Aprimorando a Detecção de Ataques Automáticos através de Decomposição Espectral de Pacotes de Rede

Lucas Airam C. de Souza, Gustavo F. Camilo, Otto Carlos M. B. Duarte

Grupo de Teleinformática e Automação (GTA)
Universidade Federal do Rio de Janeiro (UFRJ)

Resumo. *A classificação de fluxos para a identificação de ataques em redes de computadores por aprendizado de máquina utiliza características quantitativas que sintetizam as informações de pacotes pertencentes a um fluxo. Entretanto, as características convencionais, como tamanho de pacote e número de bytes, geram redundâncias e não representam as correlações temporais entre os pacotes de um fluxo. Ataques de rede automatizados geram padrões periódicos observáveis através da decomposição espectral, o que facilita a classificação. Este artigo propõe o FENED[1], um método para extrair características de dados de rede considerando a ordem de chegada dos pacotes dentro de um mesmo fluxo através da transformada rápida de Fourier para a classificação binária. O vetor de características proposto contém o módulo das componentes espectrais do fluxo. Os resultados mostram que a proposta é melhor ou igual às propostas convencionais de extração de características que desconsideram a ordem de chegada dos pacotes em um fluxo.*

1. Introdução

As redes de robôs (*botnets*) são as maiores ameaças para as redes de computadores atuais, infectando milhares de dispositivos para aumentarem a efetividade de ataques e com potenciais para causar prejuízos devastadores. O cenário é muito preocupante devido à massiva quantidade de dispositivos de Internet das coisas (*Internet of Things* - IoT), que usualmente possuem baixa capacidade de processamento e segurança, sendo facilmente comprometidos e adicionados às redes de robôs [Bezerra et al. 2018]. Os padrões de comunicação e os ataques de redes de robôs são automatizados, o que permite a rápida disseminação. Nesse cenário, o uso da tecnologia de aprendizado de máquina surge como uma importante contramedida para detectar padrões de tráfego e mitigar ameaças de rede. A extração de características do conjunto de dados é a etapa mais crítica da criação de um algoritmo de aprendizado de máquina, pois as características afetam diretamente o ajuste dos parâmetros do modelo e o desempenho de classificação. Entre os métodos para extração de características e representação dos fluxos de rede existem os convencionais, que quantificam informações dos pacotes, como número de bytes e quantidade de pacotes [Lobato et al. 2017], ou representações mais complexas através de grafos [Sanz et al. 2018, Bian et al. 2019] ou até por imagem [Liu et al. 2019]. Entretanto, essas características não refletem a dependência temporal entre os pacotes, e nem comportamentos periódicos gerados pela automatização dos ataques.

Este trabalho foi realizado com recursos do CNPq, CAPES, FAPERJ e FAPESP (18/23292-0, 2015/24485-9 e 2014/50937-1).

[1]Do inglês Feature Extraction by Network spEctrum Decomposition.

Este artigo propõe o FENED[2], um método para extração de características de fluxos de rede através da transformada rápida de Fourier (*Fast Fourier Transform* - FFT). O FENED extrai um vetor de características com as componentes espectrais dos fluxos de rede para o treino e teste de algoritmos de aprendizado de máquina. A representação dos fluxos no domínio da frequência reflete a dependência temporal entre os pacotes e evidencia comportamentos periódicos típicos de redes de robôs e ataques automatizados. O método proposto utiliza o tamanho dos pacotes para gerar as componentes espectrais, sendo adaptável para análise de tráfego encriptado e preservando a privacidade das amostras por não manipular dados sensíveis, que é uma preocupação atual [Guzman et al. 2021, Camilo et al. 2020]. Além disso, a classificação dos fluxos por meio das características propostas é agnóstica ao tipo de classificador adotado.

Os resultados mostram que o método supera ou iguala na maioria dos casos o desempenho na classificação de fluxos de rede quando comparado ao método convencional aplicado para extração de características. A dimensão do vetor aplicado no cálculo da FFT não altera estatisticamente as métricas de classificação obtidas, permitindo a proposta operar janelas menores para reduzir gastos computacionais enquanto mantém alto desempenho de classificação. Por fim, a fixação da janela em $s = 10$ pacotes torna a complexidade computacional $O(f.N)$, dependendo linearmente do número f de fluxos analisados com N pacotes em cada fluxo.

O restante do artigo é organizado da seguinte forma. A Seção 2 exibe o estado da arte em métodos para extrair características de fluxos de redes de computadores. A Seção 3 apresenta o método proposto e analisa sua complexidade computacional. A Seção 4 detalha o desenvolvimento do protótipo para aplicar o método, além de analisar os resultados obtidos. Por fim, a Seção 5 conclui o artigo e apresenta as direções futuras.

2. Trabalhos Relacionados

A extração de características é uma etapa primordial do aprendizado de máquina, na qual são selecionados os dados utilizados para ajustar os parâmetros do modelo e testá-lo. Em análise de tráfego de redes, é muito comum usar como identificadores de fluxo a quíntupla dos protocolos TCP/IP, que são o endereço IP origem, o endereço IP destino, protocolo de transporte, a porta TCP/UDP origem e a porta TCP/UDP destino; e como características os dados dos pacotes, tais como tamanho em número de bytes do pacote e número de pacotes de um fluxo etc. As ferramentas mais populares do mercado para análise de tráfego, como Netflow e o sFlow, seguem esta abordagem. O flowtbag[3] é uma aplicação que processa um arquivo de captura de pacotes de rede e extrai 40 características numéricas de fluxos identificados pela quíntupla [Guimarães et al. 2020]. A maioria das propostas para a detecção de intrusão [Viegas et al. 2019, Pelloso et al. 2018, Lobato et al. 2017, Possebon et al. 2019] é baseada em características de tráfegos de pacotes deste tipo. Entretanto, os tópicos de pesquisa em engenharia de características propõem abordagens alternativas para representação de um tráfego de rede, como grafos, imagens e análises espectrais.

