Avaliação de Atributos Estatísticos na Detecção de Ataques DDoS Baseada em Aprendizado de Máquina

Eduardo da Costa da Silva, Sidney Cunha de Lucena
Programa de Pós-Graduação de Informática – Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Av. Pasteur – Urca – CEP 22290-240 – Rio de Janeiro - RJ – Brasil
eduardo.dasilva@uniriotec.br, sidney@uniriotec.br

RESUMO

Identificar ataques a serviços de TI é tarefa difícil, dada a grande quantidade de fluxos de dados numa rede. Ademais, há diversos casos onde o tráfego de ataque tem padrão similar ao tráfego legítimo. Logo, modelar eficientemente as características do tráfego da rede se faz necessário para destacar anomalias que possam indicar um ataque. Este trabalho analisa o impacto que determinados atributos do tráfego em uma rede exercem no desempenho de alguns sistemas de detecção de ataques baseados em aprendizado de máquina. Em especial, propõe-se o uso de atributos obtidos a partir de medidas estatísticas envolvendo o agregado de fluxos de dados coexistentes num intervalo de tempo. Para esta análise, foi implementado um sistema de detecção baseado na arquitetura *lambda* e os experimentos usaram *datasets* realísticos. Os resultados demonstram que a inclusão dos atributos propostos pode melhorar o desempenho de certos algoritmos de aprendizado de máquina na detecção de ataques.

PALAVRAS CHAVE: Detecção de anomalia,aprendizado de máquina,entropia.

TEL&SI - PO em Telecomunicações e Sistemas de Informações.

ABSTRACT

Identifying attacks towards IT services can be very difficult due to the large amount of data flow in a network. In addition, there are several cases where malicious and legitimate traffics have similar patterns. Therefore, it is necessary to efficiently model the network traffic in order to highlight any anomaly that may indicate an attack. This work analyzes the impact of certain network traffic attributes on the performance of some attack detection systems based on machine learning. In particular, it proposes the use of attributes obtained from statistical measures involving the aggregated data flows that coexist in a given time interval. The analysis was performed through the implementation of a detection system based on the *lambda* architecture and experiments used realistic datasets. The results show that the inclusion of the proposed attributes may increase the performance of certain machine learning algorithms for attack detection.

KEYWORDS: Anomaly detection,Machine learning, Entropy.

TEL&SI - OR in Telecommunications and Information Systems.

https://proceedings.science/p/85281?lang=pt-br

1. Introdução

Durante os últimos anos, o tráfego de dados na Internet tem apresentado um acentuado crescimento. Em 2010, por exemplo, foram gerados 1 bilhão de *gigabytes* de dados a cada 2 dias e espera-se que até 2020 sejam criados e manipulados mais de 40 ZB [Yi et al., 2014]. Paralelamente, vem aumentando também a quantidade de tráfego ilícito gerado por atividades maliciosas. É o caso, dentro outros, dos ataques de negação de serviço distribuídos (DDoS - *Distributed Denial of Service*), que visam esgotar os recursos computacionais da vítima ou da rede que a conecta a partir de fluxos de pacotes de dados com requisições forjadas e origens distribuídas pela Internet. Logo, garantir que uma rede de computadores esteja operando satisfatoriamente, livre de tráfegos maliciosos oriundos de ataques, tornou-se uma atividade primordial. No entanto, há dificuldades no que tange à análise do tráfego de rede para a construção de padrões comportamentais que permitam uma correta identificação dos ataques, que podem ser caracterizados como anomalias no tráfego. Tais dificuldades derivam não apenas da grande quantidade de fluxos de dados e suas peculiares, mas também do avanço das técnicas empregadas nos ataques, que objetivam camuflar o tráfego ilícito para dificultar sua identificação [Ficco and Rak, 2015]. Deste modo, esta modelagem deve considerar atributos e métricas, relacionadas aos fluxos de dados que passam por uma rede, que sejam relevantes ao desempenho dos mecanismos empregados na análise do tráfego.

Diante desse contexto, este artigo investiga certos mecanismos de detecção de ataques baseados em em árvore de decisão, no modo *on-line*, para analisar o desempenho destes sistemas com relação a certos atributos associados aos fluxos de dados que passam pela rede que se deseja proteger. Em especial, objetiva-se verificar se é possível melhorar este desempenho a partir do acréscimo de atributos relacionados a medidas estatísticas extraídas para o tráfego agregado da rede num dado intervalo de tempo. Para tal, foi proposto e implementado um sistema baseado na arquitetura *lambda* [Marz and Warren, 2015], que considera o atual cenário tráfego de dados e alinha-se com o conceito de *Big Data* para gerar respostas em tempo hábil. A avaliação do sistema utilizou um *dataset* realístico formado por uma mescla de tráfego real legítimo, ataques DDoS gerados a partir de um ambiente controlado e dados relativos a três diferentes cenários do *dataset* CTU-13 [Garcia et al., 2014]. Os resultados obtidos demonstram que a inclusão dos atributos propostos pode melhorar o desempenho de certos algoritmos de aprendizado de máquina na detecção de ataques DDoS.

