Coleta e Análise de Características de Fluxo para Classificação de Tráfego em Redes Definidas por Software

Rodolfo Vebber Bisol[1], **Anderson Santos da Silva**[1], **Cristian Cleder Machado**[1], **Lisandro Zambenedetti Granville**[1], **Alberto Egon Schaeffer-Filho**[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

Email: {rvbisol, assilva, ccmachado, granville, alberto}@inf.ufrgs.br

Resumo. *Uma classificação acurada de fluxos de tráfego é uma tarefa muito importante no âmbito de resiliência de redes de computadores, podendo ser aplicada para diversos objetivos, por exemplo, criar mecanismos para detecção de fluxos de tráfego maliciosos. Estes mecanismos de classificação de fluxos de tráfego, apesar de já serem muito sofisticados, ainda precisam ser aprimorados, pois certos ataques podem se camuflar entre os tráfegos benignos, que normalmente existem na rede, o que pode dificultar a sua classificação. O objetivo deste trabalho é aprimorar as técnicas de seleção de características e classificação de tráfego propostas em trabalhos anteriores pela adição dos algoritmos Sequential Backward Selection (SBS) e K-means. Além disso, como validação dessa extensão, este trabalho realiza experimentos em um ambiente mais realístico, onde diversos tipos de tráfego coexistem e camuflam-se na rede. Os resultados dos experimentos indicam que a combinação do algoritmo SBS com os algoritmos de classificação de tráfego SVM e K-means permite a identificação de características adequadas para identificar tráfego malicioso.*

Abstract. *An accurate flow classification is a very important task in network resilience and can be applied with many purposes such as to create mechanisms for malicious traffic detection. These mechanisms although sophisticated still need to be refined because certain attacks can hide amongst benign flows that normally exist in the network which can hinder their classification. The main objective of this work is to enhance the feature selection and flow classifier techniques proposed in previous work by the addition of the Sequential Backward Selection (SBS) and K-means algorithms. Moreover, as validation of our extension, this work conducts the experimental evaluation on a more realistic environment, where different flow profiles coexist and camouflage in the network. The experiment results indicate that the combination of SBS with the SVM and K-means traffic classification algorithms allows the identification of appropriate features to identify malicious traffic.*

1. Introdução

Redes de computadores e principalmente a Internet se tornaram essenciais para a vida moderna, sendo utilizadas por governos, empresas e instituições de ensino. Por isso, é muito importante identificar comportamentos anômalos e ataques de usuários maliciosos a fim de manter o adequado funcionamento das redes.

A classificação de fluxos de tráfego pode ser utilizada para diversos objetivos, como detecção de ataques, realocação de recursos de redes e modelagem de perfil de usuários [Nguyen 2008]. Para se obter uma classificação acurada, é necessário identificar diversas características, tais como duração do fluxo e número de pacotes transmitidos. Essas características são capazes de descrever fielmente os diferentes perfis de tráfego que possam coexistir na rede, por exemplo, o número de pacotes e bytes transmitidos em fluxos de tráfego HTTP é caracterizado por rajadas, diferentemente de fluxos de tráfego FTP que apresentam-se de maneira contínua. A operação de coleta de características é dificultada por limitações existentes na organização de redes IP tradicionais, tais como: *(i)* o acesso à informação sobre o estado da rede é heterogêneo para dispositivos de diferentes fabricantes; *(ii)* a grande complexidade da adição de novas funcionalidades aos *switches* ou roteadores de rede, já que cada equipamento deve ser modificado individualmente; e *(iii)* a lógica distribuída em cada elemento de rede pode alterar um fluxo durante seu trânsito pela rede, por exemplo, fazendo alterações na prioridade dos pacotes ou em cabeçalhos TCP/IP ou IP.

A adoção de Redes Definidas por Software (Software Defined Networking - SDN), neste contexto, elimina as limitações mencionadas e facilita o estudo e implementação de novas técnicas que tornam a classificação de fluxos de tráfego mais acurada [ONF 2012]. O fato de SDN ser caracterizada pela separação dos planos de controle e encaminhamento faz com que a lógica esteja centralizada em controladores SDN, como o NOX [Gud 2008] e FloodLight [Erickson 2012]. Assim, alterações na lógica da rede são realizadas apenas nos controladores. Além disso, a coleta de dados sobre os fluxos de tráfego existentes na rede também se torna uma tarefa menos complexa, sendo realizada através dos controladores que possuem visão de grandes porções da rede, ou até mesmo global quando utilizados em conjunto [Machado et al. 2014]

Mesmo SDN tornando esta tarefa de coleta de informações mais simples e acurada, a adoção de SDN por si só não torna os mecanismos de classificação de fluxos de tráfego mais acurados. Outras medidas devem ser tomadas para aumentar a precisão destes mecanismos, tais como uma análise detalhada das características criadas através do processo de seleção de características. Esse processo consiste em identificar quais características são mais relevantes para a classificação de fluxos de tráfego e eliminar características correlacionadas que podem adicionar ruídos à classificação ou gerar desperdício de recursos computacionais [Kim et al. 2008].

Dentro deste cenário, o objetivo deste artigo é aprimorar o trabalho desenvolvido anteriormente em [da Silva et al. 2015] que, através de contadores nativos tais como número de pacotes e bytes, estendeu um conjunto de características (*e.g.*, tamanho médio dos pacotes e tempo entre chegada dos pacotes). Este artigo apresenta a implementação de um novo algoritmo de seleção de características, o *Sequential Backward Selection* (SBS), que é analisado e comparado com estratégias baseadas em algoritmo genético e Análise de Componente Principal (*Principal Component Analysis* - PCA). Além disso, é proposta a utilização de técnicas de seleção de características para aprimorar a precisão de dois algoritmos de classificação de fluxos de tráfego, SVM e K-means. Como estudo de caso e validação da extenção proposta, foram realizados experimentos utilizando transferências FTP e *streaming* de vídeo como tráfegos benignos e ataques DDoS como tráfegos maliciosos.