As características extraídas de grafos permitem obter relações entre fluxos di-

[2]Disponível em: https://github.com/GTA-UFRJ-team/FENED.git

[3]Disponível em: https://github.com/DanielArndt/flowtbag

ferentes aos invés de informações de um único fluxo. Sanz *et al.* e Sagirlar *et al.* propõem uma representação da rede por grafos para detectar fluxos maliciosos [Sanz et al. 2018, Sagirlar et al. 2018]. Porém, o processo de construção do grafo envolve um maior custo computacional dificultando o emprego destas propostas em detecção de ataques em tempo real. Outra abordagem para representar um fluxo de rede é através de uma imagem que represente a diferença entre os tráfegos malicioso e normal [Liu et al. 2019]. Esta técnica também sofre com o alto custo computacional, mas tem sido usada com algum sucesso com as modernas técnicas de aprendizado profundo e com processamento paralelo de processadores gráficos (*Graphical Processing Unit* - GPU). Por fim, a análise espectral dos pacotes também permite identificar comportamentos descontínuos, como ataques de negação de serviço com baixas taxas [Aiello et al. 2014], ou periódicos.

Os padrões de comunicação de ataques automatizados provenientes de uma rede de robôs são periódicos, enquanto os fluxos normais possuem um espectro similar ao de um ruído branco [Chen and Hwang 2007]. Portanto, a análise espectral de fluxos de rede permite caracterizar a periodicidade e identificar ataques [Bottazzi et al. 2016]. Zhou *et al.* propõem a análise do espectro de fluxos em conjunto com a medida de variância e média dos pacotes para detectar comportamentos maliciosos gerados de forma automática [Zhou and Lang 2003]. Chimedstseren *et al.* empregam a transformada de Fourier para detecção de intrusão [Chimedtseren et al. 2014]. Os autores definem um fluxo através do par endereço IP origem e endereço IP destino e criam uma série temporal contendo o tamanho em bytes dos pacotes de cada fluxo, atribuindo um sinal de magnitude positivo para os pacotes enviados pela origem e recebidos pelo destino ou negativo caso a comunicação ocorra na direção oposta. A finalidade do sinal de magnitude é obter um sinal com oscilação temporal e verificar a periodicidade no padrão de comunicação entre cliente e servidor.

AsSadhan e Moura propõem analisar o periodograma de tráfegos de rede para a detecção de redes de robôs [AsSadhan and Moura 2014], enquanto que Manasrah *et al.* utilizam a densidade espectral [Manasrah et al. 2020]. Os comportamentos automatizados da arquitetura da rede de robôs implicam componentes periódicas com alta energia. Os autores utilizam a função de autocorrelação no periodograma extraído para identificar os IPs pertencentes à rede de robôs. PsyBoG [Kwon et al. 2014] é um mecanismo que analisa a periodicidade do tráfego DNS para detectar redes de robôs. Porém, as propostas se limitam a buscar no tráfego de rede componentes periódicas com alta energia que ultrapassam um limiar para a detectar ataques ao invés de analisar outras propriedades do espectro.

Yu *et al.* utilizam a propriedade da transformada discreta de fourier (*Discrete Fourier Transform* - DFT) que preserva a distância euclidiana para medir a similaridade entre os fluxos e identificar ataques [Yu et al. 2009]. A partir da DFT os autores criam um vetor com um número reduzido de componentes espectrais para o cálculo da distância, pois os primeiros coeficientes das frequências mais baixas retêm a maior parte da energia espectral. Fouladi *et al.* propõem aplicar a transformada de Fourier para detecção de negação de serviço distribuída através da medida de similaridade [Fouladi et al. 2019]. As propostas consideram que os fluxos pertencentes a uma mesma rede de robôs possuem alta similaridade. Assim, a distância entre dois fluxos maliciosos é menor do que entre

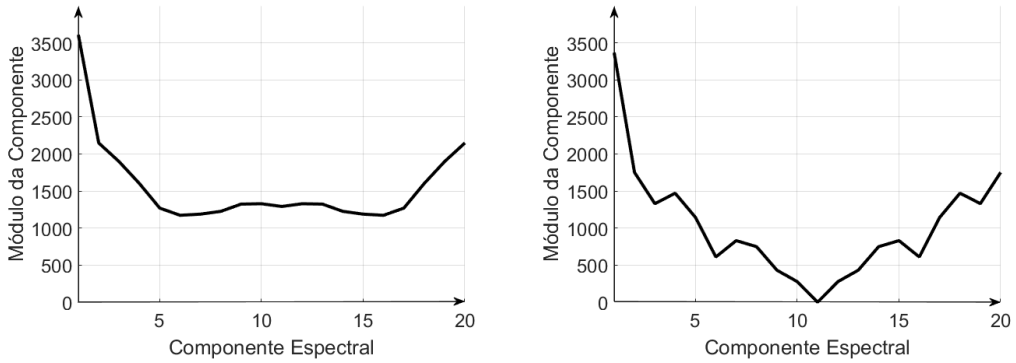
fluxos normais. Portanto, os autores medem a distância entre as componentes de dois fluxos e classificam como suspeito todo fluxo que estiver abaixo do limiar estabelecido. As propostas possuem um baixo número de falsos positivos, entretanto os autores utilizam somente a medida de distância entre componentes extraídas para identificar os ataques.

Powell propõe o uso da transformada discreta de Fourier para a detecção de extrusão de dados [Powell 2019]. A transformada de Fourier extrai características que possuem correlações temporais e permite uma classificação do tráfego com baixas taxas de falso positivo. Entretanto, o autor limita a avaliação da proposta a um conjunto de dados de detecção de extrusão de dados e não avalia a proposta para um conjunto de dados de rede de robôs. O uso de técnicas de decomposição espectral voltadas para botnets apresenta vantagens, já que ataques automatizados possuem padrões espectrais facilmente distinguíveis de padrões de tráfego normal.

Ao contrário dos artigos citados, este artigo extrai um vetor de características que representa as componentes espectrais dos fluxos de rede para o treino e teste de algoritmos de aprendizado de máquina supervisionado. A representação dos fluxos através de componentes espectrais extrai características que refletem a sequência temporal entre os pacotes do fluxo. Além disso, a classificação dos fluxos não se restringe ao uso de medidas de similaridade e periodicidade como as propostas anteriores e é agnóstica ao tipo de classificador adotado.

3. O Método Proposto para Extração de Características Espectrais

O artigo propõe um método de extração de características que retorna um vetor contendo os módulos de cada componente da transformada discreta de Fourier para um fluxo de pacotes. A ideia-chave para a detecção de fluxos maliciosos tem como base o comportamento espectral distinto do tamanho dos pacotes de um fluxo normal e de um fluxo malicioso. Os fluxos gerados através de uma comunicação normal possuem um espectro similar ao de um ruído branco [Chen and Hwang 2007], com as componentes



(a) Espectro de um fluxo, com 20 componentes extraídas, escolhido de forma aleatória entre as amostras de fluxos normais. O espectro com os valores dos módulos das componentes próximos uns dos outros se aproxima do ruído branco.

