UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Samuel Lautert Jardim

DETECÇÃO DE ATAQUES DDOS FLASH CROWD BASEADO NA ANÁLISE COMPORTAMENTAL DE USUÁRIOS

Santa Maria, RS
2017

Samuel Lautert Jardim

DETECÇÃO DE ATAQUES DDOS FLASH CROWD BASEADO NA ANÁLISE COMPORTAMENTAL DE USUÁRIOS

Dissertação apresentada ao Curso de Pós-Graduação em Informática, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Ciência da Computação**

Orientador: Prof. Dr. Raul Ceretta Nunes

Santa Maria, RS
2017

Ficha catalográfica elaborada através do Programa de Geração Automática da Biblioteca Central da UFSM, com os dados fornecidos pelo(a) autor(a).

Lautert Jardim, Samuel
DETECÇÃO DE ATAQUES DDOS FLASH CROWDBASEADO NA ANÁLISE COMPORTAMENTAL DE USUÁRIOS / Samuel Lautert Jardim.- 2017.
62 p.; 30 cm

Orientador: Raul Ceretta Nunes
Dissertação (mestrado) - Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-Graduação em Informática, RS, 2017

1. Ataques DDoS 2. Comportamento Humano 3. Flash Crowd 4. HTTP.Solicitações Web I. Ceretta Nunes, Raul II. Título.

Samuel Lautert Jardim

DETECÇÃO DE ATAQUES DDOS FLASH CROWD BASEADO NA ANÁLISE COMPORTAMENTAL DE USUÁRIOS

Dissertação apresentada ao Curso de Pós-Graduação em Informática, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Ciência da Computação.**

Aprovado em 03 de Março de 2017

Raul Ceretta Nunes, Dr. (UFSM)
(Orientador - Presidente)

Carlos Raniery Paula dos Santos, Dr. (UFSM)

Eduardo Luzeiro Feitosa, Dr. (UFAM)

Santa Maria, RS

2017

AGRADECIMENTOS

Gostaria de agradecer inicialmente ao professor Raul Ceretta Nunes, pela oportunidade em estarmos juntamente realizando este trabalho. Pelo apoio nos diversos desafios enfrentados, na excelente orientação e especial atenção as revisões e sugestões, fatores fundamentais para a conclusão deste trabalho.

Aos meus pais Adriane e Rildo pelo apoio incondicional em estar buscando maior conhecimento e preparação para quaisquer desafios a serem enfrentados. A minha irmã Manuelle que sempre esteve ao meu lado, para juntos alcançarmos nossos objetivos.

A toda minha família que sempre apoiou e torceu pelas minhas almejadas conquistas, aos meus avós Alci, Lindolfo, Irmgard e Donival que sempre desejavam ter os netos graduados e com esta etapa consigo realizar além dos seus desejos.

A minha querida namorada Luciana, por me apoiar muito e ser bastante compreensiva nos momentos que tive que me ausentar para a realização deste trabalho.

Aos colegas e agora novos amigos do GTSeg e ao Programa de Pós-Graduação em Informática (PPGI) da Universidade Federal de Santa Maria em proporcionar a realização da pós-graduação.

“Aprender é a única coisa de que a mente nunca se cansa,
nunca tem medo e nunca se arrepende.”
Leonardo da Vinci

RESUMO

DETECÇÃO DE ATAQUES DDOS FLASH CROWD BASEADO NA ANÁLISE COMPORTAMENTAL DE USUÁRIOS

AUTOR: Samuel Lautert Jardim
ORIENTADOR: Raul Ceretta Nunes (UFSM)

A Internet vem sendo alvo de ataques por diversas razões, tais como gratificações financeiras, guerras cibernéticas, entre outras. Os ataques de negação de serviço distribuído (*DDoS*) se destacam por serem uma ameaça para o bom funcionamento da Internet e, quando presentes na camada de aplicação, como *DDoS mimic Flash Crowd*, podem servir como alternativa para os *botmasters* tornarem seus ataques ainda mais indetectáveis. A principal dificuldade encontrada na identificação desse ataque, deve-se principalmente à similaridade com o tráfego de rede benigno do tipo *Flash Crowd* (surto de visitas inesperadas). Ferramentas de detecção de ataques necessitam diferenciar um tráfego *Flash Crowd* de um tráfego com ataque *DDoS*. Dessa forma, o objetivo deste trabalho é apresentar um método capaz de detectar ataque *DDoS mimic Flash Crowd*, bem como diferenciar usuários maliciosos disfarçados de usuários legítimos (humano). Esse trabalho propõe um método de detecção baseado na observação do padrão de interatividade nas solicitações dos usuários, diferenciando um usuário humano de um *bot* (programa malicioso) modelando o comportamento através da taxa de interatividade, número de solicitações e do tempo entre elas. Os experimentos demonstram a eficácia do método na detecção, comprovando que o padrão de interatividade esperado pode ser aplicado como mecanismo de detecção.

Palavras-chave: Ataques DDoS. Comportamento Humano.Flash Crowd. HTTP. Solicitações Web.

ABSTRACT

DETECTION OF DDOS FLASH CROWD ATTACKS BASED ON BEHAVIORAL ANALYSIS OF USERS

AUTHOR: Samuel Lautert Jardim
ADVISOR: Raul Ceretta Nunes (UFSM)

The Internet has been target of attacks for several reasons, such as financial gratuities, cyber wars, among others. Distributed denial of service (DDoS) attacks stand out as a threat to the proper functioning of the Internet, and when present in application layer, such as DDoS mimic Flash Crowd, can serve as an alternative for botmasters to make their attacks even more undetectable. The main difficulty found in the identification of this attack is mainly due to the similarity with benign network traffic of the Flash Crowd type (outbreak of unexpected visits). Attack-detection tools need to differentiate a Flash Crowd traffic from a traffic with DDoS attack. Thus, the purpose of this work is to present a method capable of detecting DDoS mimic Flash Crowd attack, as well as distinguishing malicious users disguised as legitimate (human) users. This work proposes a method of detection based on the observation of the interactivity pattern in user requests, differentiating a human user from a bot (malicious program) by modeling the behavior through the interactivity rate, the number of requests and the time between them. The experiments demonstrate the effectiveness of the detection method, proving that the expected interactivity pattern can be applied as a detection mechanism.

Keywords: DDoS attack. Flash Crowd. HTTP. Human behavior. Web requests.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CAIDA	*Center for Applied Internet Data Analysis*
DAPV	Diferença Acumulativa das Porcentagens de Variação
DAPVH	Diferença Acumulativa das Porcentagens de Variação Do Host
DDOS	*Distributed Denial of Service*
DVM	Diferença os Valores Médios
DVMH	Diferença os Valores Médios do Host
FN	Falso Negativo
FP	Falso Positivo
HTTP	*Hypertext Transfer Protocol*
HTTPS	*Hyper Text Transfer Protocol Secure*
In	Interatividade
NumS	Número de Solicitações
NumSG	Número de Solicitações Global
Pat	Período de Atividade
PV	Porcentagens e Variação
Ta	Taxa de Acurácia
Teei	Tempo entre Estados Interativos
TeeiG	Tempo entre Estados Interativos Global
TxIn	Taxa de Interatividade
TxInG	Taxa de Interatividade Global
URL	*Uniform Resource Locator*
VIn	Vetor de Interatividade
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo
WITS	*Waikato Internet TrafficStorage*

SUMÁRIO

1 INTRODUÇÃO

A praticidade de se trabalhar em redes tornou essa ferramenta imprescindível para a comunicação global, promovendo importantes recursos e mecanismos como pesquisas, compartilhamento de arquivos, acesso remoto e interação entre pessoas. Porém, a Internet vem sendo alvo de ataques de muitos grupos motivados por diversas razões, como: recompensas financeiras, ativismo, guerras entre países, divergências ideológicas, entre outras (YU et al.,2012; SHIAELES e PAPADAKI, 2015). Sendo assim, inúmeras formas de ataques são conduzidas por grupos maliciosos para tornar falho ou inoperante um *Website* ou serviço específico podendo afetar uma cadeia de pessoas que usufruem do serviço de maneira direta ou indireta.

No intuito de tornar-se um serviço falho ou inoperante, são utilizados muitos métodos de ataques aos computadores servidores, onde se destaca o ataque de Negação de Serviço Distribuído *(Distributed Denial of Service - DDoS)* (DANTAS, 2013; DHINGRA e SACHDEVA, 2014; YU et al., 2012). Para a realização de ataques *DDoS,* são utilizadas redes denominadas *botnets,* formadas por computadores distribuídos geograficamente e comprometidos com softwares maliciosos (*bots*) comandados por criminosos (*botmasters*), sem o consentimento e conhecimento dos proprietários dos computadores (hosts infectados). Segundo Feily et al. (2009) o ataque possui dois métodos principais. O primeiro se aproveita de uma vulnerabilidade e envia pacotes mal formados ludibriando um protocolo ou aplicação. Já o segundo tem como objetivos: *inundação de redes*, interrompendo a conectividade do usuário utilizando o esgotamento da largura de banda, da capacidade dos roteadores e/ou recursos de rede; e *esgotamento dos recursos* do servidor, esse conhecido como ataque de inundação de aplicação.

Com o objetivo de dificultar a detecção de *botnets*, os *botmaster s*utilizam-se de técnicas anti-forenses como ofuscação de código, criptografia, comunicação *peer-to-peer* e imitação de eventos *Flash Crowd*, para disfarçar peculiaridades que possam diferenciar traços de um ataque dos dados legítimos (YU et al., 2012).

Um evento *Flash Crowd* corresponde a um pico de acesso legítimo a um dado *Website*, tal como pode ocorrer em sites de notícias durante grandes eventos. Este tipo de evento costuma gerar degradação de desempenho ou indisponibilidade do *Website* com consequente insatisfação de usuários (ARI et al., 2003). Um ataque DDoS *mimic Flash Crowd* é um tipo de ataque que procura imitar um comportamento *Flash Crowd*, ou seja, procura tornar o serviço

indisponível a partir de rajadas de tráfego legítimo (KE et al., 2009; PRASAD et al., 2013; XIE e YU, 2009). Tanto *Flash Crowd* quanto DDoS *mimic Flash Crowd* podem ser detectados analisando as características estatísticas do tráfego (Thapngam et al., 2011). O desafio é adotar uma abordagem que seja capaz de realizar a diferenciação adequada entre o comportamento de um *Flash Crowd* (gerado por humanos) e o de um ataque (gerado por *bots*) (PRASAD et al., 2013; XIE e YU, 2009; YU et al., 2012).

O cálculo da correlação de *Pearson* aplicado às séries temporais do número de solicitações *Hypertext Transfer Protocol* (HTTP) (XU et al., 2012) tem sido o método mais aplicado para diferir o comportamento humano de um *bot* no caso de ataques DDoS *mimic Flash Crowd*. Esse método assume que os traços gerados pelos *bots* possuem entre sí uma correlação maior que os traços gerados por humanos (THAPNGAM et al., 2011). Entretanto, cada página Web possui um número de solicitações secundárias aleatórias, podendo fazer com que esse método apresente muitos resultados falso-positivos. Para minimizar o problema, Thapngam et al. (2011) propuseram um método no qual calcula a correlação de *Pearson* da taxa de chegada de pacotes dos *hosts* ao longo do tempo. Porém, na medida em que um servidor Web estiver trabalhando com alta carga, o comportamento da taxa de pacotes dos *hosts* pode se assemelhar bastante com a taxa do servidor, em função do *throughput* minimizado para cada *host*, o que pode levar a um comprometimento do método.

Observa-se, assim, que existem alguns métodos já estudados para detecção e diferenciação de ataques DDoS e fenômenos de *Flash Crowd*, porém, com os avanços e a sofisticação das técnicas de ataque, há uma grande dificuldade de distingui-los com precisão e propostas para identificar os ataques são cada vez mais importantes.

1.1 JUSTIFICATIVA

Tanto os eventos de *Flash Crowd* quanto os ataques *DDoS mimic Flash Crowd* são duas grandes preocupações para a segurança dos sites da Web e precisam ser tratados de maneira rápida para que não haja indisponibilidade de acesso ao usuário. Para isso, esses fenômenos precisam ser identificados e abordados diferentemente, para que os acessos legítimos possam ser permitidos e acessos mal-intencionados bloqueados. Um dos problemas identificados nesse contexto é a diferenciação de um ataque *DDoS mimic Flash Crowd* a um fenômeno de *Flash Crowd*, visto que ambos têm características muito parecidas (DANTAS et al., 2013). Como esses vêm se equiparando ao longo do tempo, é um desafio distingui-los. Dada a imprevisibilidade e o aumento na frequência de *Flash Crowds* e a facilidade de elaboração de

ataques *DDoS* por meio de *botnets*, é de fundamental importância que exista um mecanismo que consiga detectar e diferenciar ataques *DDoS mimic Flash Crowd* de eventos *Flash Crowd*.

1.2 OBJETIVOS GERAL E ESPECÍFICOS

O objetivo deste trabalho é desenvolver um método capaz de detectar um ataque *DDoS mimic Flash Crowd*, identificando usuários maliciosos disfarçados de usuários legítimos (humano). Neste sentido o trabalho apresenta um método para diferenciação entre humano e *bot*, baseado na análise comportamental da interação nas solicitações HTTP(S) de cada *host* cliente. O comportamento é modelado através do número de solicitações e do tempo entre solicitações, resultando na análise da taxa de interatividade. O método de diferenciação contrasta os comportamentos médios esperados para humanos com os de *bots*.

Os objetivos específicos desta dissertação foram:

- Identificar e analisar o padrão comportamental do usuário humano na interação das solicitações Web utilizando base de dados com traços de redes existentes;
- Apresentar o método de detecção de ataques *DDoS mimic Flash Crowd*, e
- Definir experimentalmente os limiares para um bom funcionamento do método de diferenciação e validar o método com traços de redes públicas e ataques sintéticos.

1.3 ORGANIZAÇÃO DO TEXTO

Esta dissertação está organizada da seguinte forma: o capítulo 1 apresenta uma breve introdução, objetivos geral e específicos e justificativa da importância da diferenciação entre eventos *Flash Crowd* e ataques *DDoS mimic Flash Crowd*; o capítulo 2 aborda os aspectos de eventos *Flash Crowd* e ataques distribuídos por negação de serviço (*DDoS*), bem como os trabalhos relacionados; o capítulo 3 descreve a proposta do método de diferenciação entre evento *Flash Crowd* e ataque *DDoS mimic Flash Crowd*; o capítulo 4 descreve os experimentos e resultados; e por fim no capítulo 5 são apresentadas as conclusões.

2 CONCEITOS BÁSICOS

Tradicionalmente, ataques de negação de serviço exploram vulnerabilidades nos protocolos de rede, visando o esgotamento de seus recursos. Diferentemente, os ataques *DDoS mimic Flash Crowd* na camada de aplicação tem por objetivo esgotar os recursos do servidor (sockets, CPU, memória, banco de dados, largura de banda de I/O) (KE et al., 2009; XIE e YU, 2009). A motivação para esta mudança de foco dos atacantes decorre da melhor eficácia dos mecanismos de segurança em redes e da crescente evolução da tecnologia, que permitiu um crescimento substancial da largura de banda das conexões de Internet (*throughput* das interconexões).

Na camada de aplicação o tráfego anormal pode ser classificado em quatro tipos (SARAVANAN et al., 2013): (1) *insistência no pedido* - o atacante concentra o ataque em uma página inicial ou uma página momentaneamente muito acessada, gerando maior fluxo no tráfego que converge para um ou mais pontos; (2) *pedido recursivo* - o tráfego é direcionado para várias páginas Web (*URLs*), fazendo o endereço fonte dos acessos convergir para um grupo de pontos, porém as metas de tráfego tornam-se dispersas; (3) *repetidas cargas de trabalho* - tráfego que resulta de um grupo de *bots* que envia repetidas solicitações, por exemplo, acesso a uma imagem grande ou operações que exijam grande carga no banco de dados; e (4) *Flash Crowd*-causa um surto de visitas inesperadas, geralmente em um anúncio de novo serviço ou download de software livre, os endereços fontes são dispersos, as solicitações são legítimas e as metas convergem para um ou dois pontos. O tráfego anormal resultante de comportamento *Flash Crowd* é similar ao de um ataque *DDoS* na camada de aplicação, tanto na maneira que os dados são transmitidos quanto na troca de mensagens de comunicação entre cliente e servidor. Ambos são volumosos, apresentam rajadas e são instáveis, causando interrupções no serviço (DHINGRA e SACHDEVA, 2014; KANDULA et al., 2005; KE et al., 2009).

A seguir, na seção 2.1, são apresentadas algumas diferenciações entre eventos *Flash Crowd* e ataques que procuram imitá-lo (*DDoS mimic Flash Crowd*). São também apresentados, na seção 2.2, trabalhos relacionados que procuram diferenciar o comportamento de eventos legítimos do de atacantes.

2.1 FLASH CROWD E DDOS FLASH CROWD

Diferenças entre fluxos *Flash Crowd* e ataques *DDoS mimic Flash Crowd* no nível de aplicação aparecem na análise comportamental dos acessos benignos e malignos, como por exemplo, na distribuição da origem dos endereços IPs e aumento e diminuição da velocidade de tráfego. De maneira geral, o ataque tem a intenção de manter o tráfego intenso de dados com o servidor por um período maior, enquanto que um usuário legítimo realiza a solicitação e encerra ou pausa sua interatividade com o servidor. Observa-se assim que o sucesso na detecção de *DDoS mimic Flash Crowd* pode ser função da interpretação de "intenção" assinada no tráfego e que pode permitir a distinção entre os fenômenos.

A diferença no tipo de tráfego também decorre da característica do *flash crowd*, no qual o aumento na quantidade de usuários e acessos se dá de forma brusca e geralmente em resposta a um evento especial num dado período, tal como na publicação de notícias de grande repercussão, lançamento de filmes, músicas, *softwares*, entre outros. No caso de acesso legítimo, no momento que o usuário recebe a informação desejada do servidor ele tende a prosseguir solicitando outra página ou finaliza seu ciclo no site. Porém no ataque *DDoS Flash Crowd* o objetivo explícito é sobrecarregar o servidor com solicitações para causar uma grande lentidão ou, preferencialmente, tornar o servidor inacessível para usuários benignos (YU et al., 2012). Em Dhingra e Sachdeva(2014) observa-se que o tráfego de dados do usuário benigno em um *Flash Crowd* é variável e tende a formar uma onda em "ziguezague". Essa tendência é em resposta ao comportamento humano de solicitar uma página e após alguns instantes solicitar outra. Outra observação pertinente é que uma eventual lentidão do servidor em função do alto volume de solicitações pode fazer com que o usuário desista e após alguns instantes insista no pedido.

Observa-se, pelo exposto, que eventos *Flash Crowd* e *DDoS* na camada de aplicação, ou *DDoS mimic Flash Crowd*, compartilham características similares, mas que a diferenciação pode ser realizada via análise comportamental do tráfego.

