Detecção de botnets baseada na análise de fluxos de rede utilizando estatística inversa

Daniele A. G. Lopes[1], **Marcelo A. Marotta**[1], **Marcelo Ladeira**[1], **João J. C. Gondim**[1]

[1]Departamento de Ciência da Computação – Universidade de Brasília (UnB)
Caixa Postal 4466 – 70.919-970 – Brasília, DF – Brasil

daniele.lopes@aluno.unb.br, {marcelo.marotta, mladeira, gondim}@unb.br

Abstract. *Botnet is a network of infected computers, which are remotely controlled by a cybercriminal, called botmaster, which aims to carry out massive cyberattacks, such as DDoS, SPAM and, information theft. Traditional botnet detection methods, usually signature-based, are unable to detect unknown botnets. The behavior-based analysis is promising for detecting current botnet trends, which are constantly evolving. This article proposes a botnet detection mechanism based on the analysis of network flow behavior. The technique used to detect botnets was recently developed and is called Energy-based Flow Classifier (EFC). This technique uses inverse statistics to detect anomalies. Two heterogeneous datasets, CTU-13 and ISOT HTTP were used to evaluate the efficiency of the generated model and the results were compared with several traditional classifiers, of one and two classes. The results obtained show that the EFC obtained more stable results, regardless of the domain, unlike the other tested algorithms.*

Resumo. *Botnet é uma rede de computadores infectados, os quais são controlados remotamente por um cibercriminoso, denominado botmaster e que tem como objetivo realizar ataques cibernéticos massivos, como DDoS, SPAM e roubo de informações. Os métodos tradicionais de detecção de botnets, normalmente baseados em assinatura, são incapazes de detectar botnets desconhecidas. A análise baseada em comportamento tem sido promissora para a detecção de tendências atuais de botnets, as quais estão em constante evolução. Este artigo propõe um mecanismo de detecção de botnet baseado na análise do comportamento de fluxo de rede. A técnica utilizada para detecção de botnets foi recentemente desenvolvida e é denominada Energy-based Flow Classifier (EFC). Essa técnica utiliza estatística inversa para detecção de anomalias. Dois conjuntos de dados heterogêneos, CTU-13 e ISOT HTTP foram utilizados para avaliar a eficiência do modelo gerado e os resultados foram comparados com diversos classificadores tradicionais, de uma e de duas classes. Os resultados obtidos mostram que o EFC obteve resultados mais estáveis, independente do domínio, ao contrário dos demais algoritmos testados.*

1. Introdução

Uma *botnet* é uma rede formada por inúmeros dispositivos infectados por algum *malware*, os quais são denominados *bots* ou zumbis e que são controlados por um atacante, denominado *botmaster* [Zhao et al. 2013]. O objetivo de uma *botnet* é realizar atividades maliciosas com base nas instruções fornecidas pelo *botmaster*. O principal componente de uma

botnet é o servidor de Comando e Controle (C&C), por ser o meio pelo qual o *botmaster* controla e envia instruções aos *bots*, iniciando vários tipos de ataques cibernéticos, como negação de serviço distribuída (DDoS), spam, *phishing* e roubo de informações [Ibrahim et al. 2021]. A estrutura do canal de C&C pode ser centralizada, na qual um servidor C&C central é responsável por enviar comandos aos *bots* ou descentralizada (P2P), onde os dispositivos infectados atuam como *bots* e como servidores de C&C ao mesmo tempo [Silva et al. 2013].

O potencial destrutivo das *botnets* têm aumentado exponencialmente com o avanço da tecnologia da Internet das Coisas (IoT) e à medida que aumenta o número de usuários e dispositivos conectados [Council to Secure the Digital Economy 2019]. Em 2016 a *botnet* Mirai foi responsável por um dos maiores ataques de negação de serviço distribuído já registrado até hoje, estimado em 1,2 Tbps (terabits por segundo). Este ataque deixou fora do ar sites como Twitter, Netflix, CNN e vários outros pela Europa e Estados Unidos [Wainwright and Kettani 2019], [Antonakakis et al. 2017]. Com a disponibilização do código-fonte da Mirai na Internet, muitos projetos variantes têm surgido. Em 2019, por exemplo, o número de variantes da *botnet* Mirai teve um crescimento de 57% em relação à 2018, ultrapassando 225.000 ocorrências [European Union Agency for Network and Information Security 2020].

Uma vez que os métodos tradicionais de detecção de *botnets* são baseados em assinaturas, estes se tornam eficientes para detectar tipos de *botnets* já conhecidos. Contudo, novos tipos de *botnets* ou de variantes de *botnets* conhecidas surgem cotidianamente e sua detecção é um grande desafio para os métodos tradicionais focados em assinaturas de ataques já conhecidos [Zhao et al. 2013]. Além disso, as *botnets* evoluem constantemente, alterando sua arquitetura e protocolos utilizados, com o intuito de evitar a detecção por sistemas de segurança. Adicionalmente, as *botnets* utilizam cada vez mais técnicas de criptografia e ofuscação, dificultando ainda mais a detecção [Ibrahim et al. 2021]. À medida que as *botnets* têm progredido e se tornado mais complexas, várias estratégias de detecção de *botnet* têm sido propostas, principalmente utilizando métodos de aprendizado de máquina para análise de comportamento e detecção de anomalias [Vormayr et al. 2017].

No contexto de detecção de ataques por *botnets*, a maioria dos métodos diferenciam-se no tipo de análise realizada, sendo elas (*i*) análise profunda de pacotes ou (*ii*) análise de fluxos. Na primeira, os pacotes são individualmente analisados considerando seu cabeçalho e os dados sendo transportados (*payload*). Na segunda, um conjunto de pacotes são agrupados de acordo com características comuns presentes em seus cabeçalhos, sendo chamados de fluxos, os quais são avaliados de acordo com essas características e métricas estatísticas, como número de bytes e tempo de duração médio. A análise de fluxos possui algumas vantagens em relação à análise profunda de pacotes, principalmente o fato de consumir menos recurso computacional, uma vez que só processa o cabeçalho dos pacotes e também o fato de possibilitar a detecção de *botnets* que utilizam técnicas de criptografia ou ofuscação, já que não requer acesso ao *payload* do pacote, o qual pode estar criptografado [Ibrahim et al. 2021]. Sendo assim, nosso estudo será baseado na análise de fluxos de rede.