A estrutura deste artigo é a seguinte: a Seção 2 apresenta alguns conceitos básicos; a Seção 3 apresenta trabalhos relacionados; na Seção 4 tem-se arquitetura do sistema desenvolvido; a avaliação experimental é descrita na Seção 5; a Seção 6 traz os resultados obtidos; e a Seção 7 conclui artigo.

2. Conceitos Básicos

Esta seção descreve como se dá uma classificação de dados utilizando aprendizado de máquina baseado em árvore de decisão e as características da entropia de *Shannon*, medida estatística usada em um dos atributos avaliados.

2.1. Árvores de Decisão

Um modelo de classificação que utiliza a estrutura de dados *árvore de decisão* é determinado como um processo de classificação que faz uma divisão recursiva de um con-

https://proceedings.science/p/85281?lang=pt-br

junto de dados em conjuntos menores, divisão esta baseada em regras simples a partir dos valores dos dados. Essas divisões ocorrem nos nós de uma estrutura em árvore, com as consequentes ramificações podendo ou não levar a outros nós até se atingir o subconjunto mínimo, representado como uma *folha*. As regras de divisão contidas nos nós são automaticamente geradas a partir de uma etapa de treinamento, que necessariamente precisa contar com um *dataset* marcado, ou seja, cujo conteúdo é completamente descrito pela inserção de *labels* em suas instâncias que informam qual categoria a instância de fato representa. Tal estratégia de aprendizado de máquina, com a necessidade de marcações, é chamada de supervisionada.

Dentre os tipos de mecanismos de aprendizado supervisionado de máquina existentes, como por exemplo, Redes Neurais e Máquina de Vetores de Suporte, optou-se pelo uso de árvores de decisão por sua simplicidade e eficiência em capturar padrões complexos [Michie et al., 1994], principalmente quando se considera a necessidade destes mecanismos trabalharem em tempo real.

Os mecanismos de árvore de decisão usados neste trabalho foram o VHT (*Vertical Hoeffding Tree*), o *Bagging* e o *Adaptive Bagging* [Quinlan, 1986], todos capazes de efetuar aprendizado de máquina com base num fluxo de dados (modo *on-line*) ou incremental. O VHT é um classificador distribuído e escalável que utiliza paralelismo vertical baseado em uma árvore de Hoeffding ou *Very Fast Decision Tree* (VFDT). O método *Bagging* busca maximizar a acurácia pelo agrupamento de diferentes árvores de decisão, superando erros de classificação ao se usar apenas um classificador [Khan et al., 2007]. O modelo final de classificação é resultado de uma combinação dos resultados preditivos individuais desses outros métodos. Já o *Adaptive Bagging* tem funcionamento análogo ao *Bagging*, exceto pelo fato de monitorar a acurácia da classificação ao longo da fase de treinamento e de refinar o seu modelo com os últimos dados capturados. É indicado para situações não-estacionárias de aprendizado, onde os padrões dos fluxos de dados podem variar com o decorrer do tempo, como pode ocorrer com os fluxos de dados numa rede.

2.2. Entropia de Shannon

A entropia de Shannon, enquanto métrica estatística, é usada neste trabalho para descrever o grau de dispersão ou concentração das distribuições de probabilidades de certas características de rede. Essa métrica é bastante adotada na literatura para a identificação de tráfego malicioso, propiciando uma visão de estado global das características de rede dentro de um intervalo de tempo. É definida pela seguinte equação [Shannon, 1948]:

$$E = -\sum_{i=1}^{N} p_i log_2(p_i), \tag{1}$$

onde N é a quantidade de diferentes ocorrências i no espaço amostral e p_i é a probabilidade associada a cada ocorrência i.

3. Trabalhos Relacionados

O trabalho de [Bhuyan et al., 2014] traz um estudo comparativo entre várias técnicas para detecção de anomalias, ressaltando a efetividade do uso de aprendizado de máquina para este fim. Seguindo a mesma tendência de utilização de aprendizado de máquina, [Singh et al., 2014] faz a detecção de anomalias oriundas de uma *botnets*. O Apache

https://proceedings.science/p/85281?lang=pt-br

Hadoop, o Apache Hive e o Apache Mahout compõem os sistemas de uma arquitetura projetada para ser escalável e lidar com grande quantidade de dados. Verificou-se que, embora seja possível analisar fluxos de dados com o Haddop, a latência provocada pelo processamento em lote comprometia a velocidade de execução do sistema.

Já utilizando o processamento em tempo real para analisar anomalias, geradas por ataques, no momento de suas ocorrências, os autores em [Du et al., 2014] usam o Apache Storm [Toshniwal et al., 2014]. Os resultados indicam que a proposta apresenta bom desempenho e escalabilidade.

Em [Robinson and Thomas, 2015], foram comparados dez algoritmos supervisionados de aprendizado de máquina, no modo *batch*, para a classificação de ataques. A avaliação usou três *datasets* antigos, já conhecidos: LLDDoS, CAIDA DDoS 2007 e CAIDA *Conficker*. O algoritmo Adaboost, com classificador base *Random Forest*, obteve o melhor desempenho, acalçando acurácia máxima de 99,96% no *dataset* CAIDA. Mais uma vez, em [Lobato et al., 2016] ataques DoS e de varredura são detectados através de classificação supervisionada usando os algoritmos C4.5, rede neural e máquina de suporte de vetores. Anomalias são sinalizadas assumindo-se uma distribuição Gaussiana nos valores dos atributos mais relevantes e um limiar empírico. Apesar do sistema proposto alcançar mais de 95% de acurácia, a classificação dos fluxos em normais, ataques DoS e de varredura não leva em consideração o estado global do tráfego da rede.