(b) Espectro de um fluxo, com 20 componentes extraídas, escolhido de forma aleatória entre as amostras de fluxos maliciosos. O espectro com os valores dos módulos das componentes difere bastante do ruído branco.

Figura 1. Comparação entre os espectros de a) um fluxo normal, que se assemelha a um ruído branco, e de b) um fluxo malicioso, no qual as componentes de frequência diferem bastante de valores.

próximas da média, como exibido na Figura 1(a). Entretanto, os fluxos de ataque gerados de modo automatizado, por uma rede de robôs, apresentam valores de algumas componentes espectrais que estão distantes da média, como exibido na Figura 1(b). A comparação da Figura 1 utiliza janelas de mesma dimensão para a decomposição espectral para não enviesar a comparação entre os fluxos. Assim, a partir do vetor de características extraído é possível obter informações sobre comportamentos periódicos e automatizados, que são padrões comuns em redes de robôs [Blaise et al. 2020]. As etapas para a construção do vetor de características são detalhadas a seguir.

3.1. Extração de Características com o FENED

A Figura 2 exibe as etapas propostas para extrair características espectrais de um fluxo utilizando o FENED. Inicialmente, o tráfego de rede é capturado e separado em fluxos[4], em que para cada fluxo é alocado um vetor que contém o tamanho em bytes dos pacotes em ordem cronológica. Os pacotes enviados pela origem e recebidos pelo destino possuem valores positivos, enquanto os pacotes enviados pelo destino em direção à origem são armazenados com um sinal negativo de magnitude. O objetivo do sinal de magnitude é produzir um sinal discreto que oscila em torno da abcissa [Chimedtseren et al. 2014].

Os algoritmos de aprendizado de máquina tradicionais, como árvore de decisão e máquina de vetores suporte, utilizam, treinam e classificam amostras que contêm vetores de características estruturados. Assim, uma restrição para a classificação de amostras é que o vetor de características possua a mesma dimensão em todo o conjunto de dados.

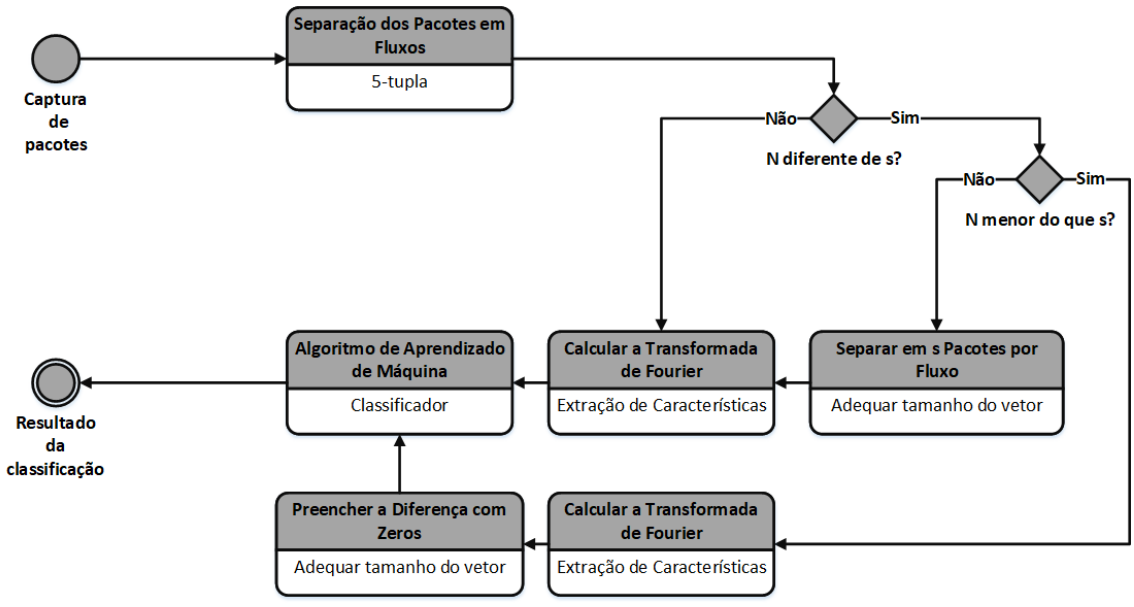


Figura 2. Diagrama de execução proposto para extração de características e classificação de fluxos de rede com *N* pacotes. Os pacotes capturados são identificados e separados em fluxos usando a quíntupla TCP/IP e são calculados os módulos do espectro que formam o vetor de características. O vetor de características é redimensionado para o tamanho *s* antes da sua classificação.

[4]Os fluxos são definidos por pacotes que contém o mesmo IP origem, porta origem, IP destino, porta destino e protocolo da camada de transporte.

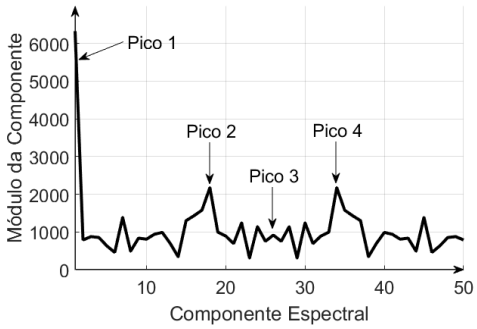
Porém, a dimensão do vetor resultante da transformada de Fourier é igual à quantidade N de pacotes utilizada no cálculo. Portanto, é necessário adequar o tamanho dos vetores gerados a fim de garantir a igualdade de dimensão após a extração de características. Para isso, é definido o parâmetro s, que representa a dimensão do vetor de características utilizado pelo classificador. O FENED processa um arquivo de captura de rede de forma contínua, sem dimensionar o vetor temporal a priori. Assim, a extração gera $\lceil \frac{N}{s} \rceil$ vetores de características para um fluxo com N pacotes em um arquivo de captura. Caso o fluxo tenha um número maior de pacotes, é necessário aplicar uma política de seleção para reduzir seu tamanho para s. Por outro lado, se o fluxo contiver menos pacotes do que o tamanho configurado, a criação do vetor de características deve introduzir elementos "de enchimento" até o vetor atingir a dimensão correta.