Este artigo está estruturado na seguinte forma. Na Seção 2, é apresentada uma visão geral dos assuntos abordados neste trabalho, incluindo SDN, seleção de características e classificação de fluxos de tráfego. Na Seção 3, é descrito o sistema de coleta e extensão de características de fluxo de tráfego e os algoritmos utilizados. Na Seção 4, são apresentados os detalhes sobre implementação, incluindo a parametrização dos algoritmos, ferramentas utilizadas e resultados obtidos. Finalmente, na Seção 5, este trabalho é concluído apresentando as considerações finais e propostas para trabalhos futuros.

2. Contextualização e Trabalhos Relacionados

Um classificador de fluxos de tráfego é um processo que captura diferentes fluxos de tráfego e determina a qual classe de tráfego tais fluxos pertencem. Essa ação é realizada através da análise dos dados dos pacotes, (*e.g.*, os endereços IP de origem e de destino) [Blake et al. 1998]. A classificação de fluxos de tráfego é utilizada para identificar os perfis de tráfego existentes na rede. A classificação dos fluxos de tráfego é importante para auxiliar a administração de redes de computadores, sendo utilizada para monitoramento, filtro de conteúdo, avaliação de qualidade de serviço (QoS), entre outras aplicações.

A utilização de um grande conjunto de características para a classificação de fluxos de tráfego possui desvantagens, tais como *(i)* o uso de características irrelevantes pode adicionar ruído, dificultando a classificação, e *(ii)* o processamento de um grande conjunto de características, possivelmente correlacionadas, pode gerar desperdício de recursos computacionais. Procurando resolver este problema, foram desenvolvidas diversas técnicas de seleção de características [Blum and Langley 1997]. Tais técnicas consistem em determinar as características, ou atributos, mais relevantes para um determinado problema e eliminar as características menos relevantes. A seleção de características oferece uma série de vantagens, tais como *(i)* um melhor entendimento e visualização dos dados, *(ii)* redução do tempo de treinamento de algoritmos de aprendizado de máquina (*Machine Learning* - ML), e *(iii)* uma maior acurácia na classificação de fluxos [Guyon 2003].

Por não ser uma tarefa trivial, existem hoje diversas técnicas de seleção de características, que podem ser divididas em três grupos principais [Sae 2007]. O primeiro grupo é o das técnicas conhecidas como filtros, que analisam as propriedades intrínsecas dos dados através de métodos matemáticos, por exemplo matrizes de correlação. Além disso, tais técnicas são independentes de algoritmos de classificação ou indução. Exemplos de filtros são Análise de Componente Principal (*Principal Component Analysis* - PCA) [Jolliffe 1986] e *Fast Correlation-based Filter* (FCBF) [Yu 2003]. O segundo grupo de técnicas é conhecido como *Wrappers*. Essas técnicas possuem como característica o uso de métodos de indução como uma sub-rotina e realizam uma busca heurística dentro do espaço de possíveis subconjuntos. Exemplos de *Wrappers* são o OBLIVION [Langley 1994] e os Algoritmos Genéticos Híbridos [Oh et al. 2004]. Por último, existem as técnicas *Embedded*. Essas técnicas são desenvolvidas especificamente para um determinado problema e possuem um funcionamento parecido com o dos *Wrappers*.

O surgimento do paradigma SDN trouxe novas possibilidades que podem solucionar problemas em relação à coleta e extensão de características de fluxos de tráfego utilizadas para a classificação dos mesmos [Feamster et al. 2013]. Isto se deve por dois motivos: o primeiro é a separação dos planos de controle e encaminhamento e, consequentemente, a centralização da lógica da rede em um controlador. Desta forma, a coleta

de informações sobre os fluxos existentes na rede é facilitada, passando a ser feita através do controlador da rede. E segundo, o protocolo OpenFlow [McKeown et al. 2008], que define a comunicação entre o controlador da rede e os *switches*, que por sua vez possuem em sua especificação uma tabela de contadores referentes aos fluxos de tráfego existentes, por exemplo contadores de *bytes* e pacotes. Esses contadores foram definidos com o objetivo de facilitar a criação de mecanismos de QoS, por exemplo mecanismos de classificação de tráfego [ONF 2014].

3. Arquitetura para Coleta e Análise de Características em SDN

Esta seção descreve uma visão geral do trabalho realizado anteriormente em [da Silva et al. 2015], indicando como é realizado o processo de coleta e extensão de características de fluxos de tráfego. Além disso, apresenta os algoritmos de seleção de características e classificação de tráfego implementados inicialmente, *i.e.*, Análise do Componente Principal (*Principal Component Analysis* - PCA), Algoritmo Genético (*Genetic Algorithm* - GA) e Máquina de Vetores de Suporte (Support Vector Machine - SVM). Finalmente, são descritos os algoritmos propostos para extensão da arquitetura. Esses algoritmos são o Seleção Sequencial Inversa (*Sequential Backward Selection* - SBS) e o K-means.