Muitas abordagens têm sido propostas, nos últimos anos, para detecção de *botnets* baseadas em fluxo de rede utilizando técnicas de aprendizado de máquina [Ibrahim et al. 2021], [Yadav and Thakur 2020], [Zhao et al. 2013]. Porém, algumas

técnicas são feitas especialmente para protocolos e estruturas específicas, sendo incapazes de detectar *botnets* que utilizem protocolos ou estruturas diferentes [Saad et al. 2011], [Khan et al. 2019], [Alauthaman et al. 2018]. Além disso, a maioria dos trabalhos utiliza algoritmos convencionais de aprendizado de máquina, como *Support Vector Machine* (SVM) e *Naive Bayes* (NB) [García et al. 2014b]. Estes algoritmos são baseados no aprendizado a partir de um conjunto de treinamento que contenha amostras das duas classes (fluxos benignos e fluxos maliciosos). Sendo assim, apenas as *botnets* encontradas durante o treinamento ou que tenham comportamento muito parecido serão detectadas, limitando o objetivo de detectar *botnets* desconhecidas. Consequentemente, a maioria das abordagens existentes não se adaptam bem a diferentes domínios, *i.e.,* o desempenho é reduzido quando treinados em um conjunto de dados específico e avaliados em outro conjunto de dados relacionado [Li et al. 2019], [Zolanvari et al. 2018]. Por fim, a obtenção de amostras de fluxos maliciosos que representem *malwares* recentes para compor o conjunto de treinamento é uma tarefa dispendiosa e inviável para largas redes, mas ao mesmo tempo cruciais para tornar os modelos atuais baseados em duas classes mais eficientes [Saad et al. 2011].

Em um estudo recente [Pontes et al. 2021] foi desenvolvido um novo classificador, denominado *Energy-Based Flow Classifier* (EFC), o qual foi inspirado no modelo inverso de *Potts* da mecânica quântica e adaptado para classificação de fluxos de rede. O EFC é um algoritmo que realiza classificação unária utilizando apenas dados benignos para realizar o treinamento e não precisa conhecer o comportamento do tráfego malicioso para realizar a detecção de anomalias, contornando assim, o problema da dificuldade de se obter amostras maliciosas rotuladas. Além disso, o EFC é um classificador intrinsecamente adaptável a diferentes domínios, uma vez que a inferência do modelo é baseada apenas em amostras benignas [Pontes et al. 2021]. Devido a essa característica, o EFC parece ser um classificador promissor para detecção de novos tipos de *botnets* ou mesmo variantes de *botnets* conhecidas, mas que ainda não foi explorado no trabalho de Pontes [Pontes et al. 2021].

Sendo assim, nossa proposta consiste em avaliar o emprego do algoritmo *Energy-Based Flow Classifier* (EFC) para detecção de *botnets* através da análise de fluxos de rede, propondo uma abordagem que seja capaz de detectar novos tipos de *botnets*, independente da estrutura ou protocolos utilizados. Para avaliar a eficiência do modelo, serão utilizados dois conjuntos de dados heterogêneos (CTU-13 e ISOT HTTP). Será realizada também a comparação de desempenho do EFC com classificadores tradicionais de uma e de duas classes. Nossos resultados evidenciaram que o EFC se mostrou mais robusto e menos sensível a mudanças na distribuição de dados do que os outros algoritmos. Nossas principais contribuições são:

- Uma análise exploratória para detecção de *botnets* utilizando o algoritmo EFC;
- Uma comparação do desempenho do EFC com classificadores clássicos de uma e de duas classes usando dois conjuntos de dados diferentes;
- Uma análise da adaptabilidade dos diferentes classificadores quando testados em um domínio diferente daquele onde foi realizado o treinamento.

O restante deste trabalho está organizado como segue. A Seção 2 apresenta os trabalhos relacionados ao tema de pesquisa. A Seção 3 apresenta a metodologia e os detalhes para entendimento do EFC e ainda descreve os conjuntos de dados utilizados. A Seção 4 apresenta os resultados obtidos. Por fim, a Seção 5 conclui o trabalho e direciona os trabalhos futuros.

2. Trabalhos Relacionados

Muitas abordagens têm sido propostas, nos últimos anos, para detecção de *botnets* baseada na análise de fluxos de rede utilizando aprendizado de máquina. Várias dessas abordagens foram desenvolvidas considerando um tipo de protocolo específico, assim como nos trabalhos de [Saad et al. 2011], [Alauthaman et al. 2018] e [Khan et al. 2019], os quais focaram na detecção de *botnets* P2P. No trabalho de [Saad et al. 2011] foi proposto um *framework* para detecção de *botnets* que possui duas fases. A primeira implementa o pré-processamento do tráfego de rede, extraindo um amplo conjunto de atributos para cada fluxo. A segunda fase implementa modelos supervisionados para classificar os fluxos em tráfego não P2P, tráfego P2P malicioso e tráfego P2P normal. Foram testados os seguintes algoritmos: SVM (*Support Vector Machine*), ANN (*Artificial Neural Network*), classificador de vizinhos mais próximos, classificador baseado em Gauss e o classificador *Naive Bayes*. Os autores concluíram que todas as cinco técnicas fornecem taxa de detecção maior que 89%, porém ANN e SVM requerem mais tempo para serem treinados e para realizar a classificação. Segundo os próprios autores, a abordagem proposta não é capaz de se adaptar às mudanças no tráfego de rede, nem de detectar novos tipos de *botnets* [Saad et al. 2011]. Diferentemente, nossa abordagem visa detectar *botnets* independente do protocolo utilizado, além de buscar a detecção de *botnets* desconhecidas.

Outras pesquisas abordaram a detecção de *botnets* baseada na estrutura do canal de C&C, como foi feito no trabalho de [Gadelrab et al. 2018], onde os autores propuseram um modelo de detecção de *botnet* denominado BotCap, utilizando os algoritmos SVM e J48 para treinar o modelo. O conjunto de dados foi gerado pelos autores, possuindo no total seis famílias de *botnets*, todas de arquitetura centralizada (HTTP e IRC), sendo assim, a detecção de *botnets* P2P não foi considerada. O nosso trabalho utiliza dois conjuntos de dados disponíveis publicamente (CTU-13 e ISOT HTTP), possuindo *botnets* de arquitetura centralizada e descentralizada e os protocolos HTTP, IRC e P2P.