2.2 COMPORTAMENTO INTERATIVO NAS SOLICITAÇÕES WEB

Com base na hipótese de que o comportamento do evento *Flash Crowd* e do ataque *DDoS mimic Flash Crowd* pode ser analisado e classificado em função do padrão de

interatividade em solicitações Web e do número de requisições, esta seção apresenta mais detalhes a respeito dos comportamentos básicos desses eventos.

A interatividade nas solicitações de um humano junto ao servidor Web apresenta um comportamento que varia de acordo com cada ação tomada pelo usuário em seu computador. Inicialmente, o usuário solicita o acesso a um *Website* pelo seu *browser*, o que gera a abertura da conexão entre os *hosts* cliente/servidor. Via protocolo HTTP ou HTTPS (*Hyper Text Transfer Protocol Secure*), neste texto representados por HTTP(S), após a conexão, é realizada a solicitação da *URL* primária e, posteriormente, das secundárias que estão incorporadas no corpo do código. Como observado em alguns estudos (OIKONOMOU e MIRKOVIC, 2009; XU et al., 2012; THAPNGAM et al., 2011; PAN et al., 2014), durante esta fase de solicitações legítimas, o comportamento pode ser definido como de *interação ativa*, ou seja, quando estão sendo enviadas solicitações HTTP(S) legítimas para o servidor. Dependendo do tamanho dos arquivos a serem carregados e da largura de banda do cliente/servidor, os dados começam a ser transmitidos logo após as solicitações, e a transmissão pode perdurar por vários segundos até sua conclusão. Entretanto, esse tempo de transmissão até o término do *download* não deve comprometer a observação do comportamento interativo do usuário, pois a troca de informações de conexão, mesmo que ocorra de maneira intercalada com as solicitações HTTP(S), não é realizada via protocolo HTTP(S). Logo, a troca de informações não interfere na frequência de mensagens das solicitações Web. Numa situação de normalidade, após o recebimento dos dados pelo *browser* é esperado que o usuário realize a apreciação das informações apresentadas e leve um tempo variável até a próxima solicitação. Em outras palavras, numa *interação ativa* é esperado que o usuário não realize solicitações em elevada frequência (HWANG et al., 2005; OIKONOMOU e MIRKOVIC, 2009; THAPNGAM et al., 2011; XIE e YU 2009). Por outro lado, o comportamento de um usuário maligno, realizando um ataque *DDoS mimic Flash Crowd* de acesso ao conteúdo do *Website*, corresponde à solicitações praticamente interruptas.

Ao modelar a interatividade como uma variável aleatória discreta, com valor 0 indicando períodos sem interatividade e 1 indicando período com interatividade, é observado que o comportamento esperado do usuário humano apresenta tempo variável entre novas solicitações, dado a existência de momentos de apreciação das informações (sem interatividade). Por outro lado, o comportamento esperado de um atacante (representado por um *bot*) apresenta maior interatividade. A Figura 1 ilustra os comportamentos típicos e esperados de um humano e de um atacante na notação discreta. Note que a interatividade, mensurada de forma binária (com interatividade e sem interatividade), o número de solicitações

(representados pelos traços horizontais) e o tempo entre os estados com interatividade (t), quando analisados por unidade de tempo descrevem um comportamento observável que também permite diferir um usuário humano de um robô de ataque (*bot*).

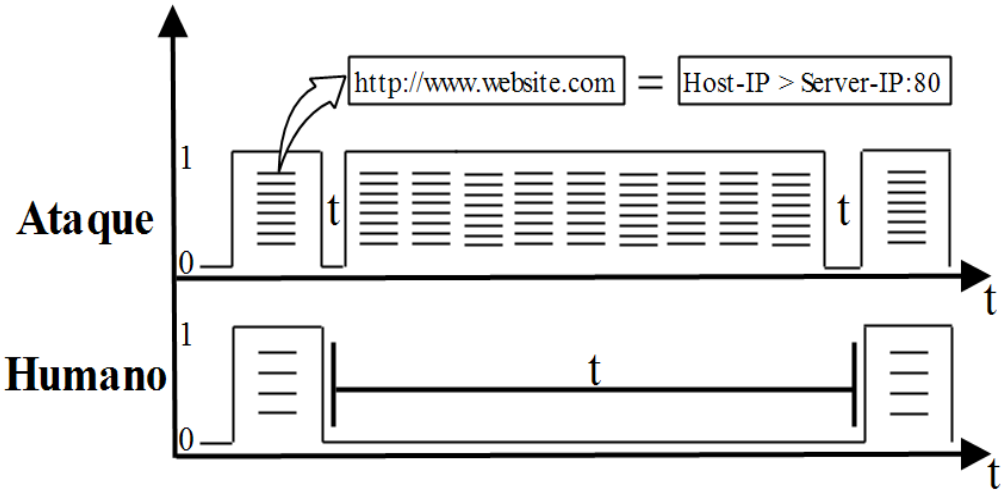

Figura 1. Padrão de interatividade de humano e de atacante. Fonte: Elaborada pelo autor.

2.3 TRABALHOS RELACIONADOS

Alguns métodos foram propostos para possibilitar a detecção de ataques *DDoS mimic Flash Crowd.* Foi apresentado em Xu et al.(2012), um algoritmo para detecção de ataques utilizando o Coeficiente de Correlação de *Pearson* do grau de atividade dos usuários, formado pelo número de solicitações HTTP. Os autores assumem que após um início de ataque os *bots* não pararam de enviar solicitações com o propósito de negar o serviço, gerando alto volume de solicitações HTTP e que os humanos têm um comportamento de altos e baixos, oscilando ao longo da análise. Os autores assumem assim, que a correlação dos *bots* deve ser mais alta que a dos humanos, permitindo a descriminação dos comportamentos. Entretanto, analisar somente a correlação entre os graus de atividades (número de requisições) pode, ao longo do tempo, acarretar em muitos resultados falso-positivos, principalmente pelo fato de que o número de solicitações na requisição de uma página Web pode variar muito. Cada requisição possuirá uma quantidade aleatória de *URLs* incorporada no seu corpo (*body*), fazendo com que o comportamento do grau de atividade não seja influenciado somente pelo usuário, mas sim pela quantidade de solicitações de cada página, comprometendo a análise discriminatória.

O trabalho de Thapngam et al. (2011) assume que os humanos não possuem uma previsibilidade no comportamento, enquanto os *bots*, por realizarem as solicitações de forma automatizada a partir de um algoritmo, possuem características que se repetem em um período curto de tempo. Desta forma, os autores apresentam um método baseado na análise da repetição da taxa de transmissão de pacotes. A análise avalia a Correlação de *Pearson* entre a taxa de chegada de pacotes x tempo e a auto correlação da taxa calculada. A limitação desta técnica é que seu foco analisa apenas a taxa de chegada de pacotes, fazendo com que a quantidade de falso-positivos possa crescer significativamente. No artigo os autores omitem esta métrica.

Segundo Yu et al. (2012) os fluxos de ataques são mais similares entre si do que os fluxos de acessos benignos. Desta forma, os autores apresentam uma técnica que consiste na discriminação utilizando o coeficiente de correlação de fluxos e o cálculo de similaridade entre os fluxos suspeitos. A descoberta que os fluxos em um ataque são bastante semelhantes é um grande avanço, porém, como destacado no trabalho, o atacante tendo conhecimento da técnica pode customizar o ataque para imitar um fluxo benigno, deixando a semelhança entre os fluxos muito equivalentes. Logo, a detecção utilizando a similaridade dos fluxos pode não ser suficiente para construção de um método eficiente.

Neste trabalho a distinção entre comportamento humano e de um *bot* explora uma análise conjunta de outras características, não só da quantidade de solicitações e similaridade entre fluxos, mas também diferencia-se dos demais por explorar a característica de interatividade e tempo entre estados interativos através de um método distinto da similaridade entre fluxos.

3 MÉTODO PROPOSTO

Neste Capítulo é apresentando um novo método para detecção de ataques *DDoS mimic Flash Crowd*, baseado na análise do comportamento interativo dos usuários nas solicitações Web. O método assume como hipótese básica a observação da interatividade comportamental de um usuário benigno (humano) distinguível do comportamento do usuário maligno (*bots* Web automatizados), possibilitando a diferenciação entre eventos *Flash Crowd* e ataques *DDoS mimic Flash Crowd.* Este capítulo também apresenta dois métodos de tomada de decisão: método de Diferença aos Valores Médios e método de Diferença Acumulativa das Porcentagens de Variação.

3.1 MODELO COMPORTAMENTAL

Para distinguir o comportamento do usuário (humano ou *bot*) três variáveis de interesse são modeladas: a Taxa de Interatividade do usuário (*TxIn*); o Número de Solicitações Web (*NumS*); e o Tempo entre Estados Interativos (*Teei*). Para estruturar a modelagem fazem parte a "*chave*" e o "*Pat*", representados de forma ilustrativa na Figura 2. Segue suas definições:

- *chave*: consiste na união do endereço IP do *host* fonte do usuário com o endereço do servidor de destino, ou seja, $chave = (IP\ do\ usuário + IP\ do\ servidor)$. A chave serve como índice para as informações alocadas; e

- *Pat*: correspondente a um período de atividade, ou seja, a uma janela discreta de tempo de tamanho Δt, discretizada em períodos Δt'. Um traço de rede é expresso por vários períodos de atividades observados (*Pat's*).

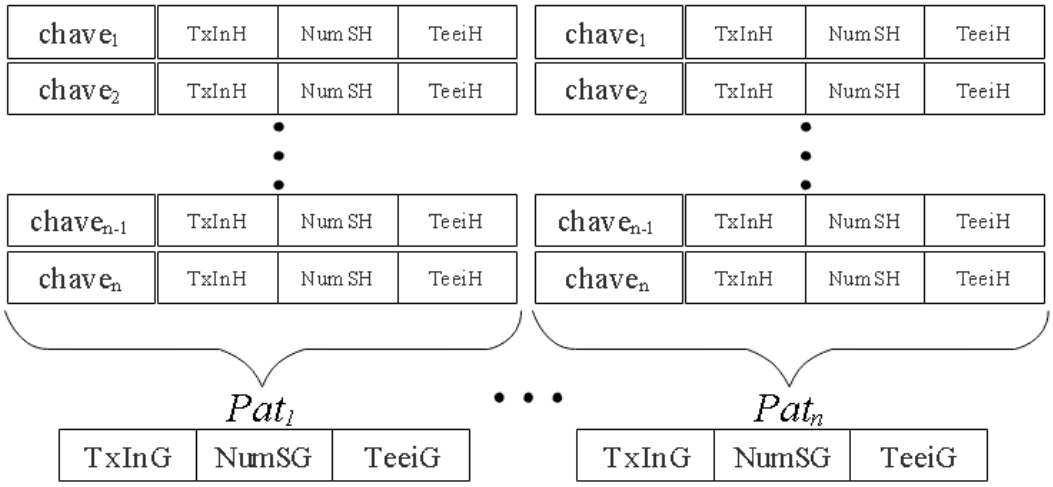

Figura 2. Estrutura da Modelagem Comportamental. Fonte: Elaborada pelo autor.

Nas subseções a seguir o comportamento de cada uma das variáveis que compõe o modelo comportamental é discutido e modelado com vistas à diferenciação entre um ataque e um tráfego legítimo.

3.1.1 Taxa de Interatividade

A interatividade de um dado usuário (*In*) é observada no momento da constatação da existência de estado interativo, ou seja, ocorrência de solicitação Web, expressa por 0 (sem interatividade) ou 1 (com interatividade). A taxa de interatividade (*TxIn*) é medida somando o número de estados interativos do usuário para um determinado servidor dentro de um período de tempo Δt. Cada usuário pode possuir várias taxas em função dos diferentes servidores no qual o mesmo realizou solicitação, ou seja, o IP de um usuário pode estar em várias *chaves*.

Dado o período de atividade observado (*Pat*), a taxa de interatividade corresponde à frequência de ocorrência de solicitações nos períodos de tempo Δt' dentro do período Δte pode ser expressa por um vetor de interatividade (*VIn*) contendo valores 0s e 1s, onde 0 indica que não existiu solicitação/interação e 1 indica que existiu, como ilustrado na Figura 3.

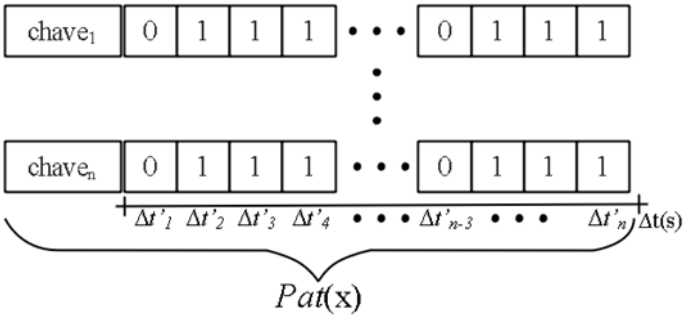

Figura 3: Vetor da Taxa de Interatividade (*VIn*). Fonte: Elaborada pelo autor.

O tamanho do vetor depende da granularidade desejada para o *Pat*, ou seja, da janela temporal de observação Δ*t* e da janela temporal considerada no computo da amostra Δ*t'*. Desta forma, a taxa de interatividade de um usuário (*TxIn*) deriva da quantidade de períodos distintos com interatividade em um *Pat*. Deste modo, o método de detecção proposto realiza duas medidas:

(*i*) a taxa de interatividade de um usuário a um determinado servidor, dentro de um *Pat*, dado por $TxInH$; e

(*ii*) a média das $TxInH$do *Pat* analisado, dado por $\overline{TxInG}$, sendo, $\overline{TxInG} = \left\{\left(\frac{\sum_i^n TxIntH_i}{n}\right) \in Pat\right\}$.

A interatividade do *bot* (*In_bot*) tenderá a ocupar todo o período *Pat* e ser maior que a média observada dos comportamentos humanos$\left(\overline{TxIn_humano}\right)$, ou seja,

$$TxIn\_bot > \overline{TxIn\_humano}$$

De maneira similar, a interatividade do humano tenderá a um comportamento de baixa interatividade no período observado, tendendo ao padrão médio dos traços observados em período sem ataques$\left(\overline{TxIn_humano} \cong \overline{TxIn_traço}\right)$. Salienta-se que em períodos de *Flash Crowd* o comportamento humano não é alterado.

3.1.2 Número de Solicitações HTTP(S)

O número de solicitações HTTP(S) (*NumS*) é a soma de todas as solicitações com porta de destino 80 e 443 de uma *chave* dentro de um período Δ*t*, como ilustrado na Figura 4. Conforme observado em Xu et al. (2012) e Singh et al. (2015)**,** em ataques *DDoS* na camada de aplicação o número de solicitações Web de um *host* atacante (*NumSH_bot*) tende a ser maior que o de um *host* operado por humano (*NumSH_humano*) em um mesmo período Δ*t*, conforme

já ilustrado na Figura 1. Desta forma, para um dado período de tempo Δt assume-se que *NumSH_bot>NumSH_humano*.

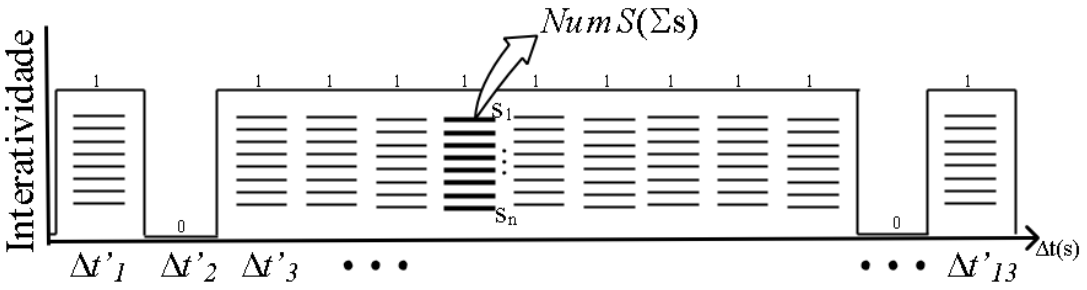

Figura 4. Ilustração do *NumS*. Fonte: Elaborada pelo autor.

A observação do número de solicitações dos *hosts* permite a análise da distinção entre comportamento humano e de um *bot*. Essa análise pode ser realizada em função da intensidade de *NumS*. Deste modo, o método de detecção proposto realiza duas medidas:

(*i*) a média do número de solicitações de uma dada *chave*, dado por $\overline{NumSH}$, que corresponde a divisão do número total de solicitações (*NumS*) pela taxa de interatividade da respectiva *chave*$(TxInH)$dentro do *Pat*, ou seja, $\overline{NumSH} = \left(\frac{\sum_i^n NumS_i}{\text{TxInH}}\right)$; e

(*ii*) a média dos$\overline{NumSH}$do *Pat* analisado, dado por $NumSG$; sendo, $\overline{NumSG} = \left\{\left(\frac{\sum_i^n \overline{\text{NumSH}_i}}{n}\right) \in Pat\right\}$.

Em termos do número de solicitações em um *Pat* é possível avaliar o quão distante um usuário sob análise $NumSH$ encontra-se da média do traço $\overline{NumSG}$. Salienta-se que para um traço livre de ataque, as duas médias tendem a ser próximas. Para o cálculo (*i*) entende-se necessário a divisão do *NumS* pela *TxInH* para que o total de solicitações seja proporcional ao tempo que o usuário esteve interagindo com o servidor. Visto que, um usuário com uma *TxInH* maior tende a apresentar resultados maiores também para $NumSH$.

3.1.3 Tempo entre Estados Interativos

Entre dois estados interativos existe um intervalo de tempo de não interatividade do usuário, dado por tempo entre estados interativos (*Teei*). Este intervalo de tempo inicia na constatação do final de um período interativo $(\Delta t' = 1)$com o início do próximo período interativo. Sendo assim, para a formação de um período *Teei* assume-se que é necessário um

conjunto de períodos como segue $\Delta t' = \{1; 0; 1\}$, onde o número de períodos inativos (0) é variável.

Dado a característica de navegação humana, num traço legítimo o *Teei* médio para um dado *host*, ou $\overline{Teei}$, tende ser maior do que para *bots*. Como observado em alguns estudos (Dhingra e Sachdeva, 2014; Singh et al., 2015; Xu et al., 2012), os períodos de não interatividade para *bots* tendem a zero segundos, e, de acordo com Hwang (2004) e Hwang et al. (2005) para humanos esses valores tendem a serem maiores do que 10 segundos. Logo, tem-se que$\overline{Teei_humano} \gg \overline{Teei_boot}$. Deste modo, o método de detecção proposto realiza duas medidas:

(*i*) a média dos$Teei$ de uma *chave*, dado por$TeeiH$, onde são divididos os valores $Teei$ de tempo pelo número *n* de ocorrências do comportamento dentro de um *Pat*, ou seja, $\overline{TeeiH} = \left(\frac{Teei}{n}\right)$; e

(*ii*)a média dos $\overline{TeeiH}$do *Pat* analisado, dado por $\overline{TeeiG}$, sendo, $\overline{TeeiG} = \left\{\left(\frac{\sum_i^n \overline{\mathrm{TeeiH}_i}}{n}\right) \in Pat(X)\right\}$.