3.1. Arquitetura

A arquitetura da solução é representada na Figura 1. A arquitetura consiste em dois componentes principais: *(i) Flow Feature Manager*, responsável pela coleta de informações da rede e criação de um conjunto de características avançadas que possam representar fielmente cada fluxo existente; e *(ii) Flow Feature Selector*, responsável por definir o melhor subconjunto de características para classificação de fluxos de tráfego. A seguir, os dois componentes e seus módulos são descritos em maiores detalhes:

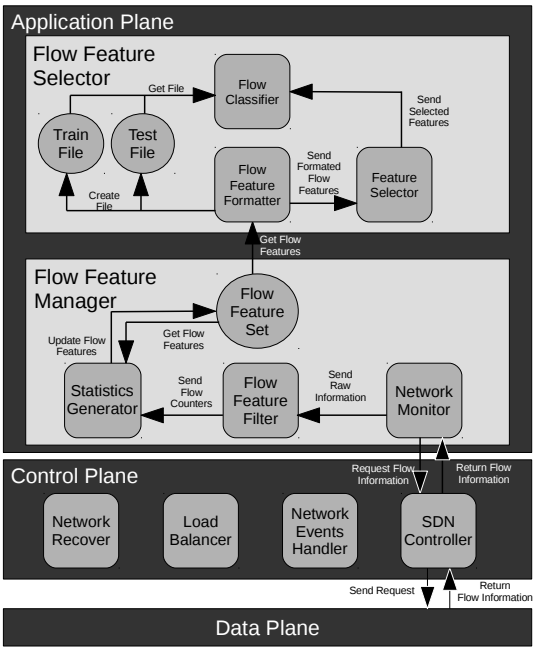

Figura 1. Arquitetura para coleta e extensão de características de fluxo.

- ***Flow Feature Manager –*** inclui os módulos: *Network Monitor Module*, responsável por coletar informações da rede; *Flow Feature Filter Module*, responsável por filtrar as informações recebidas selecionando somente as informações relevantes ao sistema; e *Statistics Generator Module*, responsável por criar o conjunto de características avançadas com as informações recebidas do *Flow Feature Filter Module*. O conjunto de características criado é representado pela estrutura de dados *Flow Feature Set*.
- ***Flow Feature Selector –*** inclui os módulos: *Flow Feature Formatter Module*, responsável por reorganizar o conjunto de características em um formato padrão e criar os arquivos de treinamento e teste; *Feature Selector Module*, responsável por selecionar as características de fluxo mais importantes e criar subconjuntos formados por estas características; e *Flow Classifier Module*, responsável por avaliar a qualidade dos subconjuntos criados através da classificação de fluxos de tráfego.

3.2. Coleta e Extensão de Características de Fluxo

Inicialmente, o *Network Monitor*[1] envia uma requisição ao controlador solicitando informações sobre os fluxos existentes na rede. O controlador, por sua vez, reúne as informações presentes nas tabelas de encaminhamento dos *switches* e as envia para o *Network Monitor*. Este, repassa as informações para o *Flow Feature Filter* que seleciona apenas as informações que serão utilizadas, por exemplo: valor dos contadores de *bytes* e pacotes que são utilizados para a criação do conjunto de características. Além desses contadores, o *Flow Feature Filter* também seleciona os endereços IP e portas TCP/UDP de origem e destino e o protocolo da camada de transporte, de modo que cada fluxo seja identificado univocamente. O *Statistics Generator* recebe os identificadores de fluxo e os dois contadores e estende estes contadores em características de fluxo mais representativas, como tamanho dos pacotes e duração do fluxo. O *Statistics Generator* também é responsável por criar e manter atualizada uma estrutura de dados, o *Flow Feature Set*, que reúne todas as características de cada fluxo de tráfego identificado na rede.

As informações utilizadas para a criação das características de fluxos de tráfego são dois contadores nativos, presentes na tabela de contadores do protocolo OpenFlow: *Byte Counter* e *Packet Counter*. Para estender os contadores nativos do OpenFlow, o sistema utiliza outros dois valores criados internamente: contador de amostras, que salva a quantidade de vezes que um determinado fluxo foi amostrado na rede, e tempo entre requisições enviadas ao controlador pelo *Network Monitor*.

A partir destas informações, são criadas quatro características básicas: *bytes* por segundo; pacotes por segundo; comprimento dos pacotes; e intervalo entre a chegada de dois pacotes. Estas quatro características são estimativas dos valores reais, uma vez que para se obter os valores reais de cada uma delas seria necessário coletar informações em nível de pacote, o que não é oferecido nativamente pelo OpenFlow e demandaria uma maior capacidade computacional por parte da aplicação. Estas características isoladamente não oferecem informações suficientes sobre o perfil ou o estado dos fluxos de tráfego. Com o objetivo de obter um maior entendimento sobre o comportamento dos tráfegos, estas quatro características são expandidas em três grupos de características: *Características Escalares*, *Características Estatísticas* e *Características Complexas*.

[1] Utiliza-se uma API REST para obter dados do controlador a cada 30s (intervalo configurável em nossa implementação).

As *Características Escalares* representam o estado dos fluxos de tráfego em um determinado tempo. As *Características Escalares* são: máximo e mínimo de cada uma das quatro características básicas, o tamanho dos fluxos em *bytes* e o número de pacotes, e a duração dos fluxos. As *Características Estatísticas* oferecem um entendimento mais preciso sobre o comportamento dos fluxos de tráfego ao longo do tempo e são elas: as médias e variâncias das quatro características básicas e o primeiro e terceiro quartis do intervalo de chegada entre dois pacotes, e tamanho dos pacotes. As *Características Complexas* são os dez primeiros componentes da Transformada Discreta de Fourier. A Transformada de Fourier representa o comportamento da função em um domínio de frequência, onde cada componente é um par complexo em que o módulo representa o quanto a frequência (ou amostra) é comum, enquanto o valor complexo representa a fase da frequência. Os dez primeiros componentes da Transformada de Fourier também são avaliadas como características relevantes para a classificação de fluxos de tráfego no trabalho de [Auld et al. 2007]. A Tabela 1 apresenta o conjunto final de trinta e três características de fluxo criadas usando os contadores nativos do OpenFlow.

