Um Sistema de Detecção de Ameaças Distribuídas de Rede baseado em Aprendizagem por Grafos

Igor Jochem Sanz, Martin Andreoni Lopez, Gabriel Antonio Fontes Rebello e Otto Carlos Muniz Bandeira Duarte

[1]Universidade Federal do Rio de Janeiro – GTA/UFRJ/PEE-COPPE – Brasil

Abstract. *The increase of Internet of Things connected devices results in vulerabilities exploitation attacks at unimaginable scales. Therefore, effectively detecting port scan techniques and distributed denial of service attacks becomes essential. This paper proposes a intrusion detection system for distributed threat detection in real time based on a graph-learning approach. Different metrics are extracted from a graph analysis of the incoming traffic samples, resumed in time windows, to be incorporated into the inital flow features before being preprocessed. The proposed system is evaluated through three traffic datasets: real traffic of a Brazlian network operator and a synthetic traffic produced in our lab. Results show that the enrichment by graph analysis improved the detection accuracy on up to 15,7%. On some scenarios, using only graph-enriched features reduced the number of false negatives on up to 1430 times.*

Resumo. *O aumento de dispositivos conectados à Internet das Coisas resulta em ataques de exploração de vulnerabilidades em escalas inimagináveis. Portanto, detectar com eficiência varredura de portas e ataques distribuídos de negação de serviço torna-se essencial. Este artigo propõe um sistema de detecção, em linha (online), de ameaças distribuídas de rede baseado em aprendizagem enriquecida por grafos. Diferentes métricas são extraídas a partir de uma análise por grafos em janelas de tempo, que são incorporadas às características originais de fluxos antes de serem pré-processadas. O sistema proposto é avaliado através de dois conjuntos de dados de tráfego: tráfego real de uma operadora de rede brasileira e tráfego sintético produzido em laboratório. Os resultados mostram que o enriquecimento pela análise de grafos melhorou em até 15,7% a acurácia de detecção. Em alguns cenários, utilizar somente as características inferidas por grafos reduziu o número de falsos negativos em até 1430 vezes.*

1. Introdução

Ataques às redes de computadores são uma das principais ameaças para um mundo totalmente conectado. O crescente aumento de dispositivos conectados à Internet, traz consigo uma gama de vulnerabilidades não exploradas. Com o advento da Internet das Coisas (*Internet of Things* - IoT), a descoberta de vulnerabilidades pode afetar milhões de dispositivos simultaneamente. Ataques de negação de serviço e de varredura de portas nestes dispositivos são pontos cruciais na exploração de vulnerabilidades e execução de

Este trabalho foi realizado com recursos do CNPq, CAPES, FAPERJ e FAPESP (2015/24514-9, 2015/24485-9 e 2014/50937-1).

ataques em larga escala. Ataques cada vez mais volumosos e com maior número de dispositivos envolvidos são registrados na Internet por ano, e este número já alcançou 10 ataques distribuídos de negação de serviço (*Distributed Denial of Service* - DDoS) com taxa de dados superior a 300 GB/s no ano de 2016 [Akamai 2017]. Além disso, redes zumbis (*botnets*) compostas de dispositivos IoT infectados foram responsáveis pelo maior ataque de negação de serviço de origem distribuída já detectado com uma taxa recorde de 1 TB/s [Kolias et al. 2017].

Portanto, mecanismos de segurança que detectem estes tipos de ataques de forma acurada são imprescindíveis para uma Internet do Futuro segura. As técnicas tradicionais para detectar estes tipos de ameaças carecem de soluções eficientes e rápidas para tratar grande volume de dados de tráfego. Entretanto, algoritmos de aprendizado de máquina para detectar intrusões vem se tornando cada vez mais utilizados, pois a alta capacidade de processamento distribuído de um aglomerado de máquinas auxiliado por arcabouços de processamento de fluxo permite a construção de algoritmos complexos e eficientes para tratar os dados em linha (*online*).

Este artigo propõe um sistema de detecção de intrusão que utiliza técnicas de aprendizado de máquina e análise de grafos em linha para detectar ameaças distribuídas de rede. A contribuição principal deste trabalho é o enriquecimento em linha de características inferidas a partir de uma análise baseada em grafos. Diferentes métricas são inferidas de *snapshots* de janelas de tempo do tráfego de entrada e incorporadas às características originais do fluxo de rede. O sistema proposto é avaliado quanto a classificação para dois conjuntos de dados de tráfego de rede: um conjunto de dados reais de uma grande operadora de telecomunicações brasileira e um conjunto de dados sinteticamente produzidos no laboratório GTA/UFRJ. Os resultados obtidos demonstram que o enriquecimento traz ganhos significativos para a detecção de ataques distribuídos de negação de serviço e de varredura de porta distribuída, sem comprometer a detecção em linha.

O restante deste artigo está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados. A Seção 3 apresenta o sistema de detecção proposto e as características de dados de tráfego inferidas para classificação. A Seção 4 detalha os conjuntos de dados rotulados utilizados na classificação. A Seção 5 introduz os métodos de classificação e de seleção de características utilizados para avaliação. A Seção 6 apresenta e discute os resultados obtidos. Por fim, a Seção 7 conclui o artigo.

2. Trabalhos Relacionados

Diferentes abordagens utilizando aprendizado de máquina para detecção de ameaças estão presentes na literatura. As técnicas de aprendizado de máquina podem ser supervisionadas ou não supervisionadas, dependendo se o conjunto de dados estiver rotulado. Na área de redes a análise supervisionada é utilizada para a classificação de ataques. Entre as técnicas de classificação mais conhecidas estão as redes neurais, as árvores de decisão e as Máquinas de Vetores de Suporte (*Support Vector Machine* - SVM) [Buczak and Guven 2015, Nguyen and Armitage 2008]. Na análise não supervisionada, não há informações sobre a classe da amostra. A detecção de padrões aplica esse tipo de análise. Lakhina *et al.* propõem o uso de entropia de amostra para detecção de anomalia. Eles mostram que a métrica, combinada com IPs e portas de origem e destino e a análise de volume, pode detectar múltiplas fontes de anomalias [Lakhina et al. 2005].