A Figura 3 exibe o espectro de dois fluxos, o primeiro contém um número maior de componentes, Figura 3(a), enquanto o segundo possui uma quantidade menor, Figura 3(b). O fluxo normal contém 50 componentes, enquanto que o fluxo malicioso possui apenas 6. Exemplificando o caso em que a configuração da dimensão do vetor de características é $s = 10$, há quatro possibilidades para adequar a dimensão do vetor de características: duas políticas para adequar o tamanho dos fluxos para o caso em que s é menor do que N e outras duas para o caso em que s é maior do que N.

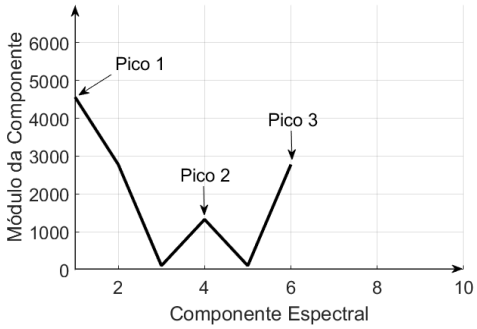
A primeira estratégia para reduzir a quantidade de componentes espectrais extraídas do fluxo é utilizar janelas com s pacotes para calcular a FFT, como mostra a Figura 3(c). Neste caso, o módulo das componentes é reduzido, porém o formato continua próximo ao espectro da Figura 3(a). A vantagem de utilizar essa estratégia é possibilitar a extração de características em tempo real, pois o cálculo do vetor de características é limitado em número de pacotes. Assim, a remoção de pacotes antes do cálculo da FFT reduz o gasto de recursos computacionais, agiliza o processo de extração de características e diminui a complexidade final da proposta.

A segunda estratégia para redimensionar o vetor envolve calcular a FFT de todo o fluxo e dividir as componentes geradas em subconjuntos após o cálculo, como mostra a Figura 3(e). A estratégia preserva a amplitude das componentes do espectro total, o que facilita a distinção de fluxos na classificação. Entretanto, é necessário processar todo o arquivo de captura para extrair as características, além do alto custo computacional conforme a quantidade de pacotes aumenta. Esses dois fatores tornam a estratégia inviável para processamento de fluxos em tempo-real. A proposta deste artigo adota a primeira estratégia, que reduz a quantidade de pacotes para s anteriormente à decomposição espectral. Dessa forma, a proposta é capaz de extrair características de maneira ágil e permite o processamento em tempo real, pois o cálculo do vetor de características utiliza um número limitado de pacotes por janela.

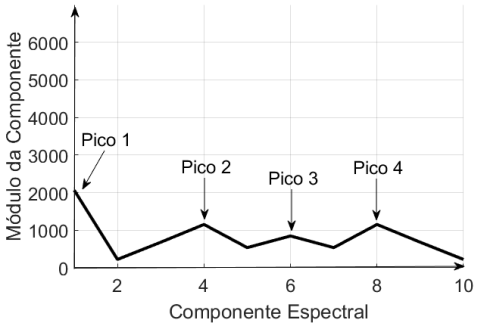
A primeira estratégia para igualar o tamanho dos vetores quando o número de pacotes N é menor do que s consiste em adicionar zeros ao vetor original e calcular a FFT. Uma segunda estratégia para essa etapa consiste em calcular a FFT do vetor original e adicionar zeros ao resultado. O resultado da primeira estratégia, na Figura 3(d), exibe que adicionar conteúdo antes do cálculo da FFT gera ruído no espectro, o que pode prejudicar o desempenho do classificador, além de introduzir gastos computacionais desnecessários. Por outro lado, a segunda estratégia mantém o espectro do fluxo inalterado, uma vez que apenas elementos nulos são adicionados ao espectro inicial, como exibido na Figura 3(f).



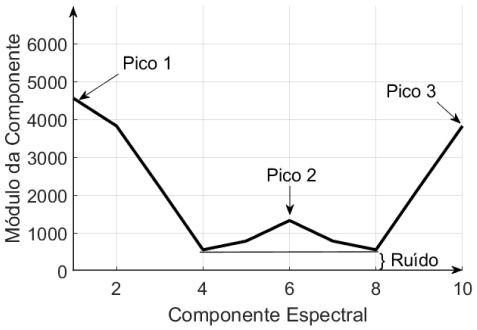
(a) Espectro do fluxo que contém 50 pacotes, que é maior do que o tamanho da janela $s = 10$.



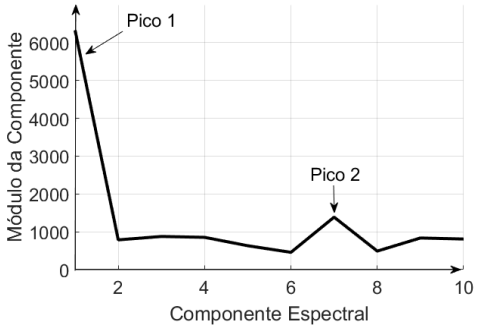
(b) Espectro do fluxo que contém 6 pacotes, que é menor do que o tamanho da janela $s = 10$.



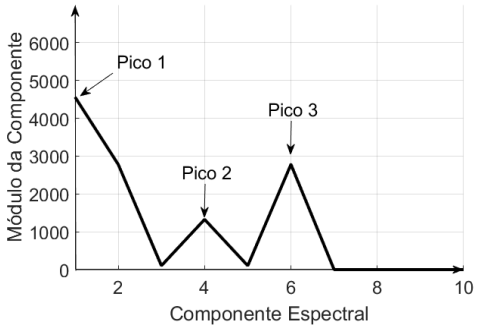
(c) Espectro do fluxo utilizando os 10 primeiros pacotes para o cálculo da FFT.



(d) Espectro do fluxo adicionando 4 pacotes nulos antes do cálculo da FFT.



(e) Espectro do fluxo utilizando as 10 primeiras componentes após o cálculo da FFT.



(f) Espectro do fluxo adicionando 4 componentes nulas após o cálculo da FFT.

Figura 3. Espectro de dois fluxos com tamanhos diferentes do estabelecido na janela de tamanho *s* = 10 no exemplo. Como o número de componentes espectrais dos fluxos deve ser uniforme, são empregadas as estratégias em 3(c) e 3(e) para a redução de componentes da Figura 3(a) e as estratégias em 3(d) e 3(f) para o preenchimento do espectro na Figura 3(b). As Figuras 3(c) e 3(f) demonstram que o emprego das estratégias antes da FFT retém mais informação do espectro original.

Além disso, a inserção de elementos no vetor possui menor impacto no tempo de execução do que a alternativa anterior e, portanto, a segunda estratégia é adotada nesse caso.