Tabela 1. Características estendidas utilizando contadores nativos do OpenFlow.

Características Escalares	**Características Estatísticas**	**Características Complexas**
Bytes por Segundo Máximo	Média de *Bytes* por Segundo	1^o Componente da Transformada Discreta de Fourier
Bytes por Segundo Mínimo	Variância de *Bytes* por Segundo	2^o Componente da Transformada Discreta de Fourier
Pacotes por Segundo Máximo	Média de Pacotes por Segundo	3^o Componente da Transformada Discreta de Fourier
Pacotes por Segundo Mínimo	Variância de Pacotes por Segundo	4^o Componente da Transformada Discreta de Fourier
Tamanho Máximo de Pacote	Tamanho Médio dos Pacotes	5^oComponente da Transformada Discreta de Fourier
Tamanho Mínimo de Pacote	Variância do Tamanho dos Pacotes	6^o Componente da Transformada Discreta de Fourier
Tempo Máximo Entre a Chegada de Dois Pacotes	Tempo Médio Entre a Chegada de Dois Pacotes	7^o Componente da Transformada Discreta de Fourier
Tempo Mínimo Entre a Chegada de Dois Pacotes	Variância do Tempo Entre a Chegada de Dois Pacotes	8^o Componente da Transformada Discreta de Fourier
Tamnaho Total em *Bytes*	1^o Quartil do Tamanho dos Pacotes	9^o Componente da Transformada Discreta de Fourier
Tamanho Total em Pacotes	3^o Quartil do Tamanho dos Pacotes	10^o Componente da Transformada Discreta de Fourier
Duração do Fluxo	1^o Quartil do Tempo Entre Chegada de Dois Pacotes	
	3^o Quartil do Tempo Entre Chegada de Dois Pacotes	

3.3. Algoritmos de Seleção de Características e Classificação de Tráfego

A seguir são descritos em maiores detalhes os algoritmos de seleção de características e classificação de fluxos de tráfego.

O primeiro algoritmo implementado para a seleção de características é chamado de Análise do Componente Principal (*Principal Component Analysis* - PCA). O método matemático PCA foi desenvolvido por Karl Pearson em 1901 [Pearson 1901]. Esse método avalia a correlação existente entre as diferentes variáveis de um problema, neste caso,

as características dos fluxos de tráfego, criando os chamados Componentes Principais. A quantidade de componentes criadas é igual ao número de características existentes no problema (no caso deste trabalho, trinta e três características). As componentes principais são compostas pelas variáveis originais multiplicadas por um peso. Esse peso varia de -1 a 1 e quanto mais próximo de 1 mais importante a variável é para a componente.

O segundo algoritmo é o Algoritmo Genético. Esse algoritmo realiza uma busca heurística no espaço de solução através da combinação de soluções (*crossover*) e inserção de novas características às soluções (*mutation*), realizando assim um processo de evolução das soluções a cada geração [Haupt 1998]. Os elementos que compõem esta classe de algoritmos são: uma população na qual cada indivíduo é uma possível solução para o problema; uma função de avaliação, sendo que esta função avalia a qualidade de cada indivíduo, ou seja, a qualidade da solução; e a quantidade de gerações que devem ser executadas.

Para a classificação de fluxos de tráfego, foi implementado o algoritmo Máquina de Vetores de Suporte (*Support Vector Machine* - SVM). O algoritmo SVM é utilizado em diversas áreas, desde o reconhecimento de imagens à bioinformática. No contexto de rede de computadores, especificamente para a classificação de tráfego, já foi comprovada a eficácia deste algoritmo para a detecção de ataques maliciosos, tais como DDoS [Mukkamala 2003]. O princípio de funcionamento do SVM é a construção de um hiperplano que separa as classes de um problema em um espaço N-dimensional, onde N é a quantidade de características do problema. Essa separação procura maximizar a distância entre cada uma das classes e é posicionada entre os dois pontos mais próximos pertencentes a duas classes diferentes [Bennett 2000].

3.4. Estendendo os Módulos de Seleção e Classificação

A seguir são descritos os algoritmos propostos para estender a arquitetura descrita em 3.1. Para a seleção de características é proposto o SBS e para a classificação de tráfego é proposto o K-means.

O algoritmo *Sequential Backward Selection* (SBS) é classificado como um filtro. Referente ao seu funcionamento, esse algoritmo elimina as características que mais interferem na classificação. O processo de escolha da característica a ser eliminada consiste em, partindo de um subconjunto, avaliar a qualidade deste subconjunto com cada característica eliminada uma a uma. O subconjunto escolhido para a próxima iteração é o subconjunto inicial com uma característica a menos que apresente o melhor desempenho. O SBS encerra quando, em alguma iteração, todos os possíveis novos subconjuntos resultam em um decréscimo na qualidade.

O SBS não garante que o subconjunto final é o melhor subconjunto possível. O subconjunto final é sempre um máximo local. Isso se deve ao fato de que uma característica, depois de removida, nunca é adicionada novamente ao subconjunto. Dessa forma, o SBS não avalia características que possam se tornar relevantes ao problema com a eliminação de outras características.