3.2 MÉTODO BÁSICO PARA DETECÇÃO DE *DDOS MIMIC FLASH CROWD*

Com base no modelo comportamental descrito na seção 3.1, esta seção apresenta o novo método básico no qual se realiza o processamento comportamental dos *hosts* e a distinção entre humanos e *bots*, possibilitando assim a detecção de ataque *DDoS mimic Flash Crowd.* O método é sintetizado na Figura 5 e a arquitetura computacional para operação na Figura 6.

O método, intitulado *DVM* – Diferença aos Valores Médios, possui duas etapas principais:

- *(i)* Preparação e Modelagem Comportamental, responsável pela coleta do traço de rede e modelagem do comportamento. A modelagem é composta por duas fases, a de formação individual, que estratifica as variáveis de interesse primárias, e a de formação global, que agrega informações; e
- *(ii)* Decisão, responsável pela tomada de decisão. Classifica o *host* como humano ou comportamento malicioso (eventos suspeitos provavelmente gerados por *bots*).

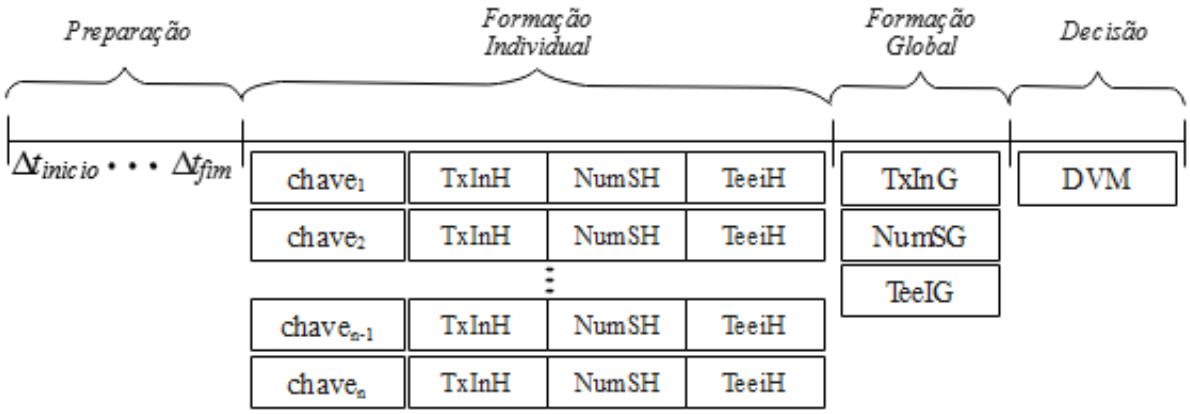

Figura 5. Etapas e fases do Método para Detecção de *DDoS mimic Flash Crowd*. Fonte: Elaborada pelo autor.

O método é desenvolvido para monitorar aplicações Web que apresentam um grau elevado de entropia no comportamento do usuário, ou seja, a demanda de solicitações HTTP(S) não apresentam-se de maneira permanente, como streaming de áudio, vídeo ou que exijam solicitações quase ininterruptas. Alguns exemplos que o método é aplicável são: Websites de instituição de ensino, empresas, organizações, softwares *open source,* portais de notícias, blogs, entre outros.

A arquitetura projetada para o método é apresentada na Figura 6, onde é representado o cliente, atacante, método e aplicação Web. Toda solicitação encaminhada a aplicação é espelhada pelo *switch* ao método, ou seja, é realizado *portmirror*. A partir destes dados é realizada a análise do traço e, caso averiguado um ataque, os *hosts* com comportamento de *bot* podem ser bloqueados na borda da Internet ou direcionados para o preenchimento de *CAPTCHA*.

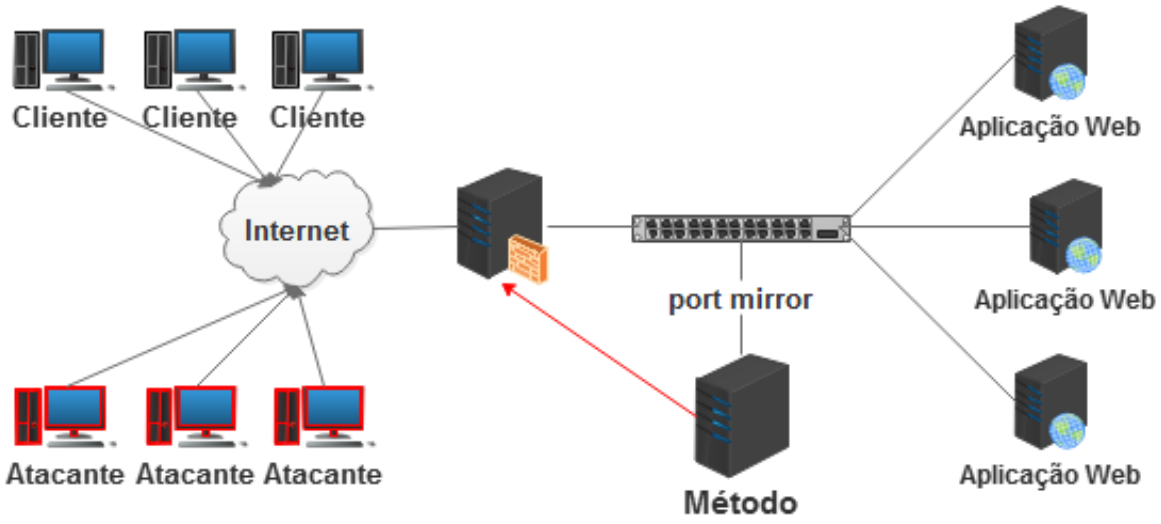

Figura 6. Arquitetura computacional do método. Fonte: Elaborada pelo autor.

A seguir são apresentados os detalhes de cada etapa (seções 3.2.1 e 3.2.2).

3.2.1 Etapa de Preparação e Modelagem Comportamental

A etapa de Preparação e Modelagem Comportamental é responsável por receber os dados brutos do traço de rede (Preparação) e transformá-los em informações comportamentais (Modelagem Comportamental) de acordo com o modelo proposto na seção 3.2. A fase de Preparação possui dois objetivos, sendo eles: (*i*) transformar dados da linguagem de captura de tráfego de rede (linguagem hexadecimal PCAP) para texto (ASCII), conforme ilustra a Figura 7; e (*ii*) buscar dentro do traço textual todas as solicitações que contenham porta de destino 80 ou 443 para gerar um novo traço filtrado para a fase de Formação Individual.

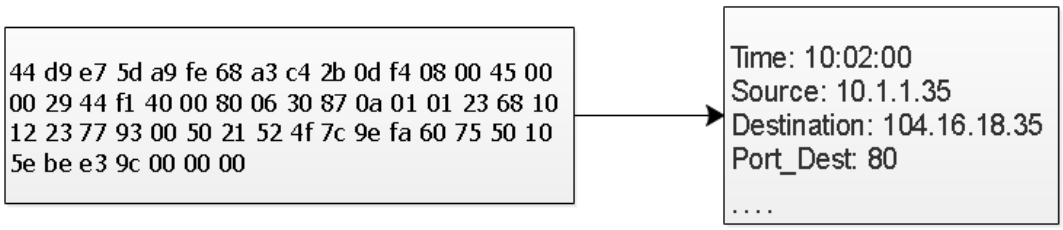

Figura 7. Transformação de Hexadecimal para texto. Fonte: Elaborada pelo autor.

A modelagem comportamental inicia com a fase de Formação Individual, conforme ilustra a Figura 5, onde as variáveis de interesse *TxIn*, *NumS* e *Teei* são processadas, e é concluída com a fase de Formação Global, onde a informação é agregada. A seguir, cada uma das fases da modelagem comportamental é detalhadamente descrita.

3.2.1.1 Fase de Formação Comportamental Individual

A fase de Formação Comportamental Individual expressa o comportamento médio de uma dada *chave*, chegando aos valores *TxInH, NumSH* e *TeeIH*. A formação é extraída conforme representação algorítmica ilustrada na Figura 8.

Inicialmente o algoritmo percorre o *Pat*, linha [2]; percorre o traço e armazena as *chaves*, linhas [3-6]; percorre as *chaves* e calcula seus respectivos comportamentos de acordo com as seções 3.1.1, 3.1.2 e 3.1.3, linhas [7-9]. Com o término das *chaves*, o algoritmo percorre o próximo *Pat.*

```
   Entrada: Traço de Rede
   Saída: Comportamento dos Hosts
 1 inicio
 2 while há Pat para analisar do
 3 |   while há novas chaves do
 4 |   |   grava_chave()
 5 |   |   verifica se há nova chave
 6 |   end
 7 |   while chave sem comportamento modelado do
 8 |   |   grava_comportamento(calcule_TxInH(),calcule_NumSH(),
   |   |    calcule_TeeIH())
 9 |   |   verifica se há chave ainda não modelada
10 |   end
11 end
```

Figura 8: Algoritmo de Formação Comportamental Individual. Fonte: Elaborada pelo autor.

3.2.1.2 Fase de Formação Comportamental Global

A fase de Formação Comportamental Global expressa o comportamento médio do traço sob análise, chegando aos valores *TxInG, NumSG* e *TeeIG*. O cálculo dos mesmos é realizado somando a média das *chaves* e dividindo pelo número de ocorrências, conforme apresentado detalhadamente na seção 3.1.

Ao término desta fase as informações comportamentais do *Pat* analisado são conhecidas, ou seja, são conhecidos valores *TxInH, NumSH* e *TeeIH* de todas as *chaves* e os valores médios do traço *TxInG, NumSG* e *TeeIG*.

3.2.2 Etapa de Decisão

A etapa de Decisão realiza a classificação dos *hosts* em humano ou *bot*, examinando os comportamentos modelados na etapa de preparação e formação comportamental, seção 3.1.1.Três variáveis de interesse são modeladas *TxIn*, *NumS* e *Teei*, cada uma delas pode expressar o comportamento médio de um dado *host TxInH, NumSH* e *TeeIH* e, juntas, o comportamento médio do traço sob análise *TxInG, NumSG* e *TeeIG*. Para um dado período de atividade amostrado (*Pat*), o cálculo das diferenças aos valores médios tem o intuito de explicitar a distância comportamental de um *host* frente aos comportamentos esperados. À distância, expressa por *host DVMH* – Diferença aos Valores Médios do Host, é função dos percentuais diferenciais relativos a taxa de interatividade, número de solicitações e tempo entre estados interativos, tal como segue.

$$\text{DVMH}=\left[\left(\frac{NumSH\text{-}NumSG}{NumSG}.100\right)+\left(\frac{TxInH\text{-}TxInG}{TxInG}.100\right)+\left(\frac{TeeiH\text{-}TeeiG}{TeeiG}.100\right).(-1)\right]$$

O valor *DVMH* assume magnitude positiva e negativa, e quando zero indica equivalência comportamental do *host* com a do traço envolvendo todos os *hosts*. Quanto mais negativo, mais distante o mesmo está de um comportamento maligno. Por outro lado, quando positivo, quanto maior sua magnitude maior é a chance de representar um comportamento malicioso. Na formulação, o cálculo percentual do *TeeiH* é multiplicado por menos um (-1), pelo fato do seu valor desejado ser o maior possível. Quanto maior mais próximo das características humanas, conforme discutido na seção 3.1.3, diferentemente de *TxInH* e *NumSH* que quanto menor mais próximo do humano.

Para avaliar se os valores de *DVMH* indicam ataque é aplicado um limiar (*threshold*). O método considera um limiar adaptativo calculado com base no *threshold* universal por nível (Donoho e Johnstone, 1995), tal como aplicado em Dalmazo et al. (2009). O limiar adaptativo é dado por:

$$\text{L}=\left(\sqrt{2\log n\sigma^2}\right)\times(2\log n\sigma^2)$$

Onde o limiar universal $\left(\sqrt{2\log n\sigma^2}\right)$ é multiplicado por $2\log n\sigma^2$, chegando no *threshold* aplicável as magnitudes do *DVMH*. De acordo com a Figura 9, com este *threshold* o

método de diferenciação do comportamento humano e de atacantes *bot* adota como parâmetro uma função de corte nas diferenças aos valores médios, que calculadas por *host* permite distinguir *bots* e identificar ataques *DDoS* mesmo em momentos de ampliação de tráfego *Flash Crowd*.

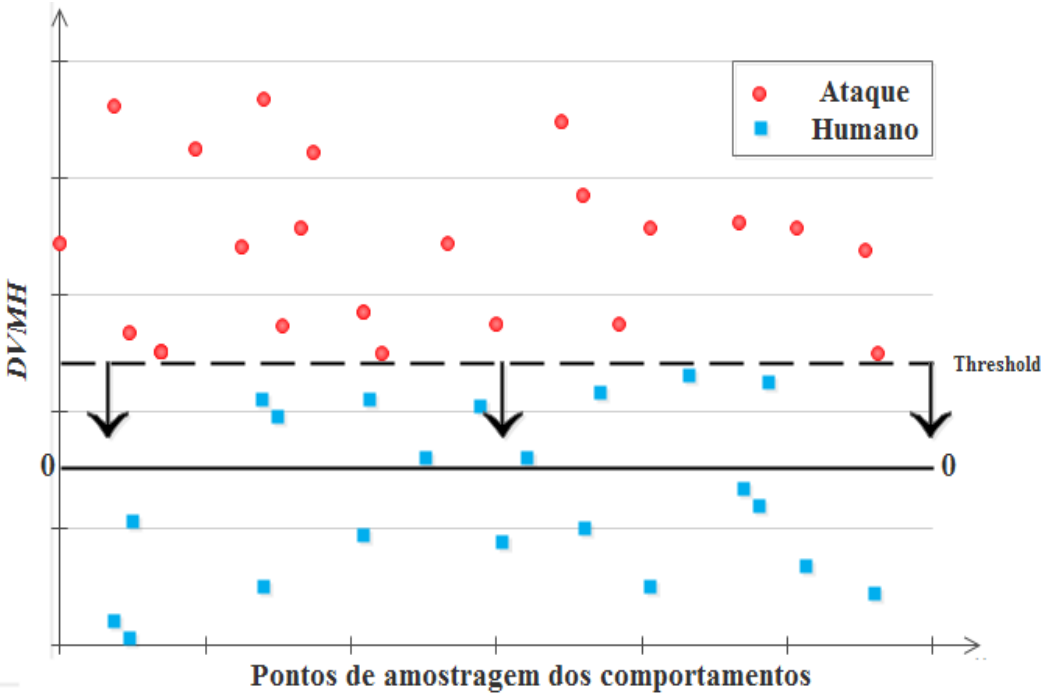

Figura 9: Diferenciação do comportamento humano e de ataque. Fonte: Elaborada pelo autor.

3.3 MÉTODO OTIMIZADO PARA DETECÇÃO DE *DDOS MIMIC FLASH CROWD*

Esta seção apresenta um segundo método para detecção de ataque *DDoS mimic Flash Crowd*, que deriva do método básico (seção 3.2) por aproveitar a etapa de preparação e modelagem comportamental, e apresenta uma nova etapa de decisão otimizada, com novo recurso de monitoramento de eventos abruptos nos servidores e novo mecanismo de cálculo que usa as porcentagens de variação ao invés dos valores médios. A ideia é evitar classificações de comportamentos não *Flash Crowd* e ser mais preciso na classificação, potencializando a redução dos falsos positivos.

O método, intitulado *DAPV* - Diferença Acumulativa das Porcentagens de Variação, é sintetizado na Figura 10 e possui duas etapas principais:

(i) Preparação e Modelagem Comportamental (idêntica ao do método básico *DVM*), responsável pela coleta do traço de rede e modelagem do comportamento. A modelagem é composta por duas fases, a de formação individual, que estratifica as variáveis de interesse primárias, e a de formação global, que agrega informações; e

(ii) Decisão, responsável pelo monitoramento dos servidores Web e seleção dos *hosts* com comportamentos maliciosos (eventos suspeitos provavelmente gerados por *bots*).

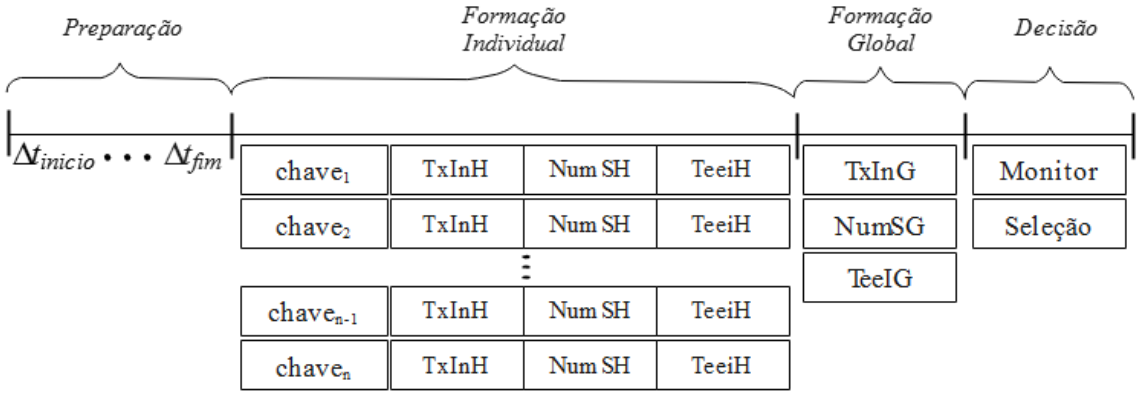

Figura 10. Etapas e fases do Método otimizado para Detecção de *DDoS mimic Flash Crowd*. Fonte: Elaborada pelo autor.

A etapa de preparação e formação comportamental é apresentada na seção 3.2.1 e a seguir são apresentado os detalhes da nova etapa de decisão na seção 3.3.1.

3.3.1Etapa de Decisão

A etapa de Decisão otimizada realiza a classificando dos *hosts (chaves)* em humano ou *bot*, examinando os comportamentos modelados na etapa de preparação e formação comportamental de acordo com duas fases: Monitoramento e Seleção Intencional. O ponto chave desta otimização é realizar a seleção de comportamentos maliciosos apenas se houver aumento repentino e demasiado de *hosts* a um dado servidor. Desta forma, o método otimizado *DAPV* difere do *DVMH* por atuar na diferenciação apenas se um comportamento equivalente a um *Flash Crowd* for percebido.

Na seção 3.3.1.1 é apresentado o detalhamento da fase de Monitoramento e na seção 3.3.1.2 o detalhamento da fase de Seleção Intencional.

3.3.1.1 Monitoramento

Um evento *Flash Crowd* corresponde a um aumento repentino e demasiado de usuários (YU et al., 2012), logo, o monitoramento tem como objetivo observar o histórico recente do número de usuários que realizaram solicitações a um dado servidor Web e apontar possíveis aumentos. Desta forma, com o uso do monitoramento, a fase de seleção somente é acionada com a observação de tal aumento, economizando recursos computacionais e realização a mitigação nos momentos necessários.

A estrutura de dados desta fase é apresentada na Figura 11, onde é composta pelo Endereço IP do servidor e o número de usuários que realizaram solicitação a este servidor no respectivo *Pat*. Sendo *Pat*(*n*) a última contagem e *Pat* (*n* - 4) a quinta.

