.0021
col_45	0.0018
hour	0.0017
Average Packet Size	0.0015
Flow IAT Std	0.0011
col_1	0.0009
Idle Mean	0.0009
Flow IAT Min	0.0008
Fwd Packet Length Max	0.0008
FIN Flag Count	0.0007
Src Port	0.0006
FWD Init Win Bytes	0.0006
Fwd IAT Total	0.0005
Flow Duration	0.0004
col_71	0.0004
Fwd Packets/s	0.0004

Tabela 3. Caracterização do tráfego na *darknet*

Atributo	Importância
col_76	0.4287
hour	0.1455
Bwd Packet Length Min	0.1262
Idle Max	0.0517
Fwd Header Length	0.0338
Idle Min	0.0335
col_58	0.0312
Packet Length Max	0.0245
Flow Duration	0.0158
col_75	0.0142
col_11	0.0128
col_21	0.0112
col_45	0.0107
Src Port	0.0083
Dst Port	0.0078
Flow IAT Max	0.0053
Fwd Seg Size Min	0.0042
Flow IAT Min	0.0039
col_91	0.0037
FWD Init Win Bytes	0.0029
Subflow Fwd Bytes	0.0029
Fwd IAT Max	0.0026

Tabela 4. Detecção do tráfego da *Darknet*

6. Conclusão

Neste trabalho abordamos os problemas da detecção e caracterização do tráfego proveniente da Darknet através da utilização de modelos de aprendizagem baseados em árvores de decisão, sendo eles a *decision tree* e a *random forest*, que se mostraram capazes de classificar novos registros de tráfego com uma acurácia superior a 98% para cada uma das tarefas de classificação.

Também foram extraídos novos atributos do *dataset* original pela busca de informações dos IPs de origem e destino do tráfego e pela codificação dos mesmos com o *hashing encoding*, outro atributo gerado foi o horário em que o tráfego ocorreu pelo *timestamp* incluıd no *dataset*, que pelas nossas análises iniciais mostraram potencial para contribuir na eficiência dos modelos treinados por mostrarem que os tráfegos da internet comum e da Darknet costumam ocorrer em horários distintos.

Por fim, também fizemos uma seleção de atributos com o RFE e verificamos que os novos atributos inseridos tiveram relevância para a predição dos modelos ficando evidente que em alguns casos é preferível o processamento de atributos que a primeira vista tem pouca relevância, além disso, foi possível obter uma grande redução do número de atributos do *dataset* original.

Fica evidente que algoritmos de aprendizagem de máquina simples como os ba-

seados em árvore decisória são bons candidatos para obtenção de resultados competitivos para problemas do mundo real e que podem ter a sua eficiência melhorada com um processamento mais cuidadoso dos atributos já existentes, apesar disso, uma desvantagem dos modelos utilizados é que eles não podem ser treinados de forma online, ou seja, não são capazes de aprender com um novo exemplo a não ser que o modelo seja retreinado com todos os dados anteriores e os novos exemplos, uma proposta de trabalho futuro é uma pesquisa com modelos de aprendizagem online.

Referências

Baiardi, F., Lipilini, J., and Tonelli, F. (2017). Using s-rules to fire dynamic countermeasures. In *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pages 371–375. IEEE.

Crotti, M., Dusi, M., Gringoli, F., and Salgarelli, L. (2007). Traffic classification through simple statistical fingerprinting. *ACM SIGCOMM Computer Communication Review*, 37(1):5–16.

Draper-Gil, G., Lashkari, A. H., Mamun, M. S. I., and Ghorbani, A. A. (2016). Characterization of encrypted and vpn traffic using time-related. In *Proceedings of the 2nd international conference on information systems security and privacy (ICISSP)*, pages 407–414.

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.

Gurdip Kaur, Arash Habibi Lashkari, A. R. (2020). Didarknet: A contemporary approach to detect and characterize the darknet traffic using deep image learning.

Lashkari, A. H., Draper-Gil, G., Mamun, M. S. I., and Ghorbani, A. A. (2017). Characterization of tor traffic using time based features. In *ICISSp*, pages 253–262.

Lotfollahi, M., Zade, R. S. H., Siavoshani, M. J., and Saberian, M. (2017). Deep packet: A novel approach for encrypted traffic classification using deep learning. *CoRR*, abs/1709.02656.

Mirea, M., Wang, V., and Jung, J. (2019). The not so dark side of the darknet: a qualitative study. *Security Journal*, 32(2):102–118.

Moore, A. W. and Papagiannaki, K. (2005). Toward the accurate identification of network applications. In Dovrolis, C., editor, *Passive and Active Network Measurement*, pages 41–54, Berlin, Heidelberg. Springer Berlin Heidelberg.

Parchekani, A., Naghadeh, S. N., and Shah-Mansouri, V. (2020). Classification of traffic using neural networks by rejecting: a novel approach in classifying vpn traffic. *arXiv preprint arXiv:2001.03665*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rudesill, D. S., Caverlee, J., and Sui, D. (2015). The deep web and the darknet: A look inside the internet's massive black box. *Woodrow Wilson International Center for Scholars, STIP*, 3.

Valenti, S., Rossi, D., Dainotti, A., Pescapè, A., Finamore, A., and Mellia, M. (2013). *Reviewing Traffic Classification*, pages 123–147. Springer Berlin Heidelberg, Berlin, Heidelberg.

Villela, S. M., Xavier, A. E., and Neto, R. F. (2011). Seleção de características com busca ordenada e classificadores de larga margem. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação.

Wang, X. and Parish, D. (2010). Optimised multi-stage tcp traffic classifier based on packet size distributions. *2010 Third International Conference on Communication Theory, Reliability, and Quality of Service*, 0:98–103.

Weinberger, K., Dasgupta, A., Langford, J., Smola, A., and Attenberg, J. (2009). Feature hashing for large scale multitask learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 1113–1120.