O segundo algoritmo proposto, para a classificação de fluxos de tráfego, é o K-means. Esse algoritmo é baseado em aprendizado não supervisionado, e sua utilização é voltada para processos de clusterização. O K-means separa as amostras em um espaço N-dimensional em *clusters*, cuja partição minimiza o erro quadrático médio entre o centro

dos *clusters* e cada ponto pertencente a este. O procedimento, de acordo com MacQueen [MacQueen 1967], consiste em iniciar k *clusters*, cada um sendo um ponto randomicamente selecionado no espaço, e a cada iteração adicionar a cada *cluster* o ponto que minimiza a média da distância entre eles.

4. Experimentos e Resultados

Nesta seção é descrito o ambiente em que o sistema apresentado anteriormente foi implementado, os experimentos realizados, e a parametrização dos algoritmos de seleção de características SBS, PCA e GA. Por fim, são apresentados os resultados obtidos com a classificação de fluxos de tráfego utilizando os algoritmos SVM e K-means.

4.1. Ambiente e Casos de Teste

Para validação do sistema proposto, foi criado um ambiente para testes composto por uma topologia em árvore com vinte e um switches, sessenta e quatro hosts, utilizando o emulador Mininet e um controlador Floodlight. Foi escolhida uma topologia em árvore por ser escalável e muito utilizada na prática. Adicionalmente, sobre esse ambiente são executados um módulo gerador de tráfego e o mecanismo de coleta, extensão e análise de características de fluxos descritos na Seção 3.

Referente à implementação, tanto o sistema quanto os programas geradores de tráfego foram implementados em Python v2.7.6. A estrutura de dados que armazena o conjunto de características é um dicionário cuja chave é a 5-tupla identificadora de fluxo descrita na Seção 3.2, e os arquivos de treinamento e teste são salvos em formato .csv. O algoritmo SBS foi implementado em Python v2.7.6 enquanto os algoritmos SVM, PCA, GA, e K-means foram implementados em R v3.0.2 utilizando as bibliotecas e1071, psych, genalg e cluster.

Os fluxos de tráfego gerados na rede são de quatro tipos diferentes: *(i)* um servidor HTTP é alvo de ataques DDoS partindo de determinados *hosts*; *(ii)* um servidor de stream de vídeo hospedado em um *host* e os demais *hosts* assinam este serviço, sendo que este tráfego é gerado utilizando o programa VLC[2] tanto no servidor quanto nos clientes; *(iii)* trocas de arquivos entre os *hosts* utilizando o protocolo FTP; e *(iv)* tráfego de *background* gerado através do Scapy[3].

Os três cenários de teste criados para a avaliação deste trabalho são compostos pelos quatro perfis de tráfego. A diferença entre cada cenário é a proporção estabelecida para cada perfil de tráfego. Tais proporções foram definidas seguindo a quantidade de fluxos que é normalmente encontrada na rede [Isolani et al. 2015]. A Tabela 2 apresenta os três cenários e a proporção que cada perfil de tráfego representa em cada cenário. Para os cenários B e C é aumentada a porcentagem de ataques DDoS e é diminuída a porcentagem dos demais perfis de tráfego. Assim, é possível observar o impacto que a proporção de fluxos utilizada no treinamento possui na acurácia do algoritmo de classificação de fluxos de tráfego.

Em cada cenário de teste são utilizados no total quinze mil amostras de treinamento e duas mil e quinhentas amostras de teste para o SVM. Igualmente, são utilizadas quinze mil amostras para a clusterização realizada pelo K-means.

[2]http://www.videolan.org/vlc/index.html

[3]http://www.secdev.org/projects/scapy/

Tabela 2. Proporção de volume de tráfego para cada cenário de teste.

	Ataques DDoS	Tráfego FTP	Stream de Vídeo	Background
Cenário A	10%	10%	50%	30%
Cenário B	30%	10%	40%	20%
Cenário C	50%	7.5%	25%	17.5%

Para permitir a comparação entre a qualidade de cada subconjunto criado, foram utilizadas as métricas de acurácia para o algoritmo SVM, ou seja, a porcentagem de amostras de ataques DDoS identificadas, e o método de Silhouettes [Rousseeuw 1987] para o algoritmo K-means. Os valores de acurácia variam de 0% a 100% e Silhouettes de -1 a 1, sendo 1 a melhor qualidade.

4.2. Parametrização de Algoritmos

O SBS só possui um parâmetro a ser definido: a quantidade de características a serem eliminadas. O valor definido é o tamanho inicial do conjunto de características, *i.e*, trinta e três, pois quanto menor o conjunto de características menor a demanda de recursos para a classificação dos fluxos de tráfego.

O PCA e o GA possuem seus parâmetros definidos de acordo com o trabalho realizado por [da Silva et al. 2015]: são gerados trinta e três componentes principais utilizando os pesos 0.025, 0.05 e 0.1 como *threshold* para o PCA; e para o GA são executadas cem gerações com uma população de duzentos indivíduos e probabilidade de mutação 0.01.

4.3. Experimentos e Resultados da Classificação de Fluxos de Tráfego

O módulo *Flow Classifier* foi instanciado com os algoritmos SVM e K-means. Esses algoritmos foram escolhidos por utilizarem duas formas diferentes de ML: enquanto o SVM utiliza aprendizado supervisionado, o K-means realiza um processo de clusterização não supervisionado. As subseções seguintes apresentam os resultados dos experimentos realizados para os dois algoritmos de classificação.