	Nº de usuários	Nº de usuários	Nº de usuários	Nº de usuários	Nº de usuários
Servidor(x)	Pat(n - 4)	Pat(n - 3)	Pat(n - 2)	Pat(n - 1)	Pat(n)
Servidor(y)	Pat(n - 4)	Pat(n - 3)	Pat(n - 2)	Pat(n - 1)	Pat(n)
⋮					
Servidor(w)	Pat(n - 4)	Pat(n - 3)	Pat(n - 2)	Pat(n - 1)	Pat(n)
Servidor(z)	Pat(n - 4)	Pat(n - 3)	Pat(n - 2)	Pat(n - 1)	Pat(n)

Figura 11. Estrutura de dados da fase de Monitoramento. Fonte: Elaborada pelo autor.

A estrutura do monitoramento proporciona uma percepção imediata na ocorrência de evento *Flash Crowd*, já que a última contagem de usuários *Pat*(*n*) irá apresentar uma sobressaliênca demasiada em relação as contagens anteriores. A análise de uma quantidade menor de valores proporciona uma maior percepção do aumento repentino, pois em um contexto maior o aumento pode não representar uma sobressaliência. Esse fato é ilustrado nas Figuras 12 e 13, onde o crescimento no instante 37 da Fig.12 não representa o maior pico. Porém, observando somente as 5 últimas contagens o pico do instante 37 fica representativo, tal como observado na Figura 13. Sendo assim, a fase de monitoramento examina um número pequeno de contagens. Os experimentos da seção 4.3 exploram os limiares para o número de contagens, ou seja, é definido a quantidade de contagens a ser utilizado.

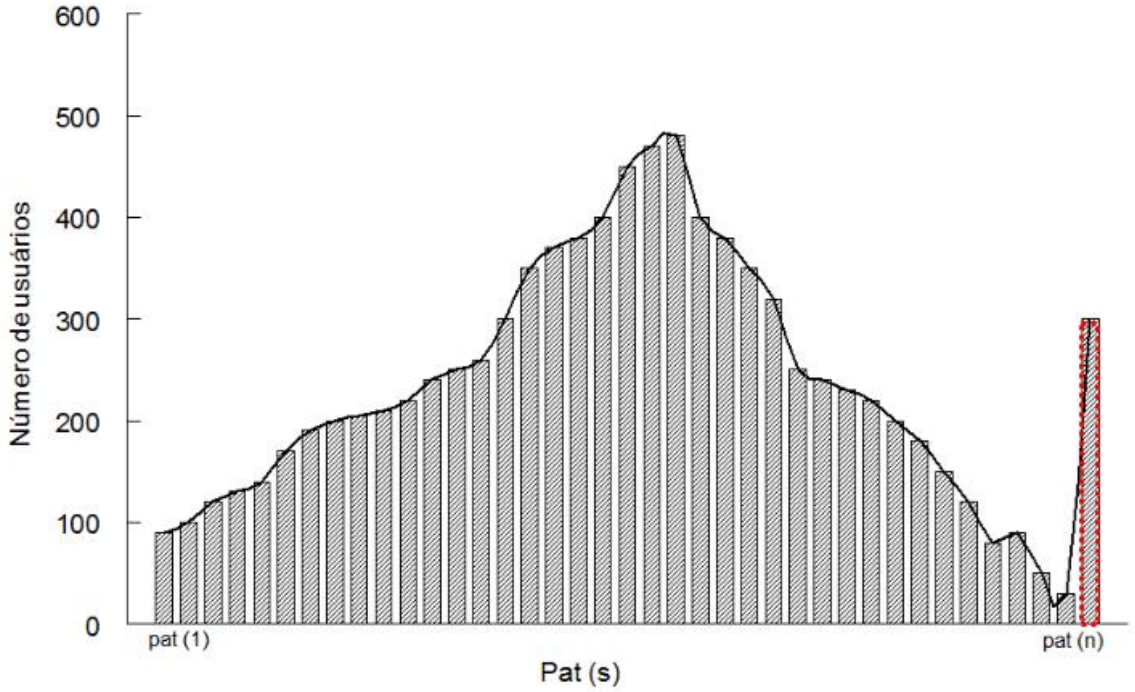

Figura 12. Ilustração da contagem de usuários por *Pat* com amostragem grande. Fonte: Elaborada pelo autor.

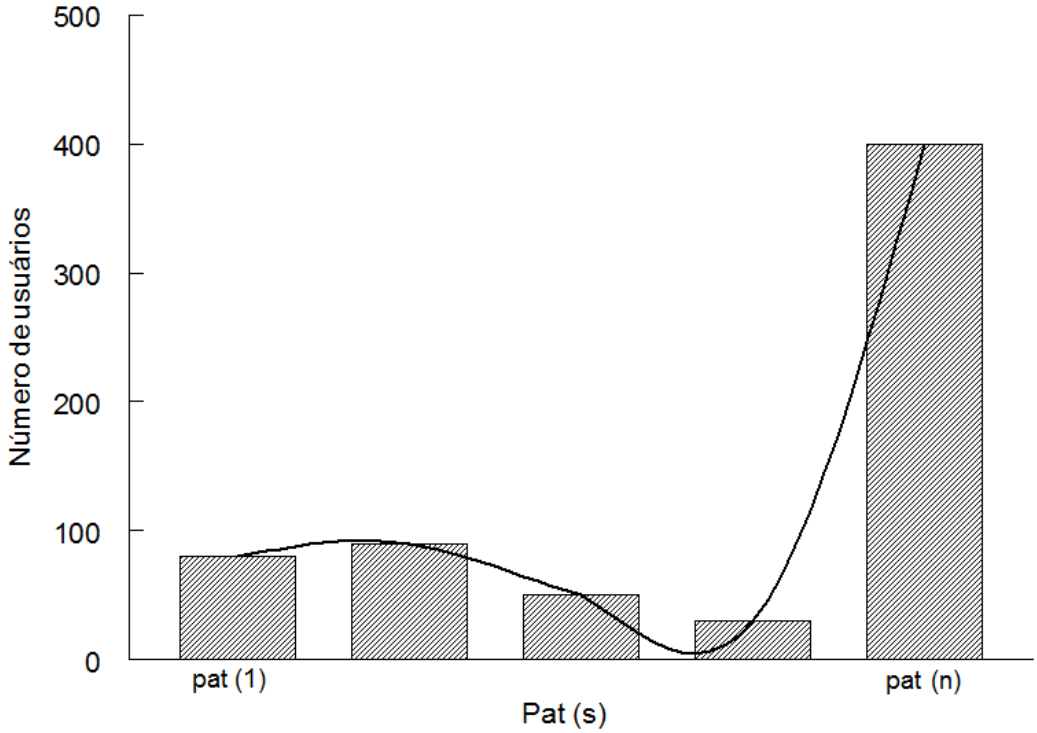

Figura 13. Ilustração da contagem de usuários por *Pat* com amostragem pequena. Fonte: Elaborada pelo autor.

O aumento repentino do número de usuários pode aparecer de duas formas:(*i*) em formato de “ziguezague” e (*ii*) crescimento contínuo, as quais podem apresentar taxa de

crescimento dentro (Figura 14 (a) e (c)) ou fora do padrão histórico (Figura 14 (b) e (d)). A taxa fora do padrão vai ocasionar um crescimento repentino e demasiado, como observado nas Figuras 14 (b) e (d).

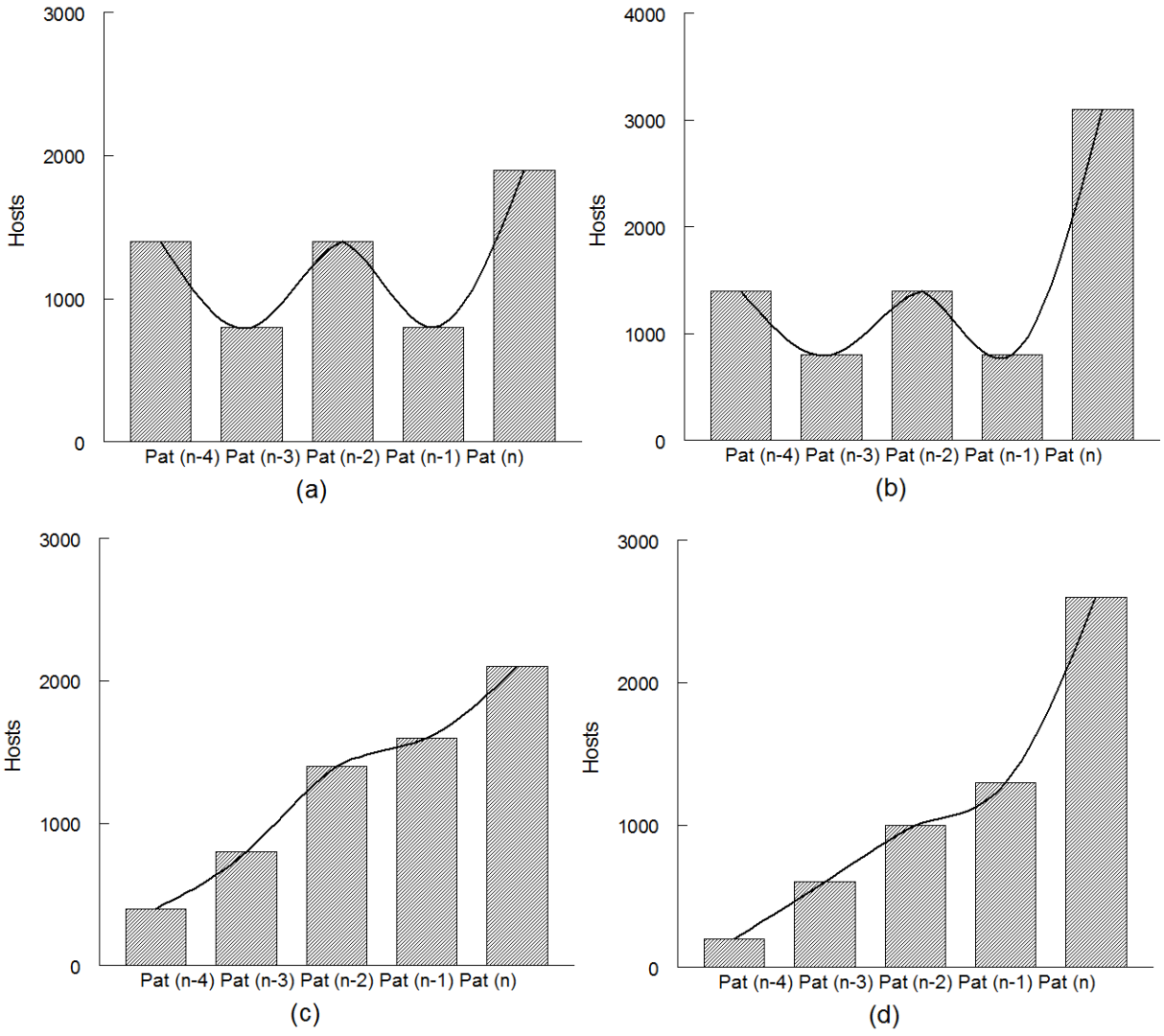

Figura 14. Ilustração do número de usuários em momentos normais, (a) estacionários e (c) não estacionários, e eventos *Flash Crowd*, (b) estacionários e (d) não estacionários. Fonte: Elaborada pelo autor.

Para detectar um pico demasiado de usuários, tanto em comportamentos estacionários (Figuras 14(a) e 14(b)) como em comportamentos não estacionários (Figuras 14(c) e 14(d)) a fase de monitoramento realiza o cálculo do Grau de Curtose de cada servidor Web. A curtose

corresponde ao grau de "achatamento" de uma curva medido em relação ao de uma curva normal (Gauss), ou ainda, o grau de concentração de valores da distribuição em torno do centro da distribuição. Segundo Kappel (2015), a curtose identifica elevadas oscilações do sinal no domínio de tempo e é muito útil para detectar descontinuidades. Num sinal com elevada curtose, a variação ocorre em um tempo curto, consequentemente, uma janela pequena de tempo é necessária para se ter uma boa observação, o que é desejado no método *DAVP*. O cálculo é baseado na formulação de *Pearson Small Sample* (XYCOON 2017), utilizando a seguinte fórmula:

$$\text{Curtose} = \left(\frac{\boldsymbol{n(n+1)}}{\boldsymbol{(n-1)(n-2)(n-3)}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \left(\frac{\boldsymbol{x_i - x}}{\boldsymbol{s}}\right)^{\boldsymbol{4}}\right) - \frac{\boldsymbol{3(n-1)^2}}{\boldsymbol{(n-2)(n-3)}}$$

Onde:

A curtose possui três classificações para as curvas juntamente dos valores para enquadrar o grau que uma curva se encaixa (DECARLO 2012; MATHWORLD 2017), ilustrado na Figura 15, sendo elas:

(i) **Mesocúrtica:** quando apresenta uma medida de curtose **igual** à da distribuição normal, Curtose = 3;

(ii) **Platicúrtica:** quando apresenta uma medida de curtose **menor** que a distribuição normal, Curtose<3 e;

(iii) **Leptocúrtica:** quando apresenta uma medida de curtose **maior** que a distribuição normal, Curtose>3.

Para exemplificar, o grau de curtose observado nos quatro gráficos da Figura 14 é de (a) -1,21 – Platicúrtica, (b) 3,10 – Leptocúrtica, (c) -1,12 – Platicúrtica e (d) 3,30 – Leptocúrtica. Este resultado demonstra a capacidade desta métrica para caracterizar a variabilidade alvo do monitoramento. Para que o cálculo seja realizado para um determinado conjunto de valores é exigido uma variância mínima dos mesmos, desta forma, eliminando o processamento para conjuntos que não possuem uma variabilidade tolerável e contribuindo para a diminuição de falsos positivos. Para exemplificar isso, o conjunto amostral [100, 102, 105, 109, 107] apresenta variância de 13,3 e o conjunto [100, 102, 150, 109, 107] apresenta 427.3, sendo assim, com uma variância mínima de 20 o primeiro conjunto não é processado. Destaca-se, que em um evento *Flash Crowd* o conjunto apresenta grande variância. O valor mínimo é definido nos experimentos.

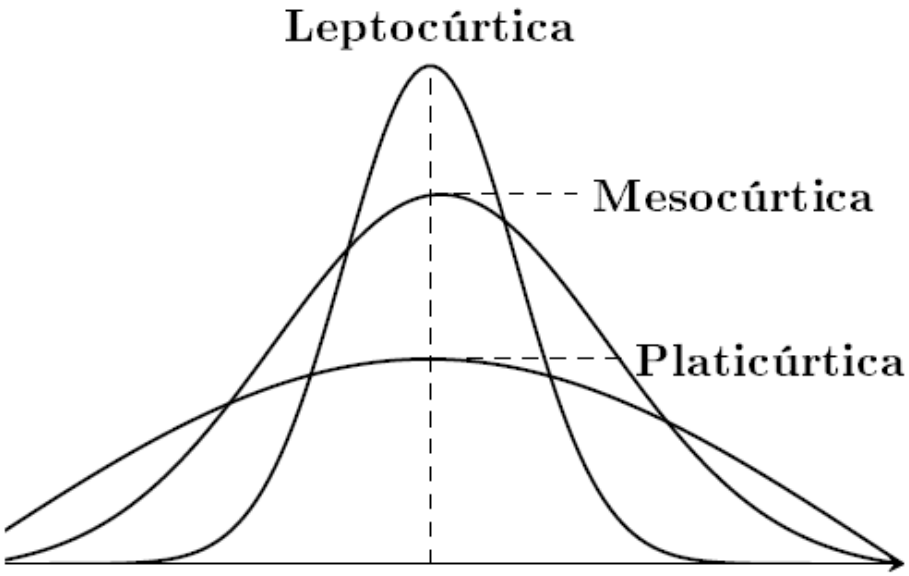

Figura 15 Classificação dos tipos de Curtoses. Fonte: Elaborada pelo autor.

Tendo como métrica o Grau de Curtose, a fase de Monitoramento aciona a fase de Seleção apenas quando um servidor apresentar um pico demasiado de usuários, ou seja, uma curva mesocúrtica ou leptocúrtica. Desta forma, a fase de Seleção fica operante apenas se os valores apresentam comportamento anômalo, reduzindo a probabilidade de um falso positivo.

3.3.1.2 Seleção Intencional

A fase de Seleção Intencional é responsável por examinar o comportamento dos *hosts* (*chaves*) modelado na etapa de preparação e modelagem comportamental (seção 3.2.1) e classifica-los como humano ou *bot*, ou seja, é a etapa final que toma a decisão. Quando classificado como *bot*, como medida de mitigação do dano o endereço de origem do *host* pode ser adicionado a uma lista de bloqueio ou direcionado para um preenchimento de *captcha.*

Esta fase tem como entrada de dados os comportamentos individuais de cada *chave* *TxInH*, *NumSH* e *TeeiH* e os valores médio do traço *TxInG*, *NumSG* e *TeeIG*. Para um dado período de atividade amostrado (*Pat*), é calculado para cada *host* o valor *DAPVH* – Diferença Acumulativa das Porcentagens de Variação do Host, que é função das porcentagens de variação (*PV*) relativas à taxa de interatividade, número de solicitações e tempo entre estados interativos, tal como segue.

$$\boldsymbol{DAPVH} = \boldsymbol{PV(TxInH)} + \boldsymbol{PV(NumSH)} + \boldsymbol{PV(TeeIH)}$$

O valor de *DAPVH* expressa de forma acumulativa as porcentagens de variação (*PV)* das três variáveis modeladas *TxInH, NumSH* e *TeeiH*, sendo *PV* um limiar adaptativo acrescido à média do traço variando de acordo com a variabilidade dos dados. O valor *DAPVH* pode ser assim classificado de acordo com três níveis: traço, PV(traço) ou *Threshold*, vide Figura 16 e detalhados a seguir.

Traço - seleciona os *hosts* que apresentam comportamento menor ou igual que a média do traço nas três variáveis, ou seja,

$$\boldsymbol{DAPVH} \leq \overline{\boldsymbol{traço}}$$

PV(*traço*) – seleciona os *hosts* com comportamento de variabilidade maior que a média do traço em uma das três variáveis de interesse, mas menor ou igual a *PV* da respectiva variável, ou seja,

$$\boldsymbol{DAPVH} \leq \boldsymbol{PV(traço)}$$

Threshold – seleciona os *hosts* com comportamento maior que o *PV* (traço) E maior ao *threshold* no respectivo *Pat*.