Após a análise de dimensão do vetor referente ao fluxo, suas componentes espectrais são calculadas. Os coeficientes da transformada discreta de Fourier são obtidos

através da Equação 1:

$$X[k] = \sum_{n=0}^{N-1} e^{-2\pi j \frac{kn}{N}} x[n], \tag{1}$$

onde n é o índice da amostra, j representa a unidade complexa, N é a quantidade de amostras presentes no cálculo da transformada e k é o índice da componente espectral resultante da transformada de Fourier. Portanto, $x[n]$ é a n-ésima amostra da sequência no domínio do tempo, enquanto $X[k]$ é a k-ésima componente complexa da sequência no domínio da frequência. A representação da frequência pode ser explicitada através da fórmula de Euler para a exponencial complexa, exibida na Equação 2:

$$X[k] = \sum_{n=0}^{N-1} \left[\cos\left(2\pi \frac{kn}{N}\right) - j \sin\left(2\pi \frac{kn}{N}\right) \right] x[n]. \tag{2}$$

Como as k componentes calculadas são complexas, o vetor $X[k]$ é transformado em um vetor $X'[k]$ que é um vetor real cujo o conteúdo é o módulo de cada componente complexa de $X[k]$. Por fim, o resultado da extração de características é enviado para o classificador que retorna a classe prevista para o fluxo.

3.2. Análise de Complexidade da Proposta

O algoritmo para o cálculo da transformada discreta de Fourier (DFT) para uma janela com N pacotes possui complexidade igual a $O(N^2)$. A complexidade de execução para um vetor com d características é igual a $O(d.N^2)$ para o cálculo da transformada em cada dimensão do vetor. A complexidade para processar o total de f fluxos, com N pacotes e d características por amostra é dada por $O(f.d.N^2)$. Uma complexidade dessa ordem de grandeza dificulta a detecção de ameças em tempo real para cenários com grandes volumes de dados, como *Big Data*. Assim, a primeira solução para reduzir a complexidade total da proposta é utilizar a transformada rápida de Fourier (FFT) para obter o vetor com as componentes espectrais do fluxo. O uso da FFT reduz a complexidade do cálculo da DFT de $O(N^2)$ para $O(N \log(N))$ quando há N pacotes, utilizando a estratégia de dividir para conquistar. A segunda ideia para reduzir o tempo de processamento da proposta é analisar uma única característica, um vetor que contém a quantidade de bytes presentes em cada pacote da janela, para o cálculo da transformada de Fourier. Por fim, estabelecendo um tamanho fixo s para a janela utilizada para extrair características, o cálculo da FFT é restrito a $\lceil \frac{N}{s} \rceil$ janelas por fluxo. Portanto, a complexidade final da proposta para extração de características para f fluxos contendo N pacotes é dada por $O(f.\frac{N}{s}.s \log(s))$. O artigo fixa o número de pacotes em uma janela o valor $s = 10$. Assim, com s constante, a complexidade assintótica do tempo de execução é dada por $O(f.N)$.

4. Desenvolvimento e Avaliação de um Protótipo do FENED

O método de extração de características proposto foi implementado em Python v3.7.7 e testado em modelos criados com a biblioteca scikit-learn v0.24.0[5]. Os experimentos foram realizados em um servidor Intel Xeon E5-2650 CPU 2.00 GHz com 16 núcleos de processamento e 252 GB de RAM.

[5]O scikit-learn [Pedregosa et al. 2011] é uma biblioteca de código aberto, bem documentada, para a criação de modelos de aprendizado de máquina, que possui um grande número de desenvolvedores. Disponível em https://scikit-learn.org/0.24/

Tabela 1. Rótulos normal e malicioso dos fluxos tipo flowtbag e tipo fluxos s_{pkt} extraídos do conjunto de dados, quando aplicados aos 13 cenários do conjunto de dados CTU-13.

	Rede de Robôs			fluxos flowtbag		fluxos s_{pkt}	
Cenário	Ataques	Robô	Nós$_{inf}$ [†]	Normal	Malicioso	Normal	Malicioso
1	SPAM e fraude de clique	Neris	1	23729	4218	121696	21744
2	SPAM e fraude de clique	Neris	1	6805	4642	38657	21185
3	Varredura de porta	Rbot	1	46866	5	154543	34893
4	Negação de serviço distribuída	Rbot	1	19906	72	123927	5951
5	SPAM e varredura de porta	Virut	1	4209	411	11289	2067
6	varredura de porta	Menti	1	6153	209	70386	4634
7	inf[‡]	Sogou	1	1460	39	2242	714
8	Varredura de porta	Murio	1	4589	351	72058	9375
9	SPAM, fraude de clique e varredura de porta	Neris	10	23393	37082	156174	120402
10	Negação de serviço distribuída	Rbot	10	5512	42	114674	128410
11	inf[‡]	Rbot	3	496	9	1367	258
12	inf[‡]	NSIS.ay	3	6368	1203	30684	10092
13	SPAM e varredura de porta	Virut	1	20914	6111	75723	37536
Todos	-	-	35	170400	54394	973420	397261

[†] Quantidade de dispositivos infectados.
[‡] O cenário contém apenas os padrões de comunicação da rede de robôs.

Os experimentos dispõem do conjunto de dados CTU-13 [Garcia et al. 2014] que possui amostras de tráfego malicioso gerado através de máquinas infectadas por redes de robôs e tráfego normal. O conjunto de dados permite avaliar os comportamentos periódicos de múltiplas redes de robôs em 13 cenários distintos, como mostra a Tabela 1.

O classificador selecionado para os experimentos é a árvore de decisão, pois é um dos mais populares algoritmos de aprendizado supervisionado para a classificação de fluxos e que apresenta bons resultados de desempenho [Guimarães et al. 2020, de Souza et al. 2020]. Além disso, os hiperparâmetros da árvore de decisão foram otimizados para as características extraídas com o flowtbag. Uma busca em grade utilizando valores mais frequentes de hiperparâmetros para a árvore de decisão e considerando a F1-score como métrica de otimização foi realizada para selecionar o melhor conjunto de hiperparâmetros para a aplicação de classificação de fluxos. Entretanto, entre todos os hiperparâmetros de árvore de decisão, apenas o hiperparâmetro de profundidade da árvore apresentou valor diferente em relação ao valor padrão, que a otimização definiu com o valor de profundidade igual a 23. A otimização dos demais hiperparâmetros influenciam pouco o desempenho de classificação [Mantovani et al. 2016] e mesmo com a busca em grade permaneceram com os valores padrões. Os resultados dos experimentos são exibidos com o valor médio das métricas de classificação com um intervalo de confiança de 95%.