4.3.1. SVM

Inicialmente, é avaliada a classificação de fluxos utilizando o SVM. Para tanto, foi comparada a acurácia da classificação utilizando o conjunto completo, com a acurácia utilizando os subconjuntos criados por cada um dos três algoritmos de seleção de características. Os gráficos da Figura 2 apresentam a evolução da acurácia do subconjunto a cada iteração do algoritmo SBS. Podem ser observadas nesses gráficos duas informações importantes: primeiro, a última iteração resulta em um decréscimo da acurácia. Isto indica o momento em que o algoritmo interrompe a execução. Segundo, existem momentos em que a acurácia não sofre alteração (cenários A e B). Isto mostra que o algoritmo sempre busca o menor subconjunto possível, mesmo quando a acurácia não se altera com a eliminação de uma característica.

A Tabela 3 compara a acurácia obtida com o melhor subconjunto de características

criado por cada algoritmo de seleção de características[4]. O tamanho de cada subconjunto de características é informado entre parênteses após a acurácia do mesmo.

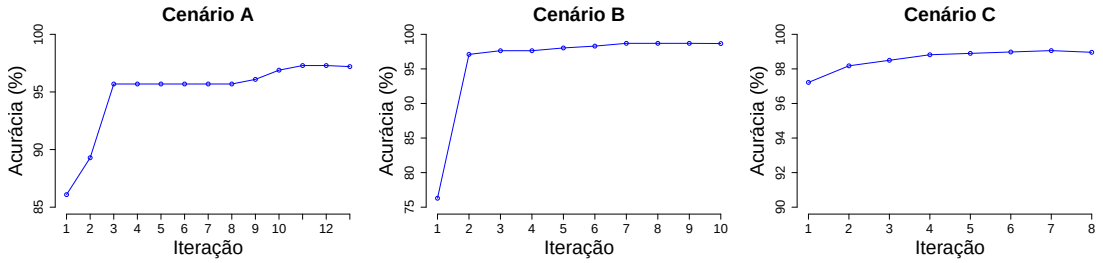

Figura 2. Evolução da acurácia do subconjunto a cada iteração do algoritmo SBS.

Tabela 3. Comparativo entre a acurácia dos subconjuntos criados para a classificação utilizando algoritmo SVM.

	Todas Características	SBS	PCA	Algoritmo Genético
Cenário A	86%	97.2% (20)	94.8% (12)	97.2% (21)
Cenário B	76.26%	98.53% (24)	96.93% (8)	98.67% (21)
Cenário C	97.2%	99.04% (26)	99.04% (10)	99.12% (21)

4.3.2. K-means

O K-means possui alguns parâmetros a serem definidos, incluindo: a quantidade de *clusters* a serem criados, e a quantidade mínima e máxima de iterações. Devido ao fato de cada um dos três cenários de teste possuir quatro perfis de tráfego diferentes, o K-means é configurado para criar quatro *clusters*, cada um representando um perfil de tráfego diferente. Além disso, para as quantidades mínimas e máximas de iterações são utilizados os valores padrão de cem e mil iterações, respectivamente.

Igualmente ao SVM, foram executados os três algoritmos de seleção de características com o objetivo de maximizar a Silhoutte da clusterização realizada pelo K-means. Os parâmetros dos algoritmos de seleção de características utilizados foram os mesmos utilizados para o SVM. A Figura 3 apresenta a evolução da Silhouette a cada iteração do algoritmo SBS. Adicionalmente, a Tabela 4 apresenta os resultados para cada um dos três algoritmos de seleção de características e o tamanho de cada subconjunto.

As figuras 4, 5 e 6 apresentam a clusterização realizada pelo K-means para os cenários A, B e C, respectivamente. Como pode ser observado, para cada figura, são comparados os resultados do conjunto de características completo (à esquerda) com os os resultados dos subconjuntos criados por cada um dos algoritmos de seleção de características.

[4]Os resultados obtidos para os algoritmos de seleção PCA e Algoritmo Genético não são detalhados nesse artigo por restrições de espaço. No entanto, informações adicionais podem ser encontradas em [da Silva et al. 2015].

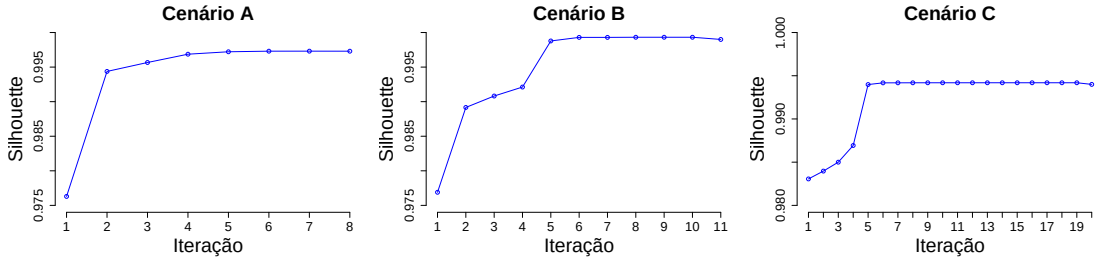

Figura 3. Evolução da qualidade da clusterização do subconjunto a cada iteração do SBS.

Tabela 4. Tabela comparativa entre a Silhueta dos subconjuntos criados para clusterização utilizando K-means.