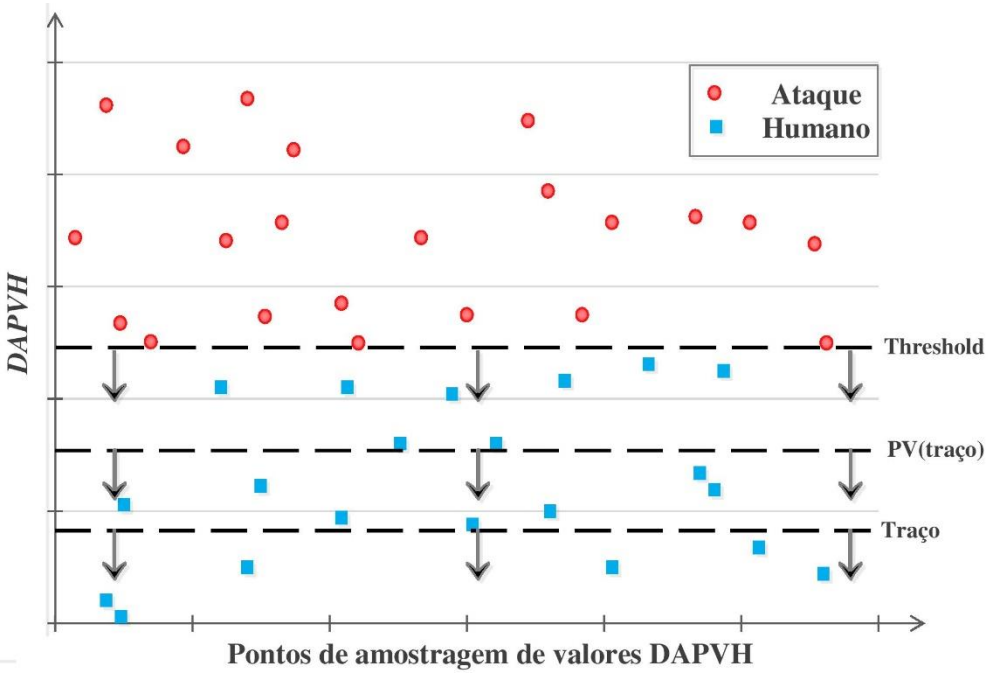

Figura 16. Gráfico ilustrativo de pontos de valores *DAPVH* e ilustração das classificações *traço*, *PV*(*traço*) e *Threshold*. Fonte: Elaborada pelo autor.

Os *hosts* classificados dentro de uma das três categorias são considerados como de comportamento de humano, enquanto os classificados acima de *threshold* como de

comportamento de *bot*, conforme ilustra a Figura 16. Os valores para os comportamentos médio do traço são obtidos na etapa de preparação e modelagem comportamental (seção 3.2.1) e os valores de *PV* (traço) e *Threshold* são obtidos conforme descrito a seguir.

O valor da *PV* tem como objetivo criar um segundo limiar acima da média do traço para classificar aqueles *hosts* que possuem um comportamento próximo da média, mas levemente acima. Para cada uma das três variáveis *TxInG, NumSG* e *TeeIG* é acrescido a variabilidade dos respectivos comportamentos dentro do *Pat*, chegando em *TxInGPV, NumSGPV* e *TeeIGPV*.

Para medir a variabilidade dos dados é utilizado o cálculo do desvio padrão (σ) que é uma medida de dispersão, o qual mede a variabilidade dos valores à volta da média. O valor mínimo é zero, indicando que não há variabilidade, ou seja, todos os valores são iguais à média, ou, quanto maior o σ mais variados são os valores. Desta forma, as oscilações comportamentais de todas as *chaves* podem ser medidas e incorporadas as medidas de *PV*. O cálculo do σ é realizado conforme segue.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}}$$

Para se chegar nos valores médios acrescido da variabilidade$(media + varialidade)$ são realizados os cálculos como segue.

$$TxInGPV = TxInG + \sqrt{\sigma(TxIn)},$$

$$NumSGPV = NumSG + \sqrt{\sigma(NumS)},$$

$$TeeIGPV = TeeIG + \sqrt{\sigma(TeeI)}.$$

É extraída a raiz quadrada do $\sigma\left(\sqrt{\sigma}\right)$para obter-se apenas uma parte desta variabilidade, ou seja, incorporar dentro do limiar somente aqueles comportamentos maiores, mas próximos da média. Com os valores *TxInGPV, NumSGPV e TeeIGPV* é calculado o valor *DAPVH* de cada *host*, somando $PV(TxInH) + PV(NumSH) + PV(TeeIH)$.

O cálculo dos valores *PV* são realizados como segue *PV* (*TxInH*) é calculado conforme ilustra a Figura 16. Se $TxInH > TxInGPV$ o retorno é o cálculo da variância percentual, linhas 2 e 3, senão, o retorno é zero (0), linha 4 e 5. O cálculo do *PV* (*NumSH*) segue a mesma lógica.

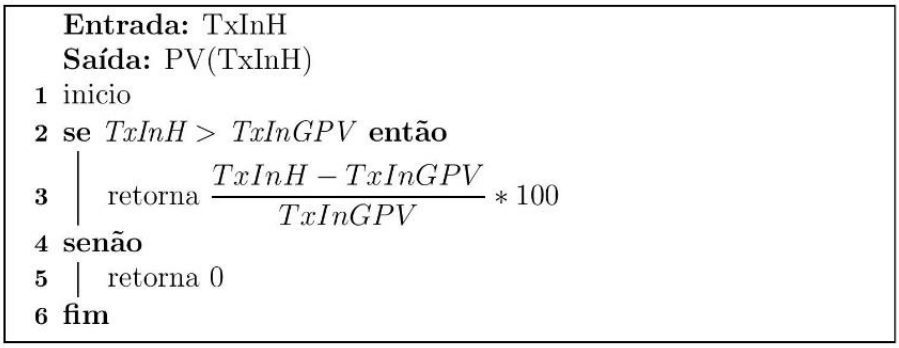

Figura 17. Algoritmo do cálculo do valor *PV* (*TxInH).* Fonte: Elaborada pelo autor.

O cálculo de *PV* (*TeeiH*) é apresentado na Figura 18(algoritmo de cálculo). Se$TeeIH < TeeIGPV$ o retorno é o módulo da variância percentual, linhas 2 e 3. O módulo é necessário pois quando o valor *TeeiH* é menor que *TeeIGPV* a porcentagem retornada é negativa, sendo que, o esperado é a variância percentual positiva para ser somado ao *DAPVH.*

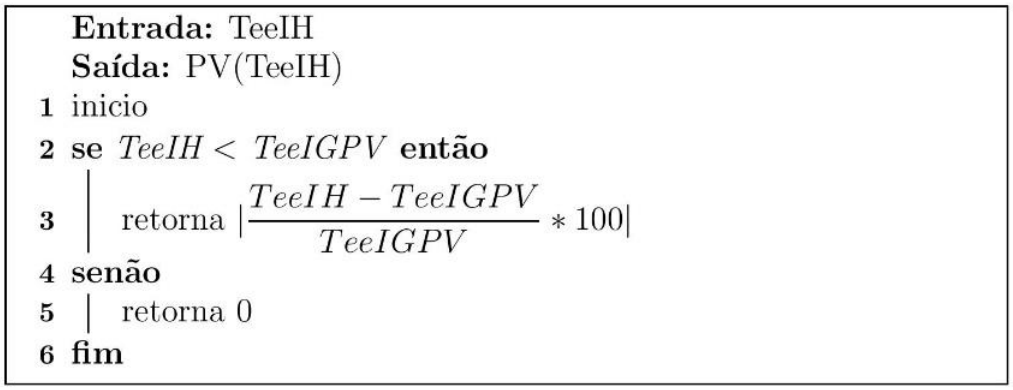

Figura 18. Algoritmo do cálculo do valor *PV*(*TeeIH*). Fonte: Elaborada pelo autor.

Para ilustrar o comportamento de *PV*, é apresentado na Figura 19 um exemplo do *NumSG* onde existem dez valores com média de $\boldsymbol{NumSG = 49}$ e $\boldsymbol{\sigma = 17}$ chegando em $\boldsymbol{NumSGPA = 53,12}$. Os resultados da *PV* das chaves são apresentados na Tabela 1. É possível observar que, utilizando o *NumSGPV* as *chaves*2 e 10 que estavam acima da média *NumSG* foram classificadas como zero (0).

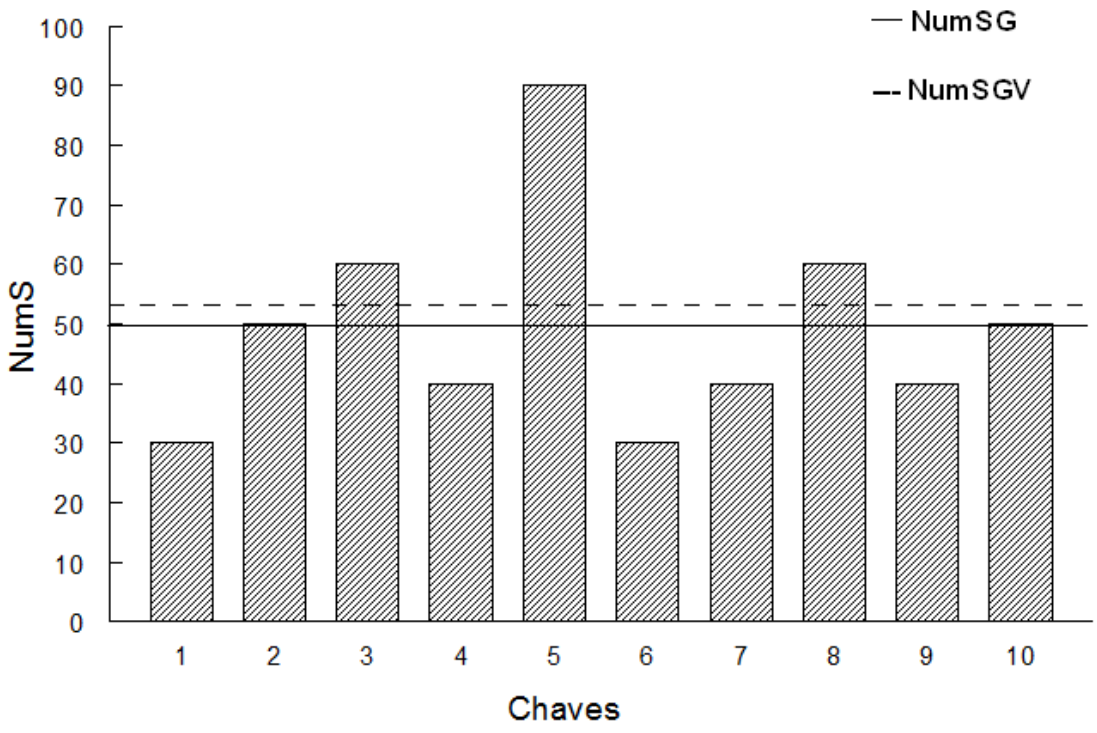

Figura 19. Exemplificação de valores para cálculo da Porcentagem de Variação. Fonte: Elaborada pelo autor.

Tabela 1- Valor da Porcentagem de Variação dos valores da Figura 19.

Chave	1	2	3	4	5	6	7	8	9	10
PV	0	0	12,95	0	69,42	0	0	12,95	0	0

Na Figura 20 é apresentada uma ilustração com quatro gráficos (a, b, c e d), contendo os valores *NumSG* em linha contínua e *NumSGPV* em linha tracejada, com seus respectivos desvio padrão (e). Observa-se na Figura 20 (a) e (c) que o limiar *NumSGPV* é variável de acordo com os valores apresentados, ou seja, mesmo ocorrendo uma variação na taxa dos comportamentos o mesmo se ajusta, contribuindo para a diminuição de resultados falso-positivos e falso-negativos. Assim, se existir uma variabilidade maior, o limiar apresentará uma maior tolerância também, como observado na Figura 20 (d), que apresentou desvio padrão de 11,26.

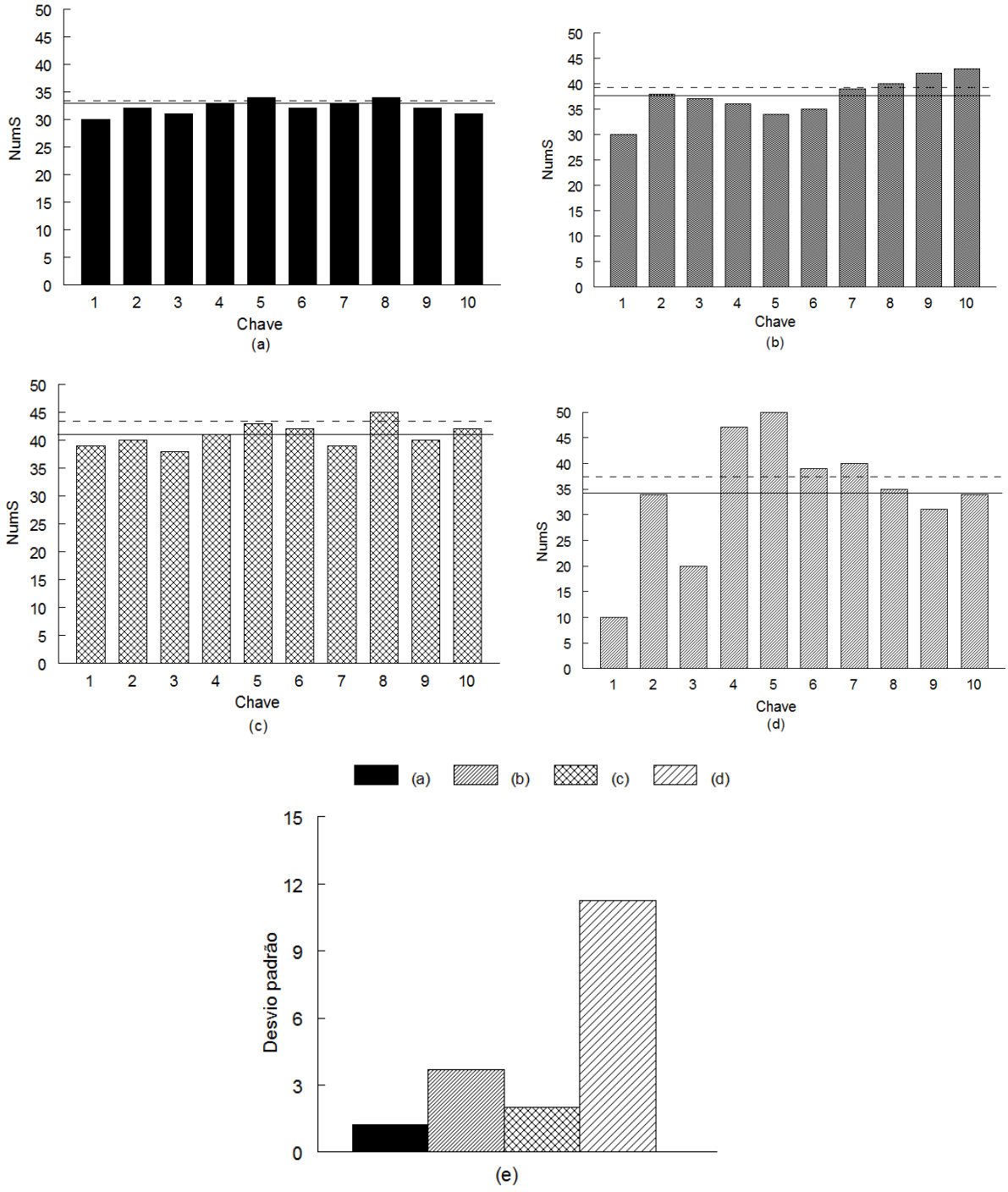

Figura 20. Ilustração do desvio padrão com diferentes valores. Fonte: Elaborada pelo autor.

Uma vez computado os três valores da *PV*, o valor de *DAPVH* para cada *chave* pode ser computado como segue.

$$\boldsymbol{DAPVH = PV(TxInH) + PV(NumSH) + PV(TeeIH)}$$

O *DAPVH* assume magnitude zero tendendo a mais infinito ($\mathbf{0} \rightarrow +\propto$), onde, zero indica comportamento abaixo dos valores *PV* nas três variáveis. Maior que zero indica comportamento se distanciando de um comportamento normal e se aproximando de *bot*. A soma das três variáveis permite uma análise em conjunto dos três comportamentos, não baseando a detecção em cima de somente um comportamento.

Novamente, para avaliar se os valores de *DAPVH* indicam ataque é aplicado um limiar (*threshold*). O método considera também um limiar adaptativo calculado com base no *threshold* universal por nível (Donoho e Johnstone, 1995), da mesma forma que foi aplicado no método básico *DVMH*.

O algoritmo da seleção é apresentado na Figura 21. O cálculo do *PV* dos comportamentos e do *threshold* são realizados nas linhas 2 e 3, respectivamente. Na linha 4 é iniciado o laço que irá percorrer todas as chaves do traço, realizando o cálculo das *PVs* (linha 5) e o *DAPVH* (linha 6) no qual é verificado se extrapola o limite aceitável (linha 7-9). Na existência de chaves classificadas como *bot* é disparado o alarme (linhas 11-12).

```
   Entrada: Formação Indivídual e Global
   Saída: Seleção de chave
 1 inicio
 2 calcula (PV dos comportamentos)
 3 calcula (threshold)
 4 while enquanto existir chave do
 5 |   calcula (PVs da chave)
 6 |   DAPVH(chave)=PV(TxInH) + PV(NumSH) + PV(TeeIH)
 7 |   se DAPVH > THRESHOLD então
 8 |   |   grava chave como BOT
 9 |   fim
10 end
11 se BOT existe então
12 |   Dispara Alarme
13 fim
```

Figura 21.Algoritmo da etapa de Seleção Intencional. Fonte: Elaborada pelo autor.

4 TESTES E RESULTADOS

Neste capítulo são apresentados os experimentos realizados para examinar e avaliar o método proposto de detecção de ataques *DDoS mimic Flash Crowd* e os padrões comportamentais do usuário em traços de rede reais.

A avaliação do método é realizada em diferentes cenários, utilizando traços de rede reais e também sintéticos, devido à grande dificuldade de obtenção de um traço de ataque *DDoS mimic Flash Crowd* real originário de uma *botnet*. Os experimentos visam validar a eficácia do método e o formato da modelagem comportamental para diferentes traços de rede.

Neste capítulo são apresentados os traços de rede e ferramentas para geração de ataques (seção 4.1), métricas para avaliação do método de detecção (seção 4.2), experimentos com seus respectivos resultados (seção 4.3) e estudos de caso (seção 4.4).

4.1 TRAÇOS DE REDE E FERRAMENTAS PARA GERAÇÃO DE ATAQUES

A obtenção de amostras reais de tráfego malicioso do tipo *DDoS Flash Crowd* é um desafio. Entretanto, a validação experimental pode também trabalhar com geração de ataques artificiais (LUCENA e MOURA, 2010; XIE e YU, 2009) inseridos ao tráfego de fundo sem ataque. Neste trabalho para tráfego de fundo foram utilizados dados das bases de dados públicas CAIDA (CAIDA, 2016) e WITS (Waikato, 2016). Os traços de WITS já foram utilizados por Behal e Kumar (2016), Jardim et al. (2016) e Lu et al. (2007) e os traços da CAIDA já foram utilizados por Babu et al. (2011), Bhandari et al. (2016), Bhuyan et al.(2015), David e Thomas (2015), Hwang et al. (2005), Jardim et al. (2016), Khundrakpam et al. (2016), Righi e Nunes (2016) e Xie e Yu (2009).