Conforme observado nos trabalhos supracitados, ao longo dos últimos anos diversas técnicas de aprendizado de máquina vêm sendo pesquisadas para fins de detecção de ataques. Porém, diferentemente desses estudos, o presente artigo visa destacar como determinados atributos estatísticos obtidos dos fluxos de dados podem influenciar certos métodos de aprendizado de máquina *on-line* utilizados para a detecção de ataques DDoS em volumosos fluxos de dados. Além disso, foram também levados em consideração aspectos operacionais relacionados com o sistema de detecção, tais como tempo disponível para análise dos fluxos, volume de tráfego e variedade de comportamento ao longo do tempo, tudo isto representado por *datasets* atuais e realísticos.

4. Arquitetura do Sistema de Detecção de Ataques

A arquitetura do sistema proposto é baseado na arquitetura *Lambda*, cujo arranjo objetiva processar, em tempo hábil, volumosos fluxos de dados. Sua estrutura é composta de 3 camadas: processamento em velocidade, processamento em lote e serviços. A camada de processamento em lote é responsável por armazenar grande quantidade de dados e processá-los em lote. A camada de processamento em velocidade é designada para a manipulação de fluxos de dados em alta velocidade. Já a camada de serviço tem a função de juntar os resultados dos processamentos das outras duas camadas e disponibilizar os dados analisados. Em adição, foi desenvolvido um módulo denominado "holístico", cuja principal função é adicionar características temporais do estado global da rede a cada intervalo de tempo. Além disso, esse módulo também computa as métricas de desempenho da classificação.

A camada de processamento em alta velocidade é composta pelas ferramentas Apache SAMOA [De Francisci Morales, 2013] e Apache Storm [Toshniwal et al., 2014], que por sua vez usam a camada de processamento em lote, representada pelo Apache Hadoop [Manikandan and Ravi, 2014], para obtenção de fluxos de treinamento e teste. Funções

https://proceedings.science/p/85281?lang=pt-br

internas do módulo holístico fazem o papel da camada de serviço, sumarizando os dados das duas outras camadas para posterior exibição das métricas de desempenho. A Figura 1, abaixo, exibe a arquitetura do sistema.

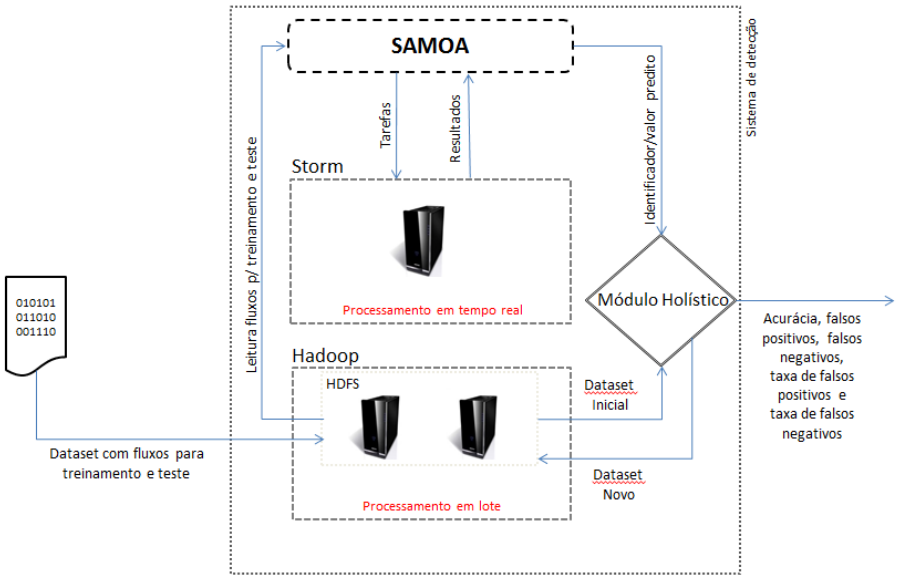

Figura 1. Arquitetura do sistema

4.1. Componentes e Funcionamento do Sistema

O Apache SAMOA é uma plataforma destinada a encontrar padrões em grandes volumes de fluxos de dados, fornecendo algoritmos para as mais comuns tarefas de mineração de dados e aprendizado de máquina. Já o Apache Storm é um sistema distribuído para processamento em tempo real de volumosos fluxos de dados, enquanto que o Apache Hadoop é destinado a armazenamento e processamento em lote, sendo que somente é utilizado seu módulo HDFS, que serve como repositório para os dados manipulados pelo sistema.