Vários outros estudos foram desenvolvidos com o objetivo de detectar *botnets* independente do protocolo e da arquitetura do C&C [Bilge et al. 2012] [Ibrahim et al. 2021]. No trabalho de [Bilge et al. 2012] foi proposto um sistema de detecção de servidores de C&C (definidos como um par de IP e porta), independente do protocolo utilizado pelas *botnets*. Os atributos utilizados para detecção foram extraídos a partir dos dados do *Netflow* e foram categorizados em três grupos: baseados no tamanho do fluxo, baseados em padrões de acesso do cliente e baseados em atributos temporais. Os seguintes modelos foram avaliados: *Random Forest*, árvore de decisão J48 e o SVM. Para reduzir a taxa de falsos positivos, foram incorporados uma série de listas de reputação externa (*blacklists*) no procedimento de detecção. A abordagem foi testada em duas redes do mundo real, com uma taxa de identificação de verdadeiro positivo de 65% e uma taxa de falso positivo de 1%.

[Ibrahim et al. 2021] propuseram um *framework* multicamadas para detecção dos servidores de Comando e Controle de *botnets*. A abordagem consiste em dois módulos principais. O primeiro é o módulo de filtragem que possui o objetivo de filtrar e reduzir o tráfego de rede para o segundo módulo, utilizando para isso o algoritmo de clusterização k-*means*. O objetivo do segundo módulo é detectar o servidor de C&C, utilizando para isso algoritmos de classificação. Foram avaliados três classificadores: kNN, SVM e *Multilayer Perceptron*, sendo que o KNN apresentou o melhor resultado com 91,51% de F1-score e uma taxa de falso negativo de 1,5%.

Os algoritmos utilizados nos trabalhos mencionados realizam a classificação do tráfego baseado no aprendizado a partir de amostras das duas classes (benigna e maliciosa). Sendo assim, é necessário conhecer o comportamento malicioso para realizar a detecção, limitando a detecção de *botnets* desconhecidas. Diferentemente, em nosso trabalho vamos utilizar um algoritmo de classificação unária, denominado *Energy-based Flow Classifier* (EFC). Este algoritmo foi proposto por [Pontes et al. 2021] com o objetivo de superar algumas limitações dos algoritmos de aprendizado de máquina, como a necessidade de se obter grandes quantidades de exemplos categorizados, e especialmente o fato de que a maioria desses algoritmos não são facilmente generalizáveis para outros conjuntos de dados, ou seja, o desempenho é reduzido quando treinados em um conjunto de dados específico e avaliados em outro conjunto de dados [Pontes et al. 2021]. Devido aos resultados obtidos por [Pontes et al. 2021] na detecção de anomalias de rede, usaremos o EFC especificamente para a detecção de *botnets*, visando detectar novos tipos de *botnets* ou mesmo variantes de *botnets* conhecidas. Por fim, a maioria dos algoritmos utilizados nos trabalhos relacionados serão implementados para comparação com o EFC.

3. Metodologia

Nesta seção apresentamos os principais conceitos sobre o algoritmo EFC e em seguida descrevemos os dois conjuntos de dados utilizados nos nossos experimentos. Por fim, apresentamos os detalhes do pré-processamento dos conjuntos de dados.

3.1. *Energy-based Flow Classifier* - EFC

O EFC é um classificador que utiliza técnicas de estatística inversa para, durante a fase de treinamento do modelo, inferir uma distribuição de probabilidade para a classe de fluxos a ser detectada, baseado apenas em amostras de fluxos benignos. Na etapa de teste do modelo, a distribuição definida na etapa anterior é usada para classificar novos fluxos calculando e comparando uma medida chamada "energia" do fluxo, a qual mede o quão improvável é a ocorrência de um fluxo na distribuição calculada [Pontes et al. 2021].

Se a energia do fluxo for alta, i.e., acima de um certo limiar, significa que esse fluxo não se assemelha aos fluxos benignos que geraram a distribuição. Da mesma forma, se a energia for baixa, é mais provável que esse fluxo exista na distribuição. O limiar do EFC é definido com base nas amostras das energias de treinamento e pode ser definido dinâmica ou estaticamente. Em nosso estudo de caso, um limiar estatístico definido pelo percentil 95 da energia das amostras de treinamento foi usado. Assim, se um fluxo tem um valor de energia inferior ao 95 percentil das amostras benignas, i.e., abaixo do limiar, é considerado normal. Caso contrário, é classificado como tráfego malicioso. Os detalhes teóricos da inferência do modelo são apresentados em [Pontes et al. 2021].

3.2. Conjuntos de Dados

Os dois conjuntos de dados utilizados nessa pesquisa serão descritos a seguir e foram selecionados por serem popularmente utilizados na literatura devido à relevância e realismo dos mesmos [Khan et al. 2019], [Ibrahim et al. 2021].

3.2.1. CTU-13

O CTU-13 é um conjunto de dados de tráfego de *botnet* que foi capturado na Universidade CTU, República Tcheca, em 2011 e armazenado em arquivos .pcap [García et al. 2014a].

O conjunto de dados CTU-13 contém 13 arquivos de captura do tráfego, os quais são denominados cenários e são rotulados como Normal, Ataque ou *Background*. Estes arquivos contêm diferentes tipos de *botnets*, incluindo estruturas centralizadas (IRC e HTTP) e descentralizadas (P2P) e vários protocolos. Dessa forma, este conjunto de dados atendeu ao nosso propósito de projetar um modelo de detecção de *botnets* que fosse independente de estrutura e protocolo.