Liu *et. al.* propõem uma abordagem na detecção de ameaças nas comunicações HTTP utilizando técnicas de análise de grafos [Liu et al. 2014]. No grafo quasi bipartido, os nós correspondem aos endereços IP clientes e servidores enquanto as arestas são as conexões, ponderadas de acordo com a saída de um classificador baseado em fluxo. O utilizado para a classificação de clientes suspeitos na rede é um algoritmo de propagação de pontuação alternada de duas fases no grafo. Grafos de Dispersão de Tráfego [Iliofotou et al. 2011] são usados para a classificação de tráfego par a par *(Peer to Peer* - P2P) na rede central da Internet. Os nós dos grafos são compostos por endereços IPs e cada aresta representa um tipo de interação entre dois nós. Como resultado a proposta melhora a detecção até seis vezes quando comparado com BLINC [Karagiannis et al. 2005], um método de classificação de aplicativos baseado nos comportamentos do hospedeiros de origem na camada de transporte. Chowdhury *et. al.* utiliza mapas auto organizados (*Self-Organizing Maps* - SOM) em grafos variantes no tempo para a detecção de *botnets* [Chowdhury et al. 2017]. Os nós utilizados são os endereços IPs e as arestas denotam a conexão entre dois endereços IP. No trabalho unicamente sete métricas dos grafos, como grau de entrada, grau de saída, peso do grau de entrada, peso do grau de saída, coeficiente de intermediação, *betweenness* do nó, centralidade do autovetor são usados como características para a detecção.

A ferramenta CATRACA [Andreoni Lopez et al. 2017a] é uma função virtual de rede para a detecção de ameaças em linha. A CATRACA utiliza as árvores de decisão para realizar a classificação de tráfego de diferentes sensores distribuídos na rede. Lobato *et. al.* implementam cinco algoritmos de detecção de ameaças com algoritmos adaptativos [Lobato et al. 2017]. O trabalho implementa o algoritmo do gradiente estocástico descendente e máquinas de vetores de suporte para a classificação de ameaças *online* e o algoritmo de entropia para a detecção de ameaças desconhecidas, *zero-day*.

Diferentemente dos trabalhos anteriores, este artigo propõe um sistema de detecção de intrusão acurado para detecção de ameaças de rede que realiza enriquecimento dos dados em linha através de uma análise baseada em grafos estáticos em conjuntos de dados atuais. Os dados enriquecidos permitem inferir padrões de comportamento de grupo do conjunto de amostras em uma janela de tempo, que não são possíveis de serem detectados ao analisar fluxos individualmente. Tais padrões são característicos de ataques de redes, como varredura de portas e negação de serviço, inclusive os ataques efetuados de forma distribuída. A análise deste artigo é feita em conjuntos de dados recentes e provê uma análise das ameaças de intrusão nas redes modernas.

3. O Sistema de Classificação Proposto

A arquitetura do sistema de detecção de intrusão em linha proposta é dividida em quatro módulos e uma base de dados histórica. Estes módulos estão compreendidos entre uma nuvem dedicada para processamento distribuído de dados e sensores que são responsáveis pela coleta dos dados.

O módulo de captura de dados consiste em ferramentas de coleta de tráfego executadas em sensores distribuídos na rede. Os sensores podem ser instanciados como máquinas virtuais em ambientes virtualizados ou através de máquinas físicas com espelhamento de tráfego de um *link* da rede. A partir da captura de tráfego em linha, este módulo ainda é responsável por abstrair os pacotes capturados em quíntuplas de fluxo em uma janela

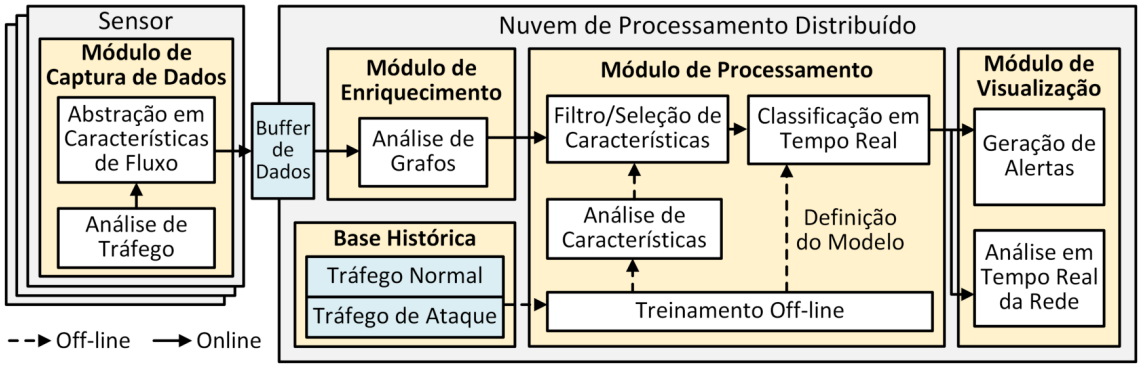


Figura 1. Arquitetura do sistema de classificação proposto. A arquitetura é dividida em quatro módulos principais. O módulo de captura de dados, o módulo de enriquecimento, o módulo de processamento e o módulo de visualização.

de tempo bem definida, além da extração de características de fluxo. Com o objetivo de detectar ameaças de rede como varredura de portas e ataques de negação de serviço, a análise de fluxos é configurada para agrupar pacotes com mesmo IP de origem e IP de destino, de forma que 26 características numéricas de cabeçalho de pacote são aferidas a partir deste agrupamento. Uma descrição detalhada das características utilizadas é apresentada na Seção 3.2. As características abstraídas anteriormente são publicadas em um sistema gerenciador de filas de dados em linha instalado na nuvem de processamento distribuído. Este sistema pode receber dados de múltiplos sensores em localizações distintas da rede. Os dados recebidos são temporariamente armazenados no *buffer* de dados para serem lidos pelo módulo de enriquecimento.