O módulo holístico tem a responsabilidade de ler os fluxos do *dataset* armazenado no HDFS e dele extrair novos atributos com características temporais, criando assim um novo *dataset* enriquecido com tais informações. Estas características temporais estão diretamente associadas ao estado global dos fluxos da rede a cada 30 segundos, que é um intervalo de tempo adotado em trabalhos similares [Joldzic et al., 2016]. Estes estados globais são representados pelas entropias associadas aos endereços IP de origem, números das portas de protocolo de origem, endereços IP de destino e numeração das portas de destino. Também são registradas as suas respectivas diferenças para os valores de entropias calculados nos quatro intervalos de tempo anteriores, registrando-se assim a variação dos valores das entropias ao longo dos últimos dois minutos [Joldzic et al., 2016]. Além das entropias, calculam-se também o tamanho médio dos fluxos em *bytes*, o número médio de pacotes dos fluxos e o desvio-padrão dos tamanhos dos pacotes e das durações dos fluxos existentes a cada 30 segundos. Portanto, a cada fluxo são acrescentados esses 20 atributos contendo características estatísticas temporais relativas ao agregado de tráfego fluindo no respectivo intervalo de 30 segundos ao qual o fluxo pertence.

O funcionamento do sistema inicia-se pelo armazenamento do *dataset* no Hadoop,

que passa a ser chamado de *dataset* inicial. O módulo holístico então lê os fluxos do *dataset* inicial, agrupando-os em intervalos de tempo de 30 segundos para cálculo dos atributos estatísticos (denominados de atributos avançados). Posteriormente, o módulo holístico acrescenta os novos atributos no *dataset* e o disponibiliza ao SAMOA. Com base nesse novo *dataset*, o SAMOA cria uma tarefa a ser enviada ao STORM para fins de processamento dos fluxos. A medida que os fluxos vão sendo analisados, os resultados das classificações - predição em "normal" ou "ataque" e identificador do fluxo - são enviados para o módulo holístico. O módulo holístico, alicerçado em marcações anteriores ao processo de classificação, que determinam se o fluxo é de fato ataque ou não, calcula as métricas de desempenho do sistema de detecção as disponibiliza em arquivo.

O sistema foi implementado em um conjunto de três maquinas virtuais, uma com 20GB de RAM e as outras duas com 6GB de RAM cada. Todas as máquinas utilizaram sistema operacional Debian 7 e quatro núcleos de processador Xeon(R) CPU E5-2643 v2 @ 3.50GHz.

5. Avaliação Experimental

Nesta seção serão descritos os procedimentos para a construção dos *datasets*, a organização dos experimentos e as métricas analisadas.

5.1. Criação dos Datasets

Os *datasets* criados para avaliar o sistema proposto derivam de um *dataset* realístico, obtido a partir da combinação de tráfego real e tráfego de ataque emulado, além de um conjunto de *datasets* publicamente disponível conhecido como CTU-13.

O CTU-13 [Garcia et al., 2014] compreende um conjunto de 13 diferentes cenários contendo *botnets* conhecidas de variados tipos. Num primeiro momento, os *datasets* gerados para o conjunto CTU-13 continham capturas de fluxos unidirecionais [Garcia et al., 2014]. Posteriormente, os autores geraram outro conjunto de capturas para os mesmos cenários, mas desta vez para fluxos bidirecionais [Garcia and Uhlir, 2017]. No presente trabalho, tanto no conjunto de fluxos unidirecionais quanto no conjunto de fluxos bidirecionais foram usados os cenários 4, 10 e 11, por conterem ataques DDoS.

O *dataset* realístico foi gerado combinando dados de tráfego legítimo (sem ataques), obtidos de uma rede real, e dados de tráfego de ataque DDoS obtidos por emulação. O tráfego legítimo foi obtido da rede de computadores do Observatório Nacional (ON) durante 7 dias. A comprovação da ausência de ataques baseia-se nas informações fornecidas pelo sistema de segurança de rede do ON. O tráfego de ataque DDoS foi gerado através da ferramenta *hping3* em uma rede isolada com duas máquinas. A origem distribuída dos ataques foi emulada forjando-se e amplamente variando os IPs de origem dos pacotes gerados pelo *hping3*. Os ataques emulados representaram dois tipos de ataques DDoS de alta incidência: tráfego UDP para a porta de destino 53 e tráfego ICMP.

Vale salientar que a literatura não define proporções exatas entre a quantidade de fluxos de tráfego normal e a quantidade de fluxos de ataque verificados no momento de um ataque. Para o *dataset* realístico, foi escolhida a proporção de 80% de fluxos normais e 20% de fluxos de ataques para a mesclagem desses arquivos. Essa proporção visa dificultar a obtenção, via mecanismos de aprendizados de máquina, dos padrões ocultos de comportamento dos fluxos em virtude do desbalanceamento entre a quantidade de fluxos de treinamento e a quantidade de fluxos de teste.

https://proceedings.science/p/85281?lang=pt-br

A partir dos *datasets* citados, foram criados três novos grupos de *datasets* contendo fluxos de treinamento e fluxos de teste, onde cada grupo possui um conjunto separado para diferentes cenários do CTU-13: Grupo 1 - tanto os fluxos de treinamento quanto os de teste são oriundos dos *datasets* CTU-13; Grupo 2 - fluxos de treinamento provenientes do *dataset* realístico e fluxos de teste provenientes do *dataset* CTU-13 com fluxos unidirecionais; e Grupo 3 - fluxos de treinamento provenientes do *dataset* realístico e fluxos de teste provenientes do *dataset* CTU-13 com fluxos bidirecionais.