As ferramentas utilizadas para geração de tráfego *Flash Crowd* e ataques *DDoS* foram a *Scorpius* (*sFlow Network Anomaly Simulator*) (Marcos e Mario, 2015) e a *Botloader* (BHATIA et al., 2014). A ferramenta *Scorpius* usa como entrada um arquivo de dados real como tráfego de fundo e injeta novos registros simulando anomalias como *DoS, DDoS* e *Flash Crowd* no intervalo desejado. O mesmo foi utilizado na geração de tráfego anômalo nos trabalhos de Amaral et al. (2016), Carvalho et al. (2016) e Fernandes et al. (2015). Já a ferramenta *Botloader* gera tráfego de rede de duas vias, ou seja, produz o tráfego real benigno e o anômalo. Como saída produz vários tipos de ataque *DDoS* e eventos *Flash Crowd*, tendo sido utilizado para geração de tráfego anômalo no trabalho de Khundrakpam et al. (2016).

4.2 MÉTRICAS

Para avaliação do método de detecção foram adotadas algumas métricas especificas nesta dissertação:

(i) Verdadeiro Positivo (*VP*): representa os comportamentos de ataque que foram classificados corretamente;

(ii) Verdadeiro Negativo (*VN*): representa os comportamentos normais que foram classificados corretamente como legítimo;

(iii) Falso Positivo (*FP*): representa os comportamentos normais classificados como ataque;

(iv) Falso Negativo (*FN*): representa os comportamentos de ataque classificados como normais;

(v) Taxa de Acurácia (*TA*): representa a capacidade do método apresentar resultados corretos. Essa medida percentual soma o número de verdadeiro positivo com verdadeiro negativo e divide pelo número de amostras normais somado com as anômalas, representado na equação a seguir:

$$TA = \left(\frac{Verdadeiro\ Positivo + Verdadeiro\ Negativo}{Amostras\ Normais + Amostras\ Anômalas}\right) \times 100$$

Para que as métricas sejam contabilizadas corretamente nos experimentos são utilizados endereços IP diferentes para cada grupo de usuários (humanos e *bot*). Desta forma, o número de *hosts* inserido no traço é confrontado com o apresentado pelo método para cada grupo.

4.3 EXPERIMENTOS

Os experimentos são divididos em análise (experimentos 1 e 2) e detecção (experimento 3). No experimento 1dois testes são feitos. O primeiro avalia se a modelagem comportamental é aplicável a traços pequenos e grandes e qual o melhor tamanho para o *Pat*. No experimento 2, da etapa de monitoramento e no terceiro a detecção de ataque *DDoS Flash Crowd.* A estrutura da realização do experimento de detecção é apresentada na Figura 22, onde, é injetado no traço de fundo eventos *Flash Crowd* e ataque *DDoS Flash Crowd*, chegando em um traço homogêneo a ser processado pelo método. Desta forma, o traço final é formado por comportamentos

normais (humano) e *bots* (ataque), possibilitando extrair as métricas desejadas. A seguir os experimentos.

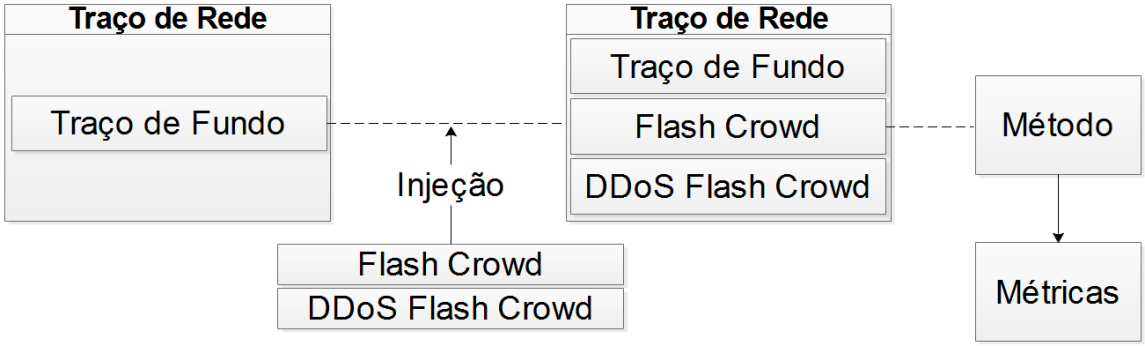

Figura 22. Estrutura dos experimentos.Fonte: Elaborada pelo autor.

4.3.1 Experimento 1: Análise da Modelagem Comportamental

Com o propósito de analisar a aplicabilidade da modelagem comportamental proposta o experimento 1 calcula TeeiG, NumSG e TxInG para traços de rede com magnitudes diferentes, um com menos de 1.000 hosts por Pat e outro com quase 200.000, conforme ilustra a Tabela 2.

Tabela 2- Dados dos traços utilizados no estudo de caso.

Traço	Data	Período	Média de hosts por Pat
CAIDA	19/02/2015	13:00h-13:10h	198,955
WITS	06/01/2010	15:30h-15:40h	734

Os traços foram avaliados e processados através da etapa de Preparação e Formação Comportamental, conforme seção 3.3.1.Os valores médios obtidos (média para todos os *Pats*) para cada traço são apresentados na tabela 3.

Tabela 3- Valores dos comportamentos médios do traço CAIDA e WITS

Traço	*TeeiG*	*NumSG*	*TxInG*
CAIDA	13,98	31,10	3,80
WITS	18,37	33,71	4,55

A regularidade e proximidade dos resultados demonstra capacidade do método para modelar o comportamento humano, permitindo, desse modo, a discrimição para não humanos (*bots*), independentemente do tamanho do traço. Ou seja, a modelagem individual de cada *host* pode ser formada tanto para traços pequenos quanto para traços grandes.

Os valores de cada *Pat* podem ser observados na Figura 23, onde a figura (a) apresenta o comportamento para os dados WITS e a figura (b) o comportamento para os dados CAIDA. A Figura 23 ilustra claramente a regularidade das medidas que descrevem o comportamento do traço.

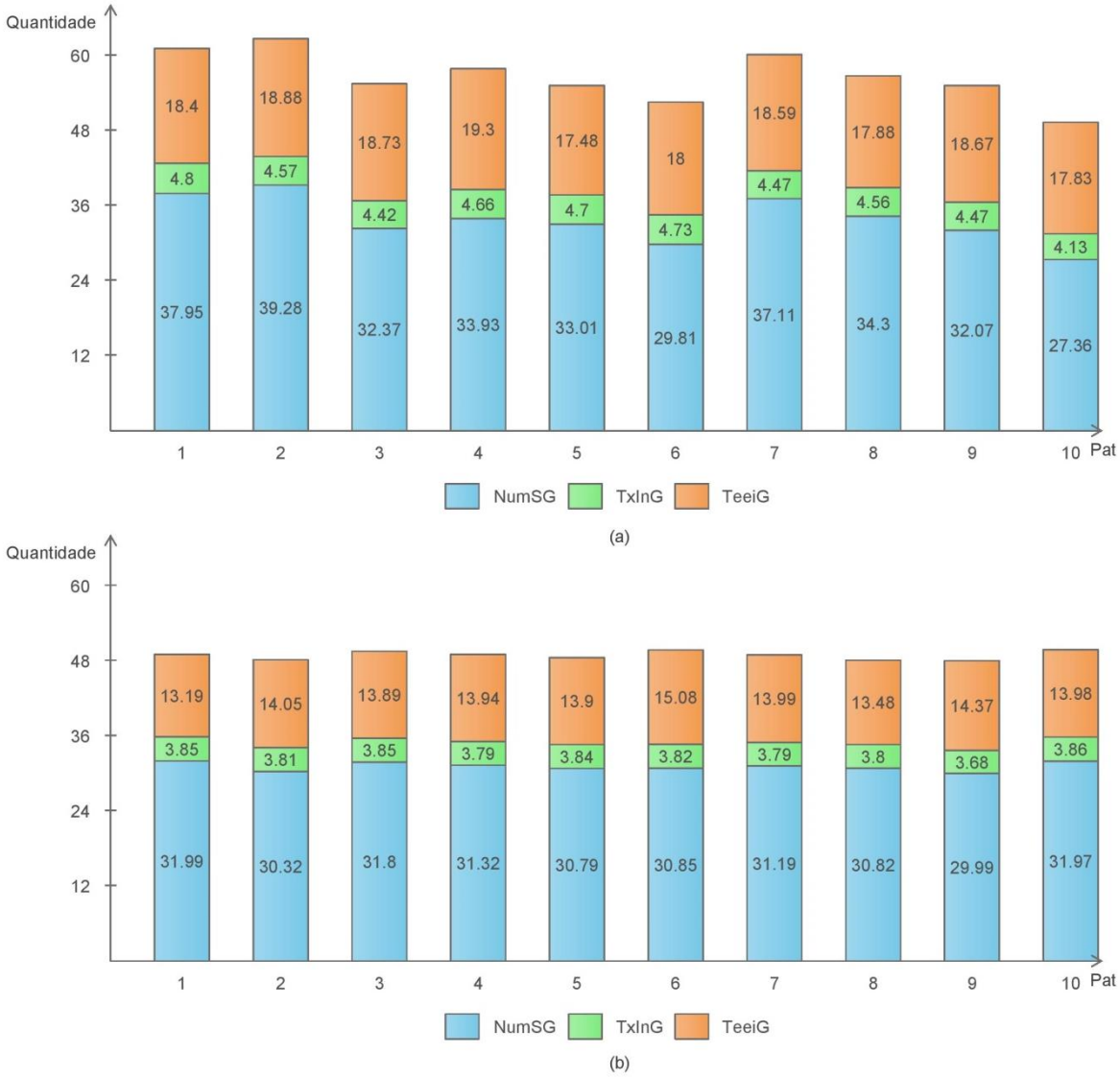

Figura 23. Comparativo do comportamento dos *hosts* nos traços CAIDA e WITS. Fonte: Elaborada pelo autor.

A segunda análise do experimento 1 tem como objetivo definir o tamanho do *Pat*, buscando a melhor representação do comportamento humano em torno do centro médio, ou

seja, com menor dispersão. A dispersão é calculada por medida de comportamento, ou seja, em cada fechamento de *Pat* é extraído o valor médio (*TxInG*, *NumSG* e *TeeiG*) e desses valores é obtido o desvio padrão. Os três desvio padrão são somados para indicar a dispersão para o tamanho de Pat analisado. O tamanho com menor somatório é escolhido como *Pat* padrão para o método. A fórmula do somatório é apresentando a seguir.

$$\sigma(TxInG) + \sigma(NumSG) + \sigma(TeeiG)$$

A experimentação dessa análise utilizou o traço WITS do dia 06/01/2010 do período das 15:30h às 16:00h. Na Figura 24 é apresentado o somatório dos desvio padrão para cada tamanho, onde é possível observar que o tamanho 60 apresenta melhor resultado que os demais, levando leve vantagem sobre o 270. Desta forma, fica evidenciado que o tamanho 60 apresenta menor dispersão nos comportamentos médios, ou seja, caso em que um número maior de *hosts* apresenta comportamento próximo da média do traço.

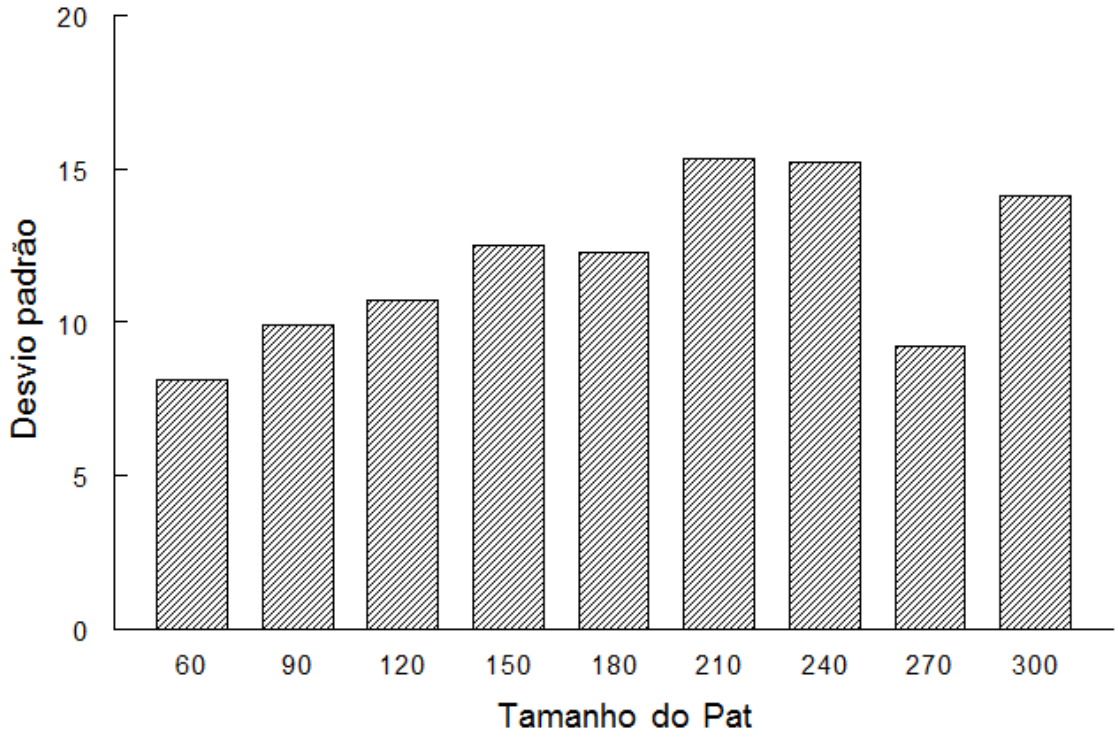

Figura 24. Gráfico com o somatório de desvio padrão para tamanhos de Pat diferentes. Fonte: Elaborada pelo autor.

4.3.2 Experimento 2: Análise da Fase de Monitoramento

Este experimento tem como propósito avaliar a fase de Monitoramento da Etapa de Decisão, seção 3.4, examinando a eficácia da constatação de um evento *Flash Crowd* (verdadeiro positivo) e de eventos normais (verdadeiro negativo). São avaliados também os seguintes tópicos: tamanho da estrutura de contagens, variância mínima do conjunto de contagens e o grau de curtose. Foi utilizado o traço de rede WITS, por ser de tamanho menor, possibilitando uma validação manual da fase de monitoramento. Foi avaliado o dia 06/01/2010 do período das 15:30h às 16:00h para a experimentação.

A fase de monitoramento é formada por uma estrutura de dados, com a contagem do número de usuários que realizaram solicitações Web para um determinado servidor. O objetivo da mesma é fornecer uma percepção da ocorrência de um evento *Flash Crowd,* observando um pico demasiado de usuários a partir do grau da curtose do conjunto de contagens.

Nesse experimento foi preparado e modelado o traço da WITS, conforme seção 3.3.1, e processado pela fase de monitoramento. É apresentada na Figura 25, a taxa de acurácia – TA para estrutura com tamanho 5, 10, 15 e 20 e variância mínima de 5, 10,15,20,25 e 30. Salienta-se, que esta análise averigua a TA sem a injeção de traço *Flash Crowd*, buscando dessa forma, um menor número de falsos positivos, ou seja, conjuntos de contagem apontado como evento abrupto. É possível observar que, todos os tamanhos obtiveram uma TA extremamente alta, porém, ode tamanho 5, obteve melhor resultado com todas variâncias, tendo com o melhor resultado a variância 25 e 30, chegando a TA de 99,98% (Figura 25).

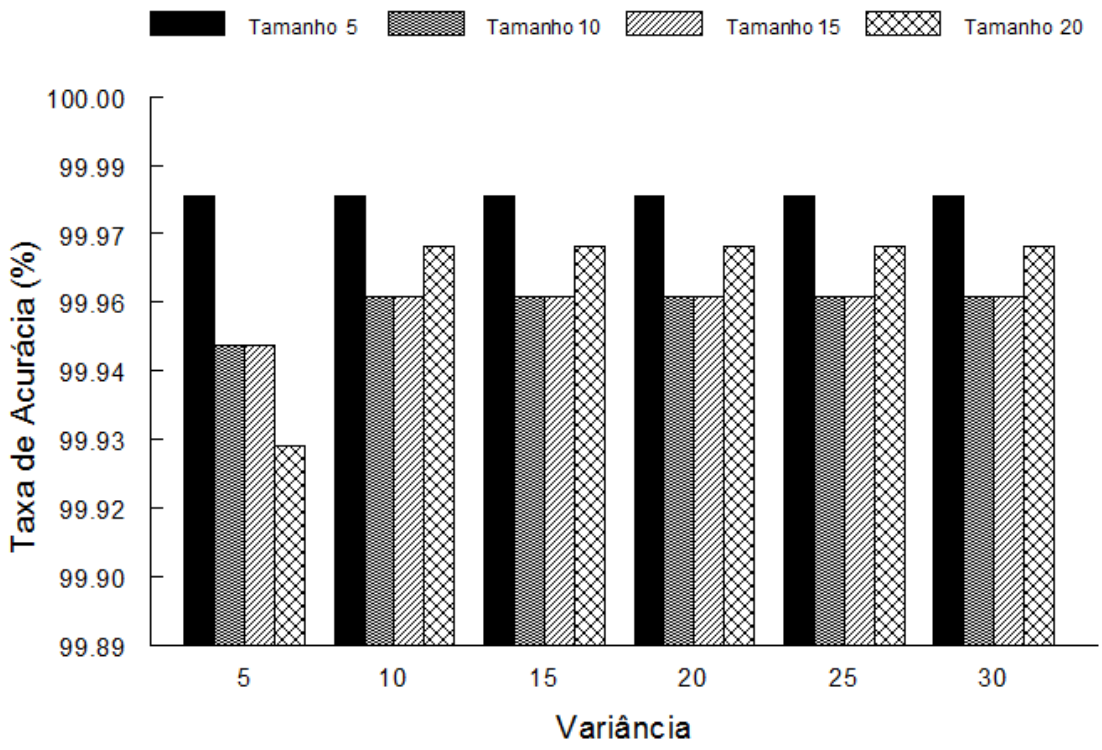

Figura 25. Representação da taxa de acurácia da fase de monitoramento analisando o traço WITS. Fonte: Elaborada pelo autor.

Como o número de amostras do tamanho 5 é de 64,450 e o número de verdadeiro negativo é 64,443, obtemos dessa forma 7 conjuntos de contagens classificados com falso positivo. A TA é calculada conforme a fórmula a seguir:

$$TA = \left(\frac{64{,}443}{64{,}450}\right) \times 100$$

O número de conjuntos analisados muda de acordo com o tamanho da estrutura do monitoramento. Por exemplo, para um determinado servidor com 30 *Pat* e estrutura de tamanho 5, o número de conjuntos de contagens analisado será de 26, já para tamanho 20 serão 11 conjuntos. Como o traço analisado apresenta 2,578 destinos com tamanho 5, o total de conjuntos será 64,450. Os detalhes dos melhores resultados para cada tamanho podem ser observados na Tabela 4.