O módulo de enriquecimento em linha é a principal contribuição deste artigo. A partir de uma análise do grafo estático de amostras em janela de tempo, 39 novas características são inferidas. O procedimento utilizado para enriquecimento e as novas características obtidas são detalhadas na Seção 3.1. Além do enriquecimento das características, a análise de grafos permite a detecção de ataques distribuídos que não são percebidas com técnicas de detecção por análises de fluxos individuais.

O módulo de processamento é responsável pela classificação das amostras enriquecidas. Um filtro de características é aplicado às amostras recebidas para reduzir a complexidade de processamento em linha. O filtro é definido a partir de algoritmos de seleção de características e de redução de dimensionalidade pré-estabelecidos e realizados de forma *off-line*. Por fim, a classificação em linha pode ser efetuada através de diferentes algoritmos de aprendizado de máquina, amostra por amostra, implementados através de arcabouços de processamento distribuído de fluxo. A definição do modelo é feita de forma *off-line* através do treinamento por um conjunto de dados rotulados e armazenados em uma base histórica. Para avaliação da proposta, são utilizados dois conjuntos de dados para treinamento e classificação.

Por fim, o módulo de visualização compreende uma interface entre o sistema e o usuário através da Internet. Esse módulo é responsável pela geração de alertas de tráfego suspeito e de uma análise da rede em linha.

3.1. Enriquecimento em linha por grafos

A proposta de enriquecimento em linha consiste em uma análise de grafos modelada a partir de amostras recebidas em uma dada janela de tempo. Um *snapshot*, ou foto, é definido como todo o conjunto de amostras coletadas durante janela de tempo de captura de pacotes. A modelagem do *snapshot* em um grafo direcionado é realizada considerando os endereços IPs como vértices e o envio de pacotes IP como aresta direcionada entre estes IPs. A Figura 2 exemplifica o grafo de um *snapshot* de uma janela de tempo contendo duas ameaças, uma varredura de porta e um ataque de negação de serviço distribuído (*Distributed Denial of Service* - DDoS).

Ressalta-se que ataques de redes distribuídos, como varredura de porta ou DDoS, possuem comportamentos que não demonstram ser maliciosos quando analisados em apenas um único fluxo desses ataques. Dessa forma, o enriquecimento baseado em grafos permite interpretar características desses ataques que só são visíveis quando um determinado comportamento de grupo é analisado. Além disso, ataques desse tipo possuem a característica de que todos os endereços IPs maliciosos envolvidos estão interligados em uma mesma componente conexa conectada, ou seja, um mesmo agrupamento de vértices que se conectam a uma distância finita. No exemplo da Figura 2, o grafo ilustrado possui 30 componentes conexas que são diferenciadas por cores, sendo duas delas diretamente relacionadas com ataques de rede.

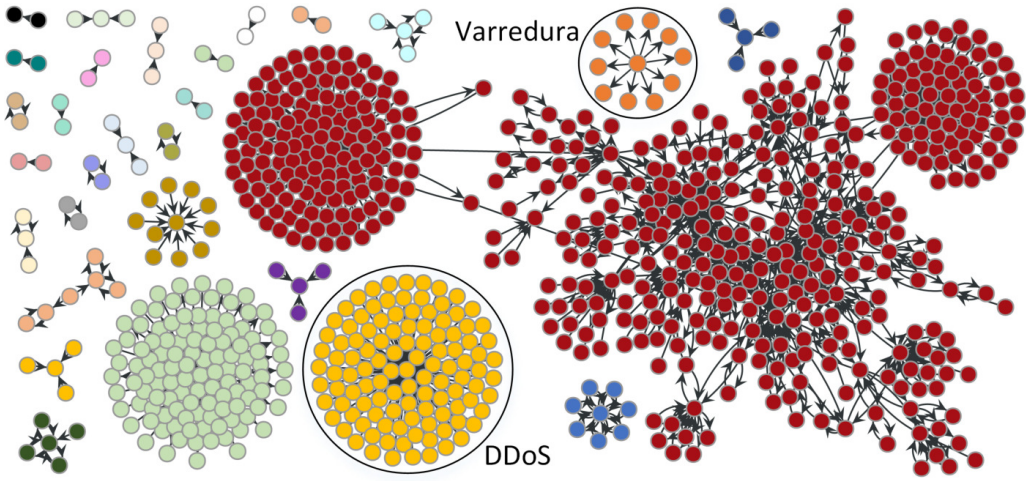


Figura 2. Grafo de um *snapshot* de uma janela de 2 segundos contendo um ataque de negação de serviço distribuído e uma varredura de porta. As diferentes cores diferenciam as 30 componentes conexas presentes.

Neste modelo de grafo, também são atribuídas às arestas um vetor de características do cabeçalho de pacotes IP, que permite inferir métricas utilizando quaisquer características do cabeçalho de pacote como peso para a aresta. Com o objetivo de detectar varredura de portas distribuídas e ataques de negação de serviço distribuídos, são selecionadas diversas características que se relacionam com a ocorrência destes ataques para ser usadas como pesos nas novas características. Estas características são utilizadas posteriormente como pesos para alguma das métricas inferidas da análise por grafos.

O Algoritmo 1 mostra o processo de extração dessas características após o recebimento do conjunto de amostras com as características originais contido na janela.