Os fluxos são definidos pela quíntupla (IP origem, IP destino, Porta Origem, Porta destino, protocolo). Nas conexões TCP, o flowtbag monitora o estado da conexão para definir o início, através do sinal SYN, e o fim de um fluxo, através do sinal FIN, retornando o vetor de características do fluxo. Esta divisão em fluxo não é atrativa para a construção de um vetor para o cálculo da FFT, por possuir um número variável de pacotes e um tamanho variável de pacotes em um fluxo. Para o cálculo simples da FFT, o melhor seria um vetor com uma dimensão fixa, correspondendo a um número s de pacotes, do tamanho de cada pacote. Para as conexões TCP, o início do fluxo também corresponde a um pacote

com o sinal SYN e é dividido em janelas de s pacotes. Doravante este fluxo é chamado fluxo s_{pkt}. Assim, um fluxo s_{pkt} com N pacotes na proposta FENED corresponde a $\lceil \frac{N}{s} \rceil$ janelas de pacotes que são convertidas em um vetor de dimensão s por janela. Os dois métodos de extração de características mantêm a distribuição entre fluxos, ou fluxos s_{pkt}, normal e maliciosos próximas, com uma diferença menor do que 5%. A Tabela 1 mostra as quantidades dos rótulos de tráfegos normais e maliciosos para diferentes ataques de robôs para os 13 cenários diferentes do conjunto de dados CTU-13 com a diferença entre o número de fluxos do flowtbag e do número do s_{pkt}.

A proposta é analisada por três experimentos objetivando: (i) determinar a dimensão do vetor de características que proporciona o maior desempenho de sensibilidade e menor tempo de processamento; (ii) avaliar o desempenho do classificador utilizando fluxos flowtbag e os fluxos s_{pkt}; e (iii) comparar o tempo de extração de características dos dois métodos. O Experimento (i) obtém o melhor tamanho da janela s para o vetor de características extraído, enquanto os Experimentos (ii) e (iii) comparam a proposta com o método convencional de extração de características. A métrica de sensibilidade foi escolhida para a análise por refletir a capacidade do classificador detectar os ataques.

O Experimento (i) analisa o desempenho de classificação conforme varia a dimensão do vetor de características. O experimento define o vetor de características espectrais com a menor dimensão que produz a melhor sensibilidade de classificação, a fim de reduzir a complexidade computacional sem prejuízos na classificação de fluxos maliciosos. A Figura 4 ilustra o resultado da sensibilidade quando a dimensão do vetor de características varia. O experimento mostrou que a capacidade de detectar os fluxos maliciosos não variou significativamente ao aumentar a dimensão do vetor de características acima de 10 dimensões. Assim, a fixação do vetor de características em dimensão 10 reduz a complexidade computacional ao mesmo tempo que atinge uma alta detecção dos ataques.

O Experimento (ii) avalia o desempenho da proposta para diferentes cenários do conjunto de dados e compara a proposta com a alternativa do flowtbag. A Tabela 2 exibe os resultados para o experimento para os dois métodos de extração de características avaliados para o algoritmo de árvore de decisão utilizando uma divisão aleatória de 70% dos dados para treino e o 30% para o teste. A dimensão do vetor de características extraído

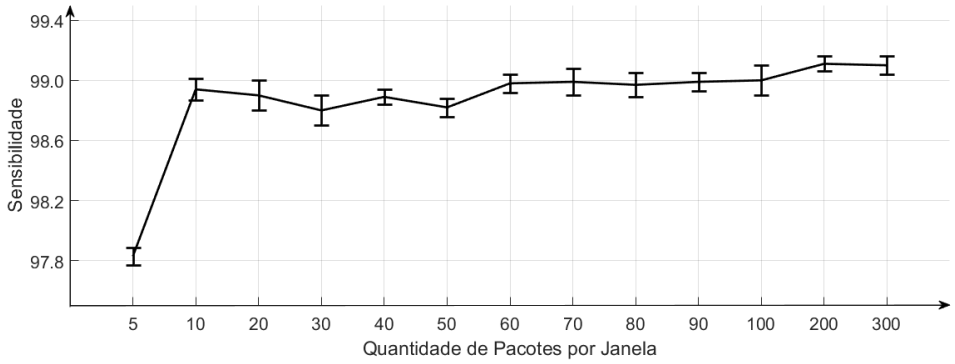


Figura 4. Métrica de sensibilidade da árvore de decisão em função da quantidade de pacotes utilizadas para extração do vetor de características.

com o FENED é igual a 10, pois é o valor da dimensão do vetor de características obtido no Experimento (i) que apresenta o melhor custo benefício.

Tabela 2. Resultados de desempenho de classificação para diferentes métricas obtidos para os dois métodos de extração de características para diferentes métricas de desempenho, avaliados para diferentes cenários do conjunto de dados CTU-13. O sistema proposto apresenta melhores resultados que o flowtbag em diversos cenários nas métricas destacadas em negrito.