	Todas Características	SBS	PCA	Algoritmo Genético
Cenário A	0.976295	0.997305 (25)	0.997305 (12)	0.983541 (24)
Cenário B	0.976893	0.999317 (23)	0.999092 (5)	0.991594 (17)
Cenário C	0.983061	0.994187 (14)	0.988336 (15)	0.983100 (24)

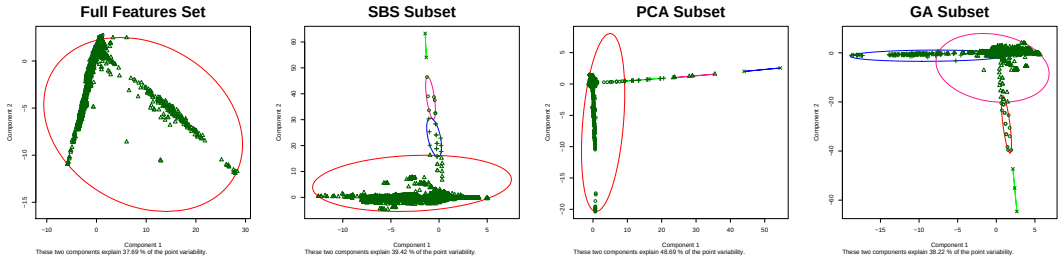

Figura 4. Resultado da clusterização para o cenário A.

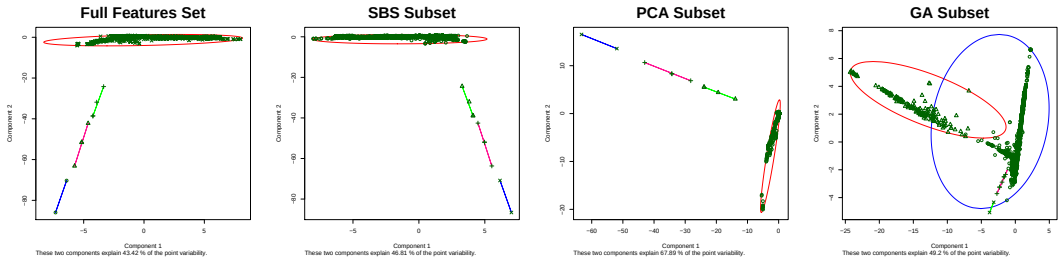

Figura 5. Resultado da clusterização para o cenário B.

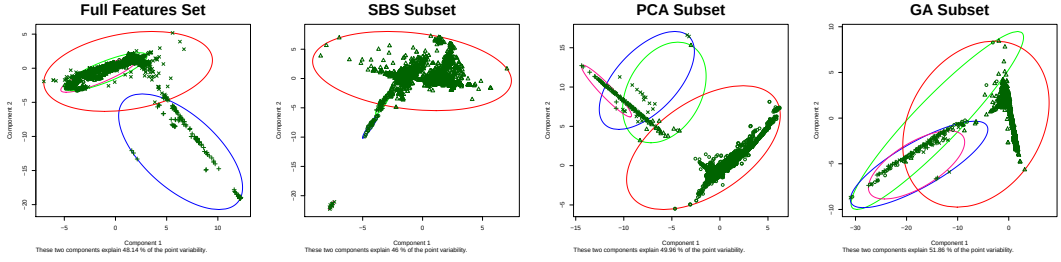

Figura 6. Resultado da clusterização para o cenário C.

4.3.3. Discussão sobre os resultados obtidos

Os gráficos apresentados na Figura 7 resumem os resultados obtidos com a execução dos algoritmos de classificação utilizando o conjunto completo de características e os subconjuntos compostos pelas características selecionadas pelos algoritmos de seleção. Em complemento, a Tabela 5 apresenta o tempo de execução de cada algoritmo de seleção de características.

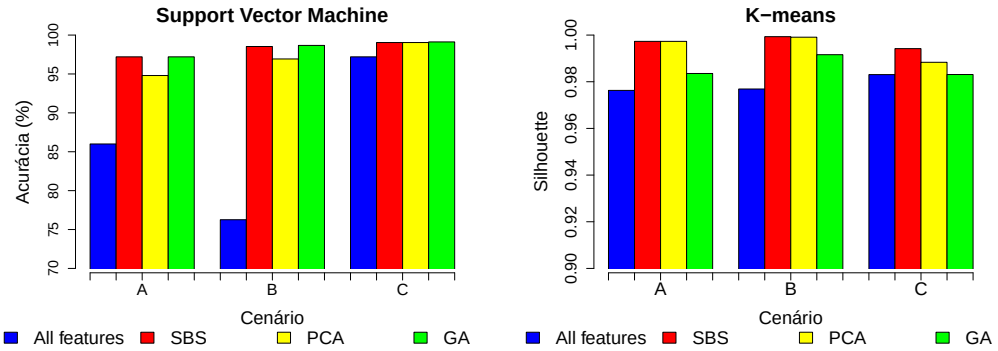

Figura 7. Gráficos comparativos da qualidade da solução oferecida pela utilização dos três algoritmos de seleção de características.

Tabela 5. Tempo de execução dos algoritmos de seleção de características.

	SVM			K-Means		
	SBS	**PCA**	**GA**	**SBS**	**PCA**	**GA**
Cenário A	21min 42s	1min 51s	16h 36min	22min 10s	3min 13s	35h 42min
Cenário B	19min 34s	2min 07s	15h 40min	26min 14s	2min 59s	29h 42min
Cenário C	12min 38s	1min 47s	11h 11min	42min 57s	3min 06s	35h 59min

Observando o gráfico à esquerda na Figura 7, pode ser observado que em todos os três cenários existe uma evolução na acurácia do algoritmo SVM, mostrando que a seleção de caraterísticas tem um impacto positivo na classificação de fluxos de tráfego utilizando o SVM. Além disso, o SBS apresenta um melhor desempenho que os outros dois algoritmos por apresentar uma acurácia parecida a do algoritmo genético e um tempo de execução de em média 17min58s.