Esta divisão em grupos busca representar situações diferentes. O Grupo 1 reflete uma prática costumeira em estudos de classificadores usando aprendizado de máquina, onde tanto os dados de treinamento quanto os de teste se originam de diferentes partes de um mesmo *dataset*. Já os Grupos 2 e 3 buscam representar uma situação de cunho prático e operacional na administração de uma rede, situação esta onde a disponibilidade de *datasets* marcados para fins de treinamento só é possível a partir de fontes externas à instituição. Trata-se de uma situação análoga às assinaturas de tráfego contendo ataques recém-descobertos que precisam ser inseridas nos sistemas de detecção de intrusão.

5.2. Organização dos Experimentos

Cada um dos três grupos de *datasets* gerou quatro diferentes conjuntos, cada qual contendo um diferente conjunto de atributos associados aos fluxos IP: (i) um conjunto chamado de *dataset* puro ou primário, contendo apenas os atributos de fluxo gerados pelo *nProbe*, aqui chamados de atributos primários; (ii) um conjunto chamado de *dataset* com estatísticas, contendo os atributos primários e incluindo o tamanhos médio dos fluxos em bytes, tamanho médio dos fluxos em pacotes, desvio-padrão das durações dos fluxos e desvio-padrão dos tamanhos dos pacotes dos fluxos, todos calculados para o tráfego agregado da rede; (iii) um conjunto chamado de *dataset* com entropias, contendo os atributos primários e incluindo as medidas de entropia do tráfego agregado e suas diferenças em relação aos quatro intervalos anteriores de 30 segundos; por fim, (iv) um conjunto chamado de *dataset* completo, contendo todos os atributos gerados. Os atributos primários, gerados pelo *nProbe*, são os seguintes: marca de tempo de início, duração do fluxo, protocolo (campo do cabeçalho IP), IP de origem, porta de origem, IP de destino, porta de destino, número de bytes e número de pacotes do fluxo.

Uma vez obtidos esses quatro conjuntos de atributos para cada um dos três grupos de *datasets*, foram aplicados, em cada conjunto, os métodos de classificação supervisionada VHT, *Bagging* e *Adaptive Bagging* - todos oferecidos pelo SAMOA. Considerando o processo de classificação de fluxos de dados cujo o fator tempo é essencial, foi utilizada a abordagem *interleaved-test-then-train* como técnica de validação da classificação - uma das mais utilizadas em cenários de classificação de fluxos de dados por não onerar demasiadamente o tempo de execução, ao contrário do que acontece, por exemplo, com a validação cruzada.

5.3. Métricas de Desempenho Utilizadas

As métricas utilizadas neste trabalho são aquelas normalmente encontradas na literatura para a aferição da eficiência de um sistema de detecção de ataques [Bhuyan et al., 2014]. Todas elas se baseiam nas seguintes variáveis: Falsos positivos (FP) - quantidade de fluxos normais classificados como fluxos de ataque; Falsos negativos (FN) - quantidade de fluxos de ataque classificados como fluxos normais; Verdadeiros positivos (VP) - quantidade de fluxos de ataque classificados como fluxos de ataques; e Verdadeiros negativos

https://proceedings.science/p/85281?lang=pt-br

(VN) - quantidade de fluxos de normais classificados como fluxos normais. A partir dessas medidas, tem-se as seguintes métricas:

- Acurácia - razão percentual entre a quantidade de fluxos corretamente classificados e a quantidade total de fluxos analisados:

$$\text{Acurácia} = \frac{\text{qtd de fluxos analisados - (FP+FN)}}{\text{qtd total de fluxos analisados}} * 100 \quad (2)$$

- Precisão - razão percentual entre a quantidade de fluxos de ataque corretamente classificados (VP) e a quantidade de fluxos classificados como ataque (VP + FP):

$$\text{Precisão} = \frac{\text{VP}}{\text{VP + FP}} * 100 \quad (3)$$

- Taxa de verdadeiros positivos, ou *Recall* - razão percentual entre o número de verdadeiros positivos (VP) e a quantidade de fluxos de ataque:

$$\textit{Recall} = \frac{\text{VP}}{\text{qtd total de fluxos de ataque}} * 100 \quad (4)$$

- F-*Measure* - média harmônica entre precisão e *recall*, que, neste trabalho, assume valores entre 0 e 10000, onde o primeiro valor representa o pior resultado e o segundo, o melhor:

$$\text{F-}\textit{Measure} = \frac{2}{\frac{1}{Preciso} + \frac{1}{\textit{Recall}}} * 100 \quad (5)$$

6. Resultados

Nesta seção serão apresentados os impactos nas métricas de desempenho elencadas, considerando a variação dos arranjos de atributos nos Grupos 1, 2 e 3. Conforme já mencionado, o VHT, o *Bagging* e o *Adaptive Bagging* foram os mecanismos de aprendizado de máquina utilizados no processo de classificação ou identificação dos fluxos como "normal" ou "ataque". A variação desses conjuntos de atributos tem a intenção representar as diferentes contribuições das medidas estatísticas provenientes do módulo holístico na detecção dos ataques.