Tabela 4-Detalhes da análise da fase de monitoramento.

Tamanho	Variância	Nº de Conjuntos Analisado	Nº de Conjuntos Apontado	Nº de Destino Apontado	Verdadeiro Negativo	Falso Positivo
5	25-30	67,028	7	1	99,98	0,2 %
10	10-30	54,138	20	1	99,96	0,4%
15	10-30	41,248	15	1	99,96	0,4%
20	10-30	28,358	8	1	99,97	0,3%

Na Figura 25 é apresentado o gráfico de dispersão do grau de curtose dos conjuntos de contagens do traço analisado utilizando o tamanho de estrutura 5 e variância 25. Aqueles que não apresentaram curtose, ou seja, com valores do conjunto idênticos, foram plotados como sendo -1 e para todos os pontos com o mesmo grau de curtose foi plotado somente um ponto no gráfico.

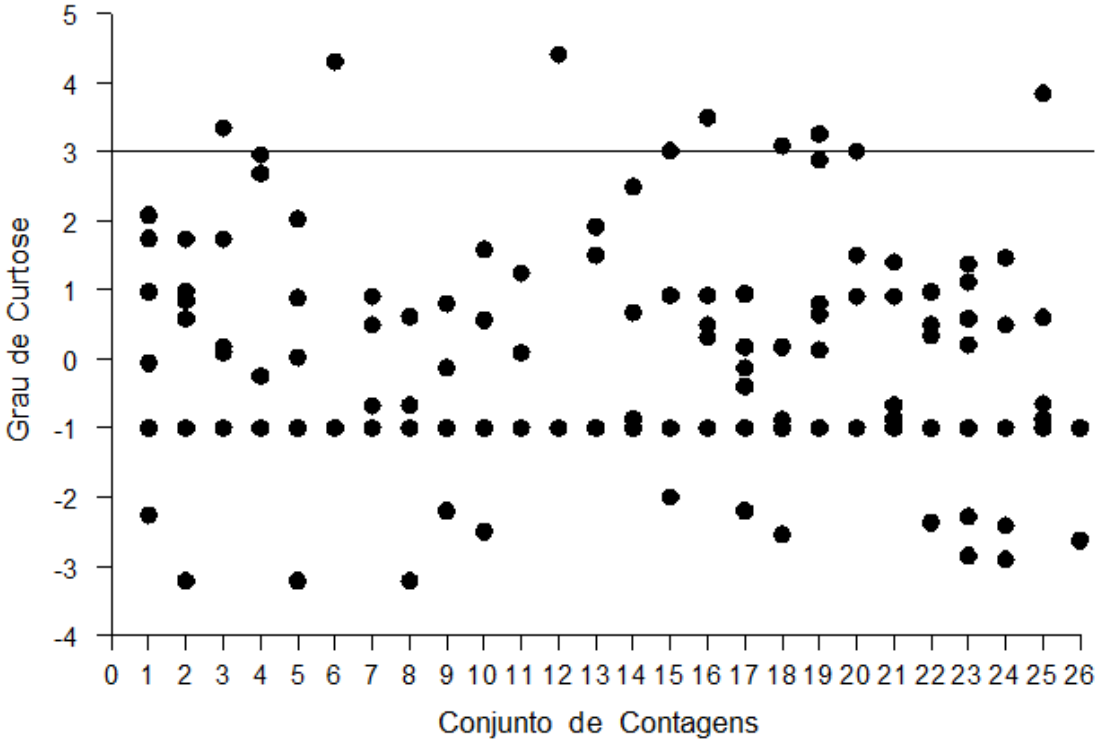

Figura 26. Gráfico de dispersão do grau de curtose do traço WITS para estrutura com tamanho 5 e variância 25.Fonte: Elaborada pelo autor.

Como pode ser observado na Tabela 3, em todas as análises foi apontado um (1) *host* de destino com o grau de curtose maior que 3. Sendo assim, mesmo apresentando excelentes resultados, é examinando o seu comportamento de forma individual para aprofundar ainda mais o experimento. Como o monitoramento utiliza o número de usuários que realizaram solicitação ao destino para o cálculo da curtose, é apresentado na Figura 27 os valores para os 1800 minutos examinados. Analisando a Figura 27, é possível observar que o destino recebeu duas cargas assoberbadas de usuários em dois momentos, acarretando dessa forma, em um grau de curtose maior que 3. O cálculo do grau de curtose dos 26 conjuntos de contagens pode ser observado na Figura 28.

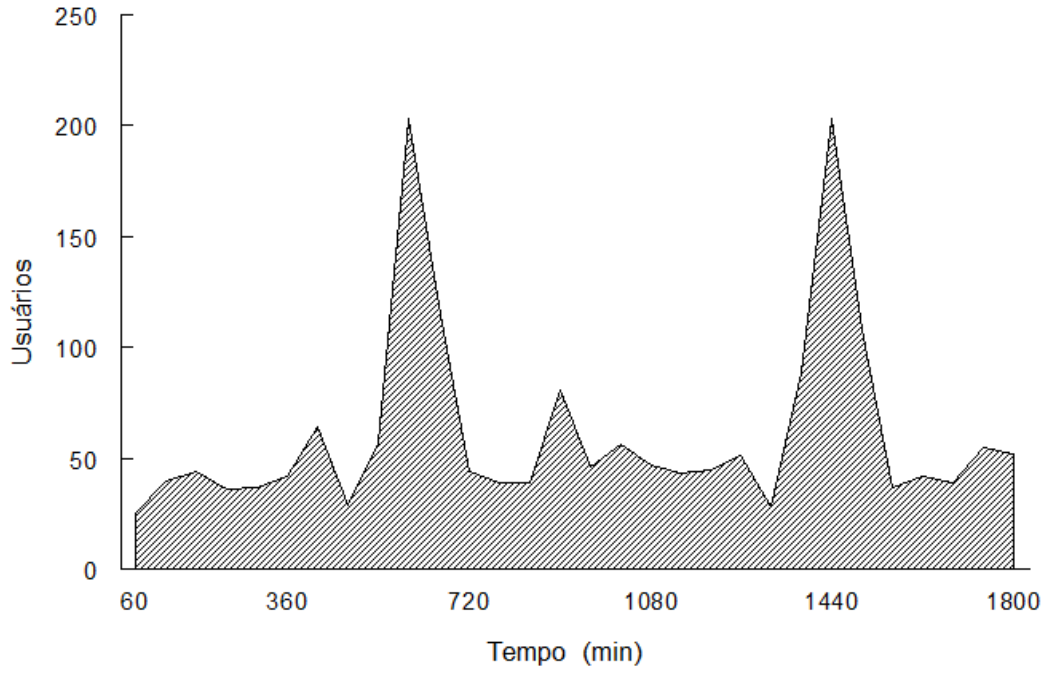

Figura 27. Número de usuários do destino com curtose maior que três. Fonte: Elaborada pelo autor.

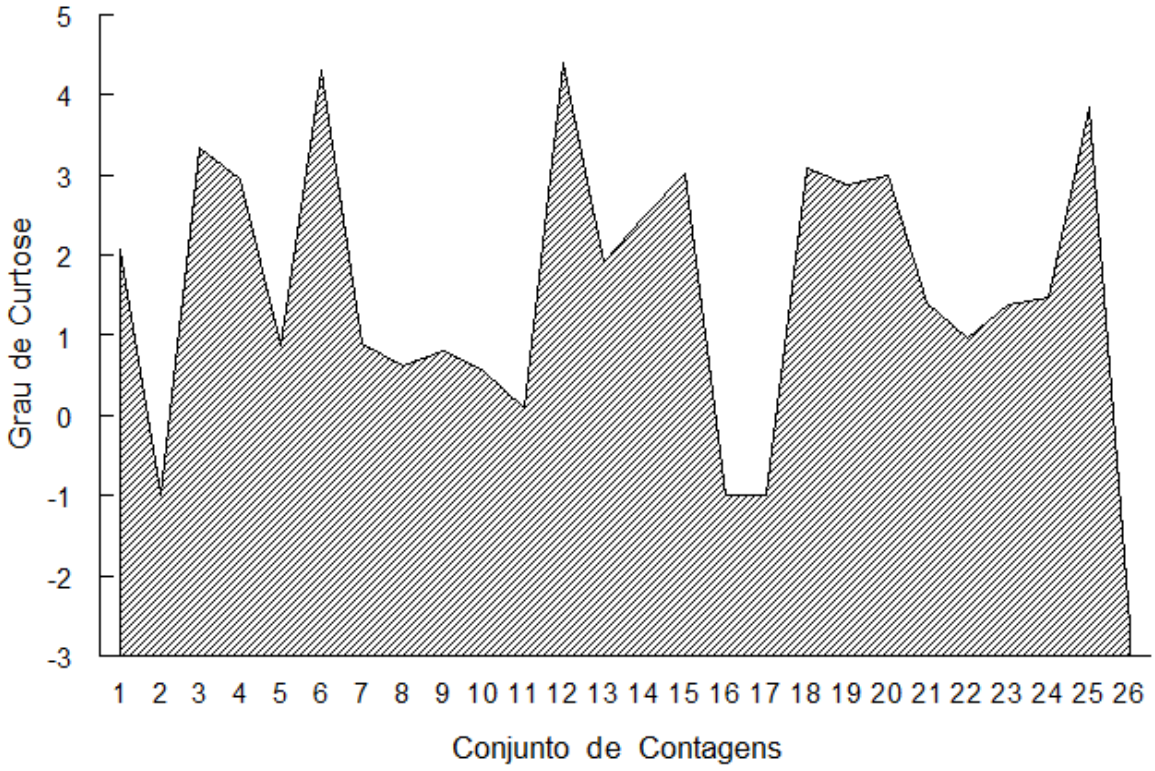

Figura 28. Gráfico do grau de curtose do destino com curtose maior que três. Fonte: Elaborada pelo autor.

A partir desta análise é possível concluir que o destino apontado pelo monitoramento apresentou um aumento abrupto no número de usuários e que tal classificação está de acordo com o objetivo do monitoramento. Sendo assim, o monitoramento além de apresentar excelentes resultados, cumpriu com a finalidade de alertar a eventos abruptos, ou seja, *Flash Crowd*. No próximo experimento, o tamanho da estrutura é padronizado em 5 com variância das contagens em 25.

4.3.3 Experimento 3: Detecção de ataque DDoS Flash Crowd

Este experimento tem como propósito avaliar os métodos propostos (básico e otimizado) realizando a detecção de ataque *DDoS Flash Crowd*. São utilizados como traço de fundo WITS e CAIDA e injetado eventos *Flash Crowd* e ataque *DDoS Flash Crowd* chegando em um traço homogêneo a ser processado por ambos os métodos. O traço WITS utilizado é do dia 06/01/2010 com duração de 30 minutos, sendo das 15h30h às 16:00h e CAIDA do dia 19/02/2015 com duração de 30 minutos, sendo das 13:00h às13:10h. A representação do traço

de fundo pode ser observada na Figura 29, representando cada *Pat* nos quadros em verde. Na primeira análise do experimento é utilizado o traço WITS e no segundo a CAIDA. Para a geração de eventos *Flash Crowd* é utilizado a ferramenta *Scorpius s*e *DDoS* a *Botloader,* na qual permite uma customização do ataque permitindo o envio de solicitações HTTP de maneira mais sutil.

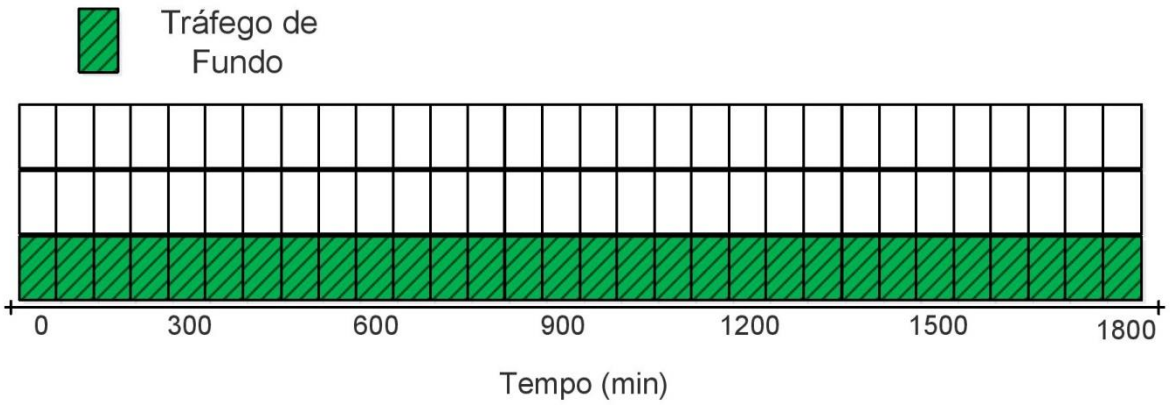

Figura 29: Ilustração do tráfego de fundo, representado nos quadros verdes. Fonte: Elaborada pelo autor.

Utilizando o traço WITS é injetado o tráfego *Flash Crowd* e o *DDoS Flash Crowd* entre o período 600 min à 1400 min, como ilustrado na Figura 30, representado nos quadros azuis e vermelhos respectivamente. Como o endereço de origem dos *hosts* injetados é distinto entre os dois eventos e também do tráfego de fundo, desta forma, ocorre a possibilidade de extração das métricas através da análise dos *hosts*. São injetados 256 hosts para cada evento que realizam as solicitações para o mesmo destino, assim, se os 256 com endereço de *DDoS* forem apontados como ataque, a taxa de verdadeiro positivo será de 100%, já se os 256 *Flash Crowd* somados com os hosts normais forem classificados como normal, a taxa de verdadeiro negativo será de 100%.

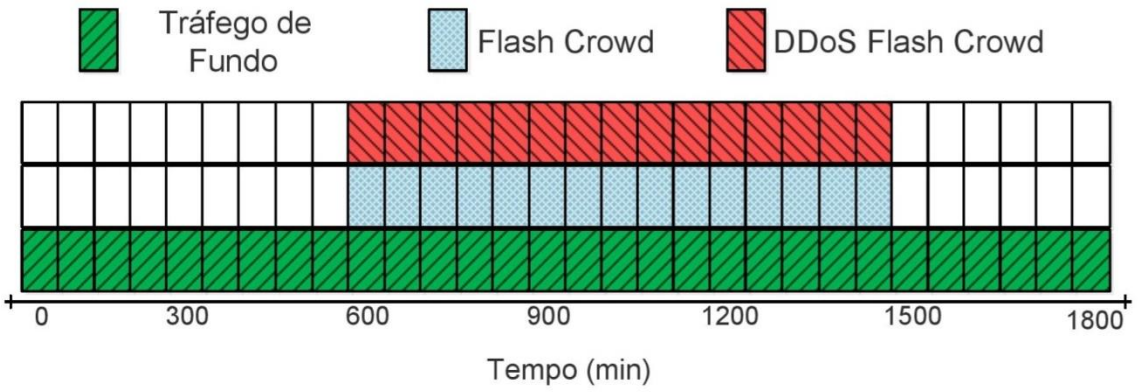

Figura 30: Ilustração do tráfego de fundo e ataque *DDoS*, representados em verde e vermelho respectivamente. Fonte: Elaborada pelo autor.

Formando o traço homogêneo, é realizada a etapa de preparação e modelagem comportamental e de decisão do método básico *DVM* e do otimizado *DAPV*. O *DVM* a cada fechamento de *Pat* apresenta as métricas para o traço, já o *DAPV*, forma a estrutura de monitoramento e na ocorrência de um evento abrupto, a seleção é acionada apresentando as métricas. O gráfico de curtose do traço homogêneo é apresentado na Figura 31, sendo possível observar que no instante que se iniciam os eventos injetados o *DAPV* apresenta uma sequência de curtose de 4,99 (destacado em vermelho). Desta forma, fica evidenciado que o monitoramento é eficaz na detecção dos eventos. São 26 conjuntos de contagens, pelo fato do monitoramento estar utilizando a estrutura com tamanho 5, como anteriormente definido no experimento 4.3.2, que apresentava somente a curtose do tráfego de fundo.

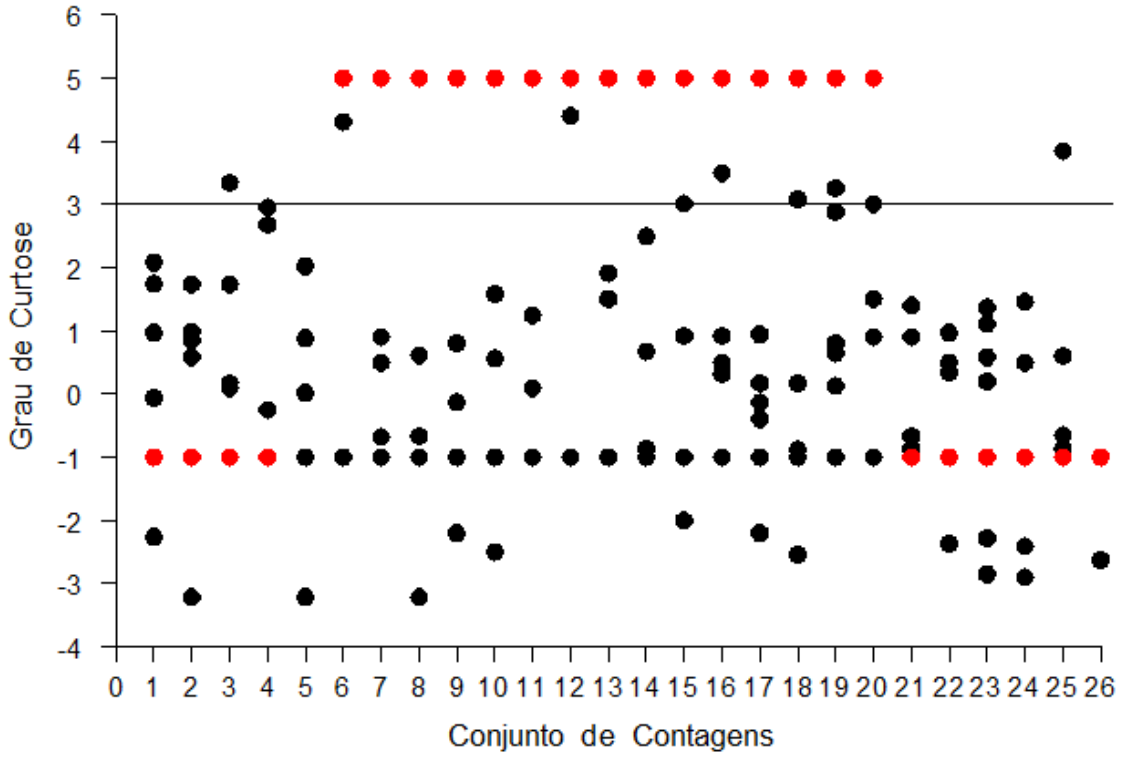

Figura 31. Gráfico de dispersão do grau de curtose. Fonte: Elaborada pelo autor.