Semelhantemente, observando os valores de Silhouette para cada um dos subconjuntos, pode-se afirmar que existe uma melhoria na qualidade da clusterização. Assim, identificando a qual classe uma das amostras de um *cluster* pertence, é possível afirmar com mais acurácia que todas as demais amostras do *cluster* pertencem à mesma classe. Além disso, observando o tempo de execução e o resultado obtido por cada algoritmo de seleção de características, é possível afirmar que o SBS apresenta um desempenho superior aos demais, por apresentar o subconjunto com a melhor Silhouette e um tempo de execução de em média 30min27s.

5. Conclusão

Neste trabalho, foi explorada uma arquitetura modular estendendo seus módulos de seleção de características e classificação de tráfego com mais dois algoritmos: Seleção

Sequencial Inversa (Sequential Backward Selection - SBS) e K-means. Os cenários e perfis de tráfego utilizados nos experimentos demonstram que a precisão dos algoritmos de classificação depende tanto do conjunto utilizado para treinamento e clusterização de algoritmos de aprendizado supervisionado (SVM) quanto da seleção de características. Através dos resultados apresentados nos experimentos, pode ser observado que é possível alcançar níveis de precisão maiores aumentando a proporção de ataques DDoS utilizados para o treinamento do SVM. Baseando-se nos resultados obtidos, também é possível concluir que o algoritmo SBS tem um desempenho muito bom por apresentar uma grande melhoria na qualidade da classificação do SVM e na clusterização do K-means e também um tempo de execução muito inferior ao algoritmo genético sem perda de qualidade.

Possíveis extensões que podem ser adicionadas a este trabalho incluem a utilização de tráfego real e não apenas tráfego sinteticamente criado. Além disso, pretende-se analisar outras abordagens para a coleta de características como *Deep Packet Inspection* (DPI) e também investigar a criação de novas características de fluxos de tráfego.

Agradecimentos

Este trabalho é apoiado por ProSeG - Segurança da Informação, Proteção e Resiliência em Smart Grids, um projeto de pesquisa financiado pelo MCTI/CNPq/CT-ENERG # 33/2013.

Referências

(2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517.

(2008). NOX: Towards an operating system for networks. *SIGCOMM Comput. Commun. Rev.*, 38(3):105–110.

Auld, T., Moore, A., and Gull, S. (2007). Bayesian neural networks for internet traffic classification. *Neural Networks, IEEE Transactions on*, 18(1):223–239.

Bennett, Kristin P. e Campbell, C. (2000). Support vector machines: Hype or hallelujah? *SIGKDD Explor. Newsl.*, 2(2):1–13.

Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W. (1998). RFC2475 - An Architecture for Differentiated Services. RFC 2475, RFC Editor.

Blum, A. L. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artif. Intell.*, 97(1-2):245–271.

da Silva, A. S., Machado, C. C., Bisol, R. V., Granville, L. Z., and Schaeffer-Filho, A. (2015). Identification and selection of flow features for accurate traffic classification in SDN. *14th IEEE International Symposium on Network Computing and Applications (NCA 2015)*, pages 137–141.

Erickson, D. (2012). Floodlight java based openflow controller. http://www.projectfloodlight.org/floodlight/.

Feamster, N., Rexford, J., and Zegura, E. (2013). The road to SDN. *Queue*, 11(12):20:20–20:40.

Guyon, Isabelle e Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182.

Haupt, Randy L. e Haupt, S. E. (1998). *Practical Genetic Algorithms*. John Wiley & Sons, Inc., New York, NY, USA.

Isolani, P. H., Araujo Wickboldt, J., Both, C., Rochol, J., and Zambenedetti Granville, L. (2015). Interactive monitoring, visualization, and configuration of openflow-based SDN. In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pages 207–215.

Jolliffe, I. (1986). *Principal Component Analysis*. Springer Verlag.

Kim, H., Claffy, K., Fomenkov, M., Barman, D., Faloutsos, M., and Lee, K. (2008). Internet traffic classification demystified: Myths, caveats, and the best practices. In *Proceedings of the 2008 ACM CoNEXT Conference*, CoNEXT '08, pages 11:1–11:12, New York, NY, USA. ACM.

Langley, Pat e Sage, S. (1994). Oblivious decision trees and abstract cases. In *Proc. AAAI-94 Workshop on case-based reasoning*, pages 113–117.

Machado, C. C., Granville, L. Z., Schaeffer-Filho, A., and Wickboldt, J. A. (2014). Towards SLA policy refinement for QoS management in software-defined networking. In *Advanced Information Networking and Applications (AINA-2014), 2014 28th IEEE International Conference on*, pages 397–404. IEEE.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif. University of California Press.

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74.

Mukkamala, S. e Sung, A. (2003). Detecting denial of service attacks using support vector machines. In *Fuzzy Systems, 2003. FUZZ '03. The 12th IEEE International Conference on*, volume 2, pages 1231–1236 vol.2.

Nguyen, T.T.T. e Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. *Comms. Surveys Tutorials, IEEE*, 10(4):56–76.

Oh, I.-S., Lee, J.-S., and Moon, B.-R. (2004). Hybrid genetic algorithms for feature selection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(11):1424–1437.

ONF (2012). Software defined netwoking: The new norm for networks. https://www.opennetworking.org.

ONF (2014). Openflow switch specification v1.5.0. https://www.opennetworking.org/.

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, 2(11):559–572.

Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65.

Yu, Lei e Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. pages 856–863.