Nos Grupo 1 e Grupo 2, nas quatro variações dos conjuntos de atributos dos cenários 4, 10 e 11 e utilizando os três mecanismos de aprendizado de máquina, foram obtidos 100% de acurácia, 100% de precisão, 100% de taxa de verdadeiros positivos, ou *recall*, 100% de taxa de verdadeiros negativos e valor máximo para a F-*Measure*, não havendo classificação errada de nenhum fluxo normal como fluxo de ataque e vice-versa.

Já para o Grupo 3, que contempla fluxos bidirecionais e que usa *datasets* de fontes diferentes para treinamento e teste, ao se usar o mecanismo VHT os resultados mostram que, pela primeira vez, o acerto máximo não foi alcançado. Verificou-se 99,84% de acurácia para todas as variações no cenário 4, 94,32% de acurácia para todas as variações no cenário 10, e 94,67% de acurácia para todas as variações no cenário 11. Tanto a acurácia, a precisão, o F-*Measure* e a taxa de falsos positivos foram influenciados pela quantidade de fluxos normais classificados erroneamente como fluxos de ataques. A maior influência observada foi na precisão, pois essa medida considera todos os casos nos quais os fluxos são classificados como positivos para ataque, sendo ou não verdadeiramente positivos, tendo

https://proceedings.science/p/85281?lang=pt-br

como resultado 59,16% para o cenário 4 e de 58,83% para os cenários 10 e 11. A precisão, por sua vez, interferiu na F-*Measure*, com 7434,09 para todas as variações no cenário 4, 7408,06 para todas as variações no cenário 10 e 7408,01 para todas as variações no cenário 11. No entanto, não houve casos de fluxos de ataques classificados erroneamente como fluxos normais, o equivalente a falsos negativos. Por conseguinte, todos os fluxos de ataques foram corretamente identificados, resultando numa taxa de verdadeiros positivos, ou *recall*, igual a 100%. Ainda como consequência da quantidade de falsos positivos, foi obtida uma taxa de verdadeiros negativos igual a 99,84% para o cenário 4, 93,82% para o cenário 10 e 94,23% para o cenário 11. Ainda com relação aos resultados do VHT para o Grupo 3, verificou-se que a variação da natureza dos atributos não interferiu nos resultados. Qualquer que fosse a combinação de atributos dentro do mesmo cenário, entre os primários e os gerados pelo módulo holístico, não houve variação nos valores das métricas empregadas.

Tal qual ocorrido para o VHT, no método *Bagging* os resultados não se modificaram com a inclusão dos atributos gerados pelo módulo holístico, a exceção do cenário 11. Neste cenário, é possível verificar uma melhora em todas as métricas de desempenho quando é usado o conjunto completo de atributos. Foi verificado, também, que a acurácia passou de 94,67% para 100%, a precisão passou de 58,83% para 99,96%, a F-*Measure* foi de 7408,01 para 9998,16, a taxa de verdadeiros negativos de 94,23% para 100%, a quantidade de falsos positivos diminuiu de 5.713 para 3, a quantidade de verdadeiros negativos sofreu um aumento de 5.710 casos e a taxa de falsos positivos apresentou uma queda de 5,77% para 0,003%.

Assim como ocorreu nos Grupos 1 e 2, quando o mecanismo *Adaptive Bagging* foi aplicado ao Grupo 3, independente do cenário analisado e das variações dos conjuntos de atributos, obteve-se eficiência máxima nas métricas de desempenho.

Os valores ótimos das métricas de desempenho obtidos para o Grupo 1 podem ser justificados pela utilização de partes de um mesmo *dataset*, gerado num mesmo ambiente de rede, para treinamento e teste do sistema, levando a crer que os fluxos IPs acabam por apresentar um padrão mais definido de comportamento. Portando, pode-se concluir que, neste caso, as variações dos conjuntos de atributos tornam-se sem efeito para os mecanismos de aprendizado de máquina, que conseguem capturar mais facilmente esses padrões comportamentais.

No caso do Grupo 2, apesar de se considerar *datasets* diferentes para treinamento e teste, o fato do *dataset* de teste ser proveniente do CTU-13 com fluxos IPs unidirecionais pode justificar o acerto total das classificações apenas usando-se os atributos primários. Os pesquisadores responsáveis pelo CTU-13 alegam em [Garcia and Uhlir, 2017] que o *dataset* contendo dados bidirecionais possui informações mais representativas a respeito do tráfego de uma rede real e explicitamente recomendam que este *dataset* seja usado no lugar do *dataset* unidirecional. Pode-se então concluir que o *dataset* unidirecional apresenta um padrão de comportamento mais simples, o que possivelmente facilita a classificação. Apesar disso, os resultados relativos ao *dataset* unidirecional foram aqui mantidos para uma maior entendimento a respeito da eficiência do sistema proposto.

No Grupo 3, onde os experimentos usaram fluxos de teste provenientes do *dataset* CTU-13 bidirecional, os dados armazenados tendem a refletir melhor a diversidade e a dinâmica dos fluxos, resultando numa maior variação do padrão comportamental dos atri-

https://proceedings.science/p/85281?lang=pt-br

butos primários. Consequentemente, a tarefa de classificação passa a ser mais complexa, ainda mais quando se considera que os dados de treinamento são originados de um *dataset* obtido de outro ambiente de rede.