Utilizamos nos nossos experimentos os arquivos correspondentes aos cenários 1, 3, 5, 6, 7, 8 e 12. Consequentemente, testamos sete tipos de *botnets*: Neris, Rbot, Virut, Menti, Sogou, Murlo e Nsis.ay, onde a combinação dessas *botnets* consistia em ambas as estruturas, centralizadas e descentralizadas. Os arquivos .pcap disponibilizados contém apenas o tráfego malicioso, uma vez que por questões de privacidade, a captura completa contendo todos os dados de *background*, normal e de *botnet* não está disponível. Sendo assim, utilizamos parte do conjunto de dados do projeto ISCX-IDS-2012[1] para obter apenas dados de tráfego normal e complementar o conjunto de dados CTU-13 [Shiravi et al. 2012]. Para isso, foi utilizado o arquivo .pcap referente à captura do dia 12/06/2010 (sábado), o qual possui 4.22 GB de tráfego normal.

3.2.2. ISOT HTTP *Botnet*

O conjunto de dados ISOT HTTP[2] foi disponibilizado pela Universidade de Victória, no Canadá e é composto por dois conjuntos de dados diferentes. O primeiro consiste em tráfego malicioso gerado por diferentes *botnets*, enquanto o segundo consiste em tráfego benigno gerado por diversas aplicações de software, como antivírus, bate-papo online e aplicativos de mensagens instantâneas (i.e, Skype, Facebook, Messenger) [Alenazi et al. 2017]. A captura do tráfego dos dois ambientes (normal e *botnet*) foi realizada no período de 14 a 21 de junho de 2017.

O tráfego malicioso foi coletado a partir de um ambiente virtual, no qual foram implementados diferentes kits de *exploits* para *botnets* HTTP, totalizando 9 (nove) servidores de Comando e Controle (C&C), um para cada tipo de *botnet*, o que gerou 5 (cinco) arquivos .pcap. O tráfego benigno também foi capturado de um ambiente virtual simulando tráfego de diversas aplicações instaladas em máquinas virtuais configuradas com o sistema operacional Windows 7, resultando em 3 (três) arquivos .pcap.

Utilizamos nos nossos experimentos os 3 arquivos contendo tráfego benigno e para o tráfego malicioso utilizamos apenas o arquivo init4.pcap, uma vez que este único arquivo contém tráfego de todos os tipos de *botnets* presentes no conjunto de dados ISOT HTTP. Os seguintes tipos de *botnets* estão presentes no arquivo utilizado: zyklon, blue, liphyra, gaudox, blackout, citadel, be.botnet e zeus. Todas essas *botnets* possuem arquitetura centralizada e utilizam o protocolo HTTP.

3.3. Extração dos atributos

Para extrair os fluxos de rede a partir das capturas de tráfego referentes aos dois conjuntos de dados descritos acima, foi utilizada a ferramenta CICFlowMeter[3], que é um

[1] https://www.unb.ca/cic/datasets/ids.html

[2] https://www.uvic.ca/ecs/ece/isot/datasets/botnet-ransomware/index.php

[3] https://www.unb.ca/cic/research/applications.html#CICFlowMeter

gerador e analisador de fluxo de tráfego de rede disponibilizado pelo Instituto Canadense de Segurança Cibernética [Lashkari et al. 2017]. O resultado gerado é um arquivo CSV contendo 84 atributos com estatísticas do tráfego, *e.g.*, total, média e mínimo de pacotes enviados e recebidos. Todos os atributos gerados foram utilizados para os experimentos iniciais, com exceção dos atributos *Flow ID, Source IP, Destination IP e Timestamp* por serem considerados muito específicos de cada fluxo. Para o CICFlowMeter cada fluxo é definido pelo primeiro pacote que determina as direções *forward* (origem para destino) e *backward* (destino para origem). Além disso, os fluxos TCP geralmente são encerrados na desconexão da conexão (pelo pacote FIN), enquanto os fluxos UDP são encerrados por um tempo limite do fluxo. O valor do tempo limite do fluxo pode ser atribuído arbitrariamente, geralmente 600 segundos para ambos (TCP e UDP).

Após a extração dos atributos, os arquivos resultantes de cada um dos conjuntos de dados foram rotulados utilizando a linguagem de programação *python* e a biblioteca *pandas*. A Tabela 1 ilustra a composição final dos dois conjuntos de dados, incluindo a quantidade de fluxos benignos e maliciosos extraídos e ainda, a quantidade por família de *botnet*. Embora em um ambiente real a quantidade de tráfego benigno seja muito superior ao tráfego malicioso, o nosso estudo de caso ficou limitado às quantidades disponibilizadas pelos dois conjuntos de dados para cada tipo de tráfego. Devido a limitação de espaço, não apresentamos a tabela com os 84 atributos extraídos, mas o leitor pode consultar a descrição desses atributos em [3]. Além disso, como o EFC trabalha com dados discretizados para realizar a classificação, discretizamos os dados apenas para implementação do EFC. Para os demais modelos utilizados nesta pesquisa foi realizada apenas a normalização dos dados, uma vez que a discretização poderia prejudicar o desempenho destes algoritmos.

O desempenho do EFC foi comparado ao desempenho dos algoritmos mais populares para detecção de anomalias baseada na análise de fluxos de rede: K-*Nearest Neighbors* (KNN), *Decision Tree* (DT), *Multilayer Perceptron* (MLP), *Naive Bayes* (NB) e *Support Vector Machine* (SVM), além dos classificadores *ensemble*: *AdaBoost* (AD) e *Random Forest* (RF). Além disso, como o EFC é um algoritmo de uma classe, ou seja, é treinado apenas com o tráfego benigno, também foram realizados experimentos com os seguintes algoritmos *One Class* disponíveis na biblioteca *scikit-learn*[4]: *One Class SVM* (OCSVM), *Isolation Forest* (iForest), *Local Outlier Factor* (LOF) e *Eliptic Envelop* (Elenv). O desempenho dos classificadores utilizados nessa pesquisa foi mensurado com base na média dos valores de AUC e F1-score sobre 5 conjuntos de teste (5 *Stratifiedk-fold*) e no erro padrão, com intervalo de confiança de 95%. Por fim, cabe salientar que todos os modelos foram implementados utilizando a configuração padrão do *scikit-learn*.