Um grafo G do conjunto de amostras é construído e são definidas todas as componentes conexas presentes em G. Dessa forma, a análise de grafos é realizada localmente por componente conexa evitando uma análise global do grafo, o que é muito mais custoso computacionalmente. Para cada componente, métricas locais são extraídas, como o total de vértices, o total de arestas, o total de *bytes* transmitidos, o total de portas TCP, etc. Essas métricas são posteriormente atribuídas como características para cada fluxo pertencente a esta componente. Além disso, métricas por vértices também são inferidas, como os graus de entrada e de saída do vértice, considerando como peso as características de cabeçalho TCP/IP. Por fim, para cada aresta são obtidas métricas relacionadas a fração de fluxos, bytes e pacotes em relação ao total da componente conexa e o coeficiente de intermediação (*betweenness*) da aresta. Após a extração de características, cada amostra inicial é enriquecida com as características locais da componente conexa que ela pertence, as dos dois vértices que representam o endereço IP de origem e o endereço IP de destino da aresta, e as inferidas da própria aresta.

Algoritmo 1: Enriquecimento de características pela análise baseada em grafos de um *snapshot* do tráfego.

Input : X: Matriz do conjunto de amostras originais
Output: Y: Matriz do conjunto de amostras enriquecidas

$G = construirGrafo(X)$
$Componentes = extrairComponentes(G)$
para cada $Subgrafo \in Componentes$ **faça**
  $LocalFeatures = extrairMetricasLocais(Subgrafo)$
  para cada $Vertice \in Subgrafo$ **faça**
    $VerticeFeatures = extrairMetricasVertices(Vertice)$
  fim
  para cada $Aresta \in Subgrafo$ **faça**
    $ArestaFeatures = extrairMetricasArestas(Aresta)$
  fim
fim
para cada $Aresta \in G$ **faça**
  $V1 = Aresta.Source$
  $V2 = Aresta.Destination$
  $Y[Aresta] = X[Aresta] + LocalFeatures +$
  $VerticeFeatures(V1) + VerticeFeatures(V2) +$
  $ArestaFeatures[Aresta]$
fim

3.2. Características Utilizadas

Neste tipo de agrupamento de fluxos de rede, as características originais inferidas do tráfego de rede são variáveis numéricas relacionadas ao conjunto de todos os pacotes transmitidos entre dois endereços IPs, como média, variância e quantidade de informações do cabeçalho TCP/IP. Nesta proposta, os pacotes de rede são abstraídos em fluxos na camada de rede, na qual um fluxo é definido como a sequência de pacotes de um endereço IP origem para o mesmo endereço de IP destino durante uma janela de tempo. Assim, cada fluxo IP-IP contém 26 características iniciais: quantidade de cada *flags* TCP (8); quantidade de pacotes TCP, UDP, ICMP e IP (4); quantidade de portas de origem e de destino (2); tamanho médio do cabeçalho e do pacote (2); quantidade de pacotes fragmentados e com bit de não fragmentação (2); número de tipos e de códigos ICMP (2);

quantidade de tipos de serviços (1); número de saltos TTL médio (1); quantidade de *bytes* transmitidos (1); quantidade de conexões TCP estabelecidas (1); quantidade de conexões UDP estabelecidas (1); e quantidade de quíntuplas distintas de fluxos (1);

Uma vez que os modelos de grafos são gerados para cada *snapshot* de uma janela conforme o Algoritmo 1, as 26 características de cada conjuntos de dados são enriquecidas com 39 métricas inferidas pela análise de grafos. Essas características são divididas em três categorias: a) **métricas locais**: número total de vértices (IPs) e arestas (conexões IP-IP) da componente conexa (2), número total de *bytes*, fluxos e pacotes transmitidos na componente conexa (3), número total de portas origem e destino (2); b) **métricas de aresta**: fração de *bytes*, de fluxos e de pacotes transmitidos na conexão IP em relação ao número total da componente conexa (3), *betweenness* da aresta (1); e c) **métricas de vértice**: grau simples de entrada e saída da origem e do destino da aresta (4), grau de entrada e saída, da origem e do destino, de pacotes TCP, UDP, ICMP e IP (16); e grau de entrada e saída considerando as portas de origem e de destino, dos vértices de origem e destino da aresta (8).

4. Conjunto de Dados Utilizados

Para avaliar o desempenho da abordagem proposta para detecção de anomalia foram realizados experimentos utilizando dois conjuntos de dados diferentes. O primeiro conjunto de dados foi criado pelos autores previamente. Esse conjunto de dados é obtido através da captura de pacotes de tráfego real do laboratório do Grupo de Teleinformática e Automação (GTA) da UFRJ. O tráfego capturado contém comportamento normal e ameaças reais de redes executadas de forma controlada. O conjunto de dados de ameaça possui 36 tipos de ameaças divididos em três categorias: a) **8 tipos de DoS**, ICMP *flood*, *land*, *nestea*, *punk*, *smurf*, SYN *flood*, UDP *flood*; b) **8 tipos de DDoS**, SYN *flood*, *teardrop*, *smurf*, *nestea*, spoofados e não spoofados; e c) **20 tipos de varredura de portas**, varreduras com *flags* de FIN, SYN, XMAS, NULL e ACK, executadas de forma horizontal e vertical, e distribuída e não distribuída [Sanz et al. 2017].

O segundo conjunto de dados é composto por dados de tráfego coletados de uma grande operadora de telecomunicações brasileira [Andreoni Lopez et al. 2017b]. O conjunto de dados contém informações de acesso real de 373 usuários residenciais de banda larga da cidade do Rio de Janeiro por um período de uma semana. O período é composto pelos dias 27 de fevereiro até o dia 5 de março de 2017. Por questões de privacidade os dados foram anonimizados. Como os dados são de clientes residenciais com acesso ADSL (*Asymmetric Digital Subscriber Line*), o tráfego capturado encontra-se encapsulado em sessões PPPoE (*Point-to-Point Protocol over Ethernet*), que foi desencapsulado utilizando a ferramenta Stripe. Além disso, como os dados são reais, não é possível assegurar que todos os fluxos são legítimos ou maliciosos. Por isso, o IDS baseado em assinaturas Suricata foi utilizado para filtrar o tráfego normal dos diferentes tipos de ameaças detectadas. Foram inseridos os 36 tipos de ataques de rede, dentre ataques de negação de serviço e de varredura de porta distribuídos, mesclando-se aos IPs pertencentes aos dos usuários domésticos da operadora.