Para demonstrar a eficiência dos métodos, foi desenvolvido e implementado o algoritmo de Xu et al. (2012) que visa detectar ataques *DDoS Flash Crowd* realizando a contagem de solicitações HTTP. O método de Xu foi então aplicado ao mesmo traço, sendo calculado o valor de Correlação do Coeficiente de Pearson (PCC) a cada Δt de 5 minutos, conforme os autores

sugerem, e definido um R de 280, que é o número de hosts com maior número de solicitações dentro do Δt, no qual, são utilizados para determinar o PCC do Δt.

Para apresentação das métricas é levado em consideração 2.388 *hosts* do tráfego de fundo e 256 *Flash Crowd*, totalizando 2.644 não atacantes e 256 *DDoS*. Na Figura 32 são apresentados os valores da taxa de verdadeiro positivo, verdadeiro negativo e acurácia para os três métodos experimentados.

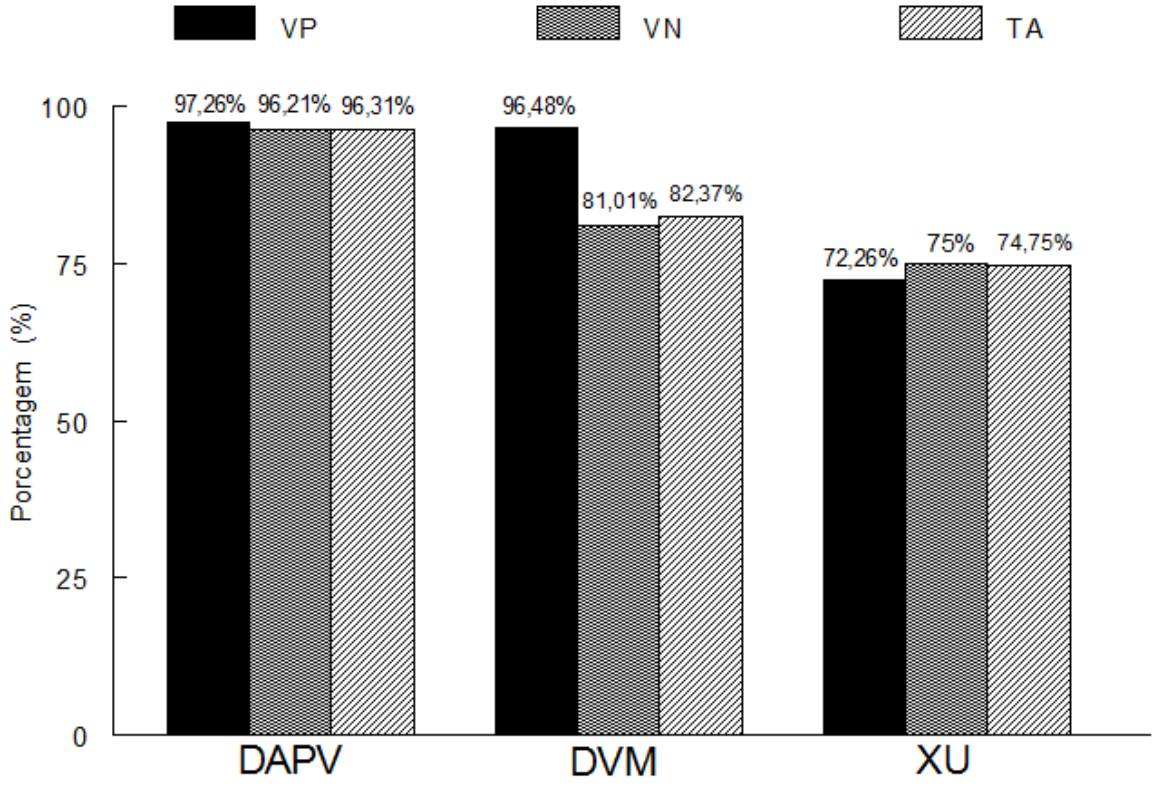

Figura 32: Métricas da experimentação utilizando o traço WITS injetado traços *Flash Crowd* e *DDoS*. Fonte: Elaborada pelo autor.

Para a segunda análise é utilizado como traço de fundo a CAIDA e mantido as mesmas características de injeção dos eventos. Para apresentação das métricas é levado em consideração uma amostra de 20.000 *hosts* e 256 *Flash Crowd*, totalizando 20.256 não atacantes e 256 *DDoS*. Na Figura 33 são apresentados os valores da taxa de verdadeiro positivo, verdadeiro negativo e acurácia para os três métodos experimentados.

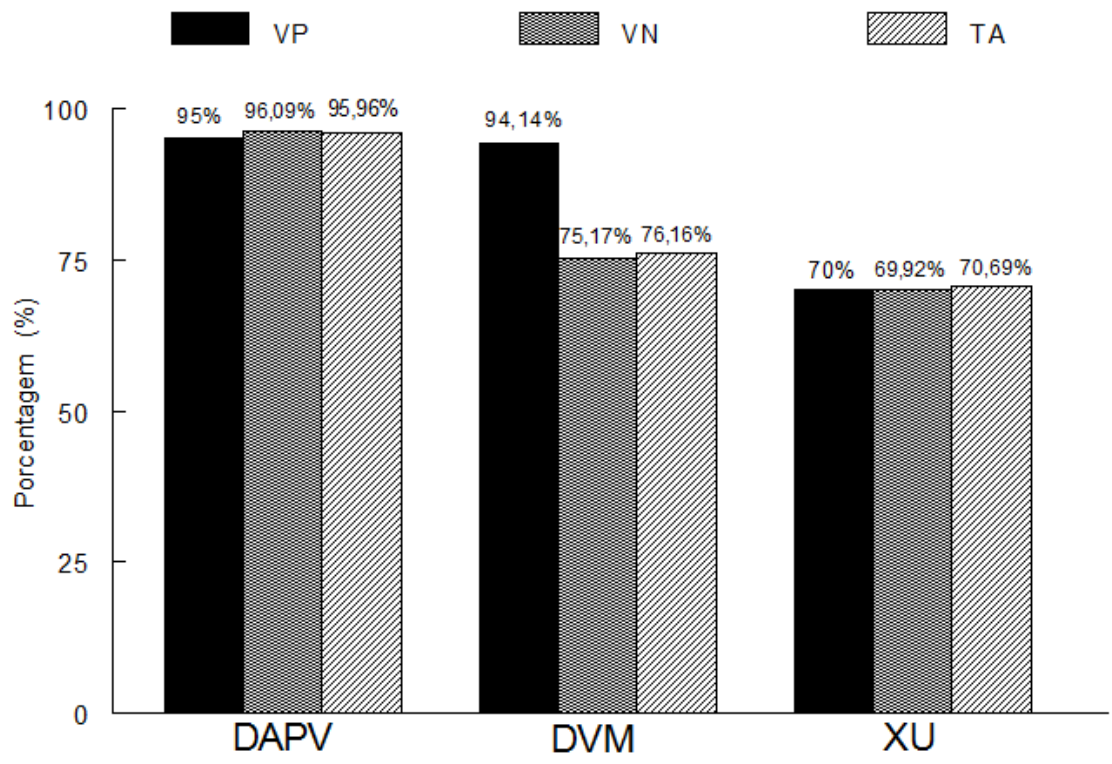

Figura 33: Métricas da experimentação utilizando o traço CAIDA injetado traços *Flash Crowd* e *DDoS*. Fonte: Elaborada pelo autor.

Apresentado as medidas para os dois traços de fundo, é possível observar que o método *DAPV* apresenta melhores resultados nas três medidas. As taxas de verdadeiro positivo nos métodos *DAPV* e *DVM* apresentaram-se muito satisfatórias, com leve vantagem para o *DAPV*. Já na taxa de verdadeiro negativo o *DAPV* apresentou grande vantagem em relação ao *DVM*, isso se deve ao fato da fase de monitoramento somente acionar a seleção para eventos abruptos. O método de Xu apresentou métricas inferiores aos métodos propostos.

5 CONCLUSÕES

Essa dissertação apresenta um novo método de modelagem comportamental para distinguir um ataque *DDoS mimic Flash Crowd* de um *Flash Crowd* legítimo através das solicitações Web, onde são observadas a taxa de interatividade, o número de solicitações e o tempo entre estados interativos. Para examinar os comportamentos e distinguir os usuários em humanos ou *bot*, são apresentadas duas soluções de tomada de decisão: uma baseada na Diferença aos Valores Médios (*DVM*) e outra baseada na Diferença Acumulativa das Porcentagens de Variação (*DAPV*).

Os experimentos revelaram que o método é capaz de gerar resultados satisfatórios na diferenciação entre o comportamento de humanos e de atacantes automatizados (*bots*). O resultado é um novo método para elaborar modelos comportamentais para detecção de ataques *DDoS mimic Flash Crowd.* Por detectar comportamentos anômalos proveniente de *bots*, o método também pode ser usado para fornecimento de uma lista de *hosts* potencialmente maliciosos ou para apoiar técnicas de Web QoS [Fang et al. 2015], que permitem priorizar o tráfego para *hosts* que não estejam incluídos em lista de suspeitos. Além disso, o método também pode ser aplicado em conjunto com métodos de redirecionamento para uma página de CAPTCHA [Singh e De 2015], afim de verificar que não se trata de um *bot*, sendo que, somente aqueles apontados como ataque pelo método seriam direcionados. Diante disto, além da diferenciação entre *Flash Crowd* e *DDoS mimic Flash Crowd*, o método proposto pode ser usado para compor soluções que visam manter a operabilidade de sistemas Web minimizando a exploração por *bots*.

Além disso, a partir dos experimentos, foi possível observar que a modelagem comportamental proposta pode ser examinada com diferentes métodos matemáticos, possibilitando a extração de novas percepções do comportamento humano. Sendo assim, o modelo proposto permite a elaboração de uma gama de novos métodos para avaliação de comportamento e detecção de ataques. Alguns métodos matemáticos aplicáveis são entropia, correlação e clusterização.

As maiores dificuldades encontradas nesse estudo foram no campo da experimentação, tanto na obtenção de traços de rede reais quanto no código fonte de métodos relacionados. Assim, a inexistência de traços reais públicos recentes de eventos *Flash Crowd* e de ataques DDoS *Flash Crowd*, nos conduziu a utilizar ferramentas para a geração de traços sintéticos. Da mesma forma, a indisponibilidade do código fonte de métodos propostos em trabalhos relacionados impossibilita uma comparação abrangente em um ambiente singular

REFERÊNCIAS

ABRAHAMSSON, H.;AHLGREN, B. Using empirical distributions to characterize Web client traffic and to generate synthetic traffic. **Global Telecommunications Conference**. Dec, 2000.

AMARAL, A. A. et al. Deep IP Flow Inspection to Detect beyond Network Anomalies.**Computer Communications**,Dec. 2016.

ARI, I. et al. Managing Flash Crowds on the Internet. **11th IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems** *(MASCOTS 2003),* Orlando, FL, Oct. 2003.

ASSIS, M. V.O.; PROENÇA, M. L. Scorpius: sFlow Network Anomaly Simulator. **Journal of Computer Science.**, n.11, v. 4, p. 662-674, July. 2015.

BEKENEVA, Y. et al. Simulation of DDoS-attacks and protection mechanisms against them. **Young Researchers in Electrical and Electronic Engineering Conference (EIConRusNW),**Russia, Feb. 2015.

BHANDARI, A.; SANGAL, A. L.; KUMAR, K. Characterizing flash events and distributed denial-of-service attacks: an empirical investigation.**Security and Communication Networks**, v. 9, p. 2222–2239, Mar. 2016.

BHATIA, S. et al. A framework for generating realistic traffic for Distributed Denial-of-Service attacks and Flash Events.**Computers& Security**, v.40, p. 95–107, Fev. 2014.

CAIDA -Anonymized Internet Traces (2015).
Disponível em: <http://www.caida.org/data/overview/>. Acesso em: Junho de 2016.

CARVALHO, L. F. et al. Unsupervised learning clustering and self-organized agents applied to help network management.**Expert Systems with Applications**,v. 54, p. 29-47, July. 2016.

DALMAZO, B. L. et al. Filtros de Alarmes de Anomalias através de Wavelets. **In: Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais – SBSeg,** Campinas/SP., v. 1. p. 85-98, 2009.

DANTAS, R. S. A. **Diferenciação de Ataques DDoS e Flash Crowds**. 2013. 75 p. Dissertação (Mestrado em Sistemas e Computação do Instituto Militar de Engenharia)- Instituto Militar de Engenharia, Rio de Janeiro, RJ, 2013.

DECARLO, L. T. (1997). On the meaning and use of kurtosis. **Psychological Methods**, v.2, n 3, p. 292–307. Setembro. 1997.

DHINGRA, A.;SACHDEVA, M. Recent Flash Events : A Study.**International Conference on Communication, Computing & Systems(ICCCS–2014),** p. 94–99, 2014.

FAWCETT, T. An introduction to ROC analysis. **Pattern Recogn. Lett**., v. 27, n. 8, p. 861-874, June. 2006.

FEILY, M., SHAHRESTANI, A.; RAMADASS, S. A survey of botnet and botnet detection. **Proceedings - 3rd International Conference on Emerging Security Information, Systems and Technologies, SECURWARE,** p. 268–273, 2009.

FERNANDES, G. et al. Autonomous profile-based anomaly detection system using principal component analysis and flow analysis.**Applied Soft Computing**, v. 34, p. 513-525, Sept. 2015.

HAMAMOTO, A. H.; CARVALHO, L.F.; PROENCA, M.L. "ACO and GA metaheuristics for anomaly detection,"**2015 34th International Conference of the Chilean Computer Science Society (SCCC)**, pp. 1-6, Nov. 2015.

HWANG, F**. Análise dos Efeitos Gerados pelo Comportamento das Aplicações e pelo Perfil das Redes na Característica Auto-Similar do Tráfego Internet.**2004.167 p. Dissertação (Mestrado em Engenharia Elétrica)- Universidade Estadual de Campinas, Campinas, SP, 2004.

HWANG, F., BIANCHI, G. R.; LING, L. L.Impacto Gerado pelo Comportamento das Aplicações (Web, FTP e E-mail) e pelo Perfil das Redes na Característica Auto-Similar. **IEEE LatinAmericaTransactions,** v.3, n.4, p.356–361, Out. 2005.

JARDIM, S.L.; NUNES, R.C.; COLOMÉ, M. Detecção de Ataques DDoS Flash Crowd Realizando Análise Comportamental de Solicitações Web. **XVI Simpósio Brasileiro em Segurança de Informática e de Sistemas Computacionais**, p. 156-169, Nov. 2016.

JENA, A. K.; POPESCU, A.; NILSSON, A. A. Modeling and evaluation of Internet applications.**Teletraffic Science and Engineering**, v. 5, p. 531–540, 2003.

MATHWORLD (2017).
Disponívelem: <http://mathworld.wolfram.com/Kurtosis.html>. Acesso em: Janeiro de 2017.

KANDULA, S. et al. Botz-4-sale: Surviving organized DDoS attacks that mimic flash crowds. **2nd Symposium on Networked Systems Design and Implementation (NSDI),** p.287–300, May. 2005.

KAPPEL, M. A. A. et al.Análise não Estacionária Aplicada a Sinais de Tribocorrosão. **Revista Virtual de Química,** v. 7, n.5, p. 1651-1662, Maio. 2015.

KE, L. et al. Distinguishing DDoS Attacks from Flash Crowds Using Probability Metrics. In: **Proc. of the International Conference on Network and System Security – NSS**, v.9, p.09-17, 2009.

KHUNDRAKPAM, J.S.; KHELCHANDRA, T.; TANMAY, D. Entropy-Based Application Layer DDoS Attack Detection Using Artificial Neural Networks. **Entropy**, v. 18, n. 10, p. 350, Oct. 2016.

LUCENA, S. C.; MOURA, A. S. Estimativa de Holt-Winters para Detecção de Ataques em Redes WAN. **X Simpósio Brasileiro Em Segurança Da Informação E De Sistemas Computacionais,** v. d, p. 157–170, 2010.

OIKONOMOU, G.; MIRKOVIC, J. Modeling Human Behavior for Defense against Flash-Crowd Attacks. **ICC'09 Proceedings of the 2009 IEEE international conference on Communications,** n. 0430228, p. 625–630, June. 2009.

PAN, J.; HU, H.; LIU, Y. Human behavior during Flash Crowd in web surfing. **Physica A: Statistical Mechanics and its Applications**, v. 413, p. 212–219, Nov. 2014.

PRASAD, K. M. et al. Discriminating DDoS Attack traffic from Flash Crowds on Internet Threat Monitors (ITM) Using Entropy variations**. AfricanJournalofComputing& ICT**, v. 6, n. 2, p. 53–62, Sept. 2013.

RIGUI, M.A.; NUNES, R.C. Detecção de DDoS Através da Análise da Quantificação da Recorrência Baseada na Extração de Características Dinâmicas e Clusterização Adaptativa.**XVI Simpósio Brasileiro em Segurança de Informática e de Sistemas Computacionais,** p. 380-393, Nov. 2016.

SARAVANAN, R.; SHANMUGANATHAN, S.; PALANICHAMY, Y. Behavior-based detection of application layer distributed denial of service attacks**. Turkish Journal of Electrical Engineering & Computer Sciences**, v. 24, n. 2, p. 1–14, Jan. 2013.

SHIAELES, S. N.; PAPADAKI, M. FHSD: An Improved IP Spoof Detection Method for Web DDoS Attacks. **The Computer Journal**, v. 58, n. 4, p. 892-903, Dec. 2015.

SINGH, B.; KUMAR, K.; BHANDARI, A. Simulation Study of Application Layer DDoS Attack. **International Conference on Green Computing and Internet of Things (ICGCIoT)**, p. 893–898, Oct. 2015.

THAPNGAM, T. et al. Discriminating DDoS Attack Traffic from Flash Crowd through Packet Arrival Patterns. **Computer Communications Workshops (INFOCOM), 2011 IEEE Conference**, p. 952–957, Apr. 2011.

WAIKATO (2016). Waikato Internet Traffic Storage- WITS
Disponível em: <http://wand.net.nz/wits/> Acesso em: Junho de 2016.

XIE, Y.; YU, S. Z. Monitoring the application-layer DDoS sttacks for popular websites. **IEEE/ACM Trans on Networking**, v. 17, n. 1, p. 15–25, Feb. 2009.

XYCOON (2017).
Disponível em: <http://www.xycoon.com/peakedness_small_sample_test_1.htm>. Acessoem: Janeiro de 2017.

XU, C.; DU, C.;KONG, X. An Application Layer DDoS Real-Time Detection Method in Flash Crowd. **International Association of Computer Science & Information Technology (IACSIT),** v. 30, p. 68–73, Feb. 2012.

YE, C.; ZHENG, K. Detection of application layer distributed denial of service. **International Conference on Computer Science and Network Technology Detection,** p. 310–314, Dec. 2011.

YU, S.; ZHOU, W.; JIA, W.; GUO, S.; XIANG, Y.; TANG, F. Discriminating DDoS Attacks from Flash Crowds Using Flow Correlation Coefficient. **IEEE Transactions on Parallel and Distributed Systems,** v23**,** p. 1073-1080, Jun. 2012.