Todos os experimentos realizados neste trabalho foram efetuados em um notebook com a seguinte configuração: processador Intel Core I7-7700HQ de 3.8 GHZ, 32 GB de memória RAM e sistema operacional Linux Debian 10. Foram realizados dois testes principais. O primeiro é o teste intra-domínio, no qual o treinamento e o teste dos modelos foram realizados utilizando o mesmo conjunto de dados. O segundo é o teste inter-domínio, no qual os modelos são treinados em um conjunto de dados e são avaliados em outro conjunto de dados. O objetivo do segundo teste é avaliar a capacidade dos modelos de se adaptarem à mudanças na rede e consequentemente, a capacidade de detectar *botnets* desconhecidas.

[4]`https://scikit-learn.org/stable/`

Tabela 1. Quantitativos dos Conjuntos de Dados

CTU-13		ISOT HTTP	
Rótulo	**Quantidade**	**Rótulo**	**Quantidade**
Normal	215.251	Normal	76.360
Virut	83.900	Cidatel	145.087
Rbot	46.540	Gaudox	90.970
Neris	22.247	Zeus	80.642
Murlo	11.536	Be.botnet	13.755
Nsis	7.645	Bluebot	13.593
Menti	4.809	Zyklon	12.008
Sogou	72	Blackout	6.881
		Liphyra	3.782
Total Malicioso	176.749	**Total Malicioso**	366.718
Total Benigno	215.251	**Total Benigno**	76.360

4. Resultados

Nessa seção apresentamos os resultados obtidos com a utilização do classificador EFC na detecção de tráfego relacionado à atividade de *botnets*, bem como os resultados obtidos utilizando diversos classificadores de uma e de duas classes.

4.1. Distribuição das Energias Calculadas pelo EFC

Para realizar a classificação de fluxos benignos e fluxos contendo atividades de *botnets*, o EFC infere um modelo estatístico baseado nas amostras de fluxos benignos durante o treinamento do modelo. Esse modelo é então utilizado para calcular as energias das amostras de fluxos benignos e maliciosos contidos no conjunto de teste e a partir dos valores calculados, realizar a classificação dos fluxos de rede.

A Figura 1 (a) ilustra os valores de energia calculados considerando parte das amostras benignas do conjunto de dados CTU-13 e a Figura 1 (b) mostra os valores das energias relativos ao conjunto de dados ISOT HTTP. Esses valores são referentes ao teste intra-domínio, ou seja, treino e teste no mesmo conjunto de dados. Pode-se observar que a separação entre as duas classes é clara, i.e., a energia de fluxos benignos está claramente deslocada para a esquerda em relação à distribuição das energias de fluxos maliciosos. A linha vertical vermelha representa o limiar de classificação do EFC, o qual foi definido como percentil 95 da distribuição de energia obtida na etapa de treino.

A energia calculada para cada fluxo é composta pela soma dos valores de acoplamento de todos os possíveis pares de atributos. Dessa forma, o acoplamento mede a probabilidade de ocorrência dos valores de pares de atributos específicos, nas amostras de treinamento que geraram o modelo [Pontes et al. 2021]. Devido a natureza caixa branca do modelo estatístico inferido pelo EFC, é possível verificar a contribuição individual de cada par de atributos para a energia total de um fluxo, identificando assim, quais são os pares que mais contribuem para um maior ou menor valor de energia.

As Tabelas 2 e 3 ilustram os três pares de atributos com menor acoplamento (indicando fluxo benigno) e os três pares de atributos com maior acoplamento (indicando fluxo malicioso), considerando os conjuntos de dados ISOT HTTP e CTU-13 respectivamente. Observa-se pelas tabelas que os atributos que caracterizam o comprimento do fluxo (pacotes) i.e., variância, desvio padrão e comprimento médio dos pacotes são determinísticos para a identificação de *botnets*. A principal suposição é que o tráfego de *botnet* gerado

1 (a) Treino/Teste CTU-13 1 (b) Treino/Teste ISOT HTTP

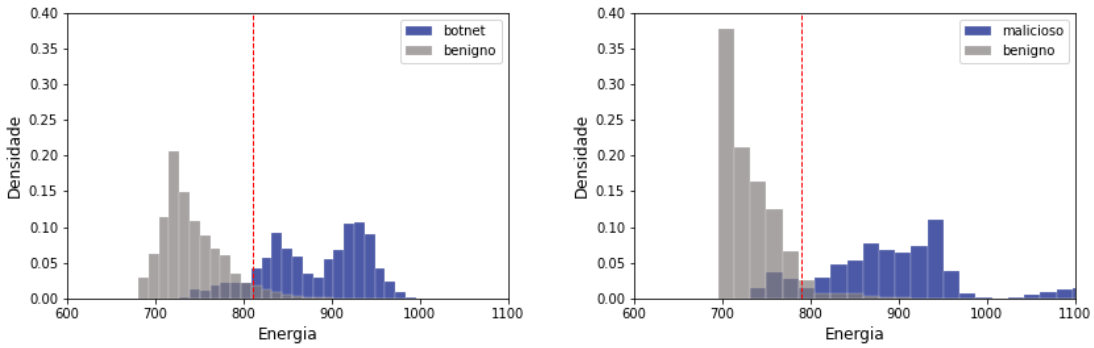

Figura 1. Histogramas das Energias Calculadas na Fase de Teste Usando os *Datasets* CTU-13 e ISOT HTTP.

por *bots* (principalmente ações predefinidas) é mais uniforme do que o tráfego gerado por usuários legítimos, exibindo um comportamento muito diversificado [Beigi et al. 2014]. Além disso, pares de atributos relacionados aos acoplamentos negativos se assemelham para os dois conjuntos de dados, o que indica uma similaridade no tráfego benigno dos mesmos. Por outro lado, os pares de atributos relacionados aos acoplamentos negativos (malicioso) são diferentes para os dois conjuntos de dados. Essa diferença pode estar relacionada a heterogeneidade dos dois conjuntos de dados, uma vez que possuem *botnets* de diferentes arquiteturas e protocolos, realizando ataques diferentes.