5. Métodos de Seleção de Características e de Classificação Utilizados

Diversas técnicas podem ser utilizadas para reduzir a carga de processamento de algoritmos de classificação de tráfego. As técnicas incluem desde algoritmos de seleção

de características até a redução de dimensionalidade por autovetores. São apresentados e avaliados dois desses métodos com o objetivo de filtrar a quantidade de características a serem processadas pelo algoritmo de classificação. Além disso, neste artigo são apresentados e implementados alguns dos métodos mais usados em problemas de classificação de tráfego com aprendizado de máquina [Buczak and Guven 2015].

5.1. Algoritmos de Seleção de Características e Redução de Dimensionalidade

A quantidade de variáveis resultante da abstração em fluxos e enriquecimento dos dados é um ponto crítico para algoritmos de classificação de tráfego. Ter 65 características é um número elevado de quantidade de variáveis a serem obtidas em linha e ainda pode criar um possível *overfitting* dos dados. Portanto, o primeiro passo na análise exploratória foi reduzir todas as dimensões possíveis sem que houvesse perda significativa de informação. Na prática, isto significou um processo em três fases: i) a normalização do conjunto de dados; ii) a retirada das características de variância nula; e iii) o cálculo da matriz de correlação entre as variáveis restantes. Uma vez que todas as variáveis do problema são numéricas, utilizou-se a correlação de Pearson, definida como

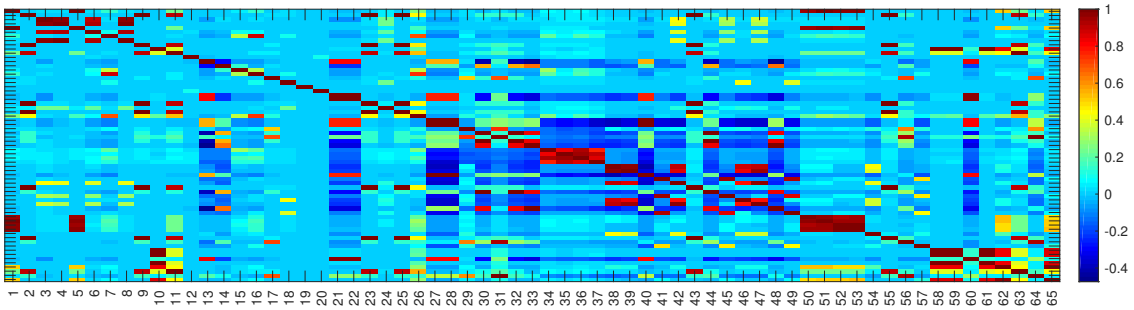
$$\rho_{X,Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}, \tag{1}$$

onde σ_X e σ_Y são os desvios padrões de cada variável, e μ_X e μ_Y suas respectivas médias aritméticas, para definir o valor das correlações entre cada par de variáveis do conjunto de dados. A matriz de correlações resultante para cada conjunto de dados são mostradas na Figura 3(a). Após o cálculo, eliminou-se todas as características de correlação maior que 0,9 e o conjunto final passou a ter 32 características para o conjunto de dados da operadora e 34 características para o conjunto de dados produzido em laboratório.

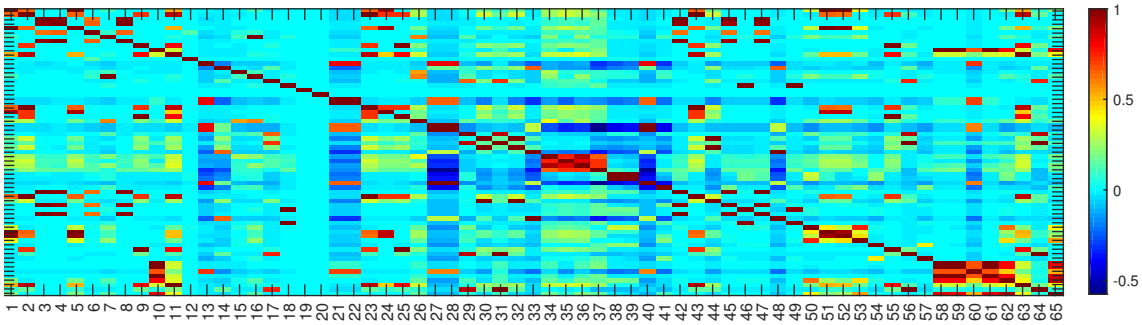
A **análise de componentes principais** (*Principal Component Analysis* - PCA) tem custo computacional baixo e é uma das técnicas de redução linear não supervisionada mais populares. O objetivo da PCA é encontrar combinações ortogonais s_i das características de entrada p que maximizam a maior variância. A direção com a maior variância projetada é chamada a primeira componente principal (PC), e cada componente em sucessão tem a maior variância sob a restrição de que é ortogonal aos componentes anteriores. As características de entrada p são uma combinação linear dos k componentes principais, e a direção que maximiza a variância é também aquela que minimiza o erro quadrático médio. Portanto, o PCA, além de reduzir a dimensionalidade dos dados de entrada, elimina a redundância causada pela correlação entre as características x_i. Como as componentes são classificadas em ordem da variância decrescente, o tamanho dos dados pode ser reduzido eliminando as principais componentes com menor variância. Usando as componentes principais mais importantes foi possível reconstruir uma boa aproximação dos dados originais, reduzindo de 65 para 51 características no conjunto de dados da operadora, e para 46 características no conjunto de dados GTA/UFRJ.

5.2. Algoritmos de Classificação

Para a classificação das ameaças discutidas, quatro classificadores de características distintas são utilizados: a árvore de decisão, o algoritmo bayesiano simples, as árvores impulsionadas por gradiente e uma rede neural. A escolha dos dois primeiros algoritmos visa estimar a eficiência de classificadores simples e de baixo processamento.