	Acurácia (%)		Precisão (%)		Sensibilidade (%)		F1-Score (%)	
Cenário	flowtbag	FENED	flowtbag	FENED	flowtbag	FENED	flowtbag	FENED
1	94,5 ± 0,1	**99,01 ± 0,03**	85,4 ± 0,6	**96,1 ± 0,2**	76,5 ± 0,6	**97,7 ± 0,1**	80,7 ± 0,5	**96,9 ± 0,1**
2	99,71 ± 0,07	98,76 ± 0,05	99,7 ± 0,1	98,01 ± 0,08	99,6 ± 0,2	98,7 ± 0,1	99,64 ± 0,09	98,34 ± 0,06
3	99,99 ± 0,01	98,63 ± 0,03	70 ± 30	**99,23 ± 0,06**	70 ± 20	**92,7 ± 0,2**	70 ± 20	**95.88 ± 0.09**
4	99,91 ± 0,03	99,12 ± 0,02	88 ± 5	**99,75 ± 0,04**	90 ± 3	88,0 ± 0,2	89 ± 4	**93.5 ± 0.2**
5	99,2 ± 0,2	97,45 ± 0,07	95 ± 1	89,4 ± 0,3	96 ± 2	95,5 ± 0,3	95.4 ± 0.8	92.3 ± 0.2
6	99,82 ± 0,05	99,70 ± 0,01	97 ± 1	97,4 ± 0,4	98 ± 1	97,8 ± 0,5	97.3 ± 0.7	97.6 ± 0.2
7	99,2 ± 0,2	97,8 ± 0,2	88,3 ± 0,4	**97,6 ± 0,4**	82,8 ± 0,6	**95,6 ± 0,6**	84.8 ± 0.3	**96.6 ± 0.3**
8	99,6 ± 0,2	99,54 ± 0,02	98 ± 1	97,6 ± 0,2	96 ± 1	**99,0 ± 0,2**	97 ± 1	**98.31 ± 0.07**
9	95,56 ± 0,09	**98,58 ± 0,02**	97,30 ± 0,07	**97,50 ± 0,03**	95,4 ± 0,2	**99,45 ± 0,01**	96.34 ± 0.08	**98.46 ± 0.02**
10	99,89 ± 0,03	99,43 ± 0,01	87,9 ± 0,5	**99,21 ± 0,01**	97,0 ± 0,2	**99,95 ± 0,03**	92 ± 0.3	**99.579 ± 0.007**
11	99,4 ± 0,6	88,6 ± 0,5	40 ± 20	**70 ± 10**	70 ± 30	60 ± 20	50 ± 20	50 ± 10
12	99,1 ± 0,2	97,97 ± 0,07	97,5 ± 0,5	96,4 ± 0,3	97 ± 1	97,1 ± 0,3	97.0 ± 0.8	96.7 ± 0.1
13	97,8 ± 0,1	**99,35 ± 0,02**	95,8 ± 0,4	**98,76 ± 0,06**	94,4 ± 0,3	**98,94 ± 0,07**	95.1 ± 0.2	**98.85 ± 0.04**
Todos	97,22 ± 0,03	**97,77 ± 0,01**	96,0 ± 0,1	95,81 ± 0,02	92,4 ± 0,2	**97,70 ± 0,03**	94.15 ± 0.07	**96.75 ± 0.01**

O método flowtbag para extração de características gera uma maior acurácia na maioria dos cenários. Entretanto, a classificação gera um maior número de falsos negativos, pois a sensibilidade foi significativamente menor do que o método proposto. Sabe-se que a acurácia não é uma medida adequada quando os conjuntos de dados estão desbalanceados, pois a classificação de diversas amostras como a classe majoritária produz uma maior acurácia sem penalizar significativamente os erros na classificação de amostras pertencentes à classe minoritária [Batista et al. 2004]. Por outro lado, o método proposto é capaz de detectar um maior número de ataques e mantém um baixo número de falsos positivos, pois sua acurácia e sensibilidade são altas. A proposta FENED de características espectrais apresenta um aumento maior do que 2% na detecção de ataques automáticos de negação de serviço distribuído realizados por redes de robôs, presentes no cenário 10. A representação no domínio da frequência dos pacotes gerados pela negação de serviço distribuída facilita a detecção dos ataques. A varredura de porta presente nos cenários 6, 8, 9 e 13 também é facilmente detectada através de decomposição espectral, pois seu comportamento é automatizado, gerando componentes periódicas com alta energia. A decomposição espectral também aumentou a detecção de ataques de SPAM e fraude de clique (*click-fraud*) nos cenários 1, 9 e 13.

Por fim, o Experimento (iii) compara a latência da proposta FENED com a aplicação flowtbag para a extração de características e comprova a simplicidade computacional da proposta. A comparação é simulada no MATLAB para mitigar as diferenças causadas pelas implementações em linguagens de programação distintas. A simplicidade do FENED se justifica pela utilização de uma janela fixa para extrair características espectrais utilizando a quantidade de bytes em cada pacote, enquanto o flowtbag extrai um vetor de 40 características convencionais que contém informações como a quantidade de pacotes e a quantidade de bytes. A estratégia de limitar a quantidade de pacotes por fluxo reduz a complexidade da proposta para $O(f.N)$, que é linear em relação ao número de

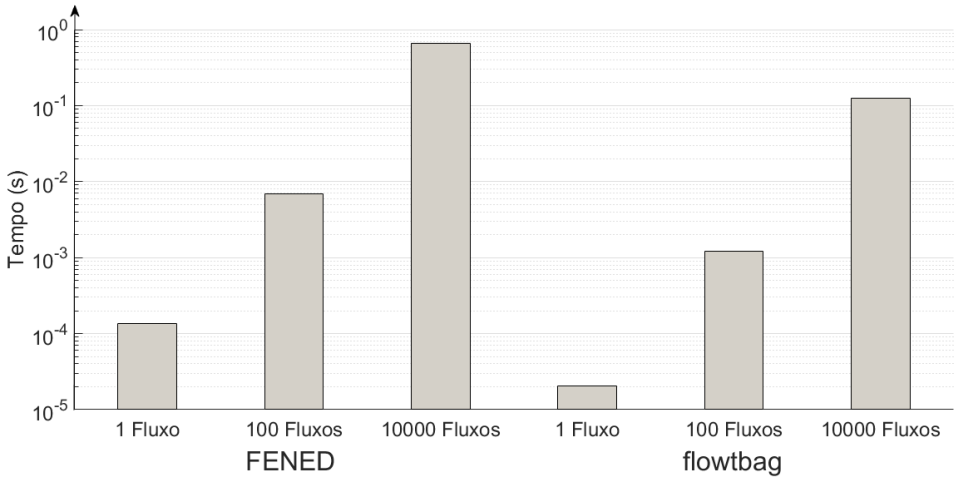


Figura 5. Comparação entre os métodos avaliados para extrair características de fluxos. Ambos possuem complexidade linear para o caso em que todos os fluxos possuem a mesma quantidade de pacotes.

fluxos f quando o número N de pacotes por fluxo é fixo. O tempo necessário para extrair as características de um único fluxo com a proposta FENED é aproximadamente 100 μs, enquanto no flowtbag é aproximadamente 10 μs. Entretanto, a Figura 5 comprova que o tempo varia linearmente para os casos em que a proposta FENED e a aplicação flowtbag extraem características de 100 e 10000 fluxos. Portanto, a proposta possui complexidade linear em relação ao número de fluxos para o caso em que todos os fluxos possuem o mesmo número de pacotes, assim como a aplicação flowtbag.