Tabela 2. Acoplamento entre Pares de Atributos - ISOT HTTP

Atributo 1	Atributo 2	Acoplamento
PacketLengthStd	PacketLengthVar	-2.22536
FwdHeaderLength	MinSegSizeForward	-2.01219
FwdPacketLengthMean	AvgFwdSegmentSize	-1.95448
FlowIATMax	PacketLengthStd	3.01095
FlowIATMax	PacketLengthVariance	3.01095
FlowPackets-s	PacketLengthStd	2.87707

Tabela 3. Acoplamento entre Pares de Atributos CTU-13

Atributo 1	Atributo 2	Acoplamento
FwdPacketLengthMean	AvgFwdSegmentSize	-2.78469
PacketLengthStd	PacketLengthVar	-2.64005
FwdPacketLengthMean	AvgFwdSegmentSize	-2.12473
FwdHeaderLength	CWEFlagCount	3.14073
FwdHeaderLength	ECEFlagCount	3.14073
FwdHeaderLength	ActiveMax	3.14073

4.2. EFC x Algoritmos de Classificação de Duas Classes

Na Tabela 4 pode ser visualizado o desempenho médio e o erro padrão (com intervalo de confiança de 95%) de cada classificador, utilizando o conjunto de dados ISOT HTTP. Na

primeira abordagem, que é o teste intra-domínio, todos os modelos propostos obtiveram resultados muito próximos, com um F1-score acima de 0.98 e AUC acima de 0.99. A única exceção foi o algoritmo NB, o qual obteve o menor desempenho, com um F1-score de 0.952 ± 0.000 e AUC de 0.799 ± 0.004.

No teste inter-domínio, onde o conjunto de dados ISOT HTTP foi utilizado para treino e o CTU-13 para teste, o EFC obteve resultado bastante superior quando comparado aos demais modelos, principalmente em relação à métrica F1-score (0.663 ± 0.001). Um fato que chamou a atenção foi que os modelos NB, RF e AD obtiveram um F1-score igual a 0. Foi observado através das respectivas matrizes de confusão, conforme a Figura 2, que estes modelos classificaram todas as instâncias maliciosas como benignas, obtendo portanto, um taxa de verdadeiro-positivo igual 0, o que explica o valor do F1-score obtido por estes modelos. Por fim, uma vez que classificadores *ensemble* utilizam uma combinação das previsões de diversos modelos e geralmente apresentam melhor desempenho em relação à classificadores simples, esperava-se que o desempenho do RF e AD fosse superior ao desempenho dos demais modelos testados.

O teste inter-domínio neste cenário é bastante desafiador, uma vez que o conjunto de dados ISOT HTTP possui instâncias somente de *botnets* que utilizam o protocolo HTTP para comunicação (arquitetura centralizada) e o conjunto de dados CTU-13 possui *botnets* que utilizam os protocolos HTTP, IRC e P2P (arquiteturas centralizadas e descentralizadas). Dessa forma, o resultado obtido pelo EFC foi bastante satisfatório, dada a grande mudança de contexto.

Tabela 4. Desempenho Médio dos Classificadores - ISOT HTTP

	Treino/Teste ISOT		**Treino ISOT/Teste CTU-13**	
Classificador	**F1-score**	**AUC**	**F1-score**	**AUC**
NB	0.952 ± 0.000	0.799 ± 0.004	0.000 ± 0.000	0.500 ± 0.000
KNN	$\mathbf{0.999 \pm 0.000}$	0.997 ± 0.000	0.074 ± 0.009	0.333 ± 0.009
DT	$\mathbf{0.999 \pm 0.000}$	0.996 ± 0.000	0.019 ± 0.031	0.473 ± 0.031
SVM	0.989 ± 0.000	0.998 ± 0.000	0.120 ± 0.002	0.636 ± 0.002
MLP	0.994 ± 0.001	$\mathbf{0.999 \pm 0.000}$	0.271 ± 0.240	0.601 ± 0.240
EFC	0.989 ± 0.000	0.995 ± 0.000	$\mathbf{0.663 \pm 0.001}$	0.535 ± 0.001
Ensemble				
RF	$\mathbf{0.999 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	0.000 ± 0.000	0.539 ± 0.000
AD	0.998 ± 0.001	$\mathbf{1.000 \pm 0.000}$	0.000 ± 0.000	$\mathbf{0.756 \pm 0.000}$

Os resultados utilizando o conjunto de dados CTU-13 podem ser visualizados na Tabela 5. Na primeira abordagem, que é o teste intra-domínio, os modelos KNN, DT, MLP e RF obtiveram um F1-score e AUC acima de 0.99, enquanto o EFC obteve um F1-score de 0.877 ± 0.000 e AUC de 0.961 ± 0.000. Novamente o NB obteve o menor desempenho, com um F1-score de 0.677 ± 0.001 e AUC de 0.864 ± 0.013.

No teste inter-domínio, onde o conjunto de dados CTU-13 foi utilizado para treino e o ISOT-HTTP para teste, o *Random Forest* obteve o maior valor de F1-score (0.794 ± 0.022), enquanto o EFC obteve o melhor AUC (0.729 ± 0.076). O desempenho dos demais classificadores foi bastante inferior. O teste inter-domínio neste cenário é menos desafiador quando comparado ao experimento anterior. Aqui, o treino é realizado em um

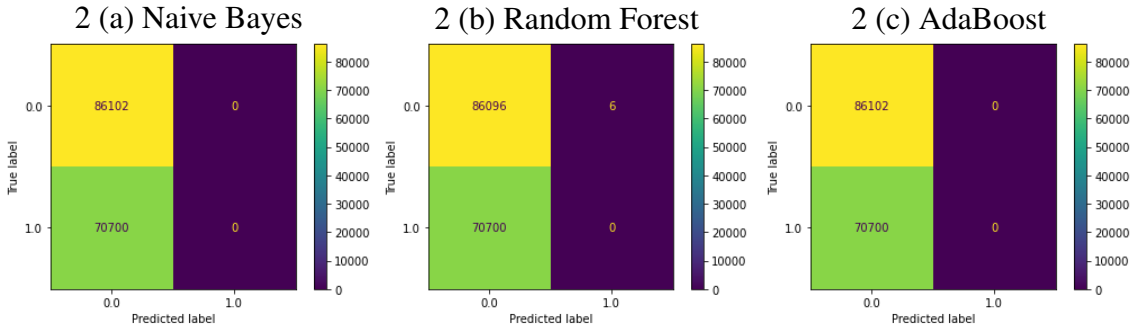

Figura 2. Matrizes de Confusão dos Classificadores NB, RF e AD nos Experimentos de Classificação Inter-Domínio.