(a) Conjunto de dados da operadora de telecomunicações.



(b) Conjunto de dados GTA/UFRJ.

Figura 3. Matriz de correlação de todas as características utilizadas em cada conjunto de dados analisado. As características 1 a 26 pertencem as características iniciais inferidas do fluxo IP-IP e as características 27 a 65 são as obtidas pelo enriquecimento de análise de grafos.

Estes classificadores são capazes de realizar o treinamento do modelo de classificação com maior rapidez e, portanto, são mais rápidos na detecção de ameaças. Os algoritmos de conjunto (*ensemble*) e as redes neurais, por outro lado, baseiam-se na comparação de modelos menores intermediários para decidir a melhor hipótese de classificação dos dados. O custo de construção do modelo de classificação destes algoritmos é maior devido à complexidade de criar modelos menores e, em geral, estes algoritmos obtêm melhores resultados nas métricas de avaliação de classificadores. A decisão por estes algoritmos vem de sua larga utilização e comportamentos bem definidos para detecção de intrusão [Buczak and Guven 2015].

O algoritmo de **árvore de decisão** (*Decision Tree* - DT) é um algoritmo supervisionado que monta uma árvore onde cada nó é responsável pelo teste de um atributo do sistema. Os valores de probabilidade de cada classe são armazenados naquele nó e servem como parâmetros para a tomada de decisão. A cada nova amostra desconhecida que entra no modelo, o algoritmo percorre os nós avaliando os respectivos atributos para estimar a probabilidade daquela amostra pertencer a uma determinada classe. Dessa forma, usualmente não é necessário percorrer todos os atributos da amostra para realizar a classificação e poupa-se tempo de processamento. A geração da árvore inicia-se pela caracterização de um nó raiz, que possui meramente a probabilidade de cada classe na amostragem. A partir de então, o nó é dividido sucessivamente, de forma que cada filho represente uma nova característica da amostragem, associado à um conjunto de probabilidades para cada classe relativos a essa característica. Esse processo é repetido para todos os nós até que estes

atinjam probabilidade de 100% para alguma classe, configurando-se como um nó folha.

Na aplicação do **algoritmo bayesiano simples** (*Naive Bayes* - NB), parte-se do pressuposto de que cada característica do sistema não influencia no valor das demais características, de modo a simplificar a predição de classificação. A partir disso, o método calcula as probabilidades *a priori* para cada característica, ou um conjunto delas, de configurar uma determinada classe, com base nos dados de treinamento. Ao entrar uma nova amostra desconhecida, o algoritmo calcula para cada atributo qual a probabilidade de configurar cada uma das classes. O produtório de todas as probabilidades de cada característica resultará em uma probabilidade *a posteriori* dessa amostragem pertencer a cada uma das classes. O algoritmo então retorna a classificação que houver maior probabilidade estimada. Segundo o teorema de Bayes, a probabilidade *a posteriori* é

$$P[C|X] = \frac{P[X|C] * P[C]}{P[X]}, \tag{2}$$

em que X é a amostra desconhecida e C é a classe a ser analisada. Portanto, a probabilidade a posteriori é dada pelo produto da probabilidade condicional de X em C e da probabilidade *a priori* da classe C na amostragem, divididos por um termo de padronização dos atributos P[X], constante para cada amostra. O objetivo é maximizar o numerador para encontrar a classe que mais se adequa ao conjunto de características da amostra desconhecida.

A utilização de classificadores simples nem sempre é suficiente para produzir uma hipótese razoável sobre a classificação dos dados processados. O algoritmo de **árvores impulsionadas por gradiente** (*Gradient Boosted Tree* - GBT) propõe utilizar uma combinação incremental de árvores de decisão, de forma a minimizar a função objetivo de erro de predição. O modelo começa com apenas uma árvore, que produz uma função $f_1(x_i)$, onde x_i é o conjunto das i entradas. A cada passo, uma nova árvore é adicionada ao modelo e sua função $f_t(x_i)$ é somada às anteriores, de forma que cada saída predita é

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) = \sum_{t=1}^{T} f_t(x_i), \tag{3}$$

onde $\hat{y}_i^{(t)}$ é a predição da entrada i no passo t de adição de árvores de decisão e T é o número total de adições desde o início do algoritmo. A principal vantagem deste tipo de modelo é que pode-se conceber cada nova adição baseando-se no gradiente da função objetivo, de maneira a construir uma árvore que se especializa nas deficiências das árvores anteriores. Assim, a próxima árvore a ser construída age de maneira complementar às demais e contribui para a qualidade de predição do classificador.

O **perceptron multicamadas** (*Multilayer Perceptron* - MLP) é um algoritmo supervisionado de classificação que utiliza o método de treinamento *backpropagation*. O método consiste em ajustar os pesos de cada neurônio em uma rede neural de acordo com o gradiente de uma função de perda arbitrária. O objetivo é minimizar o erro de predição dos neurônios em relação à saída real de forma a otimizar seus pesos associados e, assim aumentar a acurácia e precisão da classificação. A função de perda mais comumente

utilizada é a média da distância euclidiana entre a saída real e a predita:

$$E = \frac{1}{2n} \sum_{x} ||y(x) - y'(x)||^2, \tag{4}$$

em que n é o número de entradas no conjunto de treinamento, x é a entrada avaliada e $y(x)$ e $y'(x)$ são, respectivamente, os valores de saída real e preditos. O ajuste dos pesos de um neurônio j da camada i é feito utilizando o método do gradiente descendente

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}, \tag{5}$$

onde η é um parâmetro de ajuste do modelo que controla a velocidade de atualização dos pesos. Os erros E_i de cada camada vão sendo retropropagados e os pesos ajustados de acordo com as derivadas parciais em relação a cada neurônio. A grande vantagem do *backpropagation* é poder separar classificações que não são divisíveis linearmente ou de forma trivial, podendo ajustar curvas que se adequem à amostragem. O MLP deve possuir ao menos três camadas: as camadas de entrada, de saída e uma ou mais camadas ocultas (internas). Cada camada oculta associa os dados de saída da camada anterior a uma função de ativação para cada neurônio e produz os dados de saída para a camada seguinte. Em particular, o MLP é um grafo totalmente conexo, e cada neurônio de uma camada está associado a todos os neurônios da camada anterior, até chegar na camada de saída que produz o resultado final da classificação. Para o escopo deste artigo, utilizou-se um MLP com 10 camadas, incluindo as camadas de entrada e saída.