Tal característica, somada ao fato da haver grande diferença (ou desbalanceamento) na quantidade de fluxos entre as duas classes (normal e ataque), causou impacto direto na métrica de precisão, que se manteve entre 58,83% e 99,96%. Vale notar que no cálculo da precisão são considerados apenas os fluxos classificados como ataque. Diferentemente, a métrica de acurácia considera as duas classes (normal e ataque) em seu cálculo, justificando assim a discrepância entre os valores dessas duas medidas de desempenho. A melhoria nos resultados que os atributos avançados (entropias e demais medidas estatísticas) proporcionaram ao método *Bagging* pode ser atribuída ao fato do método em questão usar mais modelos de classificação do que o método VHT, que usa apenas um. Isso faz com que o mesmo consiga capturar com maior exatidão a contribuição, em termos de ganho de informações, de cada conjunto de atributos assinalados aos seus modelos de classificação. Assim sendo, no passo de decisão da classificação final de um fluxo, onde existe uma votação entre os modelos, fará diferença escolher aqueles atributos com maior contribuição para a formação do modelo de classificação.

Já os resultados ótimos a partir da utilização do método *Adaptive Bagging*, para o Grupo 3, podem ser explicados em função do mecanismo de funcionamento do método. Trata-se de um método mais otimizado para fluxos que mudam suas características ou padrões comportamentais ao longo do tempo. Na fase de construção do modelo, a acurácia é monitorada e, caso exista alguma pequena queda no valor da mesma, o modelo é refeito utilizando como base os últimos fluxos analisados. Dessa forma, tende-se a construção de um modelo bem ajustado aos padrões comportamentais dos fluxos, mesmo quando se usa um conjunto menor de atributos (no caso, os atributos primários). Entretanto, vale ressaltar que, do ponto de vista genérico, a estratégia usada por este método nem sempre pode ser tão eficiente, pois depende da dinâmica de variações entre as classes ao longo do *dataset* de treinamento.

Objetivando fornecer uma leitura mais fluida dos resultados alcançados, os gráficos da Figura 2 sumarizam-nos para as métricas Precisão, *Recall* e F-*Measure* aplicadas ao Grupo 3. Nela, cada coluna representa o mecanismo de aprendizado de máquina utilizado (VHT, *Bagging* ou *Adaptive Bagging*) nos cenários CEN 4, CEN 10 e CEN 11 de modo a exibir o impacto nas métricas de desempenho apresentadas em função da variação dos conjuntos de atributos (Primários, Primários+entropias, Primários+estatísticos ou Primários+entropias+estatísticos).

Quanto às considerações de ordem operacional, o sistema consumiu, em média, 15 segundos com o método VHT, 17 segundos com o *Bagging* e 19 segundos com o *Adaptive Bagging*. Obteve-se um tempo médio de 42 microssegundos para a classificação de mais de 407.242 fluxos com 34 atributos cada um. Tais valores para o tempo de execução sugerem que, do ponto de vista da operação do sistema proposto em regime de produção num ambiente real, é possível a realização de uma etapa de treinamento extremamente rápida e capaz de gerar um sistema de detecção altamente acurado e com baixo tempo de reação. Vale ressaltar que os tempos médios calculados já incluem o tempo gasto com a construção do modelo de classificação.