Tabela 5. Desempenho Médio dos Classificadores - CTU-13

	Treino/Teste CTU-13		Treino CTU-13/Teste ISOT	
Classificador	**F1-score**	**AUC**	**F1-score**	**AUC**
NB	0.677 ± 0.001	0.864 ± 0.013	0.109 ± 0.214	0.675 ± 0.214
KNN	0.997 ± 0.000	0.999 ± 0.000	0.005 ± 0.007	0.457 ± 0.007
DT	$\mathbf{0.999 \pm 0.000}$	$\mathbf{0.999 \pm 0.000}$	0.544 ± 0.004	0.275 ± 0.004
SVM	0.912 ± 0.003	0.962 ± 0.001	0.481 ± 0.010	0.665 ± 0.003
MLP	0.994 ± 0.000	1.000 ± 0.000	0.325 ± 0.240	0.711 ± 0.240
EFC	0.877 ± 0.000	0.961 ± 0.000	0.758 ± 0.076	$\mathbf{0.729 \pm 0.076}$
Ensemble				
RF	$\mathbf{0.999 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.794 \pm 0.022}$	0.702 ± 0.022
AD	0.980 ± 0.002	0.998 ± 0.000	0.262 ± 0.172	0.573 ± 0.172

conjunto de dados contendo *botnets* com protocolos de comunicação variados (HTTP, IRC e P2P) e o teste em um conjunto de dados contendo somente *botnets* HTTP. Sendo assim, todos os modelos obtiveram um desempenho superior ao obtido no experimento inter-domínio apresentado anteriormente e observa-se ainda que, nenhum modelo obteve F1-score igual a 0.

4.3. EFC X Algoritmos de Classificação de Uma Classe

Na Tabela 6 pode ser visualizado o desempenho médio e o erro padrão (com intervalo de confiança de 95%) de cada classificador *One Class*, utilizando o conjunto de dados ISOT-HTTP. Na primeira abordagem, que é o teste intra-domínio, o EFC obteve desempenho bem superior aos demais classificadores, tanto em relação ao F1-score (0.989 ± 0.000) quanto ao AUC (0.995 ± 0.000). O menor desempenho foi obtido pelo classificador Elenv com um F1-score de 0.035 ± 0.034 e AUC de 0.781 ± 0.005.

No teste inter-domínio, onde o conjunto de dados ISOT HTTP foi utilizado para treino e o CTU-13 para teste, o EFC manteve o melhor desempenho em relação ao F1-score (0.663 ± 0.001), seguido pelos classificadores OCSVM (0.627 ± 0.005) e LOF (0.621 ± 0.000). Em relação ao AUC, o LOF obteve o melhor resultado (0.770 ± 0.000), seguido pelo OCSVM (0.726 ± 0.005) e pelo EFC (0.535 ± 0.001).

A Tabela 7 mostra o desempenho médio e o erro padrão (com intervalo de

Tabela 6. Desempenho Médio dos Classificadores *One Class* - ISOT HTTP

	Treino/Teste ISOT		Treino ISOT/Teste CTU-13	
Classificador	**F1-score**	**AUC**	**F1-score**	**AUC**
EFC	**0.989 ± 0.000**	**0.995 ± 0.000**	**0.663 ± 0.001**	0.535 ± 0.001
OCSVM	0.731 ± 0.001	0.540 ± 0.002	0.627 ± 0.005	0.726 ± 0.005
iForest	0.670 ± 0.040	0.768 ± 0.009	0.443 ± 0.043	0.342 ± 0.043
Elenv	0.035 ± 0.034	0.781 ± 0.005	0.234 ± 0.247	0.466 ± 0.247
LOF	0.630 ± 0.011	0.721 ± 0.029	0.621 ± 0.000	**0.770 ± 0.000**

confiança de 95%) de cada classificador, utilizando o conjunto de dados CTU-13. Na primeira abordagem, que é o teste intra-domínio, o modelo LOF obteve o melhor desempenho, tanto em F1-score (0.923 ± 0.002), quanto em AUC (0.983 ± 0.000), seguido pelo EFC, o qual obteve um F1-score de 0.879 ± 0.003 e AUC de 0.962 ± 0.001. O classificador iForest obteve o pior desempenho, com um F1-score de 0.084 ± 0.002 e AUC de 0.595 ± 0.025.

Tabela 7. Desempenho Médio dos Classificadores *One Class* - CTU13

	Treino/Teste CTU13		Treino CTU-13/Teste ISOT	
Classificador	**F1-score**	**AUC**	**F1-score**	**AUC**
EFC	0.879 ± 0.003	0.962 ± 0.001	0.705 ± 0.002	**0.736 ± 0.003**
OCSVM	0.762 ± 0.001	0.751 ± 0.003	**0.906 ± 0.000**	0.338 ± 0.001
iForest	0.084 ± 0.002	0.595 ± 0.025	0.732 ± 0.198	0.628 ± 0.002
Elenv	0.257 ± 0.143	0.705 ± 0.043	0.411 ± 0.314	0.412 ± 0.143
LOF	**0.923 ± 0.002**	**0.983 ± 0.000**	0.893 ± 0.001	0.435 ± 0.002

No teste inter-domínio, onde o conjunto de dados CTU-13 foi utilizado para treino e o ISOT-HTTP para teste, o OCSVM obteve o melhor F1-score (0.906 ± 0.000), porém o AUC foi o menor dentre todos os outros classificadores (0.338 ± 0.001). Por outro lado, o EFC obteve o melhor AUC (0.736 ± 0.003).

5. Conclusão e Trabalhos Futuros

Neste trabalho foi proposta uma nova abordagem para detecção de *botnets* através da utilização do classificador de fluxos *Energy-based Flow Classifier* (EFC), o qual possui como característica principal a adaptabilidade a diferentes domínios [Pontes et al. 2021]. Além disso, comparamos o desempenho do EFC com diversos classificadores de uma e de duas classes. Os resultados parciais obtidos demonstraram que os modelos baseados em duas classes sofreram fortes variações nos testes inter-domínio, principalmente no cenário mais desafiador, no qual o conjunto de treinamento possuía somente *botnets* centralizadas (HTTP) e o conjunto de testes possuía *botnets* com arquiteturas centralizadas (HTTP e IRC) e descentralizadas (P2P). Neste cenário o EFC foi bem superior aos demais modelos, obtendo um F1-score de 0.663 ± 0.0010 e um AUC de 0.535 ± 0.001.