6. Avaliação e Resultados

Um protótipo do sistema proposto é implementado e avaliado quanto a acurácia de classificação. A configuração de *hardware* utilizada na implementação dos algoritmos consiste em uma máquina Intel Xeon E5-2650 @ 2.00GHz, 16-core (32) com 512GB de memória e 21TB de disco. A janela de dois segundos foi utilizada por apresentar uma boa precisão com algoritmos para treinamento *offline* [Lobato et al. 2016]. São definidos cinco conjuntos de características para avaliação: i) o conjunto de 26 características de cabeçalho TCP/IP que são inferidos em linha diretamente da abstração de dados de tráfego em janelas de tempo; ii) o conjunto de 39 características inferidas a partir do enriquecimento em linha da análise por grafos; iii) o conjunto total enriquecido incluindo todas as 65 características inferidas; iv) o conjunto de características obtido através da redução de dimensionalidade pelo método PCA, que resultou em 51 características para o conjunto de dados da operadora e 46 características para o conjunto de dados GTA/UFRJ; e v) o conjunto de características obtidas após um filtro das características que apresentam o coeficiente de correlação de Pearson maior que 0,9, que resultou em 32 características para o conjunto de dados da operadora e 34 características para o conjunto de dados GTA/UFRJ.

Os conjuntos de características são utilizados para classificação através dos quatro métodos de aprendizado de máquina apresentados, árvores impulsionadas por gradiente (GBT), árvore de decisão (DT), algoritmo bayesiano simples (NB) e perceptron multicamadas (MLP). O treinamento de todos os algoritmos foi realizado utilizando o algoritmo k-fold de validação cruzada para melhorar a generalização dos modelos e evitar *overfitting* dos dados. Esta técnica consiste em separar sucessivamente o conjunto de dados

em k partições aleatórias, onde $k-1$ partições são utilizadas no treinamento e apenas uma na validação do modelo. Desta forma, após k ciclos, o modelo haverá iterado por todo o conjunto inicial e podemos avaliar todas as estatísticas de validação dos modelos de acordo com a média de todos os ciclos, garantindo maior confiança estatística no resultado. Neste trabalho, a validação foi realizada com 10 ciclos. Ainda, o conjunto de dados foi balanceado igualmente entre ameaças e tráfego normal, de forma a evitar uma classificação tendenciosa dos dados em função da classe mais presente.

Tabela 1. Acurácia de classificação do conjunto de dados da operadora de telecomunicações.

	GBT	**DT**	**NB**	**MLP**
Características TCP/IP (26)	99.97%	99.95%	81,46%	99,12%
Características de grafos (39)	99,99%	99.96%	95,38%	99,96%
Conjunto enriquecido (65)	99,99%	99,98%	94,26%	99,99%
Redução PCA (51)	99,99%	99.96%	96,03%	99,19%
Filtro Linear $\rho > 0,9$ (32)	99,99%	99.94%	94,60%	99,99%

Tabela 2. Acurácia de classificação do conjunto de dados GTA/UFRJ.

Algoritmo ML	**GBT**	**DT**	**NB**	**MLP**
Características TCP/IP (26)	99,98%	99,98%	75,14%	99,29%
Características de grafos (39)	99,98%	99,96%	82,60%	99,70%
Conjunto enriquecido (65)	99,97%	99,96%	80,32%	99,96%
Redução PCA (46)	99,94%	99,96%	94,65%	96,53%
Filtro Linear $\rho > 0,9$ (34)	99,97%	99,96%	90,24%	99,95%

As Tabelas 1 e 2 mostram os resultados de acurácia obtidos em todos os cenários de classificação avaliados para os dois conjunto de dados. Os resultados mostram que o enriquecimento das características aumentou a acurácia em relação às características TCP/IP sem enriquecimento para todos os algoritmos usando o conjunto de dados da operadora, e em dois dos quatro algoritmos para o conjunto de dados GTA/UFRJ. Ainda, as caraterísticas de grafos isoladas tiveram acurácia superior às características enriquecidas, mostrando que, neste caso, incluir certas características TCP/IP também introduzem informação não útil à classificação. Os algoritmos GBT, DT e MLP já apresentavam acurácias próximas a 100% com as 26 características TCP/IP, denotando que, nestes casos, o enriquecimento por grafos agrega pouco à detecção de ameaças. De todos os algoritmos avaliados, o algoritmo bayesiano simples apresentou o maior ganho de acurácia após o enriquecimento, 15,7% para o conjunto de dados da operadora e 9,9% para o conjunto de dados GTA/UFRJ. Além disso, após a aplicação do filtro de correlação linear de características, obteve-se uma acurácia de classificação muito próxima da acurácia do conjunto enriquecido, utilizando aproximadamente metade da quantidade de características. A única exceção foi para o algoritmo bayesiano simples no conjunto de dados GTA/UFRJ, que apresentou um ganho de 12% mesmo após a seleção de características. Os resultados indicam, portanto, que o enriquecimento agrega valor à detecção e, em especial, é mais eficiente para algoritmos com treinamento mais ingênuo, como o bayesiano simples. Por fim, uma melhora signficativa na detecção de ameaças ocorreu para o algoritmo MLP. As Tabelas 3, 4, 5 e 6 apresentam as estatísticas completas da classificação para os dois

Tabela 3. MLP no conjunto de dados originais da operadora. Recall: 98,4% e precisão: 99,9%

	Normal	Ameaça
Normal	351869	301
Ameaça	5721	357289

Tabela 4. MLP no conjunto de dados enriquecidos da operadora. Recall: 99,99% e precisão: 99,98%

	Normal	Ameaça
Normal	357586	62
Ameaça	4	357528

Tabela 5. MLP no conjunto de dados originais GTA/UFRJ. Recall: 99,3% e precisão: 99,3%.