5. Conclusão

O artigo avalia a eficácia do uso de técnicas de processamento de sinais para detectar ameaças de rede produzidas de forma automática por um rede de robôs. A proposta usa a transformada rápida de Fourier para gerar um vetor de características que expõem a dependência temporal entre os pacotes pertencentes a um fluxo. A visualização do espectro resultante demonstra que os fluxos normais possuem representações frequenciais equivalentes ao ruído branco, enquanto os ataques de redes de robôs desviam desse padrão. Essa forma de representar os fluxos permite a identificação correta de ataques automatizados e periódicos, como ataques de negação de serviço distribuída, varredura de porta e fraudes de clique, característicos de ataques produzidos por redes de robôs. O método de extração de características é agnóstico ao algoritmo de aprendizado de máquina utilizado, pois produz vetores de dimensões iguais e definidos previamente. Além disso, a privacidade dos usuários é preservada, pois o método proposto utiliza apenas o tamanho dos pacotes, não revelando o conteúdo do pacote e informações sensíveis pertencentes aos usuários. Por fim, os resultados demonstram que fixar a quantidade de pacotes inspecionados por janela não altera significativamente as métricas de classificação. Logo, a complexidade da proposta é reduzida a $O(f.N)$, quando há f fluxos com N pacotes por fluxo, para o caso em que o cálculo da FFT utiliza janelas de tamanho fixo s. Portanto, a proposta de extração de características possui a mesma complexidade dos métodos convencionais de extração de características e análise de rede, sendo ágil e permitindo o processamento de fluxos em tempo real. Em trabalhos futuros, a proposta será avaliada em outros conjuntos de dados.

Referências

Aiello, M. et al. (2014). An on-line intrusion detection approach to identify low-rate DoS attacks. In *2014 International Carnahan Conference on Security Technology (ICCST)*, pages 1–6.

AsSadhan, B. and Moura, J. M. (2014). An efficient method to detect periodic behavior in botnet traffic by analyzing control plane traffic. *Journal of advanced research*, 5(4):435–448.

Batista, G. E., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29.

Bezerra, V. et al. (2018). Providing IoT host-based datasets for intrusion detection research. In *Anais do XVIII SBSeg*, pages 15–28.

Bian, H. et al. (2019). Host in danger? detecting network intrusions from authentication logs. In *15th International Conference on Network and Service Management (CNSM)*, pages 1–9. IEEE.

Blaise, A., Bouet, M., Conan, V., and Secci, S. (2020). Botfp: Fingerprints clustering for bot detection. In *NOMS IEEE/IFIP Network Operations and Management Symposium*, pages 1–7. IEEE.

Bottazzi, G. et al. (2016). Frequency domain analysis of large-scale proxy logs for botnet traffic detection. In *Proceedings of the 9th International Conference on Security of Information and Networks*, pages 76–80.

Camilo, G. F. et al. (2020). Autavailchain: Automatic and secure data availability through blockchain. In *GLOBECOM*, pages 1–6. IEEE.

Chen, Y. and Hwang, K. (2007). Spectral analysis of TCP flows for defense against reduction-of-quality attacks. In *IEEE International Conference on Communications*, pages 1203–1210.

Chimedtseren, E. et al. (2014). Intrusion detection system using Discrete Fourier Transform. In *Seventh Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, pages 1–5. IEEE.

de Souza, L. A. C. et al. (2020). DFedForest: Decentralized Federated Forest. In *2020 IEEE Blockchain*, pages 90–97.

Fouladi, R. F., Ermiş, O., and Anarim, E. (2019). Anomaly-Based DDoS Attack Detection by Using Sparse Coding and Frequency Domain. In *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1–6. IEEE.

Garcia, S., Grill, M., Stiborek, J., and Zunino, A. (2014). An empirical comparison of botnet detection methods. *Computers & Security*, 45:100–123.

Guimarães, L. C. et al. (2020). TeMIA-NT: ThrEat Monitoring and Intelligent data Analytics of Network Traffic. In *2020 4th Conference on Cloud and Internet of Things (CIoT)*, pages 9–16. IEEE.

Guzman, J. A. d., Seneviratne, A., and Thilakarathna, K. (2021). Unravelling Spatial Privacy Risks of Mobile Mixed Reality Data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(1):1–26.

Kwon, J. et al. (2014). PsyBoG: Power spectral density analysis for detecting botnet groups. In *2014 9th International Conference on Malicious and Unwanted Software: The Americas (MALWARE)*, pages 85–92. IEEE.

Liu, W., Liu, X., Di, X., and Qi, H. (2019). A novel network intrusion detection algorithm based on Fast Fourier Transformation. In *2019 1st International Conference on Industrial Artificial Intelligence (IAI)*, pages 1–6.

Lobato, A., Lopez, M. A., Rebello, G., and Duarte, O. (2017). Um sistema adaptativo de detecção e reação a ameaças. *Anais do XVII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais-SBSeg*, 17:400–413.

Manasrah, A. M., Domi, W. B., and Suppiah, N. N. (2020). Botnet detection based on DNS traffic similarity. *International Journal of Advanced Intelligence Paradigms*, 15(4):357–387.

Mantovani, R. G. et al. (2016). Hyper-parameter tuning of a decision tree induction algorithm. In *2016 5th BRACIS*, pages 37–42.

Pedregosa, F. et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pelloso, M. et al. (2018). A self-adaptable system for DDoS attack prediction based on the metastability theory. In *IEEE GLOBECOM*, pages 1–6.

Possebon, I. P., Silva, A. S., Granville, L. Z., Schaeffer-Filho, A., and Marnerides, A. (2019). Improved network traffic classification using ensemble learning. In *2019 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6. IEEE.

Powell, B. A. (2019). Malicious Overtones: Hunting data theft in the frequency domain with one-class learning. *Transactions on Privacy and Security (TOPS)*, 22(4):1–34.

Sagirlar, G., Carminati, B., and Ferrari, E. (2018). AutoBotCatcher: Blockchain-Based P2P Botnet Detection for the Internet of Things. In *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, pages 1–8.

Sanz, I. J. et al. (2018). Um sistema de detecção de ameaças distribuídas de rede baseado em aprendizagem por grafos. In *Anais do XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. SBC.

Viegas, E., Santin, A., Bessani, A., and Neves, N. (2019). BigFlow: Real-time and reliable anomaly-based intrusion detection for high-speed networks. *FGCS*, 93:473–485.

Yu, X. et al. (2009). Online botnet detection by continuous similarity monitoring. In *2009 International Symposium on Information Engineering and Electronic Commerce*, pages 145–149. IEEE.

Zhou, M. and Lang, S.-D. (2003). A frequency-based approach to intrusion detection. In *Proc. of the Workshop on Network Security Threats and Countermeasures*.