Em relação aos modelos baseados em uma classe, no teste intra-domínio o EFC apresentou resultados superiores aos demais algoritmos considerando o conjunto de dados ISOT HTTP, obtendo um F1-score de 0.989 e um AUC de 0.995. Nos testes com o *dataset*

CTU-13 o EFC obteve um F1-score de 0.879 e um AUC de 0.962, sendo superado apenas pelo algoritmo LOF. Já no teste inter-domínio, o EFC se mostrou superior, ou em relação ao F1-score (0.663 ± 0.001) em um dos experimentos, ou em relação ao AUC (0.736 ± 0.003) em outro experimento. Sendo assim, no contexto geral, o EFC foi o modelo que se mostrou menos sensível a mudanças na distribuição de dados, apresentando resultados mais robustos e se mostrando um classificador promissor para a detecção de novos tipos de *botnets*. Como trabalhos futuros pretendemos realizar uma análise e seleção dos atributos que melhor caracterizam o comportamento de atividades de *botnets*, visando assim, obter um melhor desempenho com o modelo proposto. Além disso, pretende-se avaliar o desempenho do EFC na detecção de *botnets* mais recentes.

6. Agradecimentos

O trabalho desenvolvido pelo Prof. Marotta e pelo Prof. Gondim faz parte do projeto "Internet do Futuro Programável para Arquiteturas e Softwares Seguros" sob processo n° 2020/05152-7, da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

Referências

Alauthaman, M., Aslam, N., Zhang, L., Alasem, R., and Hossain, M. A. (2018). A P2P Botnet detection scheme based on decision tree and adaptive multilayer neural networks. *Neural Computing and Applications*, 29(11):991–1004.

Alenazi, A., Traore, I., Ganame, K., and Woungang, I. (2017). Holistic Model for HTTP Botnet Detection Based on DNS Traffic Analysis. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10618 LNCS:1–18.

Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J. A., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., and Zhou, Y. (2017). Understanding the mirai botnet. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1093–1110, Vancouver, BC. USENIX Association.

Beigi, E. B., Jazi, H. H., Stakhanova, N., and Ghorbani, A. A. (2014). Towards effective feature selection in machine learning-based botnet detection approaches. *2014 IEEE Conference on Communications and Network Security, CNS 2014*, pages 247–255.

Bilge, L., Balzarotti, D., Robertson, W., Kirda, E., and Kruegel, C. (2012). Disclosure: detecting botnet command and control servers through large-scale netflow analysis. In *Proceedings of the 28th Annual Computer Security Applications Conference*, pages 129–138.

Council to Secure the Digital Economy (2019). International Botnet and Iot Security Guide 2020.

European Union Agency for Network and Information Security (2020). Botnet - ENISA Threat Landscape 2019/20. (April).

Gadelrab, M. S., ElSheikh, M., Ghoneim, M. A., and Rashwan, M. (2018). BotCap: Machine learning approach for botnet detection based on statistical features. *International Journal of Communication Networks and Information Security*, 10(3):563–579.

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

García, S., Zunino, A., and Campo, M. (2014b). Survey on network-based botnet detection methods. *Security and Communication Networks*, 7(5):878–903.

Ibrahim, W. N. H., Anuar, S., Selamat, A., Krejcar, O., Gonzalez Crespo, R., Herrera-Viedma, E., and Fujita, H. (2021). Multilayer Framework for Botnet Detection Using Machine Learning Algorithms. *IEEE Access*, 9:48753–48768.

Khan, R. U., Zhang, X., Kumar, R., Sharif, A., Golilarz, N. A., and Alazab, M. (2019). An adaptive multi-layer botnet detection technique using machine learning classifiers. *Applied Sciences (Switzerland)*, 9(11).

Lashkari, A. H., Gil, G. D., Saiful, M., Mamun, I., and Ghorbani, A. A. (2017). Characterization of Tor Traffic using Time based Features. (Cic):253–262.

Li, H., Chen, Z., Spolaor, R., Yan, Q., Zhao, C., and Yang, B. (2019). DART: Detecting Unseen Malware Variants using Adaptation Regularization Transfer Learning. *IEEE International Conference on Communications*, 2019-May.

Pontes, C. F., De Souza, M. M., Gondim, J. J., Bishop, M., and Marotta, M. A. (2021). A New Method for Flow-Based Network Intrusion Detection Using the Inverse Potts Model. *IEEE Transactions on Network and Service Management*, 18(2):1125–1136.

Saad, S., Traore, I., Ghorbani, A., Sayed, B., Zhao, D., Lu, W., Felix, J., and Hakimian, P. (2011). Detecting P2P botnets through network behavior analysis and machine learning. *2011 9th Annual International Conference on Privacy, Security and Trust, PST 2011*, pages 174–180.

Shiravi, A., Shiravi, H., Tavallaee, M., and Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Computers and Security*, 31(3):357–374.

Silva, S. S., Silva, R. M., Pinto, R. C., and Salles, R. M. (2013). Botnets: A survey. *Computer Networks*, 57(2):378–403.

Vormayr, G., Zseby, T., and Fabini, J. (2017). Botnet Communication Patterns. *IEEE Communications Surveys and Tutorials*, 19(4):2768–2796.

Wainwright, P. and Kettani, H. (2019). An analysis of botnet models. *ACM International Conference Proceeding Series*, pages 116–121.

Yadav, J. and Thakur, J. (2020). BotEye: Botnet detection technique via traffic flow analysis using machine learning classifiers. *PDGC 2020 - 2020 6th International Conference on Parallel, Distributed and Grid Computing*, pages 154–159.

Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A., and Garant, D. (2013). Botnet detection based on traffic behavior analysis and flow intervals. *Computers and Security*, 39(PARTA):2–16.

Zolanvari, M., Teixeira, M. A., and Jain, R. (2018). Effect of imbalanced datasets on security of industrial IoT using machine learning. *2018 IEEE International Conference on Intelligence and Security Informatics, ISI 2018*, pages 112–117.