	Normal	Ameaça
Normal	9508	69
Ameaça	66	9505

Tabela 6. MLP no conjunto de dados enriquecidos GTA/UFRJ. Recall: 99,99% e precisão: 99,99%

	Normal	Ameaça
Normal	9571	3
Ameaça	3	9571

conjunto de dados utilizados. O enriquecimento reduziu em até 1430 vezes a quantidade de falsos negativos para o conjunto da operadora e em até 22 vezes para o conjunto de dados GTA/UFRJ.

7. Conclusão

Este artigo propôs um sistema de detecção de intrusão em linha para detecção de ameaças distribuídas de rede utilizando uma abordagem baseada na análise de grafos. Para isso, uma arquitetura de detecção de intrusão, que inclui um módulo de enriquecimento em linha, é proposta para inferir características de análise de grafos a partir das amostras do tráfego coletadas em uma janela de tempo. Para cada amostra recebida, um grafo do *snapshot* desta janela é gerado e repartido em subgrafos de componentes conexas. Um algoritmo percorre as componentes conexas inferindo 39 novas características a partir de métricas locais, de vértices e de arestas das componentes.

Para todos os conjuntos de dados e algoritmos de classificação utilizados, o enriquecimento por uma análise baseada em grafos apresentou uma melhora na classificação de ameaças em relação às características analisando apenas o cabeçalho de pacote TCP/IP. Para algoritmos mais simples, o método proposto melhorou em até 15,7% a taxa de acerto. Em alguns cenários, utilizar somente as características inferidas por grafos demonstrou ser mais eficaz na detecção de ameaças de rede mesmo ao reduzir o número de falsos negativos em até 1430 vezes. Os resultados mostraram ainda que ao aplicar uma seleção de características através de filtro linear de Pearson, ou uma redução de dimensionalidade através na análise de componentes principais, pode-se reduzir o tempo para classificação a um custo pequeno na acurácia da classificação. Como trabalhos futuros, pretende-se estudar o enriquecimento baseado em grafos para a detecção de anomalias de tráfego de rede, e também testar novos conjuntos de características e métodos de classificação.

Referências

Akamai (2017). Q4 2016 State of the internet / security report. Technical Report 4, Akamai. Disponível em: https://content.akamai.com/pg7967-q4-soti-security-report.html.

Andreoni Lopez, M., Sanz, I. J., Menezes, D. M., Duarte, O. C. M. B., and Pujolle, G. (2017a). CATRACA: uma Ferramenta para Classificação e Análise Tráfego Escalável

Baseada em Processamento por Fluxo. In *Salão de Ferramentas do XVII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais - SBSeg'2017*, pages 788–795.

Andreoni Lopez, M., Silva, R. S., Alvarenga, I. D., Rebello, G. A. F., Sanz, I. J., Lobato, A. G. P., Mattos, D. M. F., Duarte, O. C. M. B., and Pujolle, G. (2017b). Collecting and Characterizing a Real Broadband Access Network Traffic Dataset. In *IEEE/IFIP 1st Cyber Security in Networking Conference (CSNet'17)*, Rio de Janeiro, Brazil.

Buczak, A. and Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys Tutorials*, (99):1–26.

Chowdhury, S., Khanzadeh, M., Akula, R., Zhang, F., Zhang, S., Medal, H., Marufuzzaman, M., and Bian, L. (2017). Botnet detection using graph-based feature clustering. *Journal of Big Data*, 4(1):14.

Iliofotou, M., Kim, H.-c., Faloutsos, M., Mitzenmacher, M., Pappu, P., and Varghese, G. (2011). Graption: A graph-based P2P traffic classification framework for the internet backbone. *Computer Networks*, 55(8):1909–1920.

Karagiannis, T., Papagiannaki, K., and Faloutsos, M. (2005). BLINC: multilevel traffic classification in the dark. In *ACM SIGCOMM Computer Communication Review*, volume 35, pages 229–240. ACM.

Kolias, C., Kambourakis, G., Stavrou, A., and Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. *Computer*, 50(7):80–84.

Lakhina, A., Crovella, M., and Diot, C. (2005). Mining anomalies using traffic feature distributions. In *ACM SIGCOMM Computer Communication Review*, volume 35, pages 217–228. ACM.

Liu, L., Saha, S., Torres, R., Xu, J., Tan, P.-N., Nucci, A., and Mellia, M. (2014). Detecting malicious clients in ISP networks using HTTP connectivity graph and flow information. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 150–157. IEEE.

Lobato, A. G. P., Andreoni Lopez, M., and Duarte, O. C. M. B. (2016). Um sistema acurado de detecção de ameaças em tempo real por processamento de fluxos. In *XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos-SBRC'2016*, Salvador, Bahia.

Lobato, A. G. P., Andreoni Lopez, M., Rebello, G. A. F., and Duarte, O. C. M. B. (2017). Um Sistema Adaptativo de Detecção e Reação a Ameaças. In *Anais do XVII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais - SBSeg'17*, pages 400–413.

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

Sanz, J. I., Andreoni Lopez, M. E., Mattos, D. M. F., and Duarte, O. C. M. B. (2017). A Cooperation-Aware Virtual Network Function for Proactive Detection of Distributed Port Scanning. In *IEEE/IFIP 1st Cyber Security in Networking Conference (CSNet'17)*.