https://proceedings.science/p/85281?lang=pt-br

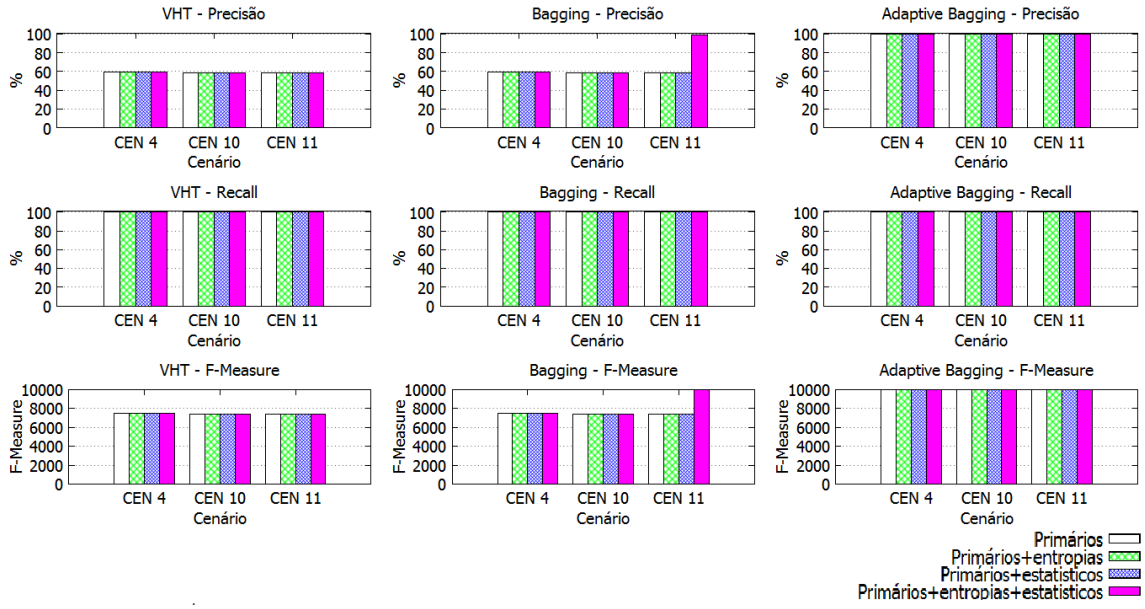

Figura 2. Resumo dos resultados para o Grupo 3

7. Conclusão

Detectar com eficiência essas atividades maliciosas numa rede de computadores é uma tarefa cada vez mais complexa que se faz necessária pela importância das mesmas nas nossas vidas. Nesse contexto, este artigo analisa o impacto que determinados atributos, associados ao conjunto de fluxos de dados que trafegam por uma rede, pode exercer sobre mecanismos de aprendizado de máquina usados para a detecção de ataques. Para tal, foi desenvolvido um sistema de detecção de ataques DDoS baseado na arquitetura *lambda* e contendo mecanismos de aprendizado de máquina. A alimentação desse sistema foi efetuada via *datasets* realísticos gerados de fontes distintas, de forma a dificultar o processo de descoberta de conhecimento dos padrões comportamentais dos fluxos de dados que passam pela rede. Os resultados mostram que a variação do conjunto de atributos associado a esses fluxos pode causar impacto nas métricas de desempenho, especialmente em casos onde há variação considerável na natureza desses fluxos. Verificou-se que é possível aprimorar a classificação dos fluxos de dados realizada com o método *Bagging* quando se incorpora, ao conjunto de atributos, medidas estatísticas que refletem a dinâmica comportamental do tráfego agregado da rede. Além disso, o sistema desenvolvido se mostrou suficientemente rápido e preciso para uso em cenários reais de operação.

Como trabalhos futuros, deseja-se ampliar a avaliação do sistema para a detecção de outros tipos de ataques que explorem a camada de aplicação. Deseja-se, também, identificar novos atributos estatísticos relacionados à dinâmica do tráfego agregado, que porventura melhorem o desempenho dos mecanismos de aprendizado de máquina na identificação dos ataques. Por fim, deseja-se implementar o sistema de detecção proposto num ambiente de produção.

Referências

Bhuyan, M., Bhattacharyya, D., and Kalita, J. (2014). Network anomaly detection: methods, systems and tools.

https://proceedings.science/p/85281?lang=pt-br

De Francisci Morales, G. (2013). Samoa: A platform for mining big data streams. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 777–778. International World Wide Web Conferences Steering Committee.

Du, Y., Liu, J., Liu, F., and Chen, L. (2014). A real-time anomalies detection system based on streaming technology. In *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2014 Sixth International Conference on*, volume 2, pages 275–279. IEEE.

Ficco, M. and Rak, M. (2015). Stealthy denial of service strategy in cloud computing. *Cloud Computing, IEEE Transactions on International Research Journal of Engineering and Technology (IRJET)*, 3(1):80–94.

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

Garcia, S. and Uhlir, V. (2017). Malware capture facility project - ctu-13 dataset. `ttp://mcfp.weebly.com/te`-ctu-13-dataset-a-labeled-dataset-with-botnet-normal- and-background-traffic.html. Accessado em 07/04/2017.

Joldzic, O., Djuric, Z., and Vuletic, P. (2016). A transparent and scalable anomaly-based dos detection method. *Computer Networks*, 104:27–42.

Khan, L., Awad, M., and Thuraisingham, B. (2007). A new intrusion detection system using support vector machines and hierarchical clustering. *The VLDB JournalThe International Journal on Very Large Data Bases*, 16(4):507–521.

Lobato, A., Lopez, M. A., 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.

Manikandan, S. G. and Ravi, S. (2014). Big data analysis using apache hadoop. In *IT Convergence and Security (ICITCS), 2014 International Conference on*, pages 1–4. IEEE.

Marz, N. and Warren, J. (2015). *Big Data: Principles and best practices of scalable realtime data systems*. Manning Publications Co.

Michie, D., Spiegelhalter, D. J., and Taylor, C. C. (1994). *Machine learning, neural and statistical classification*. Citeseer.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

Robinson, R. R. and Thomas, C. (2015). Ranking of machine learning algorithms based on the performance in classifying ddos attacks. In *Intelligent Computational Systems (RAICS), 2015 IEEE Recent Advances in*, pages 185–190. IEEE.

Shannon, C. E. (1948). A mathematical theory of communication. In *Bell System Technical Journal, 27:379-423 and 623-656.*

Singh, K., Guntuku, S. C., Thakur, A., and Hota, C. (2014). Big data analytics framework for peer-to-peer botnet detection using random forests. *Information Sciences*, 278:488–497.

Toshniwal, A., Taneja, S., Shukla, A., Ramasamy, K., Patel, J. M., Kulkarni, S., Jackson, J., Gade, K., Fu, M., Donham, J., et al. (2014). Storm@ twitter. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 147–156. ACM.

Yi, X., Liu, F., Liu, J., and Jin, H. (2014). Building a network highway for big data: architecture and challenges. *Network, IEEE*, 28(4):5–13.

https://proceedings.science/p/85281?lang=pt-br